MikroTik RouterOS™ V2.7 Reference Manual

MikroTik

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

MikroTik RouterOS™ V2.7 Reference Manual

PDF version (for printing)

Document revision 1.99 (30−Dev−2003)
This document applies to the MikroTik RouterOS™ V2.7

MikroTik RouterOS™ V2.7 Basic Setup Guide

PDF version

Document revision 1.3 (09−Jun−2003)
This document applies to the MikroTik RouterOS™ V2.7

Table Of Contents

Summary

MikroTik RouterOS™ is independent Linux−based Operating System for PC−based routers and thinrouters. It does not require any additional components and has no software prerequirements. It is designed with easy−to−use yet powerful interface allowing network administrators to deploy network structures and functions, that would require long education elsewhere simply by following the Reference Manual (and even without it).

Related Documents

Software Package Installation and Upgrading
Device Driver List
License Management
Ping
Queues and Data Rate Management
Packet Filter (Firewall) and NAT (Network Address Translation)

Description

MikroTik RouterOS™ turns a standard PC computer into a powerful network router. Just add standard network PC interfaces to expand the router capabilities.

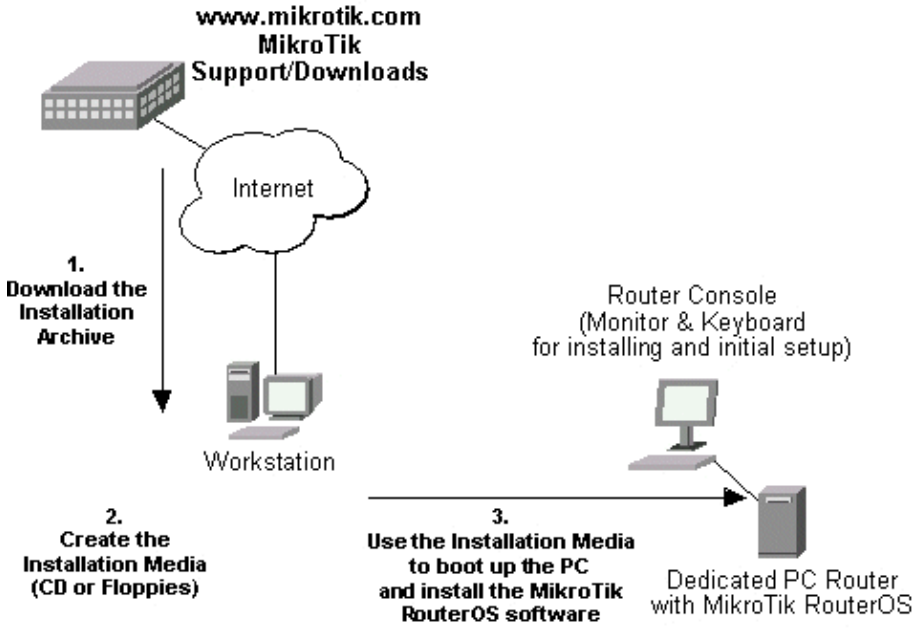
- Remote control with easy real−time Windows application (WinBox)
- Telnet/SSH/console/serial console control with RADIUS authentication
- Advanced bandwidth control
- Network firewall with packet−filtering, masquerading, network address translation, logging and connection monitoring
- DHCP support
- HotSpot gateway with RADIUS authentication
- Ethernet 10/100/1000Mb/s
- Wireless client and Access Point 2.4GHz 11Mb/s (IEEE802.11b), 5GHz 54Mb/s (IEEE802.11a) and 2.4GHz 54Mb/s (IEEE802.11g) with RADIUS authentication for AP
- V.35 synchronous 8.448Mb/s with Sync−PPP, HDLC or Frame Relay
- X.21 synchronous 8.448Mb/s with Sync−PPP, HDLC or Frame Relay
- Async PPP (up to 128 ports) with RADIUS authetication for modem pools
- E1/T1 support
- IP Telephony Gateway
- Built−in Web−proxy
- And much more

The Guide describes the basic steps of installing and configuring a dedicated PC router running MikroTik RouterOS™.

Setting up MikroTik RouterOS™

Downloading and Installing the MikroTik RouterOS™

The download and installation process of the MikroTik RouterOS™ is described in the following diagram:



1. Download the basic installation archive file.

Depending on the desired media to be used for installing the MikroTik RouterOS™ please chose one of the following archive types for downloading:

- **ISO image** of the installation CD, if you have a CD writer for creating CDs. The ISO image is in the MTcdimage_v2−7−x_dd−mmm−yyyy.zip archive file containing a bootable CD image. The CD will be used for booting up the dedicated PC and installing the MikroTik RouterOS™ on its hard−drive or flash−drive.
- **MikroTik Disk Maker**, if you want to create 3.5" installation floppies. The Disk Maker is a self−extracting archive DiskMaker_v2−7−x_dd−mmm−yyyy.exe file, which should be run on your Windows 95/98/NT4/2K/XP workstation to create the installation floppies. The installation floppies will be used for booting up the dedicated PC and installing the MikroTik RouterOS™ on its hard−drive or flash−drive.
- **Netinstall**, if you want to install RouterOS™ over a LAN with one floppy boot disk, or alternatively using PXE−boot option supported by some network interface cards, that allows truly networked installation. Netinstall program works on Windows 95/98/NT4/2K/XP.

2. Create the installation media

Use the appropriate installation archive to create the Installation CD or floppies.

- For the CD, write the ISO image onto a blank CD.
- For the floppies, run the Disk Maker on your Windows workstation to create the installation floppies. Follow the instructions and insert the floppies in your FDD as requested, label them as

Disk 1,2,3, etc.

3. Install the MikroTik RouterOS™ software.

Your dedicated PC router hardware should have:

- An advanced 4th generation (core frequency 100MHz or more), 5th generation (Intel Pentium, Cyrix 6X86, AMD K5 or comparable) or newer Intel IA−32 (i386) compatible **motherboard and processor** (uniprocessor only, dual processors and other SMP configurations are not supported);
- from 32MB to 1GB **RAM** (from 48MB suggested);
- 30MB or more PRIMARY MASTER **IDE HDD or IDE flashdrive**.
- A **network adapter** (NE2000 compatible PCI or ISA Ethernet card, or any other supported NIC, see the supported device list on our web page);

For installation purposes (and only for that time) you should also have:

- A SECONDARY MASTER **CD drive** set as **primary boot** device, if you want to use the created CD for installing the MikroTik RouterOS™ onto the primary master HDD;
- A **3.5" FDD** set as primary boot device, if you want to use the created set of floppies for installing the MikroTik RouterOS™;
- A **monitor and keyboard** for installation and initial setup of the MikroTik Router. The monitor and keyboard do not need to be connected to the router after it is set up for connecting to it over the network.

Boot up your dedicated PC router from the Installation Media you created and follow the instructions on the console screen while the HDD is reformatted and MikroTik RouterOS™ installed on it.

After successful installation please remove the installation media from your CD or floppy disk drive and hit 'Enter' to reboot the router. While the router will be starting up for the first time you will be given a **Software ID** for your installation and asked to supply a valid software license key (**Software Key**) for it. Write down the Software ID. You will need it to obtain the Software License through the MikroTik Account Server. If you need extra time to obtain the Software License Key, you may want to power off the router. Type **shutdown** in the Software key prompt and power the router off when the router is halted.

Notes

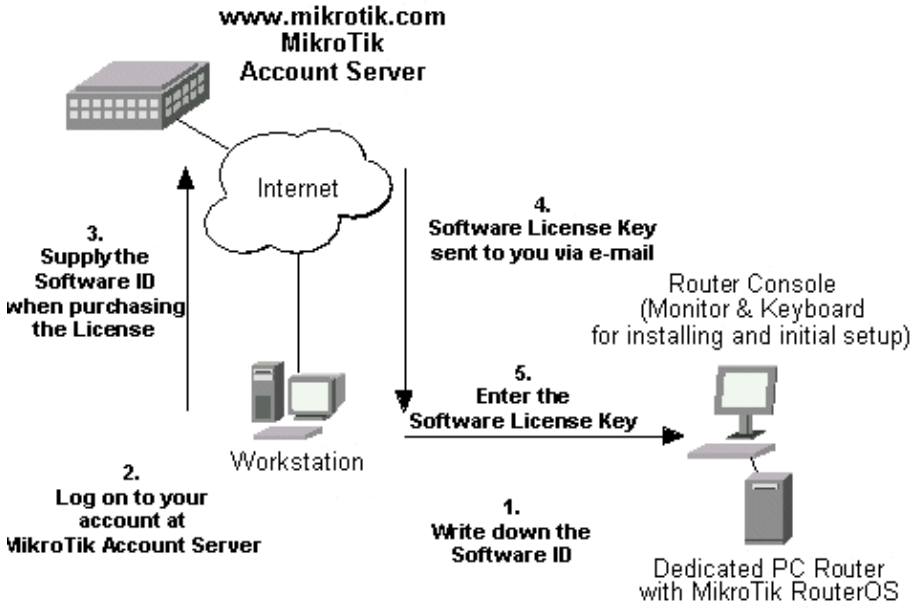
The installation from CD or network requires Base (paid) License. If you intend to obtain the Free Demo License, you should use the floppy installation media.

The hard disk will be entirely reformatted during the installation and all data on it will be lost!

You can move the hard drive with MikroTik RouterOS™ installed to a new hardware without loosing a license, but you cannot move the RouterOS™ to a different hard drive without purchasing an another license (except hardware failure situations). For additional information write to key−support@mikrotik.com

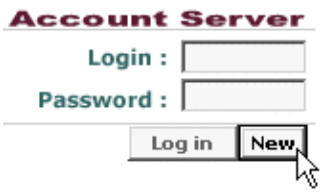
Obtaining the Software License

The MikroTik RouterOS™ Software licensing process is described in the following diagram:



After installing the router and starting it up for the first time you will be given a Software ID.

1. Write down the Software ID reported by the RouterOS™.
2. If you have an account with MikroTik, follow to the next step.
   If you do not have an account at www.mikrotik.com, just press the 'New' button on the upper right−hand corner of the MikroTik's web page to create your account.



   You will be presented with the Account Sign−Up Form where you chose your account name and fill in the required information.
3. To obtain the Software License Key, log on to your account at www.mikrotik.com entering your account name and password (upper right−hand corner on this webpage), for example:

4. After logging on to the Account Server select "Free Demo License" or "Order Software License" in the Account Menu.
5. The Software Key will be sent to the email address, which has been specified in your account setup.
6. Read your email and enter the Software Key at the router's console, for example:

```
Software ID: 5T4V-IUT
Software key: 4N7X-UZ8-6SP
```

Instead of entering the license key you can enter **shutdown** to shut down the router and enter the license key later, or enter **display** to read the License Agreement, or **help** to see a help message.

After entering the correct Software License Key you will be presented with the MikroTik Router's login prompt.

Notes

The CD or Netinstall installation cannot be 'unlocked' with the Free Demo Key. Use the Floppy installation or purchase a Licensed Key.

Logging into the MikroTik Router

When logging into the router via terminal console, you will be presented with the MikroTik RouterOS™ login prompt. Use 'admin' and no password (hit 'Enter') for logging on to the router for the first time, for example:

```
MikroTik v2.7
Login: admin
Password:
```

The password can be changed with the **/password** command.

Adding Software Packages

The basic installation comes with only the "system" package and few other packages. This includes basic IP routing and router administration. To have additional features such as IP Telephony, OSPF, wireless and so on, you will need to download additional software packages.

The additional software packages should have **the same version** as the system package. If not, the package won't be installed. Please consult the MikroTik RouterOS™ Software Package Installation and Upgrading Manual for more detailed information about installing additional software packages.

Software Licensing Issues

If you want to upgrade your 'free' version of MikroTik RouterOS™ installation to a 'paid' version, please purchase the new Software License KEY for the Software ID you used when getting the 'free' demo license. Similarly, if additional license is required to enable the functionality of a software package, the license should be obtained for the Software ID of your system. The new key should be entered using the **/system license set key** command, and the router should be rebooted afterwards:

```
[admin@MikroTik] ip firewall src-nat> /system license print
         software-id: "SB6T-R8T"
                 key: "3YIV-ZW8-DH2"
    upgradable-unitl: apr/01/2004
[admin@MikroTik] system license> feature print
Flags: X - disabled
  #   FEATURE
  0 X AP
  1 X synchronous
  2 X radiolan
  3 X wireless-2.4gHz
  4   licensed
```

```
[admin@MikroTik] system license> set key=D46G-IJ6-QW3
[admin@MikroTik] system license>/system reboot
Reboot, yes? [y/N]: y
system will reboot shortly
```

Notes

If there is no appropriate license, the appropriate interfaces wont show up under the interface list, even though the packages can be installed on the MikroTik RouterOS™ and corresponding drivers loaded.

Navigating the Terminal Console

Welcome Screen and Command Prompt

After logging into the router you will be presented with the MikroTik RouterOS™ Welcome Screen and command prompt, for example:

```
  MMM      MMM       KKK                          TTTTTTTTTTT      KKK
  MMMM    MMMM       KKK                          TTTTTTTTTTT      KKK
  MMM MMMM MMM  III  KKK  KKK  RRRRRR     OOOOOO      TTT     III  KKK  KKK
  MMM  MM  MMM  III  KKKKK     RRR  RRR  OOO  OOO     TTT     III  KKKKK
  MMM      MMM  III  KKK KKK   RRRRRR    OOO  OOO     TTT     III  KKK KKK
  MMM      MMM  III  KKK  KKK  RRR  RRR   OOOOOO      TTT     III  KKK  KKK

  MikroTik RouterOS v2.7 (c) 1999-2003       http://www.mikrotik.com/

Terminal xterm detected, using multiline mode
[admin@MikroTik] >
```

The command prompt shows the identity name of the router and the current menu level, for example:

```
[admin@MikroTik] >               Base level menu
[admin@MikroTik] interface>      Interface configuration
[admin@MikroTik] ip address>     IP Address management
```

Commands

The list of available commands at any menu level can be obtained by entering the question mark '?', for example:

```
[admin@MikroTik] > ?

     driver  Driver management
       file  Local router file storage.
     import  Run exported configuration script
  interface  Interface configuration
        log  System logs
   password  Change password
       ping  Send ICMP Echo packets
       port  Serial ports
       quit  Quit console
     radius  Radius client settings
       redo  Redo previosly undone action
      setup  Do basic setup of system
       snmp  SNMP settings
```

```
       undo  Undo previous action
       user  User management
        ppp  Point to Point Protocol
         ip  IP options
      queue  Bandwidth management
     system  System information and utilities
       tool  Diagnostics tools
    routing  Various routing protocol settings
     export  Print or save an export script that can be used to restore
             configuration


[admin@MikroTik] > ip ?

      accounting  Traffic accounting
         address  Address management
             arp  ARP entries management
             dns  DNS settings
        firewall  Firewall management
        neighbor  Neighbors
         packing  Packet packing settings
            pool  IP address pools
           route  Route management
         service  IP services
  policy-routing  Policy routing
            upnp
     dhcp-client  DHCP client settings
     dhcp-server  DHCP server settings
       dns-cache  DNS cache management
           ipsec  IP security
          export  Print or save an export script that can be used to restore
                  configuration
[admin@MikroTik] > ip
```

The list of available commands and menus has short descriptions next to the items. You can move to the desired menu level by typing its name and hitting the [Enter] key, for example:

```
[admin@MikroTik] >                      Base level menu
[admin@MikroTik] > driver               Enter 'driver' to move to the driver level
                                        menu
[admin@MikroTik] driver> /              Enter '/' to move to the base level menu
                                        from any level
[admin@MikroTik] > interface            Enter 'interface' to move to the interface
                                        level menu
[admin@MikroTik] interface> /ip         Enter '/ip' to move to the IP level menu
                                        from any level
[admin@MikroTik] ip>
```

A command or an argument does not need to be completed, if it is not ambiguous. For example, instead of typing 'interface' you can type just 'in' or 'int'. To complete a command use the [Tab] key.

The commands may be invoked from the menu level, where they are located, by typing its name. If the command is in a different menu level than the current one, then the command should be invoked using its full (absolute) or relative path, for example:

```
[admin@MikroTik] ip route> print                  Prints the routing table
[admin@MikroTik] ip route> .. address print       Prints the IP address table
[admin@MikroTik] ip route> /ip address print      Prints the IP address table
```

The commands may have arguments. The arguments have their names and values. Some commands, may have a required argument that has no name.

Summary on executing the commands and moving between the menu levels

```
       Command                              Action
command [Enter]      Execute the command
[?]                  Show the list of all available commands
command [?]          Display help on the command and the list of arguments
command argument [?] Display help on the command's argument
[Tab]                Complete the command/word. If the input is ambiguous, a
                     second [Tab] gives possible options
/                    Move up to the base level
/command             Execute the base level command
..                   Move up one level
""                   Enter an empty string
"word1 word2"        Enter 2 words that contain a space
```

You can abbreviate names of levels, commands and arguments.

For the IP address configuration, instead of using the 'address' and 'netmask' arguments, in most cases you can specify the address together with the number of true bits in the network mask, i.e., there is no need to specify the 'netmask' separately. Thus, the following two entries would be equivalent:

```
/ip address add address 10.0.0.1/24 interface ether1
/ip address add address 10.0.0.1 netmask 255.255.255.0 interface ether1
```

Notes

You **must** specify the size of the network mask in the address argument, even if it is the 32−bit subnet, i.e., use 10.0.0.1/32 for address 10.0.0.1 and netmask 255.255.255.255

Accessing the Router Remotely Using Web Browser and WinBox Console

Summary

The MikroTik router can also be accessed remotely using **http** and **WinBox Console**, for example, using the web browser of your workstation.

Description

The Winbox Console is used for accessing the MikroTik Router configuration and management features using graphical user interface.

All Winbox interface functions are as close as possible to Console functions: all Winbox functions are exactly in the same place in Terminal Console and vice versa (except functions that are not implemented in Winbox). That is why there are no Winbox sections in the manual.

The Winbox Console plugin loader, the winbox.exe program, can be retrieved from the MikroTik router,

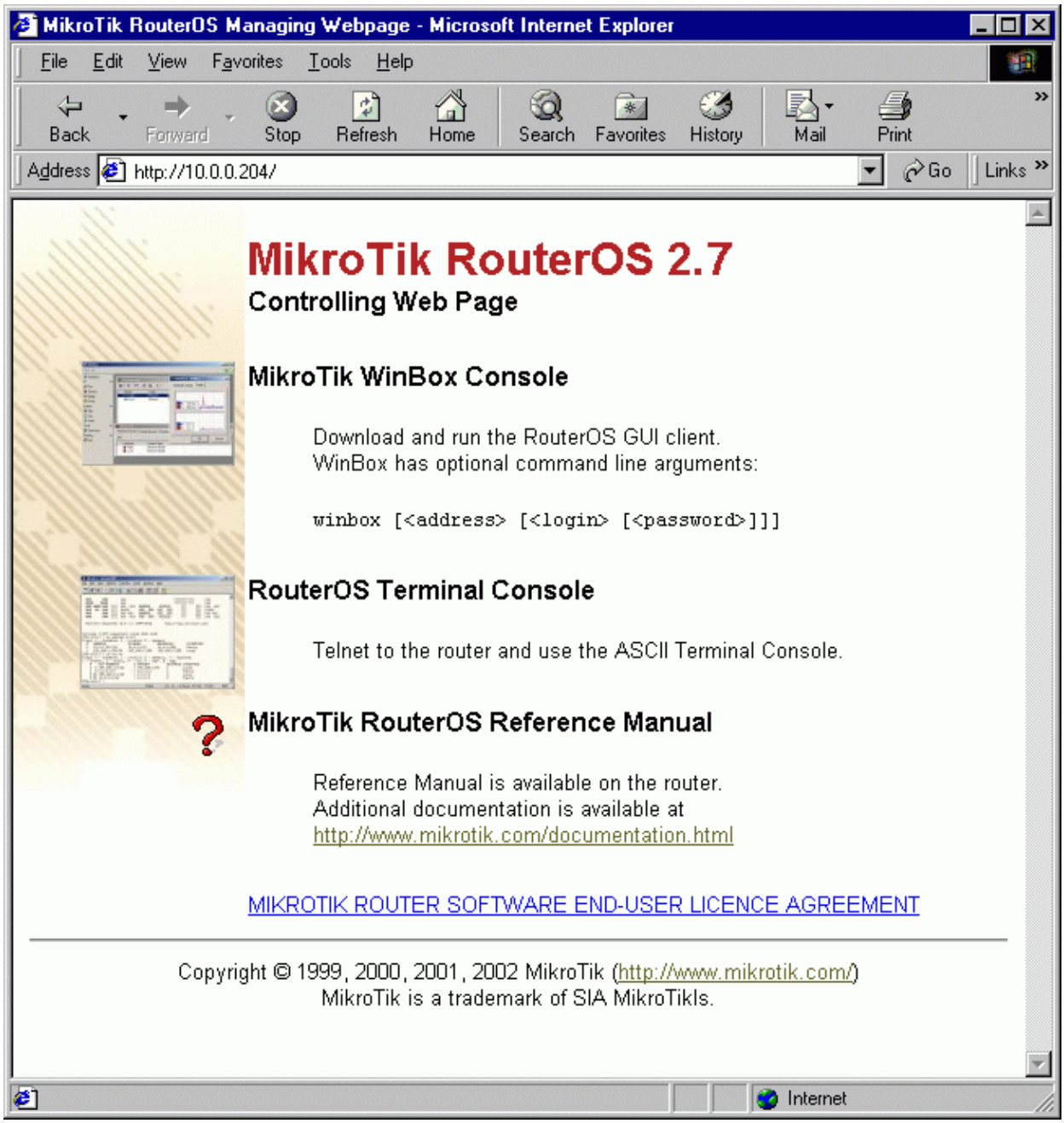
the URL is **http://*router_address*/winbox/winbox.exe** Use any web browser on Windows 95/98/ME/NT4.0/2000/XP to retrieve the router's web page with the mentioned link.

Note that if you change the default port for www service on the router, you will have to specify it just after the IP address separated by column (eg. 10.0.0.1:8080).

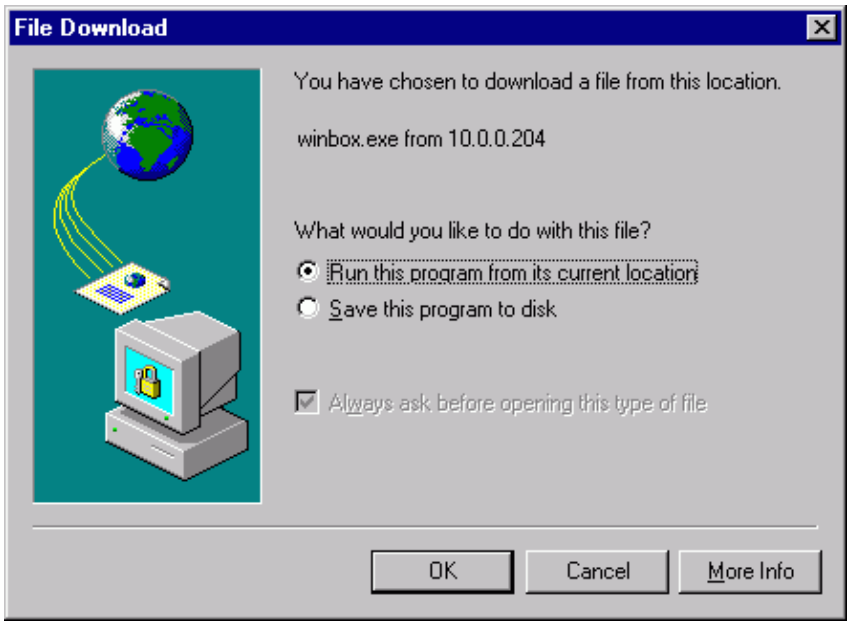
The winbox plugins are cached on the local disk for each MikroTik RouterOS™ version. The plugins are not downloaded, if they are in the cache, and the router has not been upgraded since the last time it has been accessed.

Starting the Winbox Console

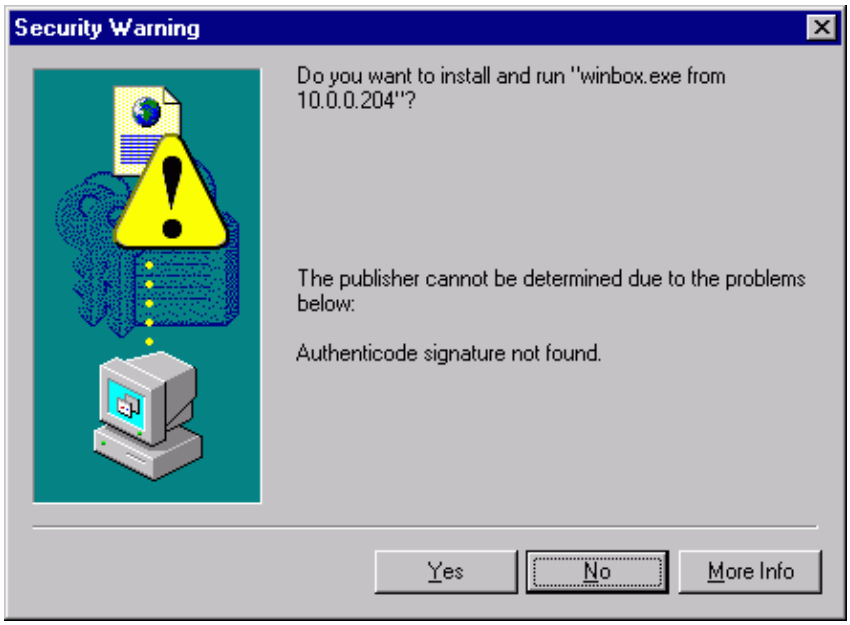
When connecting to the MikroTik router via http (TCP port 80 by default), the router's Welcome Page is displayed in the web browser, for example:



By clicking on the Winbox Console link you can start the winbox.exe download. Choose the option "Run this program from its current location" and click "OK":

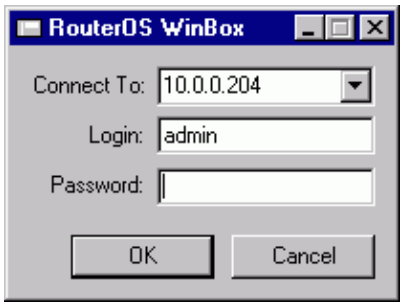


Accept the security warning, if any:

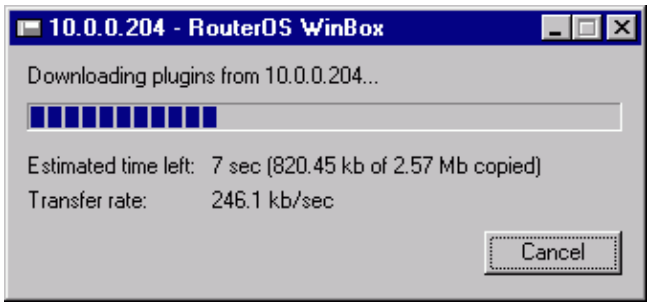


Alternatively, you can save the winbox.exe program to your disk and run it from there.

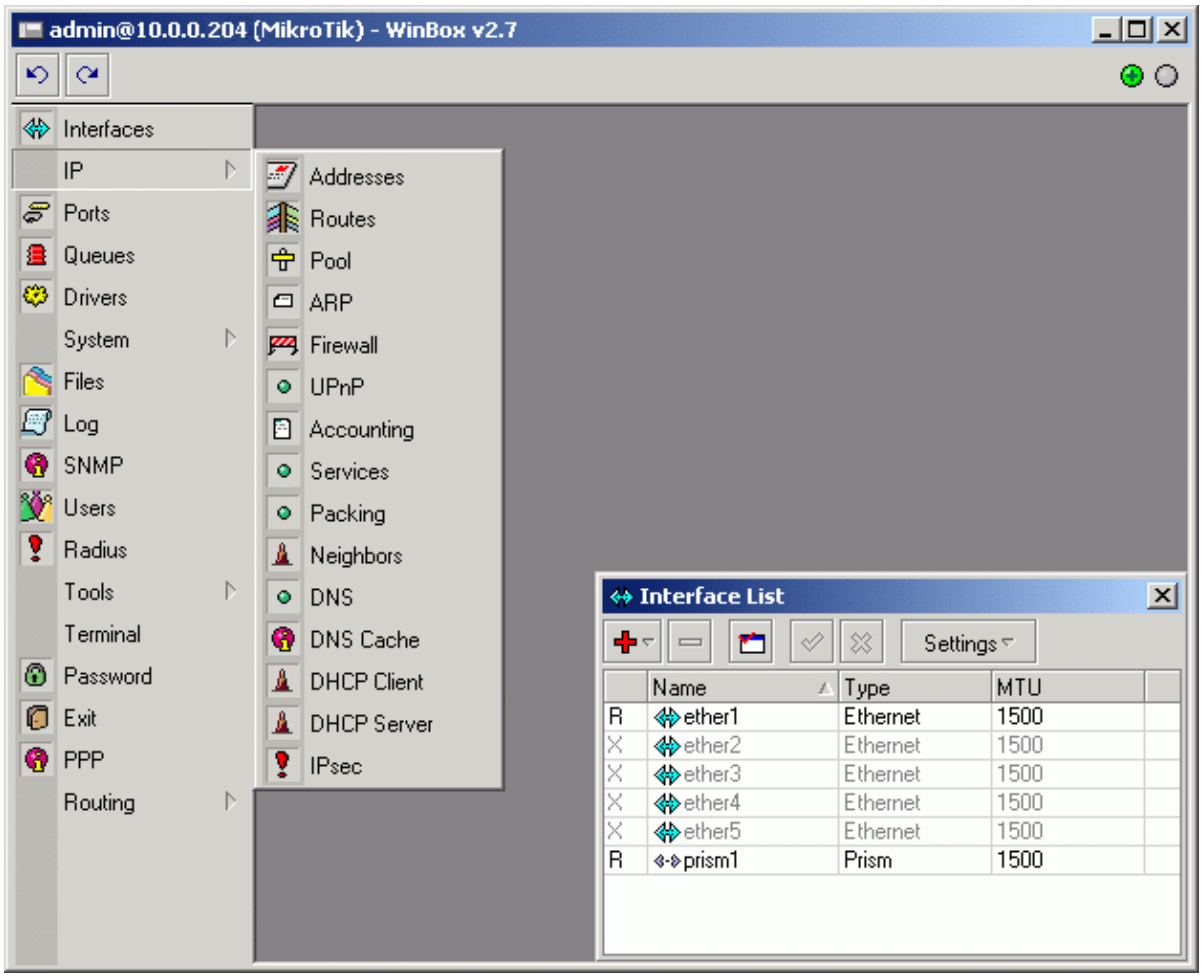
The winbox.exe program opens the Winbox login window. Login to the router by specifying the IP address (and the port number if you have changed it from the default value of 80), user name, and password, for example:



Watch the download process of Winbox plugins:



The Winbox console is opened after the plugins have been downloaded:



The Winbox Console uses TCP port 3986 (not secure) or 3987 (secure; requires **security** package to be installed). After logging on to the router you can work with the MikroTik router's configuration through the Winbox console and perform the same tasks as using the regular console.

Overview of Common Functions

You can use the menu bar to navigate through the router's configuration menus, open configuration windows. By double clicking on some list items in the windows you can open configuration windows for the specific items, and so on.

There are some hints for using the Winbox Console:

- To open the required window, simply click on the corresponding menu item.
- To add a new entry you should click on the icon in the corresponding window.
- To remove an existing entry click on the icon.
- To enable an item, click on the icon.
- To disable an item, click on the icon.
- To make or edit a comment for a selected item, click on the icon.
- To refresh a window, click on the icon.
- To undo an action, click on the icon above the main menu.
- To redo an action, click on the icon above the main menu.
- To logout from the Winbox Console, click on the icon.

Troubleshooting for Winbox Console

- *I cannot open the Winbox Console*
  - Check the port and address for **www** service in **/ip service print** list. Make sure the address you are connecting from matches the network you've specified in **address** field and that you've specified the correct port in the Winbox loader. The command **/ip service set www port=80 address=0.0.0.0/0** will change these values to the default ones so you will be able to connect specifying just the correct address of the router in the address field of Winbox loader
  - The Winbox Console uses TCP port 3986 (not secure) or 3987 (secure; requires **security** package to be installed). Make sure you have access to it through the firewall.

Configuring Basic Functions

Working with Interfaces

Before configuring the IP addresses and routes please check the **/interface** menu to see the list of available interfaces. If you have Plug−and−Play cards installed in the router, it is most likely that the device drivers have been loaded for them automatically, and the relevant interfaces appear on the **/interface print** list, for example:

```
[admin@MikroTik] interface> print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE            MTU
  0  R ether1              ether           1500
  1  R ether2              ether           1500
  2  R ether3              ether           1500
  3  R ether4              ether           1500
```

```
  4  R ether5               ether            1500
  5  R sync1                sync             1500
  6  R pc1                  pc               1500
  7  R ether6               ether            1500
  8  R prism1               prism            1500
[admin@MikroTik] interface>
```

The interfaces need to be enabled, if you want to use them for communications. Use the **/interface enable name** command to enable the interface with a given name or number, for example:

```
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
  #   NAME                TYPE             MTU
  0 X  ether1              ether            1500
  0 X  ether2              ether            1500
[admin@MikroTik] interface> enable 0
[admin@MikroTik] interface> enable ether2
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
  #   NAME                MTU   TYPE
  0  R ether1              ether            1500
  0  R ether2              ether            1500
[admin@MikroTik] interface>
```

The interface name can be changed to a more descriptive one by using the **/interface set** command:

```
[admin@MikroTik] interface> set 0 name=Public
[admin@MikroTik] interface> set 1 name=Local
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
  #   NAME                MTU   TYPE
  0  R Public              ether            1500
  0  R Local               ether            1500
[admin@MikroTik] interface>
```

Use of the 'setup' Command

The initial setup of the router can be done by using the **/setup** command which enables an interface, assigns an address/netmask to it, and configures the default route. If you do not use the **setup** command, or need to modify/add the settings for addresses and routes, please follow the steps described below.

Notes

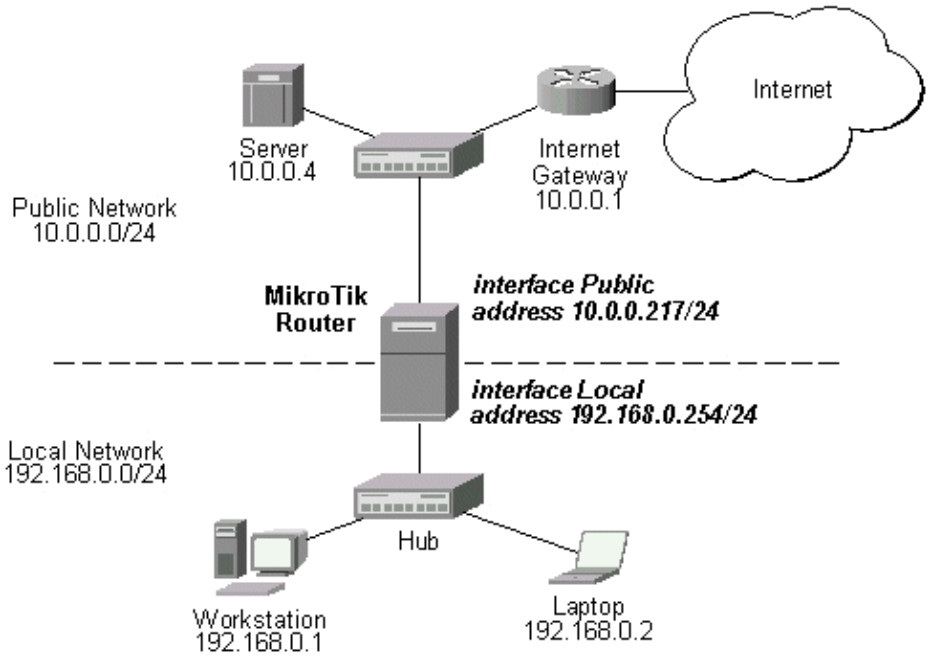
The device drivers for NE2000 compatible ISA cards need to be loaded using the **add** command under the /drivers menu. For example, to load the driver for a card with IO address 0x280 and IRQ 5, it is enough to issue the command:

```
[admin@MikroTik] driver> add name=ne2k-isa io=0x280
[admin@MikroTik] driver> print
Flags: I - invalid, D - dynamic
  #   DRIVER                              IRQ IO       MEMORY   ISDN-PROTOCOL
  0 D RealTek 8139
  1 D Intel EtherExpressPro
  2 D PCI NE2000
  3   ISA NE2000                          280
  4   Moxa C101 Synchronous                            C8000
[admin@MikroTik] driver>
```

There are some other drivers that should be added manually. Please refer to the respective manual sections for the detailed information on how drivers are to be loaded.

Adding Addresses

Assume you need to configure the MikroTik router for the following network setup:



In the current example we use two networks:

- The local LAN with network address 192.168.0.0 and 24−bit netmask 255.255.255.0 The router's address is 192.168.0.254 in this network.
- The ISP's network with address 10.0.0.0 and 24−bit netmask 255.255.255.0 The router's address is 10.0.0.217 in this network.

The addresses can be added and viewed using the following commands:

```
[admin@MikroTik] ip address> add address 10.0.0.217/24 interface Public
[admin@MikroTik] ip address> add address 192.168.0.254/24 interface Local
[admin@MikroTik] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.217/24      10.0.0.217      10.0.0.255      Public
  1   192.168.0.254/24   192.168.0.0     192.168.0.255   Local
[admin@MikroTik] ip address>
```

Here, the network mask has been specified in the value of the address argument. Alternatively, the argument 'netmask' could have been used with the value '255.255.255.0'. The network and broadcast addresses were not specified in the input since they could be calculated automatically.

Notes

Please note that the addresses assigned to different interfaces of the router should belong to different networks.

Configuring the Default Route

You can see two dynamic (D) and connected (C) routes, which have been added automatically when the addresses were added in the example above:

```
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0 DC 192.168.0.0/24     r 0.0.0.0         0        Local
    1 DC 10.0.0.0/24        r 0.0.0.0         0        Public
[admin@MikroTik] ip route> print detail
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    0 DC dst−address=192.168.0.0/24 preferred−source=192.168.0.254
         gateway=0.0.0.0 gateway−state=reachable distance=0 interface=Local

    1 DC dst−address=10.0.0.0/24 preferred−source=10.0.0.217 gateway=0.0.0.0
         gateway−state=reachable distance=0 interface=Public

[admin@MikroTik] ip route>
```

These routes show, that IP packets with destination to 10.0.0.0/24 would be sent through the interface Public, whereas IP packets with destination to 192.168.0.0/24 would be sent through the interface Local. However, you need to specify where the router should forward packets, which have destination other than networks connected directly to the router.

Example

In the following example the **default route** (destination 0.0.0.0, netmask 0.0.0.0) will be added. In this case it is the ISP's gateway 10.0.0.1, which can be reached through the interface Public:

```
[admin@MikroTik] ip route> add gateway=10.0.0.1
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        Public
    1 DC 192.168.0.0/24     r 0.0.0.0         0        Local
    2 DC 10.0.0.0/24        r 0.0.0.0         0        Public
[admin@MikroTik] ip route>
```

Here, the default route is listed under #0. As we see, the gateway 10.0.0.1 can be reached through the interface 'Public'. If the gateway was specified incorrectly, the value for the argument 'interface' would be unknown.

Notes

You cannot add two routes to the same destination, i.e., destination−address/netmask! It applies to the default routes as well. Instead, you can enter multiple gateways for one destination. For more information

on IP routes, please read the relevant topic in the Manual.

If you have added an unwanted static route accidentally, use the **remove** command to delete the unneeded one. You will not be able to delete dynamic (DC) routes. They are added automatically and represent routes to the networks the router connected directly.

Testing the Network Connectivity

From now on, the **/ping** command can be used to test the network connectivity on both interfaces. You can reach any host on both connected networks from the router.

Example

In the example below it's seen, hows does **ping** command work:

```
[admin@MikroTik] ip route> /ping 10.0.0.4
10.0.0.4 64 byte ping: ttl=255 time=7 ms
10.0.0.4 64 byte ping: ttl=255 time=5 ms
10.0.0.4 64 byte ping: ttl=255 time=5 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 5/5.6/7 ms
[admin@MikroTik] ip route>
[admin@MikroTik] ip route> /ping 192.168.0.1
192.168.0.1 64 byte ping: ttl=255 time=1 ms
192.168.0.1 64 byte ping: ttl=255 time=1 ms
192.168.0.1 64 byte ping: ttl=255 time=1 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1/1.0/1 ms
[admin@MikroTik] ip route>
```

The workstation and the laptop can reach (ping) the router at its local address 192.168.0.254, If the router's address 192.168.0.254 is specified as the default gateway in the TCP/IP configuration of both the workstation and the laptop, then you should be able to ping the router:

```
C:\>ping 192.168.0.254
Reply from 192.168.0.254: bytes=32 time=10ms TTL=253
Reply from 192.168.0.254: bytes=32 time<10ms TTL=253
Reply from 192.168.0.254: bytes=32 time<10ms TTL=253

C:\>ping 10.0.0.217
Reply from 10.0.0.217: bytes=32 time=10ms TTL=253
Reply from 10.0.0.217: bytes=32 time<10ms TTL=253
Reply from 10.0.0.217: bytes=32 time<10ms TTL=253

C:\>ping 10.0.0.4
Request timed out.
Request timed out.
Request timed out.

C:\>
```

Notes

You cannot access anything beyond the router (network 10.0.0.0/24 and the Internet), unless you do the one of the following:

- Use source network address translation (masquerading) on the MikroTik router to 'hide' your private LAN 192.168.0.0/24 (see the information below), or
- Add a static route on the ISP's gateway 10.0.0.1, which specifies the host 10.0.0.217 as the gateway to network 192.168.0.0/24. Then all hosts on the ISP's network, including the server, will be able to communicate with the hosts on the LAN.

To set up routing, it is required that you have some knowledge of configuring TCP/IP networks. There is a comprehensive list of IP resources compiled by Uri Raz at http://www.private.org.il/tcpip_rl.html We strongly recommend that you obtain more knowledge, if you have difficulties configuring your network setups.

Application Examples

Next will be discussed situation with 'hiding' the private LAN 192.168.0.0/24 'behind' one address 10.0.0.217 given to you by the ISP.

Application Example with Masquerading

If you want to 'hide' the private LAN 192.168.0.0/24 'behind' one address 10.0.0.217 given to you by the ISP, you should use the source network address translation (masquerading) feature of the MikroTik router. Masquerading is useful, if you want to access the ISP's network and the Internet appearing as all requests coming from the host 10.0.0.217 of the ISP's network. The masquerading will change the source IP address and port of the packets originated from the network 192.168.0.0/24 to the address 10.0.0.217 of the router when the packet is routed through it.

Masquerading conserves the number of global IP addresses required and it lets the whole network use a single IP address in its communication with the world.

To use masquerading, a source NAT rule with action 'masquerade' should be added to the firewall configuration:

```
[admin@MikroTik] ip firewall src-nat> add action=masquerade out-interface=Public
[admin@MikroTik] ip firewall src-nat> print
Flags: X - disabled, I - invalid, D - dynamic
  0   src-address=0.0.0.0/0:0-65535 dst-address=0.0.0.0/0:0-65535
      out-interface=Public protocol=all icmp-options=any:any flow=""
      connection="" content="" limit-count=0 limit-burst=0 limit-time=0s
      action=masquerade to-src-address=0.0.0.0 to-src-port=0-65535

[admin@MikroTik] ip firewall src-nat>
```

Notes

Please consult the **Firewall Manual** for more information on masquerading.

Application Example with Bandwidth Management

Mikrotik RouterOS™ V2.7 offers extensive queue management.

Assume you want to limit the bandwidth to 128kbps on downloads and 64kbps on uploads for all hosts on the LAN. Bandwidth limitation is done by applying queues for outgoing interfaces regarding the traffic

flow. It is enough to add two queues at the MikroTik router:

```
[admin@MikroTik] queue simple> add interface=Local max-limit=128000
[admin@MikroTik] queue simple> add interface=Public max-limit=64000
[admin@MikroTik] queue simple> print
Flags: X - disabled, I - invalid, D - dynamic
  0   name="queue1" src-address=0.0.0.0/0 dst-address=0.0.0.0/0
      interface=Local limit-at=0 queue=default priority=8 max-limit=128000

  1   name="queue2" src-address=0.0.0.0/0 dst-address=0.0.0.0/0
      interface=Public limit-at=0 queue=default priority=8 max-limit=64000

[admin@MikroTik] queue simple>
```

Leave all other parameters as set by default. The limit is approximately 128kbps going to the LAN (download) and 64kbps leaving the client's LAN (upload).

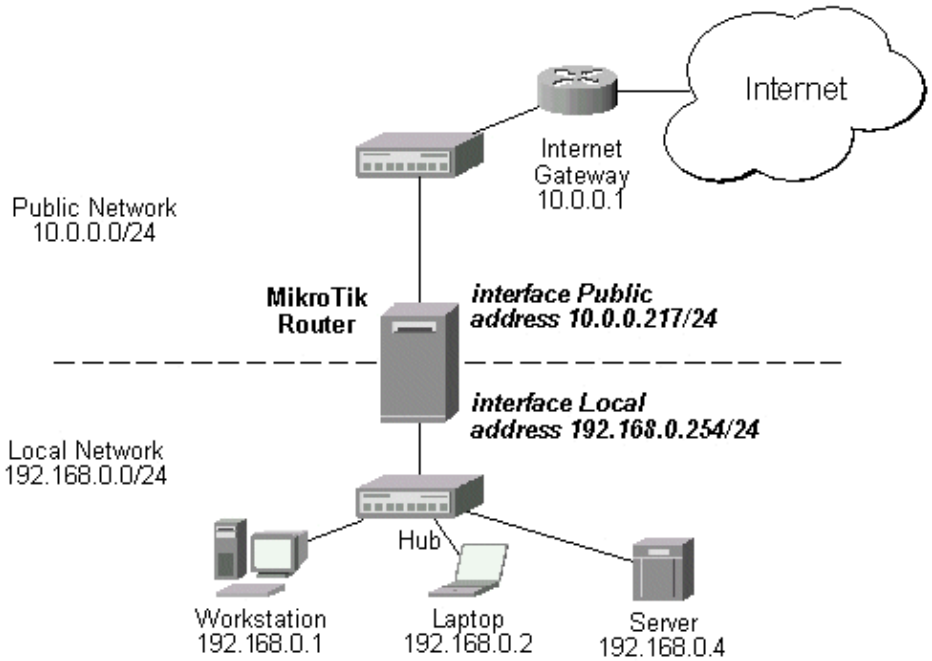
Notes

The queues have been added for the outgoing interfaces regarding the traffic flow.

Please consult the **Queues Manual** for more information on bandwidth management and queuing.

Application Example with NAT

Assume we have moved the server in our previous examples from the public network to our local one:



The server'would have been s address now is 192.168.0.4, and we are running web server on it that listens to the TCP port 80. We want to make it accessible from the Internet at address:port 10.0.0.217:80. This can be done by means of Static Network Address translation (NAT) at the MikroTik Router. The Public address:port 10.0.0.217:80 will be translated to the Local address:port 192.168.0.4:80. One destination NAT rule is required for translating the destination address and port:

```
[admin@MikroTik] ip firewall dst-nat> add action=nat protocol=tcp \
dst-address=10.0.0.217/32:80 to-dst-address=192.168.0.4
[admin@MikroTik] ip firewall dst-nat> print
Flags: X - disabled, I - invalid, D - dynamic
  0   src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=10.1.0.217/32:80 protocol=tcp icmp-options=any:any flow=""
      src-mac-address=00:00:00:00:00:00 limit-count=0 limit-burst=0
      limit-time=0s action=nat to-dst-address=192.168.0.4 to-dst-port=0-65535

[admin@MikroTik] ip firewall dst-nat>
```

Notes

Please consult the **Firewall Manual** for more information on NAT.

Terminal Console Manual

Document revision 1.1 (29−Jan−2003)
This document applies to the MikroTik RouterOS v2.7

Table of Contents

Summary

The Terminal Console is used for accessing the MikroTik Router configuration and management features using text terminals, i.e., remote terminal clients, as well as local monitor and keyboard. The Terminal Console is used for writing scripts. This manual describes the general console operation principles. Please consult the Scripting Manual on some advanced console commands and on how to write scripts.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : *None*
Protocols utilized : *None*
Hardware usage: *not significant*

Related Documents

Scripting Manual

Overview of Common Functions

The console allows configuration of the router settings using text commands. The command structure is similar to the Unix shell. Since there's a lot of available commands, they're split into hierarchy. For example, all (well, almost all) commands that work with routes start with **ip route**:

```
[admin@MikroTik] > ip route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        ether6
                            r 192.168.1.254            ether4
    1 DC 192.168.1.0/24     r 0.0.0.0         0        ether4
    2 DC 10.10.10.0/24      r 0.0.0.0         0        prism1
    3 DC 10.0.0.0/24        r 0.0.0.0         0        ether6
[admin@MikroTik] > ip route set 0 gateway=10.0.0.1
[admin@MikroTik] > ip route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        ether6
    1 DC 192.168.1.0/24     r 0.0.0.0         0        ether4
    2 DC 10.10.10.0/24      r 0.0.0.0         0        prism1
    3 DC 10.0.0.0/24        r 0.0.0.0         0        ether6
[admin@MikroTik] >
```

Instead of typing **ip route** before each command, **ip route** can be typed once to "change into" that particular branch of command hierarchy. Thus, the example above could also be executed like this:

```
[admin@MikroTik] > ip route
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        ether6
    1 DC 192.168.1.0/24     r 0.0.0.0         0        ether4
    2 DC 10.10.10.0/24      r 0.0.0.0         0        prism1
    3 DC 10.0.0.0/24        r 0.0.0.0         0        ether6
[admin@MikroTik] ip route>
```

Notice that prompt changes to show where in the command hierarchy you are located at the moment. To change to top level, type /

```
[admin@MikroTik] ip route> /
[admin@MikroTik] >
```

To move up one command level, type **..**

```
[admin@MikroTik] ip route> ..
[admin@MikroTik] ip>
```

You can also use / and **..** to execute commands from other levels without changing the current level:

```
[admin@MikroTik] ip route> /ping 10.0.0.10
10.0.0.10 64 byte pong: ttl=128 time=5 ms
10.0.0.10 64 byte pong: ttl=128 time=6 ms
2 packets transmitted, 2 packets received, 0% packet loss
```

```
round−trip min/avg/max = 5/5.5/6 ms
[admin@MikroTik] ip route>
```

Or alternatively, to go back to the base level you could use multiple **..** commands:

```
[admin@MikroTik] ip route> .. .. ping 10.0.0.10
10.0.0.10 64 byte pong: ttl=128 time=8 ms
10.0.0.10 64 byte pong: ttl=128 time=6 ms
2 packets transmitted, 2 packets received, 0% packet loss
round−trip min/avg/max = 6/7.0/8 ms
[admin@MikroTik] ip route>
```

Lists

Many of the command levels operate with arrays of items: interfaces, routes, users etc. Such arrays are displayed in similarly looking lists. All items in the list have an item number followed by its parameter values. For example:

```
[admin@MikroTik] > interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE            MTU
  0  R ether1              ether           1500
  1  R ether2              ether           1500
  2  R ether3              ether           1500
  3  R ether4              ether           1500
  4  R prism1              prism           1500
[admin@MikroTik] >
```

To change parameters of an item (interface settings in this particular case), you have to specify it's number to the **set** command:

```
[admin@MikroTik] interface> set 0 mtu=1460
[admin@MikroTik] interface> print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE            MTU
  0  R ether1              ether           1460
  1  R ether2              ether           1500
  2  R ether3              ether           1500
  3  R ether4              ether           1500
  4  R prism1              prism           1500
[admin@MikroTik] interface>
```

Numbers are assigned by **print** command and are not constant − it is possible that two successive **print** commands will order items differently. But the results of last **print** commands are memorized and, thus, once assigned item numbers can be used even after **add**, **remove** and **move** operations (after **move** operations, item numbers are moved with the items). Item numbers are assigned for sessions, they will remain the same until you quit the console or until the next **print** command is executed. Also, numbers are assigned separately for every item list, so **ip address print** won't change numbers for interface list.

Let's assume **interface prism print** hasn't been executed in this session. In this case:

```
[admin@MikroTik] interface> prism set 0 ssid=mt
ERROR: item numbers not assigned
```

Console is telling that there has been no **interface prism print** command, and thus, it cannot (and also you)

know which PRISM interface number 0 corresponds to.

To understand better how do item numbers work, you can play with **from** argument of **print** commands:

```
[admin@MikroTik] interface> print from=1
Flags: X - disabled, D - dynamic, R - running
  #    NAME                TYPE            MTU
  0  R ether2               ether           1500
[admin@MikroTik] interface>
```

The **from** argument specifies what items to show. Numbers are assigned by every **print** command, thus, after executing command above there will be only one item accessible by number − interface **ether2** with number 0.

Item Names

Some lists have items that have specific names assigned to each. Examples are **interface** or **user** levels. There you can use item names instead of numbers:

```
[admin@MikroTik] interface> set prism1 mtu=1460
```

You don't have to use the **print** command before accessing items by name. As opposed to numbers, names are not assigned by the console internally, but are one of the items' properties. Thus, they won't change on their own. However, there are all kinds of obscure situations possible when several users are changing router configuration at the same time. Generally, item names are more "stable" than numbers, and also more informative, so you should prefer them to numbers when writing console scripts.

Quick Typing

There are two features in router console that help entering commands much quicker and easier − the [TAB] key completions, and abbreviations of command names. Completions work similarly to the bash shell in UNIX. If you press the [TAB] key after part of a word, console tries to find the command in current context that begins with this word. If there's only one match, it is automatically appended, followed by space character:

```
/inte[TAB]_ becomes /interface _
```

Here, "_" is the cursor position. And [TAB] is pressed TAB key, not '[TAB]' character sequence.

If there's more than one match, but they all have a common beginning, which is longer than that what you have typed, then the word is completed to this common part, and no space is appended:

```
/interface set e[TAB]_
```

becomes

```
/interface set ether_
```

because "e" matches both "ether5" and "ether1" in this example

If you've typed just the common part, pressing the tab key once has no effect. However, pressing it for the second time shows all possible completions in compact form:

```
[admin@MikroTik] > interface set e[TAB]_
[admin@MikroTik] > interface set ether[TAB]_
[admin@MikroTik] > interface set ether[TAB]_
ether1 ether5
[admin@MikroTik] > interface set ether_
```

The tab key can be used almost in any context where the console might have a clue about possible values − command names, argument names, arguments that have only several possible values (like names of items in some lists or name of protocol in firewall and NAT rules).You can't complete numbers, IP addresses and similar values.

Note that pressing [TAB] key while entering IP address will do a DNS lookup, instead of completion. If what is typed before cursor is a valid IP address, it will be resolved to a DNS name (reverse resolve), otherwise it will be resolved directly (i.e. to an IP address). To use this feature, DNS server must be configured and working. To avoid input lockups any such lookup will timeout after half a second, so you might have to press [TAB] several times, before name is actually resolved

It is possible to complete not only beginning, but also any distinctive substring of name: if there is no exact match, console starts looking for words that have string being completed as first letters of a multiple word name, or that simply contain letters of this string in the same order. If single such word is found, it is completed at cursor position. For example:

```
[admin@MikroTik] > interface x[TAB]_
[admin@MikroTik] > interface export _
```

x is completed to **export**, because no other word in this context contains 'x'.

```
[admin@MikroTik] > interface mt[TAB]_
[admin@MikroTik] > interface monitor-traffic _
```

No word begins with letters 'mt', but it is an abbreviation of **monitor−traffic**.

Another way to press fewer keys while typing is to abbreviate command and argument names. You can type only beginning of command name, and, if it is not ambiguous, console will accept it as a full name. So typing:

```
[admin@MikroTik] > pi 10.1 c 3 s 100

equals to:

[admin@MikroTik] > ping 10.0.0.1 count 3 size 100
```

Help

The console has a built−in help, which can be accessed by typing '?'. General rule is that help shows what you can type in position where the '?' was pressed (similarly to pressing tab key twice, but in verbose form and with explanations).

Internal Item numbers

Items can also be addressed by their internal numbers. These numbers are generated by console for scripting purposes and, as the name implies, are used internally. Although you can see them if you print return values of some commands (internal numbers look like hex number preceeded by '*' − for example "*100A"), there's no reason for you to type them in manually.

Note: As an implication of internal number format, you should not use item names that begin with asterisk (*).

Multiple Items

You can specify multiple items as targets of some commands. Almost everywhere, where you can write the number of items, you can also write a list of numbers:

```
[admin@MikroTik] > interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE            MTU
  0  R ether1               ether           1500
  1  R ether2               ether           1500
  2  R ether3               ether           1500
  3  R ether4               ether           1500
[admin@MikroTik] > interface set 0,1,2 mtu=1460
[admin@MikroTik] > interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE            MTU
  0  R ether1               ether           1460
  1  R ether2               ether           1460
  2  R ether3               ether           1460
  3  R ether4               ether           1500
[admin@MikroTik] >
```

This is handy when you want to perform same action on several items, or do a selective export. However, this feature becomes really useful when combined with scripting.

General Commands

Most command groups have some or all of these commands: **print**, **set**, **remove**, **add**, **find**, **get**, **export**, **enable**, **disable**, **comment**, **move**. These commands have similar behavior in all hierarchy.

print

The **print** command shows all information that's accessible from particular command level. Thus, **/system clock print** shows system date and time, **/ip route print** shows all routes etc. If there's a list of items in this level and they are not read−only, i.e. you can change/remove them (example of read−only item list is **/system history**, which shows history of executed actions), then **print** command also assigns numbers that are used by all commands that operate on items in this list.

If there's list of items then **print** usually can have a **from** argument. The **from** argument accepts space separated list of item numbers, names (if items have them), and internal numbers. The action (printing) is performed on all items in this list in the same order in which they're given.

Output can be formatted either as a table, with one item per line or as a list with **property=value** pairs for each item. By default **print** uses one of these forms, but it can be set explicitly with **brief** and **detail** arguments. In **brief** (table) form, **column** argument can be set to a list of property names that should be shown in the table:

```
[admin@MikroTik] interface ethernet> print
Flags: X − disabled, R − running
  #    NAME                MTU   MAC−ADDRESS       ARP
```

```
  0  R ether1               1460  00:50:08:00:00:F5 enabled
  1  R ether2               1460  00:50:08:00:00:F6 enabled
[admin@MikroTik] interface ethernet> print brief
Flags: X - disabled, R - running
  #    NAME                MTU   MAC-ADDRESS       ARP
  0  R ether1               1460  00:50:08:00:00:F5 enabled
  1  R ether2               1460  00:50:08:00:00:F6 enabled
[admin@MikroTik] interface ethernet> print detail
Flags: X - disabled, R - running
  0  R name="ether1" mtu=1460 mac-address=00:50:08:00:00:F5 arp=enabled
       disable-running-check=yes

  1  R name="ether2" mtu=1460 mac-address=00:50:08:00:00:F6 arp=enabled
       disable-running-check=yes


[admin@MikroTik] interface ethernet> print brief column=mtu,arp
Flags: X - disabled, R - running
  #    MTU   ARP
  0  R 1460  enabled
  1  R 1460  enabled
[admin@MikroTik] interface ethernet> print
```

Rules that do some accounting (for example, **ip firewall** or **queue** rules) may have two additional views of packets and of bytes matched these rules:

```
[admin@MikroTik] ip firewall rule forward> print packets
Flags: X - disabled, I - invalid
  #   SRC-ADDRESS                   DST-ADDRESS                   PACKETS
  0   0.0.0.0/0:0-65535             0.0.0.0/0:0-65535             0
[admin@MikroTik] ip firewall rule forward> print bytes
Flags: X - disabled, I - invalid
  #   SRC-ADDRESS                   DST-ADDRESS                   BYTES
  0   0.0.0.0/0:0-65535             0.0.0.0/0:0-65535             0
[admin@MikroTik] ip firewall rule forward>
```

To reset these counters **reset-counters** command is used.

Some items might have statistics other than matched **bytes** and **packets**. You can see it by using **print stats** command:

```
[admin@MikroTik] ip ipsec> policy print stats
Flags: X - disabled, I - invalid
  0   src-address=10.0.0.205/32:any dst-address=10.0.0.201/32:any
      protocol=icmp ph2-state=no-phase2 in-accepted=0 in-dropped=0
      out-accepted=0 out-dropped=0 encrypted=0 not-encrypted=0 decrypted=0
      not-decrypted=0


[admin@MikroTik] ip ipsec>
```

There migtht also be **print status** command:

```
[admin@MikroTik] routing bgp peer> print status
  # REMOTE-ADDRESS  REMOTE-AS STATE          ROUTES-RECEIVED
  0 159.148.42.158  2588      connected      1
[admin@MikroTik] routing bgp>
```

Normally, the **print** command pauses after the screen is full and asks whether to continue or not. Press any key other from [Q] or [q] to continue printing.

The **without−paging** argument suppresses prompting after each screen of output.

You can specify interval for repeating the command until [Ctrl]+[C] is pressed. Thus, you do not need to repeatedly press the [Up−Arrow] and [Enter] buttons to see repeated printouts of a changing list you want to monitor. Instead, you use the argument **interval=2s** for **print**.

The other useful parameter is **count−only**, that shows the total number of items in the table.

```
[admin@MikroTik] interface> print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ether1               ether            1460
  1  R ether2               ether            1460
  2  R ether3               ether            1460
  3  R ether4               ether            1500
[admin@MikroTik] interface> print count−only
4
[admin@MikroTik] interface>
```

set

The **set** command allows you to change values of general parameters or item parameters. The **set** command has arguments with names corresponding to values you can change. Use **?** or double [TAB] to see list of all arguments. If there is list of items in this command level, then set has one unnamed argument that accepts the number of item (or list of numbers) you wish to set up. **set** does not return anything.

Examples are given above.

add

The **add** command usually has the same arguments as **set**, minus the unnamed number argument. It adds new item with values you've specified, usually to the end of list (in places where order is relevant). There are some values that you have to supply (like interface for new route), and other values that are set to defaults if you don't supply them. The **add** command returns internal number of item it has added.

You can create a copy of an existing item by using **copy−from** argument. It takes default values of new item's properties from another item. If you don't want exact copy, you can specify new values for some properties. When copying items that have names, you will usually have to give new name to a copy.

You can place a new item before an existing item by using **place−before** argument. Thus, you do not need to use the **move** command after adding an item to the list. You can control disabled/enabled state of new items by using **disabled** argument, if present. You can supply description for new item using **comment** argument, if present:

```
[admin@MikroTik] ip route> set 0 comment="our default gateway"
[admin@MikroTik] ip route> set 1 comment="wireless network gateway"
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
```

```
    0  S ;;; our default gateway
         0.0.0.0/0          r 10.0.0.1        1        ether6
    1  S ;;; wireless network gateway
         10.100.0.0/16      r 10.0.0.254      1        ether6
    2 DC 192.168.1.0/24     r 0.0.0.0         0        ether4
    3 DC 10.10.10.0/24      r 0.0.0.0         0        prism1
[admin@MikroTik] ip route>
```

remove

The **remove** command has one unnamed argument, which contains number(s) or name(s) of item(s) to remove.

```
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S ;;; our default gateway
         0.0.0.0/0          r 10.0.0.1        1        ether6
    1  S ;;; wireless network gateway
         10.100.0.0/16      r 10.0.0.254      1        ether6
    2 DC 192.168.1.0/24     r 0.0.0.0         0        ether4
    3 DC 10.10.10.0/24      r 0.0.0.0         0        prism1
[admin@MikroTik] ip route> remove 0
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S ;;; wireless network gateway
         10.100.0.0/16      r 10.0.0.254      1        ether6
    1 DC 192.168.1.0/24     r 0.0.0.0         0        ether4
    2 DC 10.10.10.0/24      r 0.0.0.0         0        prism1
[admin@MikroTik] ip route>
```

move

If the order of items is relevant, command level will also contain **move** command. First argument is a list of items, whose order will be changed, second argument specifies item before which to place all items being moved (they are placed at the end of the list if second argument is not given). Item numbers after **move** command are left in a consistent, but hardly intuitive order, so it's better to resync by using **print** after each **move** command.

```
[admin@MikroTik] ip firewall mangle> print brief
Flags: X − disabled, I − invalid, D − dynamic
  #   SRC−ADDRESS                    DST−ADDRESS
  0   0.0.0.0/0:80                   0.0.0.0/0:0−65535
  1   1.1.1.1/32:80                  0.0.0.0/0:0−65535
  2   2.2.2.2/32:80                  0.0.0.0/0:0−65535
  3   3.3.3.3/32:80                  0.0.0.0/0:0−65535
[admin@MikroTik] ip firewall mangle> move 0
[admin@MikroTik] ip firewall mangle> print brief
Flags: X − disabled, I − invalid, D − dynamic
  #   SRC−ADDRESS                    DST−ADDRESS
  0   1.1.1.1/32:80                  0.0.0.0/0:0−65535
  1   2.2.2.2/32:80                  0.0.0.0/0:0−65535
  2   3.3.3.3/32:80                  0.0.0.0/0:0−65535
  3   0.0.0.0/0:80                   0.0.0.0/0:0−65535
[admin@MikroTik] ip firewall mangle> move 0 2
```

```
[admin@MikroTik] ip firewall mangle> print brief
Flags: X − disabled, I − invalid, D − dynamic
  #   SRC−ADDRESS                     DST−ADDRESS
  0   2.2.2.2/32:80                   0.0.0.0/0:0−65535
  1   3.3.3.3/32:80                   0.0.0.0/0:0−65535
  2   1.1.1.1/32:80                   0.0.0.0/0:0−65535
  3   0.0.0.0/0:80                    0.0.0.0/0:0−65535
[admin@MikroTik] ip firewall mangle> move 3,2,0 0
[admin@MikroTik] ip firewall mangle> print brief
Flags: X − disabled, I − invalid, D − dynamic
  #   SRC−ADDRESS                     DST−ADDRESS
  0   0.0.0.0/0:80                    0.0.0.0/0:0−65535
  1   1.1.1.1/32:80                   0.0.0.0/0:0−65535
  2   2.2.2.2/32:80                   0.0.0.0/0:0−65535
  3   3.3.3.3/32:80                   0.0.0.0/0:0−65535
[admin@MikroTik] ip firewall mangle>
```

find

The **find** command has the same arguments as **set**, and an additional **from** argument which works like the **from** argument with the **print** command. Plus, **find** command has flag arguments like **disabled**, **invalid** that take values **yes** or **no** depending on the value of respective flag. To see all flags and their names, look at the top of **print** command's output. The **find** command returns internal numbers of all items that have the same values of arguments as specified.

```
[admin@MikroTik] interface> print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ether1              ether            1500
  1  R ipip1               ipip             1480
  2  R eoip−tunnel1        eoip−tunnel      1500

[admin@MikroTik] interface> print from=1
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ipip1               ipip             1480

[admin@MikroTik] interface> print from=[find mtu=1500]
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ether1              ether            1500
  1  R eoip−tunnel1        eoip−tunnel      1500

[admin@MikroTik] interface> print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ether1              ether            1500
  1  R ipip1               ipip             1480
  2  R eoip−tunnel1        eoip−tunnel      1500

[admin@MikroTik] interface> print from=[find mtu=1500 from=0,1]
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ether1              ether            1500

[admin@MikroTik] interface>
```

export

The **export** command prints a script that can be used to restore configuration. If it has the argument **from**, then it is possible to export only specified items. Also, if the **from** argument is given, **export** does not descend recursively through the command hierarchy. The **export** command also has the argument **file**, which allows you to save the script in file on router to retrieve it later via ftp. **Note** that it is not possible to bring back router configuration after reset just from the export scripts. Some important things like interface name assignment, or user passwords just cannot be saved in export script. To back up all configuration, use **/system backup save** command.

enable/disable

You can enable/disable some items (like ip address or default route). If an item is disabled, it is marked with the **X** flag. If an item is invalid, but not disabled, it is marked with the **I** flag. All such flags, if any, are described at the top of the **print** command's output.

```
[admin@MikroTik] > ip route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        ether6
    1 DC 192.168.1.0/24     r 0.0.0.0         0        ether4
    2 DC 10.10.10.0/24      r 0.0.0.0         0        prism1
    3 DC 10.0.0.0/24        r 0.0.0.0         0        ether6
[admin@MikroTik] >
```

Safe Mode

It is possible to change router configuration in a way that will make it not accessible except from local console. Usually this is done by accident, but there is no way to undo last change when connection to router is already cut. Safe mode can be used to minimize such risk.

Safe mode is entered by pressing [Ctrl]+[X]. To quit safe mode, press [Ctrl]+[X] again.

```
[admin@MikroTik] ip firewall rule input> [Ctrl]+[X]
[Safe Mode taken]
[admin@MikroTik] ip firewall rule input<SAFE>
```

Message **Safe Mode taken** is displayed and prompt changes to show that session is now in safe mode. All configuration changes that are made (also from other login sessions), while router is in safe mode, are automatically undone if safe mode session terminates abnormally. You can see all such changes that will be automatically undone tagged with an **F** flag in system history:

```
[admin@MikroTik] ip firewall rule input<SAFE> add
[admin@MikroTik] ip firewall rule input<SAFE> /system history print
Flags: U − undoable, R − redoable, F − floating−undo
   ACTION              BY            POLICY
 F rule added          admin         write
[admin@MikroTik] ip firewall rule input<SAFE>
```

Now, if telnet connection is cut, then after a while (TCP timeout is 9 minutes) all changes that were made while in safe mode will be undone. Exiting session by [Ctrl]+[D] also undoes all safe mode changes, while **/quit** doesn't.

If another user tries to enter safe mode, he's given following message:

```
[admin@MikroTik] >
Hijacking Safe Mode from someone − unroll/release/don't take it [u/r/d]:
```

- Pressing [u] will undo all safe mode changes, and put the current session in safe mode.
- Pressing [r] will keep all current safe mode changes, and put current session in a safe mode. Previous owner of safe mode is notified about this:

  ```
  [admin@MikroTik] ip firewall rule input
  [Safe mode released by another user]
  ```
- Pressing [d] will leave everything as−is.

If too many changes are made while in safe mode, and there's no room in history to hold them all (currently history keeps up to 100 most recent actions), then session is automatically put out of the safe mode, no changes are automatically undone. Thus, it is best to change configuration in small steps, while in safe mode. Pressing [Ctrl]+[X] twice is an easy way to empty safe mode action list.

Software Package Management

Document revision 1.3 (06−Sep−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The MikroTik RouterOS is distributed in the form of software packages. The basic functionality of the router and the operating system itself is provided by the **system** software package. Other packages contain additional software features as well as support to various network interface cards (NICs).

Specifications

Packages required : *None*
License required : *Any*
Home menu level : ***/system package***
Standards and Technologies : *FTP (RFC 959)*
Hardware usage : *not significant*

Additional Documents

Basic Setup Guide
Device Driver Management
License management

Features

The modular software package system of MikroTik RouterOS has the following features:

- Ability to add RouterOS functions by installing additional software packages

- Optimal usage of the storage space by the modular/compressed system
- Unused software packages can be uninstalled
- The RouterOS functions and software can be easily upgraded
- Multiple packages can be installed at once
- The package dependency is checked before installing a software package. The package will not be installed, if the required software package is missing
- The version of the software package should be the same as that of the system package
- The packages can be uploaded on the router using ftp and installed only when the router is going for shutdown during the reboot process.
- If the software package file can be uploaded to the router, then the disk space is sufficient for the installation of the package

Software Package Installation (Upgrade)

Description

Installation of the MikroTik RouterOS software packages can be done by uploading the newer version of the package to the router and rebooting it.

The software package files are compressed binary files, which can be downloaded from the MikroTik's web page www.mikrotik.com download section. The full name of the package file consists of a descriptive name, version number, and file extension '.npk'. For example, **system−2.7rc4.npk**, **ppp−2.7rc4.npk**, etc.

You should check the available hard disk space prior to downloading the package file by issuing **/system resource print** command. If there is not enough free disk space for storing the upgrade packages, it can be freed up by uninstalling some software packages, which provide functionality not required for your needs. If you have a sufficient amount of free space for storing the upgrade packages, connect to the router using ftp. Use user name and password of a user with full access privileges.

- Select the BINARY mode file transfer.
- Upload the software package files to the router and disconnect (**Note** that the packages uploaded should retain the original name and also be in lowercase)
- View the information about the uploaded software packages using the **/file print** command.
- Reboot the router by issuing the **/system reboot** command or by pressing **Ctrl+Alt+Del** keys at the router's console.

After successful installation the software packages installed can be viewed using **/system package print** command.

Notes

The installation/upgrade process is shown on the console screen (monitor) attached to the router.

The Free Demo License do not allow software upgrades using ftp. You should use complete reinstall from floppies, or purchase the license.

Before upgrading the router, please check the current version of the system package and of the additional software packages. The versions of additional packages should match the version number of the system software package. The version of the MikroTik RouterOS system software (and the build number) are shown before the console login prompt. Information about the version numbers and build time of the installed MikroTik RouterOS software packages can be obtained using the **/system package print** command, for

example:

```
[admin@MikroTik] system license> .. package print
Flags: I – invalid
  #   NAME                  VERSION             BUILD-TIME           UNINSTALL
  0   web-proxy             2.7.11              sep/04/2003 17:22:32 no
  1   ppp                   2.7.11              sep/04/2003 17:18:26 no
  2   dhcp                  2.7.11              sep/04/2003 17:13:37 no
  3   telephony             2.7.11              sep/04/2003 17:51:46 no
  4   system                2.7.11              sep/05/2003 13:17:40 no
  5   routing               2.7.11              sep/04/2003 17:20:20 no
  6   security              2.7.11              sep/04/2003 17:12:36 no
  7   advanced-tools        2.7.11              sep/04/2003 17:09:35 no
  8   ntp                   2.7.11              sep/04/2003 17:52:46 no
  9   dns-cache             2.7.11              sep/04/2003 17:20:49 no

[admin@MikroTik] system license>
```

The list shows the number, name, version, and build time of the installed software packages. If the functions provided by a software package are not required for the router implementation, the package can be scheduled for uninstallation at the next shutdown/reboot of the router. Use the **/system package set** command to mark the packages for uninstallation.

If a package is marked for uninstallation, but it is required for another (dependent) package, then the marked package cannot be uninstalled. You should uninstall the dependent package too. For package dependencies see the section about contents of the software packages below. The system package will not be uninstalled even if marked for uninstallation.

Software Package List

System Software Package

The **system** software package provides the basic functionality of the MikroTik RouterOS, namely:

- IP address, ARP, static IP routing, policy routing, firewall (packet filtering, content filtering, masquerading, and static NAT), traffic shaping (queues), IP traffic accounting, MikroTik Neighbour Discovery, IP Packet Packing, DNS client settings, IP service (servers)
- Ethernet interfaces
- IP over IP tunnel interfaces (IPIP)
- Ethernet over IP tunnel interfaces (EoIP)
- driver management for Ethernet ISA cards
- serial port management
- local user management
- export and import of router configuration scripts
- backup and restore of the router's configuration
- undo and redo of configuration changes
- network diagnostics tools (ping, traceroute, bandwidth tester, traffic monitor)
- bridge configuration
- system resource management
- package management
- telnet client and server
- local and remote logging facility

It also includes winbox server as well as winbox executable with some plugins

After installing the MikroTik RouterOS, a license should be obtained from MikroTik to enable the basic system functionality.

Additional Software Feature Packages

The table below shows additional software feature packages, the provided functionality, the required prerequisites and additional licenses, if any.

Name	Contents	Prerequisites	Additional License
advanced−tools	Provides email client, pingers, netwatch and other utilities	−	−
arlan	Provides support for DSSS 2.4GHz 2mbps Aironet ISA cards	−	2.4GHz/5GHz Wireless Client
dhcp	Provides DHCP server and client support	−	−
dns−cache	DNS cache	−	−
hotspot	HotSpot gateway	−	any additional license
isdn	Provides ISDN support	ppp	−
lcd	Provides LCD monitor support	−	−
ntp	Provides network time protocol support	−	−
ppp	Provides support for PPP, PPTP, L2TP, PPPoE and ISDN PPP	−	−
radiolan	Provides support for 5.8GHz RadioLAN cards	−	2.4GHz/5GHz Wireless Client
routing	Provides support for RIP, OSPF and BGP4	−	−
security	Provides support for IPSEC, SSH and secure connectivity with WinBox	−	−
synchronous	Provides support for framerelay and Moxa C101, Moxa C512, Farsync, Cyclades PC300 and XPeed synchronous cards	−	Sync and Hotspot
ups	Provides APC Smart Mode UPS support	−	−
web−proxy	HTTP Web proxy package	−	−
wireless		−	

	Provides support for Cisco Aironet cards and PrismII and Atheros wireless stations and APs		2.4GHz/5GHz Wireless Client / 2.4GHz/5GHz Wireless Server (optional)

Software Package Uninstalling

Description

Usually, you do not need to uninstall software packages. However, if you have installed a wrong package, or you need additional free space to install new one, you have to uninstall some unused packages.

Installed software packages can be viewed using **/system package print** command.

In order to uninstall software package, you have to set **uninstall** property for that package to **yes** and reboot the router.

Example

Suppose we need to uninstall **web−proxy** package from a router.

```
[admin@MikroTik] > /system package print
Flags: I − invalid
  #   NAME                 VERSION              BUILD−TIME           UNINSTALL
  0   web−proxy            2.7.11               sep/04/2003 17:22:32 no
  1   ppp                  2.7.11               sep/04/2003 17:18:26 no
  2   dhcp                 2.7.11               sep/04/2003 17:13:37 no
  3   telephony            2.7.11               sep/04/2003 17:51:46 no
  4   system               2.7.11               sep/05/2003 13:17:40 no
  5   routing              2.7.11               sep/04/2003 17:20:20 no
  6   security             2.7.11               sep/04/2003 17:12:36 no
  7   advanced−tools       2.7.11               sep/04/2003 17:09:35 no
  8   ntp                  2.7.11               sep/04/2003 17:52:46 no
  9   dns−cache            2.7.11               sep/04/2003 17:20:49 no

[admin@MikroTik] > /system package set 0 uninstall=yes
[admin@MikroTik] > /system reboot
```

Troubleshooting

- *Is it possible to upgrade from 2.5 to 2.7 without configuration loss?*
  No, you will loose Point−to−Point interface, DHCP and bridge interface settings. Also, you will have to copy all the PPP users in the new location manually. **Please Note** that you should uninstall **telephony** package before the upgrade. After the upgrade you can put it back and you will not loose the configuration.
- *I have Free Demo license for V2.3 of MikroTik RouterOS, and I want to upgrade to V2.7*
  You will need to obtain a new demo license after the upgrade, or purchase the license. It can be done prior the upgrade.
- *Not enough space to upgrade the system package.*
  Uninstall some packages not in use.
- *The system package does not install because of incorrect version.*
  Use system package version that is greater than the currently installed one.

- *Additional packages do not install because of incorrect version of the system package.*
  Upgrade the system package first, then install the additional packages. The packages should be of the same version as your system package.
- *The package file is corrupted after upload.*
  Use BINARY mode for file transfer.
- *The package has been successfully installed and the driver loaded, but there is no interface in the interface list*
  Obtain the required license to enable the functionality of provided by the software package.
- *The package files have been uploaded to the router, but they have not been installed.*
  Reboot the router!
- *The version 2.2 has been upgraded to the version 2.7, but the connection to the router was lost after the reboot. The configuration has been lost.*
  Using the console (monitor and keyboard attached to the router), restore the configuration.

MikroTik RouterOS™ V2.7 Specifications Sheet

Document revision 1.6 (09−Jul−2002)
This document applies to the MikroTik RouterOS™ V2.7

Hardware

CPU and motherboard − advanced 4th generation (core frequency 100MHz or more), 5th generation (Intel Pentium, Cyrix 6X86, AMD K5 or comparable) or newer uniprocessor Intel IA−32 (i386) compatible (multiple processors are not supported);
RAM − minimum 32 MB, maximum 1 GB; 48 MB or more recommended
hard disk/Flash IDE − minimum 32 MB; 48MB or more recommended
for installation time − floppy drive, CD reader or PXE−compatible NIC (depending on installation method), keyboard, monitor

Basic Network Platform

TCP/IP protocol suite

- **Firewall and NAT**
  packet filtering; source and destination NAT; classification by source MAC, IP addresses, ports, protocols, protocol options, interfaces, internal marks, content, matching frequency
- **Routing**
  RIP v1 / v2, OSPF v2, BGP v4; Equal cost multi−path routing; Policy based routing; firewall marked packet routing
- **Bridging**
  spanning tree protocol; multiple bridge interfaces; bridge firewalling
- **Data Rate Management**
  per IP / protocol / subnet / port / firewall mark; HTB, RED, SFQ, byte limited queue, packet limited queue; hierarchical limitation, CIR, MIR, contention ratios
- **Point−to−Point links**
  ISDN dial−out and dial−in; RADIUS authentication/accounting; onboard serial ports; modem pool; PPTP and L2TP encrypted tunnel (VPN); PPTP, PPPoE and L2TP Access Concentrator and client
- **Tunnels**
  IPIP tunnels, EoIP (Ethernet over IP)
- **IPsec**
  IP encryption (IP security)
- **VLAN**
  Virtual LAN support
- **DHCP**
  DHCP server per interface; DHCP client
- **HotSpot**
  HotSpot Gateway with RADIUS authentication/accounting
- **Universal Client**
  Transparent address translation not depending on the client's setup
- **NTP**
  Network Time Protocol server and client

- **Monitoring/Accounting**
  IP traffic accounting, firewall actions logging
- **Tools**
  ping; traceroute; bandwidth test; ping flood; telnet; SSH; packet sniffer
- **DNS client**
  name resolving for local use; Dynamic DNS Client
- **SNMP**
  read−only access

Special Protocols

- **MikroTik Packet Packer Protocol (M3P)**
  For Wireless links and for Ethernet
- **MikroTik Neighbor Discovery Protocol (MNDP)**

Caching Features

- DNS cache
- SQUID caching proxy

Administration

General

History undo / redo / display; multiple administrator connections; safe−mode operations
Real time updates in WinBox GUI; real time configuration

- **Web/GUI**
  Uses GUI tool for remote administration,
  graphing of traffic, statistics and resource monitoring
  multiple internal configuration windows
- **Terminal Console**
  standard keyboard and monitor connection, scripting
  import/export of configuration scripts to screen / file
  command history, hierarchical command structure
  monitoring of interface status and traffic, context specific hints
- **Telnet**
  all terminal console features, SSH option, cut/paste of configuration
- **Serial terminal console**
  all terminal console features
- **System**
  date/time setting, identity setting, upgrade, ftp upload, users, access levels, groups, PPP access, UPS monitoring APC, router safe−mode on power outage, LCD hardware option, 2 X 16 or 4 X 24 character backlit displays, configurable revolving system status / statistics
- **FTP**
  For uploading upgrade packages, uploading and downloading scripts, HotSpot authorization servlet pages.
- **Upgrading**
  Remote upgrading by uploading the software packages to the router

Scripting

Scripts can be scheduled for executing at certain times, periodically, or on events. All Terminal Console commands are supported in scripts.

Hardware Supported

See Device Driver List for more complete supported device list.

Wireless Interfaces

(additional license purchase required)

- **2.4 GHz Wireless 11Mbit/s IEEE802.11b clients**
  Aironet Arlan 655
  Aironet 4800 ISA/PCI/PC
  Cisco 340/352 ISA/PCI/PC
  WaveLAN Bronze/Gold/Silver ISA/PC
- **2.4 GHz Wireless 11Mbit/s IEEE802.11b Access Point and clients**
  Prism II chipset based cards
  Atheros AR5212 chipset based cards
- **2.4 GHz Wireless 54Mbit/s IEEE802.11g Access Point and clients**
  Atheros AR2111 chipset based cards
  Atheros AR5212 chipset based cards
- **5.2 GHz Wireless 54Mbit/s IEEE802.11a Access Points and clients**
  Atheros AR5000/AR5001A chipset series based cards
  Atheros AR5111 chipset based cards
  Atheros AR5212 chipset based cards
- **5.8 GHz 10Mbit/s 10BaseRadio Wireless**
  10Mbps RadioLAN

Synchronous

(additional license purchase required)

- **Protocols**
  PPP Synchronous, HDLC, Cisco HDLC, Frame Relay
- **Synchronous Interfaces**
  Moxa C101 V.35 (4 Mbit/s)
  Moxa C502 PCI 2−port V.35 (8 Mbit/s)
  Cyclades PC−300 V.35 (5 Mbit/s)
  Cyclades PC−300 E1/T1
  FarSync V.35/X.21 (8.448 Mbit/s)

Asynchronous Interfaces

- Standard Communication Ports Com1 and Com2
- Moxa Smartio C104H, C168H, CP−114, CP−132, CP−168U, CP−104U, CP−134U, CP−132U PCI 2/4/8 port up to 4 cards (up to 32 ports)

- Cyclades Cyclom−Y and Cyclades−Z Series up to 32 ports per card, up to 4 cards (up to 128 ports)
- TCL DataBooster 4 or 8 PCI cards

Ethernet Interfaces

Most widely used single and multiport Ethernet interface cards including:

- ISA and PCI NE2000 compatible (most common network cards)
- 3Com 509 Series (3Com EtherLink III ISA)
- 3Com 3c59x/3c90x series
- Intel EtherExpress Pro 100
- Intel PRO/1000 series
- DEC 'Tulip' compatible
- Realtec RTL8139 based
- Winbond w89c840 based
- Davicom DM9102 based
- ADMtek Pegasus based USB adapters
- AMD PCnet/PCnetII/PCnet32
- National Semiconductor DP8381x based
- National Semiconductor DP8382x based Gigabit Ethernet cards
- VIA Rhine based
- TI ThunderLAN based

ISDN Interfaces

- **Most ISDN PCI Cards**
  Data connections at 64...128kbps, client and server

VoIP Interfaces

- **H.323 Protocol VoIP Analog Gateways**
  QuickNet LineJack ISA
  QuickNet PhoneJack for IP telephones
  Voicetronix V4PCI − 4 analog telephone lines cards
  Zaptel X.100P IP telephony card (1 analog line)
- **H.323 Protocol VoIP Digital Gateways**
  ISDN cards for VoIP gateways
- **H.323 Protocol IP Telephones**
  QuickNet LineJack and PhoneJack ISA

xDSL Interfaces

(additional license purchase required − 'Synchronous')

- **Xpeed 300 SDSL cards**
  Up to 6.7km twisted pair wire connection, max 2.3Mbps

HomePNA Interfaces

- **Linksys HomeLink PhoneLine Network Card**
  Up to 10Mbps home network over telephone line.

Phone: +371 7 317 700
Fax: +371 7 317 701
URL: http://www.mikrotik.com
E-mail: mt@mikrotik.com
Call the office using our H.323 gateway: VoIP.MikroTik.COM
Office hours: Monday-Friday 9AM-5PM local time (GMT + 2)

Device Driver List

Document revision 1.29 (04−Sep−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The document lists the drivers, included in MikroTik RouterOS and the devices that are tested to work with MikroTik RouterOS. If a device is not listed here, it does not mean the device is not supported, it still may work. It just means that the device is not tested.

Ethernet (system)

- **3Com 509 Series**
  Load the driver by specifying the I/O base address. IRQ is not required.
  Chipset type: 3Com 509 Series ISA 10Base
  Compatibility: 3Com EtherLink III
- **3Com FastEtherLink**
  Chipset type: 3Com 3c590/3c900 (3Com FastEtherLink and FastEtherLink XL) PCI 10/100Base
  Compatibility:

  3c590 Vortex 10Mbps
  3c592 chip
  3c595 Vortex 100baseTx
  3c595 Vortex 100baseT4
  3c595 Vortex 100base−MII
  3c597 chip
  3Com Vortex
  3c900 Boomerang 10baseT
  3c900 Boomerang 10Mbps Combo
  3c900 Cyclone 10Mbps Combo
  3c900B−FL Cyclone 10base−FL
  3c905 Boomerang 100baseTx
  3c905 Boomerang 100baseT4

3c905B Cyclone 100baseTx
3c905B Cyclone 10/100/BNC
3c905B−FX Cyclone 100baseFx
3c905C Tornado
3c980 Cyclone
3cSOHO100−TX Hurricane
3c555 Laptop Hurricane
3c575 Boomerang CardBus
3CCFE575 Cyclone CardBus
3CCFE656 Cyclone CardBus
3c575 series CardBus
3Com Boomerang

- **ADMtek Pegasus**
  Chipset type: ADMtek Pegasus/Pegasus II USB 10/100BaseT
  Compatibility:

  Planet 10/100Base−TX USB Ethernet Adapter UE−9500
  Linksys Instant EtherFast 10/100 USB Network Adapter USB100TX
- **AMD PCnet**
  For ISA cards load the driver by specifying the I/O base address. IRQ is not required.
  Chipset type: AMD PCnet/PCnet II ISA/PCI 10BaseT
  Compatibility:

  AMD PCnet−ISA
  AMD PCnet−ISA II
  AMD PCnet−PCI II
  AMD 79C960 based cards
- **AMD PCnet32**
  Chipset type: AMD PCnet32 PCI 10BaseT and 10/100BaseT
  Compatibility:

  AMD PCnet−PCI
  AMD PCnet−32
  AMD PCnet−Fast
- **Davicom DM9102**
  Chipset type: Davicom DM9102 PCI 10/100Base
  Compatibility:

  Davicom DM9102
  Davicom DM9102A
  Davicom DM9102A+DM9801
  Davicom DM9102A+DM9802
- **DEC 21x4x "Tulip"**
  Chipset type: DEC 21x4x "Tulip" PCI 10/100Base
  Compatibility:

  Digital DC21040 Tulip
  Digital DC21041 Tulip
  Digital DS21140 Tulip
  21140A chip
  21142 chip

Digital DS21143 Tulip
D−Link DFE 570TX 4−port
Lite−On 82c168 PNIC
Macronix 98713 PMAC
Macronix 98715 PMAC
Macronix 98725 PMAC
ASIX AX88140
Lite−On LC82C115 PNIC−II
ADMtek AN981 Comet
Compex RL100−TX
Intel 21145 Tulip
IMC QuikNic FX
Conexant LANfinity

- **Intel EtherExpressPro**
  Chipset type: Intel i82557 "Speedo3" (Intel EtherExpressPro) PCI 10/100Base
  Compatibility: Intel i82557/i82558/i82559ER/i82801BA−7 EtherExpressPro PCI cards
- **Intel PRO/1000**
  Chipset type: Intel i8254x (Intel PRO/1000) PCI 10/100/1000Base
  Compatibility:

  Intel PRO/1000 Gigabit Server Adapter (i82542, Board IDs: 700262−xxx, 717037−xxx)
  Intel PRO/1000 F Server Adapter (i82543, Board IDs: 738640−xxx, A38888−xxx)
  Intel PRO/1000 T Server Adapter (i82543, Board IDs: A19845−xxx, A33948−xxx)
  Intel PRO/1000 XT Server Adapter (i82544, Board IDs: A51580−xxx)
  Intel PRO/1000 XF Server Adapter (i82544, Board IDs: A50484−xxx)
  Intel PRO/1000 T Desktop Adapter (i82544, Board IDs: A62947−xxx)
  Intel PRO/1000 MT Desktop Adapter (i82540, Board IDs: A78408−xxx, C91016−xxx)
  Intel PRO/1000 MT Server Adapter (i82545, Board IDs: A92165−xxx, C31527−xxx)
  Intel PRO/1000 MT Dual Port Server Adapter (i82546, Board IDs: A92111−xxx, C29887−xxx)
  Intel PRO/1000 MT Quad Port Server Adapter (i82546, Board IDs: C32199−xxx)
  Intel PRO/1000 MF Server Adapter (i82545, Board IDs: A91622−xxx, C33915−xxx)
  Intel PRO/1000 MF Server Adapter (LX) (i82545, Board IDs: A91624−xxx, C33916−xxx)
  Intel PRO/1000 MF Dual Port Server Adapter (i82546, Board IDs: A91620−xxx, C30848−xxx)
- **National Semiconductor DP83810**
  Chipset type: National Semiconductor DP83810 PCI 10/100BaseT
  Compatibility:

  RouterBoard 200 built−in Ethernet
  RouterBoard 24 4−port Ethernet
  NS DP8381x−based cards
- **National Semiconductor DP83820**
  Chipset type: National Semiconductor DP83820 PCI 10/100/1000BaseT
  Compatibility:

  Planet ENW−9601T
  NS DP8382x−based cards
- **NE2000 ISA**
  Load the driver by specifying the I/O base address. IRQ is not required.
  Chipset type: NE2000 ISA 10Base
  Compatibility: various ISA cards

- **NE2000 PCI**
  Chipset type: NE2000 PCI 10Base
  Compatibility:

  RealTek RTL−8029
  Winbond 89C940 and 89C940F
  Compex RL2000
  KTI ET32P2
  NetVin NV5000SC
  Via 86C926
  SureCom NE34
  Holtek HT80232
  Holtek HT80229
  IMC EtherNic/PCI FO
- **NS8390**
  Chipset type: NS8390 PCMCIA/CardBus 10Base
  Compatibility:

  D−Link DE−660 Ethernet
  NE−2000 Compatible PCMCIA Ethernet
  NS8390−based PCMCIA cards
- **RealTek RTL8129**
  Chipset type: RealTek RTL8129 PCI 10/100Base
  Compatibility:

  RealTek RTL8129 Fast Ethernet
  RealTek RTL8139 Fast Ethernet
  RTL8139A/B/C chip
  RTL8130 chip
  SMC1211TX EZCard 10/100 (RealTek RTL8139)
  Accton MPX5030 (RealTek RTL8139)
  D−Link DFE 538TX
- **Sundance ST201 "Alta"**
  Chipset type: Sundance ST201 "Alta" PCI 10/100Base
  Compatibility

  D−Link DFE−550TX Fast Ethernet Adapter
  D−Link DFE−550FX 100Mbps Fiber−optics Adapter
  D−Link DFE−580TX 4−port Server Adapter
  D−Link DFE−530TXS Fast Ethernet Adapter
  D−Link DL10050−based FAST Ethernet Adapter
  Sundance ST201 "Alta" chip
  Kendin KS8723 chip
- **TI ThunderLAN**
  Chipset type: TI ThunderLAN PCI 10/100Base
  Compatibility:

  Compaq Netelligent 10 T
  Compaq Netelligent 10 T/2
  Compaq Netelligent 10/100 TX
  Compaq NetFlex−3/P

Olicom OC−2183
Olicom OC−2185
Olicom OC−2325
Olicom OC−2326

- **VIA vt86c100 "Rhine"**
  Chipset type: VIA vt86c100 "Rhine" PCI 10/100Base
  Compatibility:

  VIA Rhine (vt3043)
  VIA Rhine II (vt3065 AKA vt86c100)
  VIA VT86C100A Rhine
  VIA VT6102 Rhine−II
  VIA VT6105 Rhine−III
  VIA VT6105M Rhine−III
  D−Link DFE 530TX
- **Winbond w89c840**
  Chipset type: Winbond w89c840 PCI 10/100Base
  Compatibility:

  Winbond W89c840
  Compex RL100−ATX

Wireless (wireless)

- **Aironet Arlan**
  Chipset type: Aironet Arlan IC2200 ISA 2Mbit/s IEEE802.11b
  Compatibility: Aironet Arlan 655
- **Atheros**
  Chipset type: Atheros AR5001X PC/PCI 11/54Mbit/s IEEE802.11a/b/g
  Compatibility:

  Intel 5000 series
  Dlink DWL−A520
  Dlink DWL−G650
  Atheros AR5000 chipset series based IEEE802.11a cards
  Atheros AR5001A chipset series based IEEE802.11a cards
  Atheros AR5001X chipset series based IEEE802.11a, IEEE802.11b/g, IEEE802.11a/b/g cards
  Atheros AR5001X+ chipset series based IEEE802.11a, IEEE802.11b/g, IEEE802.11a/b/g cards
- **Cisco/Aironet**
  Chipset type: Cisco/Aironet ISA/PCI/PC 11Mbit/s IEEE802.11b
  Compatibility:

  Aironet ISA/PCI/PC4800 2.4GHz DS 11Mbps Wireless LAN Adapters (100mW)
  Aironet ISA/PCI/PC4500 2.4GHz DS 2Mbps Wireless LAN Adapters (100mW)
  CISCO AIR−PCI340 2.4GHz DS 11Mbps Wireless LAN Adapters (30mW)
  CISCO AIR−PCI/PC350/352 2.4GHz DS 11Mbps Wireless LAN Adapters (100mW)
- **Intersil Prism II**
  Chipset type: Intersil Prism II PC/PCI 11Mbit/s IEEE802.11b
  Compatibility:

  Intersil PRISM2 Reference Design 11Mb/s IEEE802.11b WLAN Card

GemTek WL−211 Wireless LAN PC Card
Compaq WL100/200 11Mb/s 802.11b WLzAN Card
Compaq iPaq HNW−100 11Mb/s 802.11b WLAN Card
Samsung SWL2000−N 11Mb/s 802.11b WLAN Card
Z−Com XI300 11Mb/s 802.11b WLAN Card
ZoomAir 4100 11Mb/s 802.11b WLAN Card
Linksys WPC11 11Mbps 802.11b WLAN Card
Addtron AWP−100 11Mbps 802.11b WLAN Card
D−Link DWL−650 11Mbps 802.11b WLAN Card
SMC 2632W 11Mbps 802.11b WLAN Card
BroMax Freeport 11Mbps 802.11b WLAN Card
Intersil PRISM2 Reference Design 11Mb/s WLAN Card
Bromax OEM 11Mbps 802.11b WLAN Card (Prism 2.5)
Bromax OEM 11Mbps 802.11b WLAN Card (Prism 3)
corega K.K. Wireless LAN PCC−11
corega K.K. Wireless LAN PCCA−11
CONTEC FLEXSCAN/FX−DDS110−PCC
PLANEX GeoWave/GW−NS110
Ambicom WL1100 11Mbps 802.11b WLAN Card
LeArtery SYNCBYAIR 11Mbps 802.11b WLAN Card
Intermec MobileLAN 11Mbps 802.11b WLAN Card
NETGEAR MA401 11Mbps 802.11 WLAN Card
Intersil PRISM Freedom 11mbps 802.11 WLAN Card
OTC Wireless AirEZY 2411−PCC 11Mbps 802.11 WLAN Card
Z−Com XI−325HP PCMCIA 200mW Card
Z−Com XI−626 Wireless PCI Card

- **RadioLAN**
  Chipset type: RadioLAN ISA/PC 10Mbit/s 5.8GHz
  Compatibility:

  RadioLAN ISA card (Model 101)
  RadioLAN PCMCIA card
- **WaveLAN/ORiNOCO**
  Chipset type: Lucent/Agere/Proxim WaveLAN/ORiNOCO ISA/PC 11Mbit/s IEEE802.11b
  Compatibility:

  WaveLAN Bronze/Gold/Silver ISA/PCMCIA

Synchronous (synchronous)

- Moxa C101 V.35 (4 Mbit/s)
- Moxa C502 PCI 2−port V.35 (8 Mbit/s)
- Cyclades PC−300 V.35 (5 Mbit/s)
- Cyclades PC−300 E1/T1
- FarSync V.35/X.21 (8.448 Mbit/s)

Asynchronous (system)

- Standard Communication Ports Com1 and Com2
- Moxa Smartio C104H, C168H, CP−114, CP−132, CP−168U, CP−104U, CP−134U, CP−132U PCI 2/4/8 port up to 4 cards (up to 32 ports)
- Cyclades Cyclom−Y and Cyclades−Z Series up to 32 ports per card, up to 4 cards (up to 128 ports)

- TCL DataBooster 4 or 8 PCI cards

ISDN (isdn)

PCI ISDN cards:

Eicon.Diehl Diva PCI
Sedlbauer Speed Card PCI
ELSA Quickstep 1000PCI
Traverse Technologie NETjet PCI S0 card
Teles PCI
Dr. Neuhaus Niccy PCI
AVM Fritz PCI
Gazel PCI ISDN cards
HFC−2BS0 based PCI cards (TeleInt SA1)
Winbond W6692 based PCI cards

VoIP (telephony)

- H.323 Protocol VoIP Analog Gateways

QuickNet LineJack ISA
QuickNet PhoneJack ISA
Voicetronix V4PCI − 4 analog telephone lines cards
Zaptel X.100P IP telephony card (1 analog line)

xDSL (synchronous)

Xpeed 300 SDSL cards (up to 6.7km twisted pair wire connection, max 2.3Mbit/s)

HomePNA (system)

Linksys HomeLink PhoneLine Network Card (up to 10Mbit/s home network over telephone line)

LCD (lcd)

- Crystalfontz (www.crystalfontz.com) Intelligent Serial LCD Module 632 (16x2 characters) and 634 (20x4 characters)
- Powertip (www.powertip.com.tw) Character LCD Module PC1602 (16x2 characters) and PC2404 (24x4 characters)

PCMCIA Adapters (system)

- Vadem VG−469 PCMCIA−ISA adapter (one or two PCMCIA ports)
- RICOH PCMCIA−PCI Bridge with R5C475 II or RC476 II chip (one or two PCMCIA ports)
- CISCO/Aironet PCMCIA adapter (ISA and PCI versions) for CISCO/Aironet PCMCIA cards only

---

How to Read Reference Manual

Document revision 1.1 (15−Apr−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

This document contains general information on how to read Reference Manual. Here you can find information about Manual purposes, structure and common conventions.

The Purpose

The Reference Manual is designed to give information about all aspects of MikroTik RouterOS installation, configuration, maintenance and upgrading as well as some tupical examples.

The Structure

The full list of covered topics can be accessed within the main Manual page. Each topic consists of:

- Main Header
- Table of Contents
- Summary
- Specifications
- Related Documents
- [Description]
- [Property Description]
- Topic 1
    - ♦ [Description]
    - ♦ [Property Description]
    - ♦ [Notes]
    - ♦ [Example]
- Topic 2
    - ♦ [Subtopic 1]
    - ♦ [Subtopic 2]
    - ♦ ...
    - ♦ [Subtopic n]
- ...
- Topic n
- [Notes]

  - [Example]
  - Additional Resources

- Note, that some items do not present in each Manual part. Such items are put in brackets [].
- **Main Header** − here the theme and document revision are shown
- **Table of Contents** − contains table of links to different subtopics of a theme
- **Summary** − short summary of functions and (or) technology.
- **Specifications** − holds information about packages and licences needed as well as utilized protocols and hardware requirements
- **Related Documents** − contains links to related entries in the Manual
- **Description** − General item description. Includes theoretical aspects and implementation specs
- **Property Description** − Describes available arguments of commands (if any)
- **Notes** − some facts worth to hold in mind
- **Example** − shows typical example or (and) application example

Each manual entry can contain subtopics which hold their own Description, Property Description, Notes and Example items.

Common Conventions

There are some common conventions through the entire Manual which are worth to know:
- All commands or arguments are in **bold**, i.e **/ip adress add address=10.10.10.1/24**
- In case instead of actual value a range has been entered, it is in *italics*, *id est* **dst−address** (*IP adress*)
- Default value of an argument is in **bold** and is prefixed by the keyword 'default' , i.e **action** (drop | accept, default: **accept**)

There are some access modifiers used in Property Description:
- read−only − the argument can not be modified by the user directly, *exempli gratia* from **set** command
- multiple choice − these arguments can be selected in combinations, *id est* **supported−rates−a=6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps**

Additional Resources

Key words for use in RFCs to Indicate Requirement Levels

Glossary

Document revision 1.0 (28−Apr−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

The Glossary consists of two parts.
The first part 'Common Properties' will give definitions to some common properties listed under 'Property Description' subtopics as well as respective values description.
The second part 'Terms and Abbreviations'' will explain the meaning of technical terms, difficult worlds or phrases and abbreviations used throughout the Reference Manual.

Common Properties

arp (disabled | enabled | proxy−arp | reply−only; default: **enabled**) − Address Resolution Protocol (ARP) setting (for more information, see IP Addresses and Address Resolution Protocol):
- **disabled** − the interface will not use ARP
- **enabled** − the interface will use ARP
- **proxy−arp** − the interface will use the ARP proxy feature
- **reply−only** − the interface will only reply to the requests originated to its own IP addresses. Neighbour MAC addresses will be resolved using **/ip arp** statically set table only

card−type (read−only: *type*) − a string with some basic information about adapter type and model

mac−address (*MAC address*) − an IEEE Media Access Control (MAC) address. This is a hardware address that uniquely identifies each node of a network. It is shown as a sequence of six two−digits hexadecimal numbers separated by colons, *exempli gratia*: **00:2f:21:c1:11:0a**. In the console it also can be entered without delimiters, *id est* **002f21c1110a**

mtu (*integer*) − Maximum Transmission Unit, the largest physical packet size, measured in bytes, that a network can transmit. Any packets larger than the MTU are fragmented into smaller ones before being sent over the network; this slows down transmission speeds. There are some tupical settings of MTU: the MTU of many PPP connections is 576 while most ethernet networks have an MTU of 1500

name (*name*) − assigned feature name. Usually is used for inner reference and scripting

Terms and Abbreviations

Access Point − see **AP**

ad−hoc mode − a network framework provided by IEEE 802.11 standard set in which all communications between wireless clients are made without the use of an Access Point (AP). This mode sometimes is referred as a peer−to−peer mode

AP − short for Access Point, a set of hardware and software that acts as a communication hub for wireless clients to connect to a wired LAN. APs are important for providing heightened wireless network security and for extending the physical range of service a wireless client has access to.
See **infrastructure mode** and **ad−hoc mode**.

ARP − short for address resolution protocol. This protocol is used to resolve **IP addresses** to **MAC addresses**

Basic Service Set − (BSS). A network setup with a set of wireless clients and one AP connected to a wired network

dlci − short for data link connection identifier. Identifies the number of the logical circuit the data travels over. DLCI is a number of a switched virtual or private circuit in a Frame Relay network, which is used to determine how to route the data.

Extended Service Set − (ESS). A set of two or more **BSS**s that for one single subnetwork

IEEE − short for Institute of Electrical and Electronics Engineers. IEEE is best known for developing various standards for the computer and electronic industry

infrastructure mode − a network framework provided by IEEE 802.11 standard set in which all communications between wireless clients are made with a help of an Access Point (AP). In this mode, wireless devices can communicate either with each other or with a wired network. There are two possible infrastructure mode configurations referred as **Basic Service Set** (BSS) and **Extended Service Set** (ESS). The infrastructure mode is widely used in corporate networks in order to gain access to wired LAN services such as file or application servers and printers

IP address − short for Internet Protocol address. This is a logical address belonging to the OSI layer 3. Consists of four (IPv4) or six (IPv6) binary octets. Usually is shown in decimal form, *exempli gratia* **159.148.60.2**.

MAC address − short for Media Access Control address. This is OSI layer 2 hardware address defined by IEEE standard and is used to deliver packets in the local network. It is sequence of six two−digits hexadecimal numbers separated by colons, *exempli gratia*: **00:2f:21:c1:11:0a**.

RFC − short for request for comments. This is a set of technical and organizational notes about the Internet. Memos in the RFC series discuss many aspects of computer networking, including protocols, procedures, programs, and concepts

ssid − short for Service Set Identifier. The SSID is a 32−character identifier which is used in wireless networking to separate different networks. All devices within the same network must have the same SSID.

EAP − short for Extensible Authentication Protocol defined in RFC 2284. It is general authentication protocol which supports various methods of authentication, such as passwords, public keys, Kerberos and smart cards.

In wireless communications using EAP, a user requests connection to a WLAN through an AP, which then requests the identity of the user and transmits that identity to an authentication server such as RADIUS. The

server asks the AP for proof of identity, which the AP gets from the user and then sends back to the server to complete the authentication.

Device Driver Management

Document revision 1.5 (15−May−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

Device drivers represent the software interface part of installed network devices. Some drivers are included in the system software package and some in additional feature packages.

For complete list of supported devices and respective device driver names please consult the **Related Documents** section.

The device drivers for PCI, miniPCI, PC (PCMCIA) and CardBus cards are loaded automatically. Other network interface cards (most ISA and PCI ISDN cards) require the device drivers to be loaded manually using the **/driver add** command.

Users cannot add their own device drivers, only drivers included in the Mikrotik RouterOS software packages can be used. If you need a support for a device, which hasn't a driver yet, you are welcome to suggest it at suggestion page on our web site.

Specifications

Packages required : *appropriate for particular device*
License required : *appropriate for particular device*
Home menu level : **/driver**
Standards and Technologies : *PCI, ISA, PCMCIA, miniPCI, CardBus*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
License Management

Device Driver List

Loading Device Drivers

Submenu level : **/driver**

Description

The drivers for PCI (except the ISDN cards) and PCMCIA cards are loaded automatically at the system startup time. You can use the **/driver print** command to see which drivers are loaded:

```
[admin@MikroTik] > /driver print
Flags: I – invalid, D – dynamic
  #   DRIVER                                IRQ IO       MEMORY   ISDN-PROTOCOL
  0 D RealTek 8139
  1 D Atheros AR5211 PCI
  2 D VIA Rhine I/II Fast Ethernet

[admin@MikroTik] >
```

As we see, the driver for the Realtek PCI card has been loaded automatically as well as drivers for Atheros wireless adapter and VIA Rhine Fast Ethernet controller.

To see system resources occupied by the devices, use the **/system resource io print** and **/system resource irq print** commands.

If an installed device requires the driver to be loaded manually, use the **/driver add** command.

Property Description

io (*integer*) − input−output port base address
irq (*integer*) − interrupt request number
isdn−protocol (euro | german; default: **euro**) − line protocol for ISDN cards
memory (*integer*; default: **0**) − shared memory base address
name (*name*) − driver name

Notes

Not all combinatios of **irq** and **io** might work on particular system. It is recommended, that you first find an acceptable irq setting and then try different i/o base addresses.

If hexadecimal values are used for the arguments, put **0x** before the number

To see the list of available drivers, enter the **/driver add name ?** command

The resource list shows only those interfaces, which are enabled

Typical **io** values for ISA cards are **0x280**, **0x300** and **0x320**

Example

To view the list of available drivers, do the following:

```
[admin@MikroTik] driver> add name ?
Name of driver to load.
     3c509  3com 3c509 ISA
      c101  Moxa C101 ISA
  ne2k-isa  ISA NE2000
    pc-isa  Aironet ISAxx00
[admin@MikroTik] driver>
```

To see system resources occupied by the devices, use the **/system resource io print** and **/system resource irq print** commands:

```
[admin@MikroTik] system resource> irq print
Flags: U - unused
   IRQ OWNER
   1   keyboard
   2   APIC
 U 3
   4   sync1
   5   pc1
 U 6
 U 7
 U 8
 U 9
   10  ether2
   11  ether1
 U 12
   13  FPU
   14  IDE 1
[admin@MikroTik] system resource> io print
 PORT-RANGE        OWNER
 20-3F             APIC
 40-5F             timer
 60-6F             keyboard
 80-8F             DMA
 A0-BF             APIC
 C0-DF             DMA
 F0-FF             FPU
 1F0-1F7           IDE 1
 300-33F           pc1
 3C0-3DF           VGA
 3F6-3F6           IDE 1
 CF8-CFF           [PCI conf1]
 1000-100F         [Silicon Integrated Systems [SiS] 5513 [IDE]]
 1000-1007         IDE 1
 1008-100F         IDE 2
 6000-60FF         [Realtek Semiconductor Co., Ltd. RTL-8139]
 6000-60FF         [8139too]
 6100-61FF         [Realtek Semiconductor Co., Ltd. RTL-8139 (#2)]
 6100-61FF         [8139too]
[admin@MikroTik] system resource>
```

Suppose we need to load a driver for a NE2000 compatible ISA card. Assume we had considered the information above and have checked avalable resources in our system. To add the driver, we must do the following:

```
[admin@MikroTik] driver> add name=ne2k-isa io=0x280
[admin@MikroTik] driver> print
Flags: I - invalid, D - dynamic
  #   DRIVER                                IRQ IO       MEMORY   ISDN-PROTOCOL
  0 D RealTek 8139
  1 D Intel EtherExpressPro
  2 D PCI NE2000
  3   ISA NE2000                            280
  4   Moxa C101 Synchronous                            C8000
[admin@MikroTik] driver>
```

Removing Device Drivers

You can remove only statically loaded drivers, *id est* those which do not have **D** flag before the driver name. The device drivers can be removed only if the appropriate interface has been disabled.

Use the **/driver remove** command to remove device drivers. Unloading a device driver is useful when you swap or remove a network device − it saves system resources avoiding to load drivers for removed devices.

Device driver needs to be removed and loaded again, if some parameters (memory range, i/o base address) have been changed for the network interface card.

Notes on PCMCIA Adapters

Currently only the following PCMCIA−ISA and PCMCIA−PCI adapters are tested to comply with MikroTik RouterOS:

- RICOH PCMCIA−PCI Bridge with R5C475 II or RC476 II chip (one or two PCMCIA ports)
- CISCO/Aironet PCMCIA adapter (ISA and PCI versions) for CISCO/Aironet PCMCIA cards only

Other PCMCIA−ISA and PCMCIA−PCI adapters might not function properly.

The Ricoh adapter might not work properly with some older motherboards. When recognized properly by the BIOS during the boot up of the router, it should be reported under the PCI device listing as "PCI/CardBus bridge". Try using another motherboard, if the adapter or the Prism card are not recognized properly.

Note that the maximum number of PCMCIA ports for a single system is equal to 8. If you will try to install 9 or more ports (no matter one−port or two−port adapters), no one will be recognized.

Troubleshooting

- *Driver for a PCI or PC card does not load automatically.*
  Check for a possible IRQ or IO conflict with other devices.
- *The driver cannot be found on the system.*
  Upload the required software package containing the required drivers and reboot the router.
- *The driver has been loaded, but the interface does not show up.*
  Obtain the required software license to enable the functionality of the interface.

General Interface Settings

Document revision 1.2 (15−Apr−2003)
This document applies to the MikroTik RouterOS V2.7

Table Of Contents

Summary

MikroTik RouterOS supports a variety of Network Interface Cards as well as some virtual interfaces (like VLAN, Bridge, etc.). Each of them has its own submenu, but there is also a list of all interfaces where some common properties can be configured.

Related Documents

Atheros 5GHz Wireless Interface
Bridge Interfaces
CISCO/Aironet 2.4GHz 11Mbps Wireless Interface
Cyclades PC300 PCI Adapters
Ethernet Interfaces
EoIP (Ethernet over IP) Tunnel Interface
FarSync X.21 Interface
FrameRelay (PVC, Private Virtual Circuit) Interface
IPIP (IP over IP) Tunnel Interfaces
ISDN Interface
L2TP (Layer 2 Tunnel Protocol) Interface
MOXA C101 Synchronous Interface
MOXA C502 Dual−port Synchronous Interface
PPP (Point to Point Protocol) and Asynchronous Interfaces
PPPoE (Point to Point Protocol over Ethernet) Interface
PPTP (Point to Point Tunnel Protocol) Interface
PrismII Wireless Interface
RadioLAN 5.8GHz Wireless Interface
VLAN (Virtual LAN) Interface
Xpeed SDSL (Single−line Digital Subscriber Line) Interface

Description

The Manual describes general settings for MikroTik RouterOS interfaces.

Interface Status

Submenu level : **/interface**

Property Description

status (*read−only*) − shows the interface status.
name (*string*) − descriptive name of interface
type (*read−only*: arlan | atheros | bridge | cyclades | eoip−tunnel | ether | farsync | ipip | isdn−in | isdn−out | l2tp−in | l2tp−out | moxa−c101− | moxa−c502− | pc | ppp−in | ppp−out | pppoe−in | pppoe−out | pptp−in | pptp−out | prism | pvc | radiolan | vlan | wlan | xpeed) − interface type
mtu (*integer*) − maximum transmit unit for the interface in bytes

Notes

In order to use the interface, its status must be 'Running'.

Example

To see the list of all available interfaces:

```
[admin@MikroTik] interface> print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ether1               ether            1500
  1  R prism1               prism            1500
[admin@MikroTik] interface>
```

Traffic monitoring

Submenu level : **/interface monitor−traffic**

Description

The traffic passing through any interface can be monitored.

Notes

One or more interfaces can be monitored at the same time.

Example

In the example below **ether1** interface will be monitored:

```
[admin@MikroTik] interface> monitor−traffic ether1
    received−packets−per−second: 271
```

```
      received-bytes-per-second: 148.4kbps
        sent-packets-per-second: 600
          sent-bytes-per-second: 6.72Mbps

[admin@MikroTik] interface>
```

In the next example we will monitor two interfaces at a time:

```
[admin@MikroTik] interface> monitor-traffic ether1,prism1
    received-packets-per-second: 2         0
       received-bits-per-second: 960.00bps 0.00bps
        sent-packets-per-second: 2         0
           sent-bits-per-second: 2.57kbps  0.00bps

[admin@MikroTik] interface>
```

Wireless Client and Wireless Access Point Manual

Document revision 1.23 (30−Dec−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The MikroTik RouterOS supports the Atheros AR5111, AR5211, AR2111 chipset based wireless adapter cards for working as wireless clients (**station** mode), wireless bridges (**bridge**) mode and wireless access points (**ap−bridge** mode).

On account of that MikroTik RouterOS provides a complete support for IEEE 802.11a, 802.11b and 802.11g wireless networking standards.

Specifications

Packages required : ***wireless***
License required : *2.4/5GHz Wireless Client, 2.4/5Ghz Wireless Access Point (optional)*
Home menu level : ***/interface wireless***

Standards and Technologies : *IEEE802.11b (IEEE802.11b), IEEE802.11a (IEEE802.11a), IEEE802.11g*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
Log Management

Wireless Networking Ranges

The Atheros card has been tested for distances up to 20 km providing connection speed up to 17Mbit/s. With appropriate antennas and cabling the maximum distance should be as far as 40 km.

These values of **ack−timeout** were approximated from the tests done by us, as well as by some of our customers:

range	**ack−timeout**		
	5GHz	**5GHz−turbo**	**2.4GHz−G**
0km	default	default	default
5km	52	30	62
10km	85	48	96
15km	121	67	133
20km	160	89	174
25km	203	111	219
30km	249	137	268
35km	298	168	320
40km	350	190	375
45km	405	−	−

Please **note** that these are not the precise values. Depending on hardware used and many other factors they may vary up to +/− 15 microseconds.

Description

Atheros 5G/ABM Wireless adapter is a new generation solution for wireless applications. This universal Multi−Band (2.4 GHz, 5.2 GHz, 5.8 GHz) PCI operates in any existing IEEE wireless standard. It minimizes any potential confusion or incompatibilities caused by having three separate wireless devices.

The Multi−Band Wireless PCI operates in both 2.4 GHz and 5 GHz wireless bands

Hardware Notes

The MikroTik RouterOS supports as many Atheros chipset based cards as many free adapter slots are there on your system. One license is valid for all cards on your system. **Note** that maximal number of PCMCIA slots is 8.

Some chipsets are not stable with Atheros cards and cause radio to stop working. Via Epia, MikroTik RouterBoard and systems based on Intel i815 and i845 chipsets are tested and work stable with Atheros cards. There might be many other chipsets that are working stable, but it has been reported that some older chipsets, and some systems based on AMD Duron CPU are not stable.

We can not be responsible for the performance and stability of Atheros−based cards that are not purchased from us. There are some cards that due to their design are unable to provide reasonably good stability and speed.

Wireless Interface Configuration

Submenu level : **/interface wireless**

Description

The wireless interface operates using IEEE 802.11 set of standards. It uses radio waves as a physical signal carrier and is capable of wireless data transmission with speeds up to 108 Mbps (in turbo−mode).

Property Description

name (*name*; default: **wlan*N***) − assigned interface name
mtu (*integer: 68..1600*; default: **1500**) − Maximum Transmission Unit
mac−address (*MAC address*) − MAC address
arp (disabled | enabled | proxy−arp | reply−only; default: **enabled**) − Address Resolution Protocol setting
sta−count (*integer: 1..2007*; default: **2007**) − maximal number of clients allowed for simultaneous connections
card−type (*read−only*: *type*) − adapter type and model
mode (station | ap−brige | bridge; default: **station**) − operating mode:

- **station** − the card is operating as a wireless client
- **ap−bridge** − the card is operating as an AP
- **bridge** − the card is operating as a bridge

ssid (*text*; default: **MikroTik**) − Service Set Identifier. Used to separate wireless networks
frequency (*integer*; default: **5180**) − operating frequency of the card
band (2.4GHz−B | 2.4GHz−G | 5GHz | 5GHz−turbo; default: **5GHz**) − operating band:

- **2.4GHz−B** − IEEE 802.11b
- **2.4GHz−G** − IEEE 802.11g
- **5GHz** − IEEE 802.11a up to 54Mbit
- **5GHz−turbo** − IEEE 802.11a up to 108Mbit

scan−list (*multiple choice*: **integer** | default−ism; default: **default−ism**) − the list of channels to scan

- **default−ism** − for 2.4GHz modes: 2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, 2462, 2467, 2472; for 5GHz mode: 5180, 5200, 5220, 5240, 5260, 5280, 5300, 5320, 5745, 5765, 5785, 5805; for 5GHz−turbo: 5210, 5250, 5290, 5760, 5800

supported−rates−a/g (*multiple choice*: 6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps; default: **6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps**) − rates to be suppored when operating in IEEE 802.11a and 802.11g standards
basic−rates−a/g (*multiple choice*: 6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps; default: **6Mbps**) − basic rates in IEEE 802.11a and 802.11g modes
supported−rates−b (*multiple choice*: 1Mbps,2Mbps,5.5Mbps,11Mbps; default: **1Mbps,2Mbps,5.5Mbps,11Mbps**) − rates to be suppored when operating in IEEE 802.11b standard

basic−rates−b (*multiple choice*: 1Mbps,2Mbps,5.5Mbps,11Mbps; default: **11Mbps**) − basic rates in IEEE 802.11b mode
ack−timeout (*integer*; default: **default**) − Acknowledgement Code timeout (transmission acceptance timeout) in microseconds
tx−power (*integer* | default; default: **default**) − transmit power in dB

- **default** − default value of the card

default−key−0 (*text*; default: "") − default key 0
default−key−1 (*text*; default: "") − default key 1
default−key−2 (*text*; default: "") − default key 2
default−key−3 (*text*; default: "") − default key 3
station−private−key (*text*; default: "") − private key of the AP
transmit−key−id (1 | 2 | 3 | 0; default: 0) − transmission key number
encryption (none | optional | required; default: **none**) − encryption to be used for connection:

- **required** − require encryption for connection
- **optional** − use encryption if possible

used−authentication (open−system | shared−key; default: **open−system**) − type of authentication to be used for connestion
accepted−authentication (both | open−system | shared−key; default: **open−system**) − accepted authentication types
default−authentication (yes | no; default: **yes**) − to enable authentication by default or not
default−forwarding (yes | no; default: **yes**) − to use forwarding by default or not
802.1x−enable (yes | no; default: **no**) − to use EAP for authentication or not

Notes

You should set **tx−power** property to an appropriate value as many cards do not have their default setting set to the maximal power it can work on. For the cards MicroTik is selling (5G/ABM), 20dB (100mW) is the maximal power in 5GHz bands and 18dB (65mW) is the maximal power in 2.4GHz bands.

For different versions of Atheros chipset there are different value range of **ack−timeout** property:

Chipset version	5GHz		5GHz−turbo		2.4GHz−B		2.4GHz−G	
	default	max	default	max	default	max	default	max
5000 (5.2GHz only)	30	204	22	102	N/A	N/A	N/A	N/A
5211 (5.2GHz and 5.8GHz)	30	409	22	204	N/A	N/A	N/A	N/A
5212 (802.11a/b/g)	25	409	22	204	30	409	52	409

Example

Let us consider a following example: a MikroTik router is connected to an AP using Atheros card and the AP is operating in IEEE 802.11b standard with **ssid=hotspot**.

```
[admin@MikroTik] interface wireless> print
Flags: X − disabled, R − running
  0  X name="wlan1" mtu=1500 mac−address=00:01:24:70:03:75 arp=enabled
       card−type=Atheros AR5211 2.4/5 GHz mode=station ssid="MikroTik"
       frequency=5180 band=5GHz scan−list=default−ism
       supported−rates−a=6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps
       basic−rates−a=6Mbps supported−rates−b=1Mbps,2Mbps,5.5Mbps,11Mbps
       basic−rates−b=1Mbps ack−timeout=default tx−power=default default−key−0=""
```

```
       default-key-1="" default-key-2="" default-key-3="" station-private-key=""
       transmit-key-id=0 encryption=none used-authentication=open-system
       accepted-authentication=open-system default-authentication=yes
       default-forwarding=yes 802.1x-enable=no

[admin@MikroTik] interface wireless>set 0 ssid=hotspot band=2.4GHz
[admin@MikroTik] interface wireless>enable 0
[admin@MikroTik] interface wireless> monitor 0
             status: connected-to-ess
               band: 2.4GHz
          frequency: 2437
            tx-rate: 11Mbps
               ssid: "hotspot"
              bssid: 00:03:2F:04:27:73
    signal-strength: 16
            rx-rate: 11Mbps

[admin@MikroTik] interface wireless>
```

The 'ess' stands for Extended Service Set (IEEE 802.11 wireless networking).

Registration Table

Submenu level : /**interface wireless registration-table**

Property Description

interface (*read-only*: *name*) - interface that client is registered to
mac-address (*read-only*: *MAC address*) - mac address of the registered client
type (*read-only*: radio) - type of the client:
parent (*read-only*: *MAC address*) - parent access point's MAC address, if forwarded from another access point
packets (*read-only*: *integer*,*integer*) - number of received and sent packets
bytes (*read-only*: *integer*,*integer*) - number of received and sent bytes
signal (*read-only*: *integer*) - average signal level
rx-rate (*read-only*: *integer*) - min/average/max receive data rate
tx-rate (*read-only*: *integer*) - transmit data rate
uptime (*read-only*: *time*) - time the client is associated with the access point

Example

To see registration table showing all clients currently associated with the access point:

```
[admin@MikroTik] interface wireless registration-table> print
  # INTERFACE   MAC-ADDRESS       TYPE      PARENT            SIGNAL     TX-RATE
  0 wlan1       00:01:24:70:03:33 radio                       20         6Mbps

[admin@MikroTik] interface wireless registration-table>
```

To get additional statistics:

```
[admin@MikroTik] interface wireless registration-table> print stats
  0 interface=wlan1 mac-address=00:01:24:70:03:33 type=radio rx-rate=36Mbps
    tx-rate=6Mbps packets=0,269 bytes=0,15190 uptime=01:49:23.020 signal=19
```

```
[admin@MikroTik] interface wireless registration-table>
```

Access List

Submenu level : /**interface wireless access-list**

Description

The access list is used by the access point to restrict / allow authentications (associations) of clients. This list contains MAC address of client and associated action to take when client attempts to connect. Also, the forwarding of frames sent by the client is controlled.

The association procedure is as follows: when a new client wants to associate to the AP that is configured on interface **wlan*N***, an entry with client's MAC address and interface **wlan*N*** is looked up in the access-list. If such entry is found then action specified in it is taken else **default-authentication** and **default-forwarding** arguments of interface **wlan*N*** are taken.

Property Description

mac-address (*MAC address*) - MAC address of the client
interface (*name*) - AP interface name
authentication (yes | no; default: **yes**) - whether to accept or to reject this client when it tries to connect
forwarding (yes | no; default: **yes**) - whether to forward the client's frames to other wireless clients
private-key (*text*; default: "") - private key of the client to validate during authentication

Notes

If you have default authentication action for the interface set to **yes**, you can disallow this node to register at the AP's interface **wlan*N*** by setting **authentication=no** for it. Thus, all nodes except this one will be able to register to the interface **wlan*N***.

If you have default authentication action for the interface set to **no**, you can allow this node to register at the AP's interface **wlan*N*** by setting **authentication=yes** for it. Thus, only the specified nodes will be able to register to the interface **wlan*N***.

Example

To allow authentication and forwarding for the client **00:40:96:37:A3:39** from the **prism1** interface:

```
[admin@MikroTik] interface wireless access-list> add mac-address=00:40:96:38:0F:
20 interface=wlan1

[admin@MikroTik] interface wireless access-list>
```

Info

Submenu level: **/interface wireless info**

Description

This facility provides you with general wireless interface information.

Property Description

tx−power−control (*read−only*: yes | no) − provides information whether this device supports transmission power control
ack−timeout−control (*read−only*: yes | no) − provides information whether this device supports transmission acceptance timeout control
supported−bands (*read−only, multiple choice*: 2GHz−B,5GHz,5GHz−turbo,2GHz−G) − the list of supported bands
2GHz−B−channels (*read−only*, *integer*) − the list of 2GHz IEEE 802.11b channels (frequences are given in MHz)
2GHz−G−channels (*read−only*, *integer*) − the list of 2GHz IEEE 802.11g channels (frequences are given in MHz)
5GHz−channels (*read−only*, *integer*) − the list of 5GHz channels (frequences are given in MHz)
5GHz−turbo−channels (*read−only*, *integer*) − the list of 5GHz−turbo channels (frequences are given in MHz)

Notes

There is a special argument for the **print** command − **print count−only**. It forces the **print** command to print only the count of information topics.

Example

```
[admin@MikroTik] interface wireless info> print
  0 tx−power−control=yes ack−timeout−control=yes
    supported−bands=2GHz−B,5GHz,5GHz−turbo,2GHz−G
    2GHz−B−channels=2412,2417,2422,2427,2432,2437,2442,2447,2452,2457,2462,
                    2467,2472
    5GHz−channels=5180,5200,5210,5220,5240,5250,5260,5280,5290,5300,5320,5745,
                  5765,5785,5805
    5GHz−turbo−channels=5180,5200,5210,5220,5240,5250,5260,5280,5290,5300,5320,
                        5745,5765,5785,5805
    2GHz−G−channels=2412,2417,2422,2427,2432,2437,2442,2447,2452,2457,2462,
                    2467,2472


[admin@MikroTik] interface wireless info> print
```

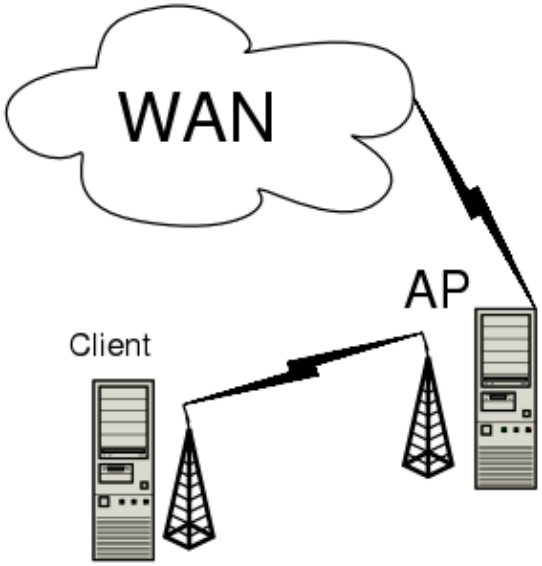
If you have the additional **Custom frequency** license (please contact support@mikrotik.com to get one), the list will be much bigger:

```
[admin@MikroTik] interface wireless info> print
  0 tx−power−control=yes ack−timeout−control=yes
    supported−bands=2GHz−B,5GHz,5GHz−turbo,2GHz−G
    2GHz−B−channels=2312,2317,2322,2327,2332,2337,2342,2347,2352,2357,2362,
                    2367,2372,2412,2417,2422,2427,2432,2437,2442,2447,2452,
                    2457,2462,2467,2472,2512,2532,2552,2572,2592,2612,2632,
                    2652,2672,2692,2712,2732,2484
    5GHz−channels=5120,5125,5130,5135,5140,5145,5150,5155,5160,5165,5170,5175,
                  5180,5185,5190,5195,5200,5205,5210,5215,5220,5225,5230,5235,
```

```
                5240,5245,5250,5255,5260,5265,5270,5275,5280,5285,5290,5295,
                5300,5305,5310,5315,5320,5325,5330,5335,5340,5345,5350,5355,
                5360,5365,5370,5375,5380,5385,5390,5395,5400,5405,5410,5415,
                5420,5425,5430,5435,5440,5445,5450,5455,5460,5465,5470,5475,
                5480,5485,5490,5495,5500,5505,5510,5515,5520,5525,5530,5535,
                5540,5545,5550,5555,5560,5565,5570,5575,5580,5585,5590,5595,
                5600,5605,5610,5615,5620,5625,5630,5635,5640,5645,5650,5655,
                5660,5665,5670,5675,5680,5685,5690,5695,5700,5705,5710,5715,
                5720,5725,5730,5735,5740,5745,5750,5755,5760,5765,5770,5775,
                5780,5785,5790,5795,5800,5805,5810,5815,5820,5825,5830,5835,
                5840,5845,5850,5855,5860,5865,5870,5875,5880,5885,5890,5895,
                5900,5905,5910,5915,5920,5925,5930,5935,5940,5945,5950,5955,
                5960,5965,5970,5975,5980,5985,5990,5995,6000,6005,6010,6015,
                6020,6025,6030,6035,6040,6045,6050,6055,6060,6065,6070,6075,
                6080,6085,6090,6095,6100
    5GHz-turbo-channels=5120,5125,5130,5135,5140,5145,5150,5155,5160,5165,5170,
                        5175,5180,5185,5190,5195,5200,5205,5210,5215,5220,5225,
                        5230,5235,5240,5245,5250,5255,5260,5265,5270,5275,5280,
                        5285,5290,5295,5300,5305,5310,5315,5320,5325,5330,5335,
                        5340,5345,5350,5355,5360,5365,5370,5375,5380,5385,5390,
                        5395,5400,5405,5410,5415,5420,5425,5430,5435,5440,5445,
                        5450,5455,5460,5465,5470,5475,5480,5485,5490,5495,5500,
                        5505,5510,5515,5520,5525,5530,5535,5540,5545,5550,5555,
                        5560,5565,5570,5575,5580,5585,5590,5595,5600,5605,5610,
                        5615,5620,5625,5630,5635,5640,5645,5650,5655,5660,5665,
                        5670,5675,5680,5685,5690,5695,5700,5705,5710,5715,5720,
                        5725,5730,5735,5740,5745,5750,5755,5760,5765,5770,5775,
                        5780,5785,5790,5795,5800,5805,5810,5815,5820,5825,5830,
                        5835,5840,5845,5850,5855,5860,5865,5870,5875,5880,5885,
                        5890,5895,5900,5905,5910,5915,5920,5925,5930,5935,5940,
                        5945,5950,5955,5960,5965,5970,5975,5980,5985,5990,5995,
                        6000,6005,6010,6015,6020,6025,6030,6035,6040,6045,6050,
                        6055,6060,6065,6070,6075,6080,6085,6090,6095,6100
    2GHz-G-channels=2312,2317,2322,2327,2332,2337,2342,2347,2352,2357,2362,
                    2367,2372,2412,2417,2422,2427,2432,2437,2442,2447,2452,
                    2457,2462,2467,2472,2512,2532,2552,2572,2592,2612,2632,
                    2652,2672,2692,2712,2732,2484

[admin@MikroTik] interface wireless info>
```

AP Configuration Example

Consider the following example:



You need both the 2.4GHz/5GHz Wireless Client and the Wireless AP Licenses to enable the AP mode. To make the MikroTik router work as an access point, the configuration of the wireless interface should be as follows:

- A unique Service Set Identificator should be chosen, say "test1"
- A frequency should be selected for the link, say 5180MHz
- The operation mode should be set to **ap−bridge**

The following command should be issued to change the settings for the wireless AP interface:

```
[admin@AP] interface wireless> set 0 mode=ap−bridge ssid=test1; enable 0
[admin@AP] interface wireless> print
Flags: X − disabled, R − running
  0  R name="wlan1" mtu=1500 mac−address=00:01:24:70:03:75 arp=enabled
       card−type=Atheros AR5211 2.4/5 GHz mode=ap−bridge ssid="test1"
       frequency=5180 band=5GHz scan−list=default−ism
       supported−rates−a=6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps
       basic−rates−a=6Mbps supported−rates−b=1Mbps,2Mbps,5.5Mbps,11Mbps
       basic−rates−b=1Mbps ack−timeout=default tx−power=default
       default−key−0="" default−key−1="" default−key−2="" default−key−3=""
       station−private−key="" transmit−key−id=0 encryption=none
       used−authentication=open−system accepted−authentication=open−system
       default−authentication=yes default−forwarding=yes 802.1x−enable=no

[admin@AP] interface wireless>
```

Then we need to configure the wireless client interface:

```
[admin@client] interface wireless> set 0 ssid=test1
[admin@client] interface wireless> enable 0
[admin@client] interface wireless> print
  0  R name="wlan1" mtu=1500 mac−address=00:01:24:70:03:33 arp=enabled
       card−type=Atheros AR5211 2.4/5 GHz mode=station ssid="test1"
       frequency=5180 band=5GHz scan−list=default−ism
       supported−rates−a=6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps
       basic−rates−a=6Mbps supported−rates−b=1Mbps,2Mbps,5.5Mbps,11Mbps
       basic−rates−b=1Mbps ack−timeout=default tx−power=default
       default−key−0="" default−key−1="" default−key−2="" default−key−3=""
```

```
       station-private-key="" transmit-key-id=0 encryption=none
       used-authentication=open-system accepted-authentication=open-system
       default-authentication=yes default-forwarding=yes 802.1x-enable=no

[admin@client] interface wireless>
```

Now we can monitor our connection both from the AP:

```
[admin@AP] interface wireless> registration-table
[admin@AP] interface wireless registration-table> print
  # INTERFACE   MAC-ADDRESS       TYPE      PARENT            SIGNAL     TX-RATE
  0 wlan1       00:01:24:70:03:33 radio                       20         6Mbps

[admin@AP] interface wireless registration-table>
```

... and the client:

```
[admin@client] interface wireless>monitor 0
             status: connected-to-ess
               band: 5GHz
          frequency: 5180
            tx-rate: 18Mbps
               ssid: "test1"
              bssid: 00:01:24:70:03:75
    signal-strength: 20
            rx-rate: 6Mbps

[admin@client] interface wireless>
```

Additional Resources

www.atheros.com

Bridge Interface

Document revision 1.2 (12−May−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Overview

MAC level bridging of Ethernet packets is supported. Ethernet, Ethernet over IP (EoIP), Prism, Atheros and RadioLAN interfaces are supported. All 802.11b and 802.11a client wireless interfaces (both ad−hoc and infrastructure or station modes) do not support this because of the limitations of 802.11 − it is possible to bridge over them using the Ethernet over IP protocol (please see documentation on EoIP).

Features include:

- Spanning Tree Protocol (STP)
- Multiple bridge interfaces
- Bridge associations on a per interface basis
- Protocol can be selected to be forwarded or discarded
- MAC address table can be monitored in real time
- IP address assignment for router access
- Bridge interfaces can be firewalled

Specifications

Packages required : *None*
License required : *None*
Home menu level : ***/interface bridge***
Standards and Technologies : *Media Access Control (MAC) Bridges (IEEE801.1D)*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
EoIP (Ethernet over IP) Tunnel Interface
Packet Filter (Firewall) and NAT (Network Address Translation)

Description

Ethernet−like networks (Ethernet, Ethernet over IP, IEEE802.11 Wireless interfaces in AP mode) can be connected together using MAC Bridges. The bridge feature allows the interconnection of stations connected to separate LANs (using EoIP, geographically distributed networks can be bridged as well if any kind of IP network interconnection exists between them) as if they were attached to a single LAN. As bridges are transparent, they do not appear in traceroute list, and no utility can make a distinction between a host working in one LAN and a host working in another LAN if these LANs are bridged (depending on the way the LANs are interconnected, latency and data rate between hosts may vary)

Bridge Interface Setup

Submenu level : **/interface bridge**

Description

To bridge a number of networks into one bridge, a bridge interface should be created, that will group all the bridged interfaces. One MAC address will be assigned to all the bridged interfaces.

Note that you may only assign IP addresses to the bridge interface (the one is created in this submenu level), not the bridged interfaces (the ones which will be grouped in the bridge).

Property Description

name (*name*; default: **bridge*N***) − a descriptive name of the interface
mtu (*integer*; default: **1500**) − Maximum Transmission Unit
arp (disabled | enabled | proxy−arp | reply−only; default: **enabled**) − Address Resolution Protocol setting
mac−address (*read−only*: *MAC address*) − Media Access Control address for the interface
forward−protocols (*multiple choice*: ip, arp, appletalk, ipx, ipv6, other; default: **ip,arp,appletalk,ipx,ipv6,other**) − list of forwarded protocols.

- **other** − all other protocols than appletalk, arp, ip, ipv6, or ipx, e.g., netbeui, vlan, etc.

priority − bridge interface priority (*integer: 0..65535*, default 1). The priority argument is used by Spanning Tree Protocol to determine, which port remains enabled if two ports form a loop

Notes

forwarded−protocols is a simple filter that also affects the locally−destined and locally−originated packets. So disabling **ip** protocol you will not be able to communicate with the router from the bridged interfaces.

Example

To add and enable a bridge interface that will forward all the protocols:

```
[admin@MikroTik] interface bridge> add
[admin@MikroTik] interface bridge> print
Flags: X − disabled, R − running
  1 X  name="bridge2" mtu=1500 arp=enabled mac−address=00:00:00:00:00:00
       forward−protocols=ip,arp,appletalk,ipx,ipv6,other priority=1

[admin@MikroTik] interface bridge> enable 0
```

Port Settings

Submenu level : **/interface bridge port**

Description

The submenu is used to group interfaces in a particular bridge interface

Property Description

interface (*read−only*: *name*) − interface name
bridge (*name*; default: **none**) − the bridge interface the respective interface is grouped in

- **none** − the interface is not grouped

Example

To group **ether1** and **prism1** in the **bridge1** bridge:

```
[admin@MikroTik] interface bridge port> set ether1,prism1 bridge=bridge1
[admin@MikroTik] interface bridge port> print
Flags: X − disabled
  #   INTERFACE BRIDGE
  0   ether1    bridge1
  1   ether2    none
  2   prism1    bridge1

[admin@MikroTik] interface bridge port>
```

Bridge Monitoring

Submenu level : **/interface bridge host**

Property Description

bridge (*read−only*: *name*) − the bridge the entry belonds to
mac−address (*read−only*: *MAC address*) − host's MAC address
on−interface (*read−only*: *name*) − which of the bridged interfaces the host is connected to
age (*read−only*: *time*) − the time since the last packet received from the host

Example

To get the active host table:

```
[admin@MikroTik] interface bridge host> print
Flags: L − local
   BRIDGE              MAC−ADDRESS       ON−INTERFACE       AGE
   bridge1             00:00:B4:5B:A6:58 ether1             4m48s
   bridge1             00:30:4F:18:58:17 ether1             4m50s
 L bridge1             00:50:08:00:00:F5 ether1             0s
 L bridge1             00:50:08:00:00:F6 ether2             0s
   bridge1             00:60:52:0B:B4:81 ether1             4m50s
   bridge1             00:C0:DF:07:5E:E6 ether1             4m46s
   bridge1             00:E0:C5:6E:23:25 prism1             4m48s
   bridge1             00:E0:F7:7F:0A:B8 ether1             1s
[admin@MikroTik] interface bridge host>
```

Bridge Firewall

Submenu level : **/interface bridge firewall**

Description

Traffic between bridged interfaces can be filtered.

Note that packets between bridged interfaces are also passed through the 'generic' **/ip firewall** rules, so they even can be NATted. These rules can be used with real, physical receiving/transmitting interfaces, as well as with bridge interface that simply groups bridged interfaces.

Property Description

mac−src−address (*MAC address*; default: **00:00:00:00:00:00**)− MAC address of the source host
in−interface (*name*; default: **all**) − interface the packet has entered the bridge through

- **all** − any interface

mac−dst−address (*MAC address*; default: **00:00:00:00:00:00**)− MAC address of the destination host
out−interface (*name*; default: **all**) − interface the packet is leaving the bridge through

- **all** − any interface

mac−protocol (all | *integer*; default: **all**) − the MAC protocol of the packet. Most widely used MAC protocols are (many other exist):

- **all** − all the MAC protocols
- **2048** − IP
- **2054** − ARP
- **32821** − RARP
- **32823** − IPX

- **32923** − AppleTalk (EtherTalk)
- **33011** − AppleTalk Address Resolution Protocol (AARP)
- **33169** − NetBEUI
- **34525** − IPv6

src−address (*IP address/netmask*; default: **0.0.0.0**) − source IP address of the packet
dst−address (*IP address/netmask*; default: **0.0.0.0**) − destination IP address of the packet
protocol (all | egp | ggp | icmp | igmp | ip−encap | ip−sec | tcp | udp | *integer*; default: **all**) − IP protocol name/number

- **all** − match all the IP protocols

action (accept | drop | passthrough; default: **accept**) − action to undertake if the packet matches the rule:

- **accept** − accept the packet. No action, i.e., the packet is passed through without undertaking any action, and no more rules are processed
- **drop** − silently drop the packet (without sending the ICMP reject message)
- **passthrough** − ignore this rule. Acts the same way as a disabled rule, except for ability to count packets

Example

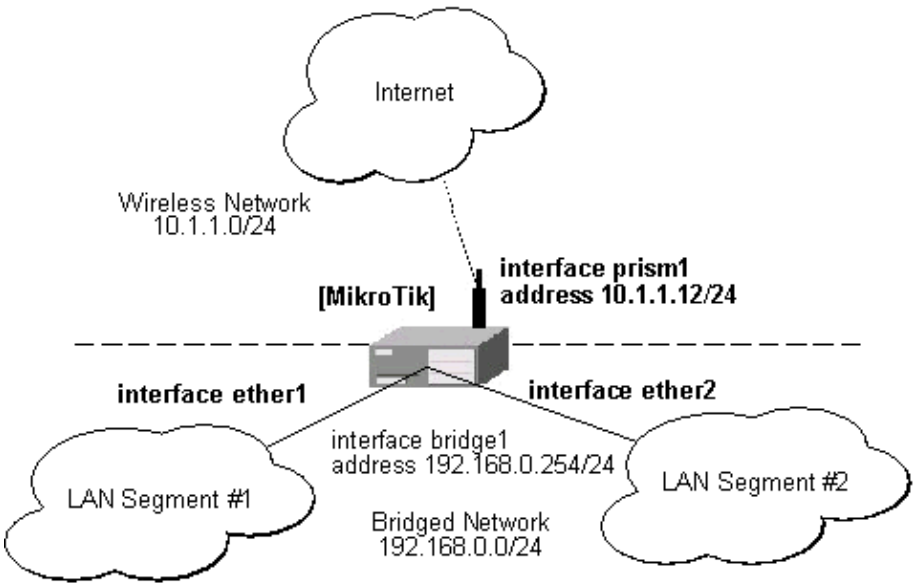
To make a brouter (the router that routes routable (IP in our case) protocols and bridges unroutable protocols), make a rule that drops IP, ARP and RARP traffic (these protocols should be disabled in bridge firewall, not in **forwarded protocols** as in the other case the router will not be able to receive IP packets itself, and thus will not be able to provide routing).

To make bridge drop IP, ARP ad RARP packets:

```
[admin@MikroTik] interface bridge firewall> add mac−protocol=2048 action=drop
[admin@MikroTik] interface bridge firewall> add mac−protocol=2054 action=drop
[admin@MikroTik] interface bridge firewall> add mac−protocol=32821 action=drop
[admin@MikroTik] interface bridge firewall> print
Flags: X − disabled, I − invalid
  0   mac−src−address=00:00:00:00:00:00 in−interface=all
      mac−dst−address=00:00:00:00:00:00 out−interface=all mac−protocol=2048
      src−address=0.0.0.0/0 dst−address=0.0.0.0/0 protocol=all action=drop

  1   mac−src−address=00:00:00:00:00:00 in−interface=all
      mac−dst−address=00:00:00:00:00:00 out−interface=all mac−protocol=2054
      src−address=0.0.0.0/0 dst−address=0.0.0.0/0 protocol=all action=drop

  2   mac−src−address=00:00:00:00:00:00 in−interface=all
      mac−dst−address=00:00:00:00:00:00 out−interface=all mac−protocol=32821
      src−address=0.0.0.0/0 dst−address=0.0.0.0/0 protocol=all action=drop

[admin@MikroTik] interface bridge firewall>
```

Application Example

Assume we want to enable bridging between two Ethernet LAN segments and have the MikroTik router be the default gateway for them:



When configuring the MikroTik router for bridging you should do the following:

1. Add bridge interface
2. Configure the bridge interface
3. Enable the bridge interface
4. Assign an IP address to the bridge interface, if needed

Note that there should be no IP addresses on the bridged interfaces. Moreover, IP address on the bridge interface itself is not required for the bridging to work.

When configuring the bridge settings, each protocol that should be forwarded should be added to the **forward-protocols** list. The **other** protocol includes all protocols not listed before (as VLAN).

```
[admin@MikroTik] interface bridge> add forward-protocols=ip,arp,other
[admin@MikroTik] interface bridge> print
Flags: X - disabled, R - running
  0 X  name="bridge1" mtu=1500 arp=enabled mac-address=00:00:00:00:00:00
       forward-protocols=ip,arp,other priority=1

[admin@MikroTik] interface bridge>
```

The priority argument is used by the Spanning Tree Protocol to determine, which port remains enabled if two ports form a loop.

Next, each interface that should be included in the bridging port table:

```
[admin@MikroTik] interface bridge port> print
Flags: X - disabled
  #   INTERFACE                           BRIDGE
  0   ether1                              none
  1   ether2                              none
  2   ether3                              none
  3   prism1                              none
[admin@MikroTik] interface bridge port> set "0,1" bridge=bridge1
[admin@MikroTik] interface bridge port> print
```

```
Flags: X - disabled
  #   INTERFACE                               BRIDGE
  0   ether1                                  bridge1
  1   ether2                                  bridge1
  2   ether3                                  none
  3   prism1                                  none
[admin@MikroTik] interface bridge port>
```

After setting some interface for bridging, the bridge interface should be enabled in order to start using it:

```
[admin@MikroTik] interface bridge> print
Flags: X - disabled, R - running
  0 X  name="bridge1" mtu=1500 arp=enabled mac-address=00:50:08:00:00:F5
       forward-protocols=ip,arp,other priority=1

[admin@MikroTik] interface bridge> enable 0
[admin@MikroTik] interface bridge> print
Flags: X - disabled, R - running
  0  R name="bridge1" mtu=1500 arp=enabled mac-address=00:50:08:00:00:F5
       forward-protocols=ip,arp,other priority=1

[admin@MikroTik] interface bridge>
```

If you want to access the router through unnumbered bridged interfaces, it is required to add an IP address to the bridge interface:

```
[admin@MikroTik] ip address> add address=192.168.0.254/24 interface=bridge1
[admin@MikroTik] ip address> add address=10.1.1.12/24 interface=prism1
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   192.168.0.254/24   192.168.0.0     192.168.0.255   bridge1
  1   10.1.1.12/24       10.1.1.0        10.1.1.255      prism1
[admin@MikroTik] ip address>
```

> ***Note!*** Assigning an IP address to bridged interfaces **ether1** or **ether2** has no sense. Thus, when you assign an interface to a bridge, you should move its IP address to the bridge interface at the same time!

Hosts on LAN segments #1 and #2 should use IP addresses from the same network 192.168.0.0/24 and have the default gateway set to 192.168.0.254 (MikroTik router).

Additional Bridge Firewall Resources

Links for Bridge Firewall documentation:

http://users.pandora.be/bart.de.schuymer/ebtables/br_fw_ia/br_fw_ia.html

Troubleshooting

- *After I configure the bridge, there is no ping response from hosts on bridged networks.*
  It may take up to 20...30s for bridge to learn addresses and start responding.
- *When I do a Bridge between the Ethernet and Wireless Interface I lost the network connection to the router via Ethernet*
  When network interface is assigned to a bridge, its ip address should be set on the bridge interface

as well. Leaving IP address on a bridged interface has no sense.

- *I have added a bridge interface, but no IP traffic is passed.*
  You should include 'arp' in forwarded protocols list, e.g., 'forward−protocols=ip,arp,other'.

MikroTik RouterOS V2.7 Arlan 655 2.4GHz 2Mbps Wireless Interface

Document revision 1.4 (25−Apr−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The MikroTik RouterOS supports Arlan 655 Wireless Interface client cards. This card fits in the ISA expansion slot and provides transparent wireless communications to other network nodes.

Specifications

Packages required : ***arlan***
License required : *2.4/5GHz Wireless Client*
Home menu level : ***/interface arlan***
Standards and Technologies : *Proprietary*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
Log Management

Installation

Assuming you have all necessary packages and licences installed, you need to load the device driver by issuing **/driver add** command.

Example

To add the driver for Arlan 655 adapter, do the following:

```
[admin@MikroTik]> driver add name=arlan io=0xD000
[admin@MikroTik]> driver print
Flags: I − invalid, D − dynamic
  #   DRIVER                              IRQ IO       MEMORY   ISDN−PROTOCOL
  0 D RealTek 8139
  1   Arlan 655                               0xD000

[admin@MikroTik] driver>
```

Wireless Interface Configuration

Submenu level : **/interface arlan**

Description

The wireless card status can be obtained from the two LEDs: the *Status LED* and the*Activity LED*.

Status	**Activity**	**Description**
Amber	Amber	ARLAN 655 is functional but nonvolatile memory is not configured
Blinking Green	Don't Care	ARLAN 655 not registered to an AP (ARLAN mode only)
Green	Off	Normal idle state
Green	Green Flash	Normal active state
Red	Amber	Hardware failure
Red	Red	Radio failure

Property Description

name (*name*; default: **arlan*N***) − assigned interface name
mtu (*integer*, default: **1500**) − Maximum Transmission Unit
mac−address (*MAC address*) − Media Access Control address
frequency (2412 | 2427 | 2442 | 2457 | 2465, default: **2412**) − channel frequency in MHz
bitrate (1000 | 2000 | 354 | 500, default: **2000**) − data rate in Kbit/s
sid (*integer*, default: **0x13816788**) − System Identifier. Should be the same for all nodes on the radio network. Must be an even number with maximum length 31 character
add−name (*text: 15 byte*, default: "test") − card name (optional)
arp (disabled | enabled | proxy−arp | reply−only, default: **enabled**) − Address Resolution Protocol setting
tma−mode (yes | no, default: **no**) − Network Registration Mode:

- yes

− ARLAN

- no

− NON ARLAN

Example

```
[admin@MikroTik] > interface print
Flags: X - disabled, D - dynamic, R - running
  #    NAME                                                TYPE            MTU
  0  R outer                                               ether           1500
  1  X arlan1                                              arlan           1500
[admin@MikroTik] interface> enable 1
[admin@MikroTik] > interface print
Flags: X - disabled, D - dynamic, R - running
  #    NAME                                                TYPE            MTU
  0  R outer                                               ether           1500
  1  R arlan1                                              arlan           1500
```

More configuration and statistics parameters can be found under the **/interface arlan** menu:

```
[admin@MikroTik] interface arlan> print
Flags: X - disabled, R - running
  0  R name="arlan1" mtu=1500 mac-address=00:40:96:22:90:C8 arp=enabled
       frequency=2412 bitrate=2000 tma-mode=no card-name="test"
       sid=0x13816788

[admin@MikroTik] interface arlan>
```

You can monitor the status of the wireless interface:

```
[admin@MikroTik] interface arlan> monitor 0
      registered: no
    access-point: 00:00:00:00:00:00
        backbone: 00:00:00:00:00:00

[admin@MikroTik] interface arlan>
```

Suppose we want to configure the wireless interface to accomplish registration on the AP with a sid **0x03816788**. To do this, it is enough to change the argument value of **sid** to **0x03816788** and **tma-mode** to **yes**:

```
[admin@MikroTik] interface arlan> set 0 sid=0x03816788 tma-mode=yes
[admin@MikroTik] interface arlan> monitor 0
         registered: yes
    access-point: 00:40:88:23:91:F8
        backbone: 00:40:88:23:91:F9

[admin@MikroTik] interface arlan>
```

Troubleshooting

Keep in mind, that not all combinations of I/O base addresses and IRQ's may work on particular motherboard. It is recommended that you choose an IRQ not used in your system, and then try to find an acceptable I/O base address setting. As it has been observed, the IRQ 5 and I/O 0x300 or 0x180 will work in most cases.

- The driver cannot be loaded because other device uses the requested IRQ.
  Try to set different IRQ using the DIP switches.
- The requested I/O base address cannot be used on your motherboard.
  Try to change the I/O base address using the DIP switches.

- *The pc interface does not show up under the interfaces list*
  Obtain the required license for 2.4/5GHz Wireless Client feature.
- *The wireless card does not register to the AP*
  Check the cabling and antenna alignment.

Additional Resources

www.aironet.com
http://www.comptek.ru:8100/wireless/files/filearlan.html

CISCO/Aironet 2.4GHz 11Mbps Wireless Interface

Document revision 1.3 (11−Jun−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The MikroTik RouterOS supports the following CISCO/Aironet 2.4GHz Wireless ISA/PCI/PC Adapter hardware:

- Aironet ISA/PCI/PC4800 2.4GHz DS 11Mbps Wireless LAN Adapters (100mW)
- Aironet ISA/PCI/PC4500 2.4GHz DS 2Mbps Wireless LAN Adapters (100mW)
- CISCO AIR−PCI340 2.4GHz DS 11Mbps Wireless LAN Adapters (30mW)
- CISCO AIR−PCI/PC350/352 2.4GHz DS 11Mbps Wireless LAN Adapters (100mW)

Specifications

Packages required : ***wireless***
License required : *2.4GHz Wireless Client*
Home menu level : ***/interface pc***
Protocols utilized : *IEEE802.11b (IEEE802.11b)*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
Log Management
Notes on PCMCIA Adapters

Wireless Interface Configuration

Submenu level : **/interface pc**

Description

CISCO/Aironet 2.4GHz card is an interface for wireless networks operating in IEEE 802.11b standard. If the wireless interface card is not registered to an AP, the green status led is blinking fast. If the wireless interface card is registered to an AP, the green status led is blinking slow. To set the wireless interface for working with an access point (register to the AP), typically you should set the following parameters:

- The **service set identifier**. It should match the ssid of the AP. Can be blank, if you want the wireless interface card to register to an AP with any ssid. The ssid will be received from the AP, if the AP is broadcasting its ssid.
- The **data−rate** of the card should match one of the supported data rates of the AP. Data rate 'auto' should work in most cases.

Property Description

name (*name*) − assigned interface name
mtu (*integer: 0..65536*, default: **1500**) − Maximum Transmission Unit
mode (infrastructure | ad−hoc, default: **infrastructure**) − operation mode of the card
rts−threshold (*integer: 0..2312*, default: **2312**) − determines the packet size at which the interface issues a request to send (RTS) before sending the packet. A low value can be useful in areas where many clients are associating with the access point or bridge, or in areas where the clients are far apart and can detect only the access point or bridge and not each other
fragmentation−threshold (*integer: 256..2312*, default: **2312**) − this threshold controls the packet size at which outgoing packets will be split into multiple fragments. If a single fragment transmit error occurs, only that fragment will have to be retransmitted instead of the whole packet. Use a low setting in areas with poor communication or with a great deal of radio interference
tx−power (1 | 5 | 20 | 50 | 100, default: **100**) − transmit power in mW
rx−antenna (both | default | left | right, default: **both**) − receive antennas
tx−antenna (both | default | left | right, default: **both**) − transmit antennas
long−retry−limit (*integer: 0..128*, default: **16**) − specifies the number of times an unfragmented packet is retried before it is dropped
short−retry−limit (*integer: 0..128*, default :**16**) − specifies the number of times a fragmented packet is retried before it is dropped
frequency (default | 2412 | 2417 | 2422 | 2427 | 2432 | 2437 | 2442 | 2447 | 2452 | 2457 | 2462 | 2467 | 2472 | 2484MHz, default: **2437**) − channel frequency in MHz
data−rate (11 | 1 | 2 | 5.5 | auto, default: **1**) − data rate in Mbit/s
ap1 (*MAC address*) − forces association to the specified access point
ap2 (*MAC address*) − forces association to the specified access point
ap3 (*MAC address*) − forces association to the specified access point
ap4 (*MAC address*) − forces association to the specified access point
ssid1 (*text*, default: **tsunami**) − establishes the adapter's service set identifier This value must match the SSID of the system in order to operate in infrastructure mode
ssid2 (*text*, default: "") − service set identifier 2
ssid3 (*text*, default: "") − service set identifier 3
modulation (cck | default | mbok, default: **cck**) − modulation mode
client−name (*text*, default: "") − client name

join−net (*time*, default **10**) − the time, during which the interface in **ad−hoc** mode will try to connect to a network rather then create a new one
- **0** − do not create own network

beacon−period (*integer*, default: **100**) − establishes a beaconing period
arp (disabled | enabled | proxy−arp | reply−only, default: **enabled**) − Address Resolution Protocol
card−type (*read−only: card type*) − your CISCO/Aironet adapter model and type

Example

```
[admin@MikroTik] > interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ether1               ether            1500
  1  X ether2               ether            1500
  2  X pc1                  pc               1500
[admin@MikroTik] interface> set 1 name aironet
[admin@MikroTik] interface> enable aironet
[admin@MikroTik] > interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ether1               ether            1500
  1  X ether2               ether            1500
  2  R aironet              pc               1500
[admin@MikroTik] > interface pc
[admin@MikroTik] interface pc> print
Flags: X − disabled, R − running
  0  R name="aironet" mtu=1500 mac−address=00:40:96:29:2F:80 arp=enabled
       client−name="" ssid1="tsunami" ssid2="" ssid3="" mode=infrastructure
       data−rate=1Mbit/s frequency=2437MHz modulation=cck tx−power=100
       ap1=00:00:00:00:00:00 ap2=00:00:00:00:00:00 ap3=00:00:00:00:00:00
       ap4=00:00:00:00:00:00 rx−antenna=right tx−antenna=right beacon−period=100
       long−retry−limit=16 short−retry−limit=16 rts−threshold=2312
       fragmentation−threshold=2312 join−net=10s card−type=PC4800A 3.65

[admin@MikroTik] interface pc>
```

You can monitor the status of the wireless interface:

```
[admin@MikroTik] interface pc> monitor 0
         synchronized: no
           associated: no
         error−number: 0

[admin@MikroTik] interface pc>
```

Suppose we want to configure the wireless interface to accomplish registration on the AP with a ssid 'mt'. To do this, it is enough to change the argument value of **ssid1** to **mt**:

```
[admin@MikroTik] interface pc> set 0 ssid1 mt
[admin@MikroTik] interface pc> monitor 0
         synchronized: yes
           associated: yes
            frequency: 2412MHz
            data−rate: 11Mbit/s
                 ssid: "mt"
         access−point: 00:02:6F:01:5D:FE
```

```
    access-point-name: ""
       signal-quality: 132
      signal-strength: -82
         error-number: 0

[admin@MikroTik] interface pc>
```

Troubleshooting

Keep in mind, that not all combinations of I/O base addresses and IRQ's may work on particular motherboard. It is recommended that you choose an IRQ not used in your system, and then try to find an acceptable I/O base address setting. As it has been observed, the IRQ 5 and I/O 0x300 or 0x180 will work in most cases.

- The driver cannot be loaded because other device uses the requested IRQ.
  Try to set different IRQ using the DIP switches.
- The requested I/O base address cannot be used on your motherboard.
  Try to change the I/O base address using the DIP switches.
- *The pc interface does not show up under the interfaces list*
  Obtain the required 2.4GHz Wireless Client license.
- *The wireless card does not register to the AP*
  Check the cabling and antenna alignment.

Wireless Network Applications

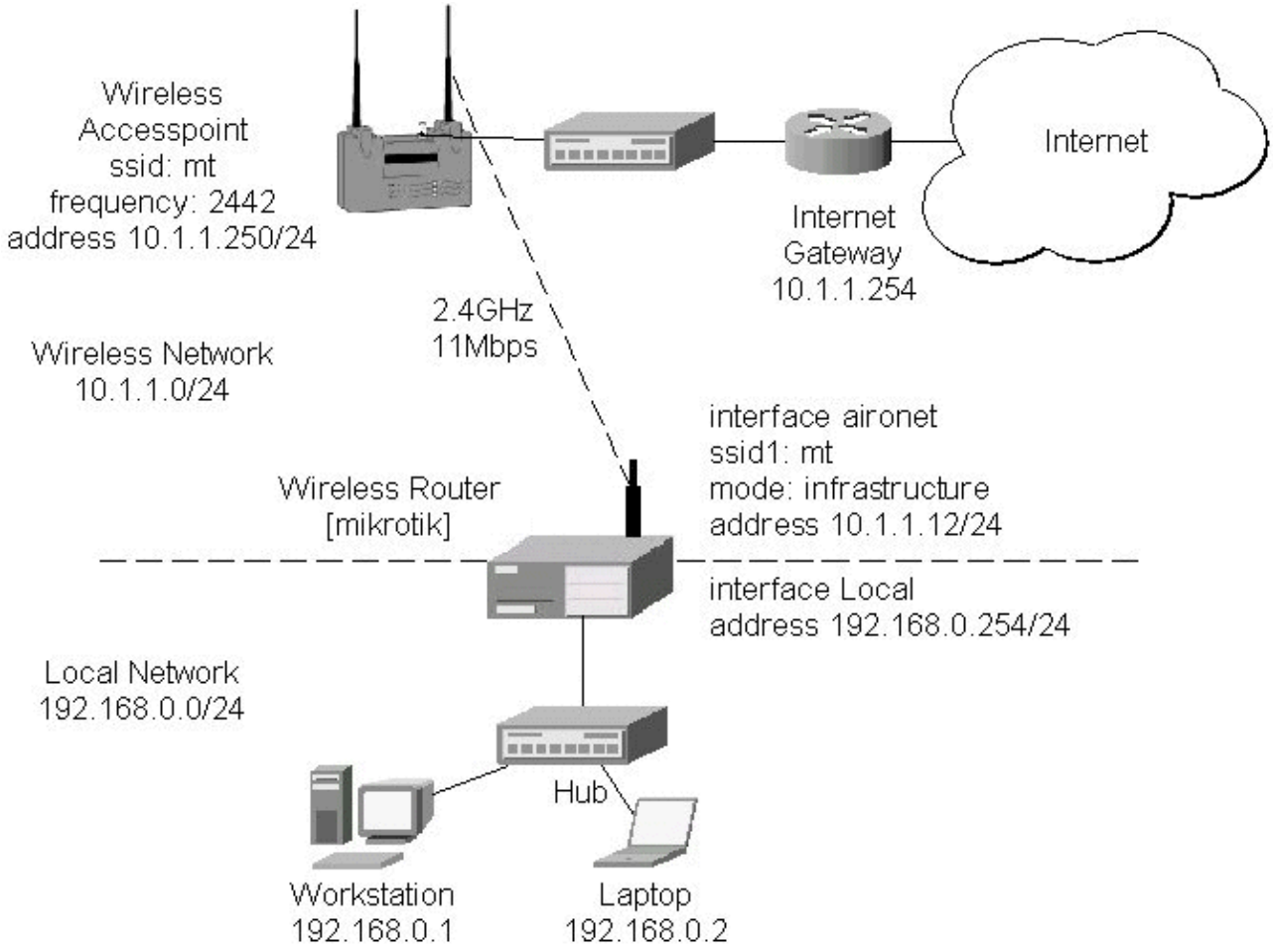
Two possible wireless network configurations are discussed in the following examples:

- Point−to−Multipoint (Wireless Infrastructure)
- Point−to−Point (Peer−to−Peer, or Ad−Hoc Wireless LAN)

Point−to−Multipoint Wireless LAN

Let us consider the following network setup with CISCO/Aironet Wireless Access Point as a base station and MikroTik Wireless Router as a client:

CISCO/Aironet 2.4GHz 11Mbps Wireless Interface



The access point is connected to the wired network's HUB and has IP address from the network 10.1.1.0/24. The minimum configuration required for the AP is:

1. Setting the Service Set Identifier (up to 32 alphanumeric characters). In our case we use ssid "mt".
2. Setting the allowed data rates at 1−11Mbps, and the basic rate at 1Mbps.
3. Choosing the frequency, in our case we use 2442MHz.
4. (For CISCO/Aironet Bridges only) Set Configuration/Radio/Extended/Bridge/mode=access_point. If you leave it to 'bridge_only', it wont register clients.
5. Setting the identity parameters Configuration/Ident: Inaddr, Inmask, and Gateway. These are required if you want to access the AP remotely using telnet or http.

Reminder! *Please note, that the AP is not a router! It has just one network address, and is just like any host on the network. It resembles a wireless−to−Ethernet HUB or bridge. The AP does not route the IP traffic! There is no need to set up the routing table under Configuration/Ident/Routing.*

The frequency argument does not have any meaning, since the frequency of the AP is used. The IP addresses assigned to the wireless interface should be from the network 10.1.1.0/24, e.g.:

```
[admin@MikroTik] ip address> add address 10.1.1.12/24 interface aironet
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.1.1.12/24       10.1.1.0        10.1.1.255      aironet
  1   192.168.0.254/24   192.168.0.0     192.168.0.255   Local
```

```
[admin@MikroTik] ip address>
```

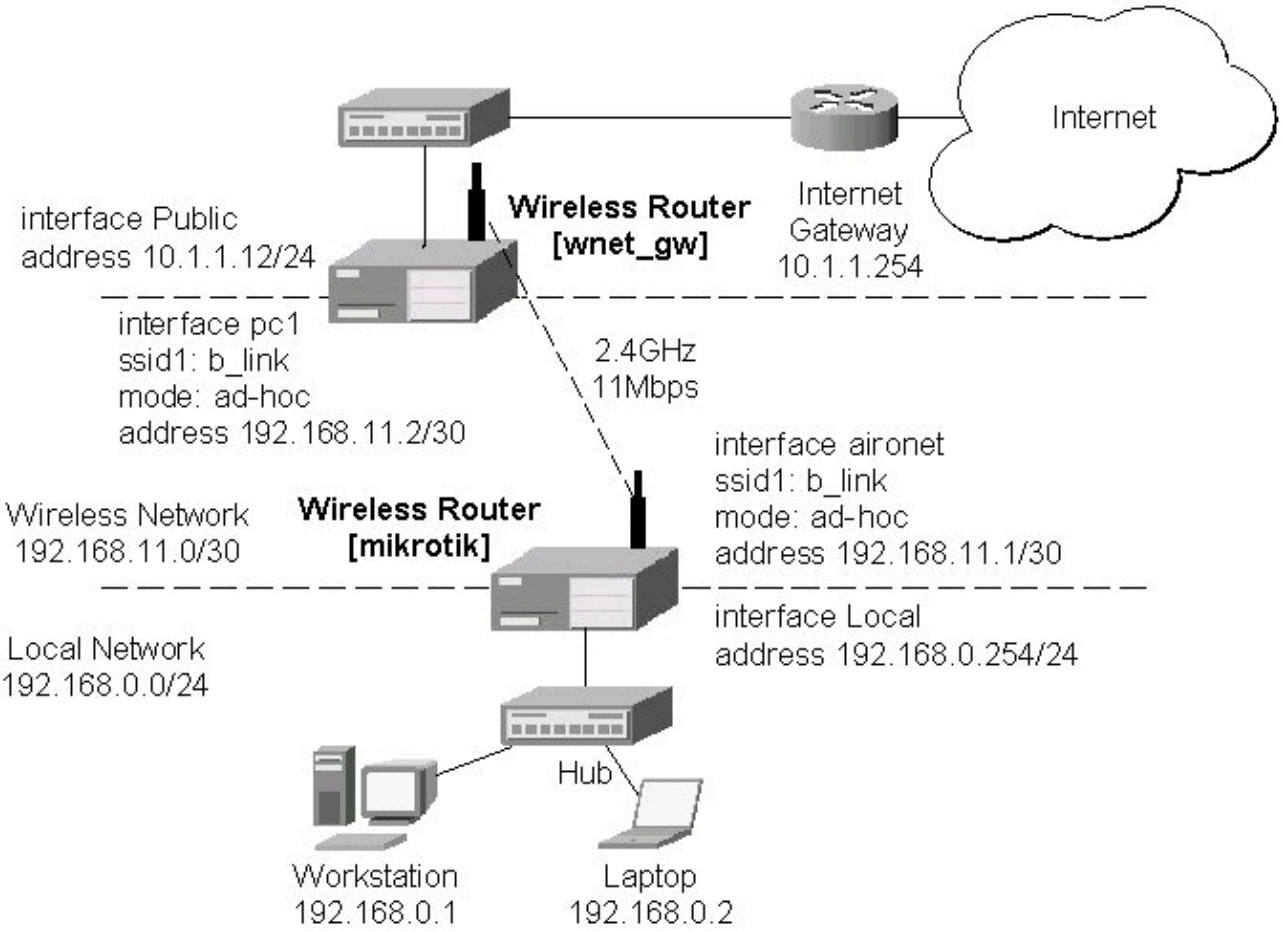
The default route should be set to the gateway router 10.1.1.254 (! not the AP 10.1.1.250 !):

```
[admin@MikroTik] ip route> add gateway=10.1.1.254
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.1.1.254      1        aironet
    1 DC 192.168.0.0/24     r 0.0.0.0         0        Local
    2 DC 10.1.1.0/24        r 0.0.0.0         0        aironet
[admin@MikroTik] ip route>
```

Point−to−Point Wireless LAN

Point−to−point connections using two wireless clients require the wireless cards to operate in **ad−hoc** mode. This mode does not provide the required timing for the cases of long distance (over 20km) links. Thus, the performance of such links is very poor on long distances, and use of infrastructure mode is required, where a wireless client registers to an access point or bridge.

Let us consider the following point−to−point wireless network setup with two MikroTik wireless routers:



To establish a point−to−point link, the configuration of the wireless interface should be as follows:

- A unique Service Set Identificator should be chosen for both ends, say "b_link"
- A channel frequency should be selected for the link, say 2412MHz
- The operation mode should be set to **ad−hoc**
- One of the units (slave) should have wireless interface argument **join−net** set to 0s (never create a network), the other unit (master) should be set to 1s or whatever, say 10s. This will enable the master unit to create a network and register the slave unit to it.

The following command should be issued to change the settings for the pc interface of the master unit:

```
[admin@MikroTik] interface pc> set 0 mode=ad−hoc ssid1=b_link frequency=2442MHz \
\... bitrate=auto
[admin@MikroTik] interface pc>
```

For 10 seconds (this is set by the argument **join−net**) the wireless card will look for a network to join. The status of the card is not synchronized, and the green status light is blinking fast. If the card cannot find a network, it creates its own network. The status of the card becomes **synchronized**, and the green status led becomes solid. The monitor command shows the new status and the MAC address generated:

```
[admin@MikroTik] interface pc> monitor 0
         synchronized: yes
           associated: yes
            frequency: 2442MHz
            data−rate: 11Mbit/s
                 ssid: "b_link"
         access−point: 2E:00:B8:01:98:01
    access−point−name: ""
       signal−quality: 35
      signal−strength: −62
         error−number: 0

[admin@MikroTik] interface pc>
```

The other router of the point−to−point link requires the operation mode set to **ad−hoc**, the System Service Identificator set to "b_link", and the channel frequency set to 2412MHz. If the radios are able to establish RF connection, the status of the card should become **synchronized**, and the green status led should become solid immediately after entering the command:

```
[admin@wnet_gw] interface pc> set 0 mode=ad−hoc ssid1=b_link frequency=2412MHz \
\... bitrate=auto
[admin@wnet_gw] interface pc> monitor 0
         synchronized: yes
           associated: no
            frequency: 2442MHz
            data−rate: 11Mbit/s
                 ssid: "b_link"
         access−point: 2E:00:B8:01:98:01
    access−point−name: ""
       signal−quality: 131
      signal−strength: −83
         error−number: 0

[admin@wnet_gw] interface pc>
```

As we see, the MAC address under the **access−point** parameter is the same as generated on the first router.

If desired, IP addresses can be assigned to the wireless interfaces of the pint−to−point linked routers using a smaller subnet, say 30−bit one:

```
[admin@MikroTik] ip address> add address 192.168.11.1/30 interface aironet
[admin@MikroTik] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   192.168.11.1/30    192.168.11.0    192.168.11.3    aironet
  1   192.168.0.254/24   192.168.0.0     192.168.0.255   Local
[admin@MikroTik] ip address>
```

The second router will have address 192.168.11.2. The network connectivity can be tested by using ping or bandwidth test:

```
[admin@wnet_gw] ip address> add address 192.168.11.2/30 interface aironet
[admin@wnet_gw] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   192.168.11.2/30    192.168.11.0    192.168.11.3    aironet
  1   10.1.1.12/24       10.1.1.0        10.1.1.255      Public
[admin@wnet_gw] ip address> /ping 192.168.11.1
192.168.11.1 pong: ttl=255 time=3 ms
192.168.11.1 pong: ttl=255 time=1 ms
192.168.11.1 pong: ttl=255 time=1 ms
192.168.11.1 pong: ttl=255 ping interrupted
4 packets transmitted, 4 packets received, 0% packet loss
round−trip min/avg/max = 1/1.5/3 ms
[admin@wnet_gw] interface pc> /tool bandwidth−test 192.168.11.1 protocol tcp
                  status: running
              rx−current: 4.61Mbps
    rx−10−second−average: 4.25Mbps
        rx−total−average: 4.27Mbps

[admin@wnet_gw] interface pc> /tool bandwidth−test 192.168.11.1 protocol udp size 1500
                  status: running
              rx−current: 5.64Mbps
    rx−10−second−average: 5.32Mbps
        rx−total−average: 4.87Mbps

[admin@wnet_gw] interface pc>
```

Additional Resources

www.aironet.com
www.cisco.com/warp/public/44/jump/wireless.shtml
Cisco − Cisco Aironet 350 Series
For more information about the CISCO/Aironet PCI/ISA adapter hardware please see the relevant User's Guides and Technical Reference Manuals in *pdf* format:

- 710−003638a0.pdf for PCI/ISA 4800 and 4500 series adapters
- 710−004239B0.pdf for PC 4800 and 4500 series adapters

Documentation about CISCO/Aironet Wireless Bridges and Access Points can be found in archives:

- AP48MAN.exe for AP4800 Wireless Access Point

- BR50MAN.exe for BR500 Wireless Bridge

Cyclades PC300 PCI Adapters

Document revision 1.1 (15−Jul−2003)
This document applies to the MikroTik RouterOS v2.7

Table of Contents

Summary

The MikroTik RouterOS supports the following Cyclades PC300 Adapter hardware:

- RSV/V.35 (RSV models) with 1 or 2 RS−232/V.35 interfaces on standard DB25/M.34 connector, 5Mbps, internal or external clock
- T1/E1 (TE models) with 1 or 2 T1/E1/G.703 interfaces on standard RJ48C connector, Full/Fractional, internal or external clock
- X.21 (X21 models) with 1 or 2 X.21 on standard DB−15 connector, 8Mbps, internal or external clock

Specifications

Packages required : ***synchronous***
License required : ***Sync and Hotspot***
Home menu level : ***/interface cyclades***
Standards and Technologies : *X.21, V.35, T1/E1/G.703, Frame Relay, PPP, Cisco−HDLC*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
Log Management

Synchronous Interface Configuration

Submenu level : **/interface cyclades**

Description

You can install up to four Cyclades PC300 PCI Adapters in one PC box, if you have so many adapter slots and IRQs available.

The Cyclades PC300/RSV Synchronous PCI Adapter comes with a V.35 cable. This cable should work for all standard modems, which have V.35 connections. For synchronous modems, which have a DB−25 connection, you should use a standard DB−25 cable.

Connect a communication device, e.g., a baseband modem, to the V.35 port and turn it on. The MikroTik driver for the Cyclades Synchronous PCI Adapter allows you to unplug the V.35 cable from one modem and plug it into another modem with a different clock speed, and you do not need to restart the interface or router.

Property Description

name (*name*; default: **cyclades*N***) − assigned interface name
mtu (*integer*; default: **1500**) − Maximum Transmission Unit
line−protocol (cisco−hdlc | frame−relay | sync−ppp; default: **sync−ppp**) − line protocol
media−type (E1 | T1 | V24 | V35 | X21; default: **V35**) − the hardware media used for this interface:
clock−rate (*integer*; default: **64000**) − internal clock rate in bps
clock−source (external | internal | tx−internal; default: **external**) − source of the clock
line−code (AMI | B8ZS | HDB3 | NRZ; default: **B8ZS**) − for T1/E1 channels only. Line modulation method:

- **AMI** − Alternate Mark Inversion
- **B8ZS** − Binary 8−Zero Substitution
- **HDB3** − High Density Bipolar 3 Code (ITU−T)
- **NRZ** − Non−Return−To−Zero

framing−mode (CRC4 | D4 | ESF | Non−CRC4 | Unframed; default: **ESF**) − for T1/E1 channels only. The frame mode:

- **Unframed** − do not check frame integrity
- **Non−CRC4** − plain Cyclic Redundancy Check
- **CRC4** − Cyclic Redundancy Check 4−bit (E1 Signaling, Europe)
- **D4** − Fourth Generation Channel Bank (48 Voice Channels on 2 T−1s or 1 T−1c)
- **ESF** − Extended Superframe Format

line−build−out (0dB | 15dB | 22.5dB | 7.5dB; default: **0**) − for T1 channels only. Line Build Out Signal Level
rx−sensitivity (long−haul | short−haul; default: **short−haul**) − for T1/E1 channels only. Receiver sensitivity
active−channels (multiple choice: *integer*; default: **all**) − for T1/E1 channels only. Numbers of active channels (up to 32 for E1 and up to 24 for T1)
chdlc−keepalive (*time*; default: **10s**) − Cisco−HDLC keepalive interval in seconds
frame−relay−dce (yes | no; default: **no**) − specifies whether the device operates in *Data Communication Equipment* mode. The value **yes** is suitable only for TE models
frame−relay−lmi−type (ansi | ccitt; default: **ansi**) − Frame Relay Line Management Interface Protocol type

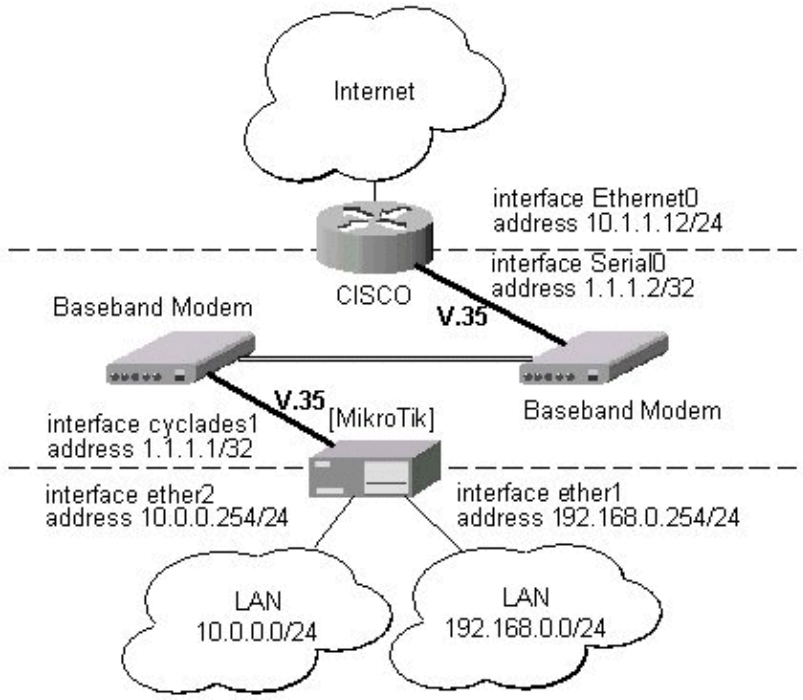
Troubleshooting

- *The cyclades interface does not show up under the interfaces list*
  Obtain the required license for synchronous feature.

- *The synchronous link does not work*
  Check the V.35 cabling and the line between the modems. Read the modem manual.

RSV/V.35 Synchronous Link Applications

Let us consider the following network setup with MikroTik Router connected to a leased line with baseband modems and a CISCO router at the other end:



The driver for the Cyclades PC300/RSV Synchronous PCI Adapter should load automatically. The interface should be enabled according to the instructions given above. The IP addresses assigned to the cyclades interface should be as follows:

```
[admin@MikroTik] ip address> add address=1.1.1.1/32 interface=cyclades1
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.219/24      10.0.0.0        10.0.0.255      ether1
  1   1.1.1.1/32         1.1.1.1         1.1.1.1         cyclades1
  2   192.168.0.254/24   192.168.0.0     192.168.0.255   ether2
[admin@MikroTik] ip address> /ping 1.1.1.2
1.1.1.2 64 byte pong: ttl=255 time=12 ms
1.1.1.2 64 byte pong: ttl=255 time=8 ms
1.1.1.2 64 byte pong: ttl=255 time=7 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 7/9.0/12 ms
[admin@MikroTik] ip address> /tool flood-ping 1.1.1.2 size=1500 count=50
        sent: 50
    received: 50
     min-rtt: 1
     avg-rtt: 1
     max-rtt: 9

[admin@MikroTik] ip address>
```

Note, that for the point−to−point link the network mask is set to 32 bits, the argument **network** is set to the IP address of the other end, and the broadcast address is set to 255.255.255.255. The default route should be set to the gateway router 1.1.1.2:

```
[admin@MikroTik] ip route> add gateway 1.1.1.2 interface cyclades1
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 1.1.1.2         1        cyclades1
    1 DC 10.0.0.0/24        r 0.0.0.0         0        ether1
    2 DC 192.168.0.0/24     r 0.0.0.0         0        ether2
    3 DC 1.1.1.2/32         r 0.0.0.0         0        cyclades1
[admin@MikroTik] ip route>
```

The configuration of the CISCO router at the other end (part of the configuration) is:

```
CISCO#show running−config
Building configuration...

Current configuration:
...
!
interface Ethernet0
 description connected to EthernetLAN
 ip address 10.1.1.12 255.255.255.0
!
interface Serial0
 description connected to MikroTik
 ip address 1.1.1.2 255.255.255.252
 serial restart−delay 1
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.254
!
...
end

CISCO#
```

Send ping packets to the MikroTik router:

```
CISCO#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100−byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round−trip min/avg/max = 28/32/40 ms
CISCO#
```

Additional Resources

For more information about the Cyclades PCI Adapter hardware please see the relevant documentation:

- http://www.cyclades.com/products/svrbas/pc300.php − The product on−line documentation
- Cyclades PC300 Installation Manual − The Installation Manual in .pdf format

Ethernet Interfaces

Document revision 1.7 (21−May−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

MikroTik RouterOS supports the following types of Ethernet Network Interface Cards:
- most NE2000 compatible ISA and PCI cards
- 3com 3c509 ISA cards
- DEC/Intel Tulip chip based cards
- Intel Pro Gigabit PCI cards
  The complete list of supported Ethernet NICs can be found in the Device Driver Management Manual.

Specifications

Packages required : *None*
License required : *Any*
Home menu level : **/interface ethernet**
Standards and Technologies : *IEEE 802.3* (grouper.ieee.org/groups/802/3/)
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
Dynamic Host Configuration Protocol (DHCP) Client and Server

Ethernet Interface Configuration

Submenu level : /**interface ethernet**

Property Description

name (*name*; default: **ether*N***) − assigned interface name
arp (disabled | enabled | proxy−arp | replay−only; default: **enabled**) − Address Resolution Protocol
mtu (*integer*; default: **1500**) − Maximum Transmission Unit
disable−running−check (yes | no; default: **yes**) − disable running check. For 'broken' ethernet cards it is good to disable running status checking (as default)
mac−address (read−only: *MAC address*) − Media Access Control address of the card
auto−negotiation (yes | no; default: **yes**) − when enabled, the interface "advertises" its maximum capabilities to achieve the best connection possible
full−duplex (yes | no; default: **yes**) − defines whether the transmission of data appears in two directions simultaneously
long−cable (no | yes; default: **no**) − changes the cable length setting (only applicable to NS DP83815/6 cards)
speed (1000Mbps | 100Mbps | 10Mbps) − sets the data transmission speed of the interface

Notes

For some Ethernet NICs it is possible to blink the LEDs for 10s. Type **/interface ethernet blink ether1** and watch the NICs to see the one which has blinking LEDs.

Examples

```
[admin@MikroTik] > interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0 X  ether1              ether            1500

[admin@MikroTik] > interface enable ether1
[admin@MikroTik] > interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ether1              ether            1500

[admin@MikroTik] > interface ethernet
[admin@MikroTik] interface ethernet> print
Flags: X − disabled, R − running
  #    NAME                MTU   MAC−ADDRESS       ARP
  0  R ether1              1500  00:50:08:00:00:F5 enabled

[admin@MikroTik] interface ethernet> print detail
Flags: X − disabled, R − running
  0  R name="ether1" mtu=1500 mac−address=00:50:08:00:00:F5 arp=enabled
       disable−running−check=yes

[admin@MikroTik] interface ethernet>
```

Monitoring the Interface Status

To monitor ethernet interface status, you can issue the **/interface ethernet monitor** command.

The output includes following statistics parameters:
status (link−ok | no−link | unknown) − status of the interface, one of the:

- **link−ok** − the card has connected to the network

- **no−link** − the card has not connected to the network
- **unknown** − the connection is not recognized
  data−rate (10 Mbps | 100 Mbps | 1000Mbps) − the actual data rate of the connection
  auto−negotiation (done | incomplete)− fast link pulses (FLP) to the adjacent link station to negotiate the SPEED and MODE of the link.
- **done** − negotiation done
- **incomplete** − negotiation failed
  full−duplex (yes | no) − whether transmission of data occurs in two directions simultaneously

Example

```
[admin@MikroTik] interface ethernet> monitor ether2
              status: link-ok
    auto-negotiation: done
                rate: 100Mbps
         full-duplex: yes
```

Notes

See the **IP Addresses and Address Resolution Protocol (ARP)** section of the manual for information how to add IP addresses to the interfaces.

Additional Resources

http://infocomp.csuchico.edu/metis/internet/topology/topo3_ethernet.htm
http://www.dcs.gla.ac.uk/~liddellj/nct/ethernet_protocol.html Ethernet Information Site

Ethernet over IP (EoIP) Tunnel Interface

Document revision 1.2 (30−May−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Overview

Ethernet over IP (EoIP) Tunneling is a MikroTik RouterOS protocol that creates an Ethernet tunnel between two routers on top of an IP connection. The EoIP interface appears as an Ethernet interface. When the bridging function of the router is enabled, all Ethernet level traffic (all Ethernet protocols) will be bridged just as if there where a physical Ethernet interface and cable between the two routers (with bridging enabled). This protocol makes multiple network schemes possible.

Network setups with EoIP interfaces:

- Possibility to bridge LANs over the Internet
- Possibility to bridge LANs over encrypted tunnels
- Possibility to bridge LANs over 802.11b 'ad−hoc' wireless networks

Specifications

Packages required : *None*
License required : *Basic (DEMO license is limited to 4 tunnels)*
Home menu level : ***/interface eoip***
Standards and Technologies : *GRE (RFC1701)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Bridge Interfaces
PPTP (Point to Point Tunnel Protocol) Interface

Description

An EoIP interface should be configured on two routers that have the possibility for an IP level connection. The EoIP tunnel may run over an IPIP tunnel, a PPTP 128bit encrypted tunnel, a PPPoE connection, or any connection that transports IP.

Specific Properties:

- Each EoIP tunnel interface can connect with one remote router which has a corresponding interface configured with the same 'Tunnel ID'.
- The EoIP interface appears as an Ethernet interface under the interface list.
- This interface supports all features of and Ethernet interface. IP addresses and other tunnels may be run over the interface.
- The EoIP protocol encapsulates Ethernet frames in GRE (IP protocol number 47) packets (just like PPTP) and sends them to the remote side of the EoIP tunnel.
- Maximal count of EoIP tunnels is 65536.

EoIP Setup

Submenu leve : **/interface eoip**

Property Description

name (*name*; default: **eoip−tunnel*N***)− interface name for reference
mtu (*integer*; default: **1500**) − Maximum Transmission Unit. The default value provides maximal compatibility
arp (disabled | enabled | proxy−arp | reply−only; default: **enabled**) − Address Resolution Protocol
tunnel−id (*integer*; default: **0**) − a unique tunnel identifier
remote−address − the IP address of the other side of the EoIP tunnel – must be a MikroTik router

Notes

tunnel−id is method of identifying tunnel. There should not be tunnels with the same **tunnel−id** on the same router. **tunnel−id** on both participient routers must be equal.

mtu should be set to 1500 to eliminate packet refragmentation inside the tunnel (that allows transparent bridging of ethernet−like networks, so that it would be possible to transport full−sized ethernet frame over the tunnrl).

Example

To add and enable an EoIP tunnel named **to_mt2** to the **10.5.8.1** router, specifying **tunnel−id** of **1**:

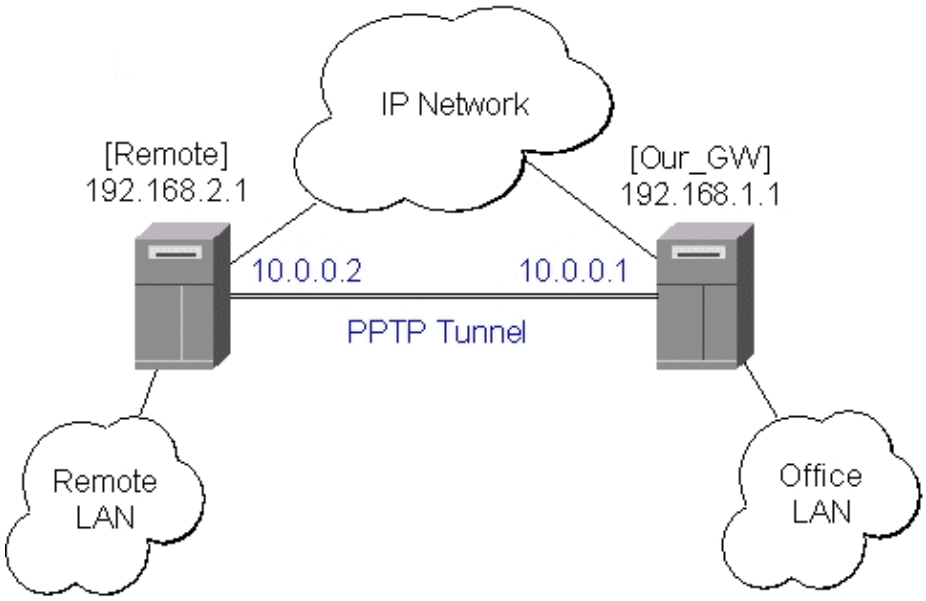
```
[admin@MikroTik] interface eoip> add name=to_mt2 remote−address=10.5.8.1 \
\... tunnel−id 1
[admin@MikroTik] interface eoip> print
Flags: X − disabled, R − running
  0 X  name="to_mt2" mtu=1500 arp=enabled remote−address=10.5.8.1 tunnel−id=1

[admin@MikroTik] interface eoip> enable 0
[admin@MikroTik] interface eoip> print
```

```
Flags: X - disabled, R - running
  0 R  name="to_mt2" mtu=1500 arp=enabled remote-address=10.5.8.1 tunnel-id=1

[admin@MikroTik] interface eoip>
```

EoIP Application Example

Let us assume we want to bridge two networks: 'Office LAN' and 'Remote LAN'. The networks are connected to an IP network through the routers [Our_GW] and [Remote]. The IP network can be a private intranet or the Internet. Both routers can communicate with each other through the IP network.

Our goal is to create a secure channel between the routers and bridge both networks through it. The network setup diagram is as follows:



To make a secure Ethernet bridge between two routers you should:

1. Create a PPTP tunnel between them. Our_GW will be the pptp server:

```
[admin@Our_GW] interface pptp-server> /ppp secret add name=joe service=pptp \
\... password=top_s3 local-address=10.0.0.1 remote-address=10.0.0.2
[admin@Our_GW] interface pptp-server> add name=from_remote user=joe
[admin@Our_GW] interface pptp-server> server set enable=yes
[admin@Our_GW] interface pptp-server> print
Flags: X - disabled, D - dynamic, R - running
  #     NAME                USER         MTU   CLIENT-ADDRESS  UPTIME   ENC...
  0     from_remote         joe
[admin@Our_GW] interface pptp-server>
```

   The Remote router will be the pptp client:

```
[admin@Remote] interface pptp-client> add name=pptp user=joe \
\... connect-to=192.168.1.1 password=top_s3 mtu=1500 mru=1500
[admin@Remote] interface pptp-client> enable pptp
[admin@Remote] interface pptp-client> print
Flags: X - disabled, R - running
```

```
  0  R name="pptp" mtu=1500 mru=1500 connect-to=192.168.1.1 user="joe"
       password="top_s2" profile=default add-default-route=no

[admin@Remote] interface pptp-client> monitor pptp
      status: "connected"
      uptime: 39m46s
    encoding: "none"

[admin@Remote] interface pptp-client>
```

See the PPTP Interface Manual for more details on setting up encrypted channels.

2. Configure the EoIP tunnel by adding the eoip tunnel interfaces at both routers. Use the ip addresses of the pptp tunnel interfaces when specifying the argument values for the EoIP tunnel:

```
[admin@Our_GW] interface eoip> add name="eoip-remote" tunnel-id=0 \
\... remote-address=10.0.0.2
[admin@Our_GW] interface eoip> enable eoip-remote
[admin@Our_GW] interface eoip> print
Flags: X - disabled, R - running
  0    name=eoip-remote mtu=1500 arp=enabled remote-address=10.0.0.2 tunnel-id=0
[admin@Our_GW] interface eoip>

[admin@Remote] interface eoip> add name="eoip" tunnel-id=0 \
\... remote-address=10.0.0.1
[admin@Remote] interface eoip> enable eoip-main
[admin@Remote] interface eoip> print
Flags: X - disabled, R - running
  0   name=eoip mtu=1500 arp=enabled remote-address=10.0.0.1 tunnel-id=0

[Remote] interface eoip>
```

3. Enable bridging between the EoIP and Ethernet interfaces on both routers.

   On the Our_GW:

```
[admin@Our_GW] interface bridge> add forward-protocols=ip,arp,other \
\... disabled=no
[admin@Our_GW] interface bridge> print
Flags: X - disabled, R - running
  0  R name="bridge1" mtu=1500 arp=enabled mac-address=00:00:00:00:00:00
       forward-protocols=ip,arp,other priority=1

[admin@Our_GW] interface bridge> port print
Flags: X - disabled
  #   INTERFACE                           BRIDGE
  0   eoip-remote                         none
  1   office-eth                          none
  2   isp                                 none

[admin@Our_GW] interface bridge> port set "0,1" bridge=bridge1
```

   And the same for the Remote:

```
[admin@Remote] interface bridge> add forward-protocols=ip,arp,other \
\... disabled=no
[admin@Remote] interface bridge> print
Flags: X - disabled, R - running
  0  R name="bridge1" mtu=1500 arp=enabled mac-address=00:00:00:00:00:00
       forward-protocols=ip,arp,other priority=1
```

```
[admin@Remote] interface bridge> port print
Flags: X - disabled
  #   INTERFACE                           BRIDGE
  0   ether                               none
  1   adsl                                none
  2   eoip-main                           none

[admin@Remote] interface bridge> port set "0,2" bridge=bridge1
```

4. Addresses from the same network can be used both in the Office LAN and in the Remote LAN

FarSync X.21 Interface

Document revision 1.4 (23−Sep−2003)
This document applies to the MikroTik RouterOS v2.7

Table of Contents

Summary

The MikroTik RouterOS supports FarSync T−Series X.21 synchronous adapter hardware. These cards provide versatile high performance connectivity to the Internet or to corporate networks over leased lines.

Specifications

Packages required : ***synchronous***
License required : ***Sync and Hotspot***
Home menu level : ***/interface farsync***
Standards and Technologies : *X.21, Frame Relay, PPP*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
Log Management

Synchronous Interface Configuration

Submenu level : **/interface farsync**

Description

You can change the interface name to a more descriptive one using the **set** command. To enable the interface, use the **enable** command.

Property Description

hdlc−keepalive (*time*; default: **10s**) − Cisco HDLC keepalive period in seconds
clock−rate (*integer*; default: **64000**) − the speed of internal clock
clock−source (external | internal; default: **external**) − clock source
disabled (yes | no; default: **yes**) − shows whether the interface is disabled
frame−relay−dce (yes | no; default: **no**) − Operate in Data Communications Equipment mode
frame−relay−lmi−type (ansi | ccitt; default: **ansi**) − Frame Relay Local Management Interface type
line−protocol (cisco−hdlc | frame−relay | sync−ppp; default: **sync−ppp**) − line protocol
media−type (V24 | V35 | X21; default: **V35**) − type of the media
mtu (*integer*; default: **1500**) − Maximum Transmit Unit
name (*text*; default: **farsync*N***) − assigned interface name

Example

```
[admin@MikroTik] > interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ether1               ether            1500
  1 X  farsync1             farsync          1500
  2 X  farsync2             farsync          1500
[admin@MikroTik] interface>
[admin@MikroTik] interface> enable 1
[admin@MikroTik] interface> enable farsync2
[admin@MikroTik] > interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0  R ether1               ether            1500
  1    farsync1             farsync          1500
  2    farsync2             farsync          1500
[admin@MikroTik] interface>farsync
[admin@MikroTik] interface farsync> print
Flags: X − disabled, R − running
  0    name="farsync1" mtu=1500 line−protocol=sync−ppp media−type=V35
       clock−rate=64000 clock−source=external chdlc−keepalive=10s
       frame−relay−lmi−type=ansi frame−relay−dce=no

  1    name="farsync2" mtu=1500 line−protocol=sync−ppp media−type=V35
       clock−rate=64000 clock−source=external chdlc−keepalive=10s
       frame−relay−lmi−type=ansi frame−relay−dce=no

[admin@MikroTik] interface farsync>
```

You can monitor the status of the synchronous interface:

```
[admin@MikroTik] interface farsync> monitor 0
           card−type: T2P FarSync T−Series
               state: running
         firmware−id: 2
    firmware−version: 0.7.0
      physical−media: V35
```

```
               cable: detected
               clock: not-detected
         input-signals: CTS
        output-signals: RTS DTR

[admin@MikroTik] interface farsync>
```

Troubleshooting

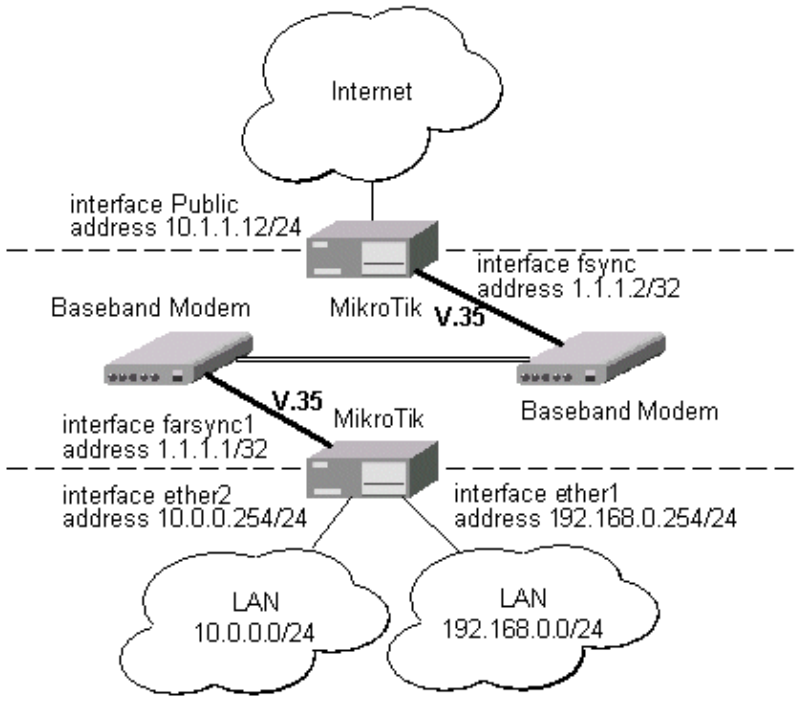
- *The farsync interface does not show up under the interface list*
  Obtain the required license for synchronous feature.
- *The synchronous link does not work*
  Check the cabling and the line between the modems. Read the modem manual.

Synchronous Link Applications

Three possible synchronous line configurations are discussed in the following examples:

MikroTik router to MikroTik router

Let us consider the following network setup with two MikroTik Routers connected to a leased line with baseband modems:



The interface should be enabled according to the instructions given above. The IP addresses assigned to the synchronous interface should be as follows:

```
[admin@MikroTik] ip address> add address 1.1.1.1/32 interface farsync1 \
\... network 1.1.1.2 broadcast 255.255.255.255
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
```

```
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.254/24      10.0.0.254      10.0.0.255      ether2
  1   192.168.0.254/24   192.168.0.254   192.168.0.255   ether1
  2   1.1.1.1/32         1.1.1.2         255.255.255.255 farsync1
[admin@MikroTik] ip address> /ping 1.1.1.2
1.1.1.2 64 byte pong: ttl=255 time=31 ms
1.1.1.2 64 byte pong: ttl=255 time=26 ms
1.1.1.2 64 byte pong: ttl=255 time=26 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 26/27.6/31 ms
[admin@MikroTik] ip address>
```

Note, that for the point−to−point link the network mask is set to 32 bits, the argument **network** is set to the IP address of the other end, and the broadcast address is set to 255.255.255.255. The default route should be set to the gateway router 1.1.1.2:

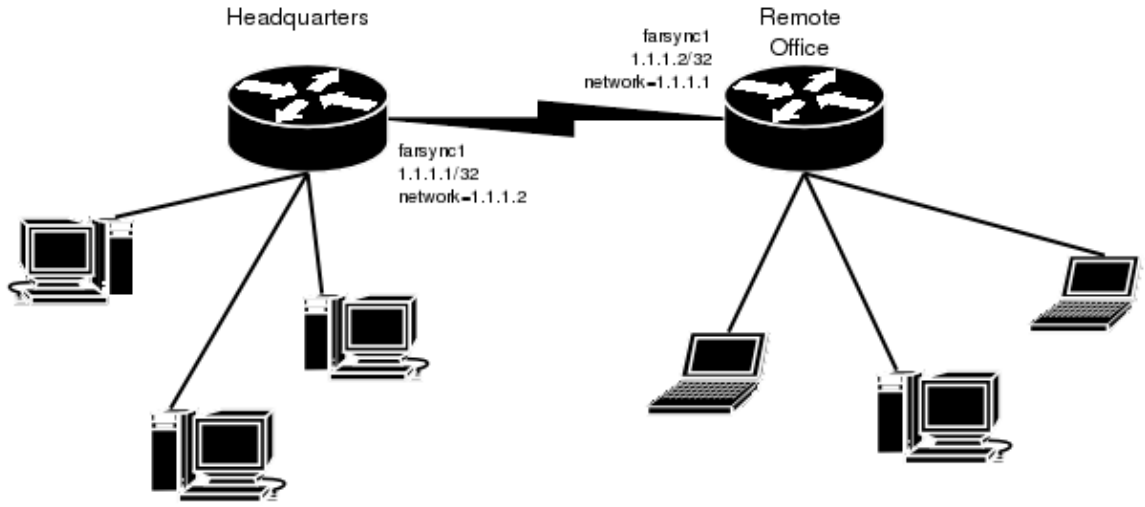
```
[admin@MikroTik] ip route> add gateway 1.1.1.2
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 1.1.1.2         1        farsync1
    1 DC 10.0.0.0/24        r 10.0.0.254      1        ether2
    2 DC 192.168.0.0/24     r 192.168.0.254   0        ether1
    3 DC 1.1.1.2/32         r 0.0.0.0         0        farsync1

[admin@MikroTik] ip route>
```

The configuration of the Mikrotik router at the other end is similar:

```
[admin@MikroTik] ip address> add address 1.1.1.2/32 interface fsync \
\... network 1.1.1.1 broadcast 255.255.255.255
[admin@MikroTik] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.1.1.12/24       10.1.1.12       10.1.1.255      Public
  1   1.1.1.2/32         1.1.1.1         255.255.255.255 fsync
[admin@MikroTik] ip address> /ping 1.1.1.1
1.1.1.1 64 byte pong: ttl=255 time=31 ms
1.1.1.1 64 byte pong: ttl=255 time=26 ms
1.1.1.1 64 byte pong: ttl=255 time=26 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 26/27.6/31 ms
[admin@MikroTik] ip address>
```

MikroTik router to MikroTik router P2P using X21 line

Consider the following example:



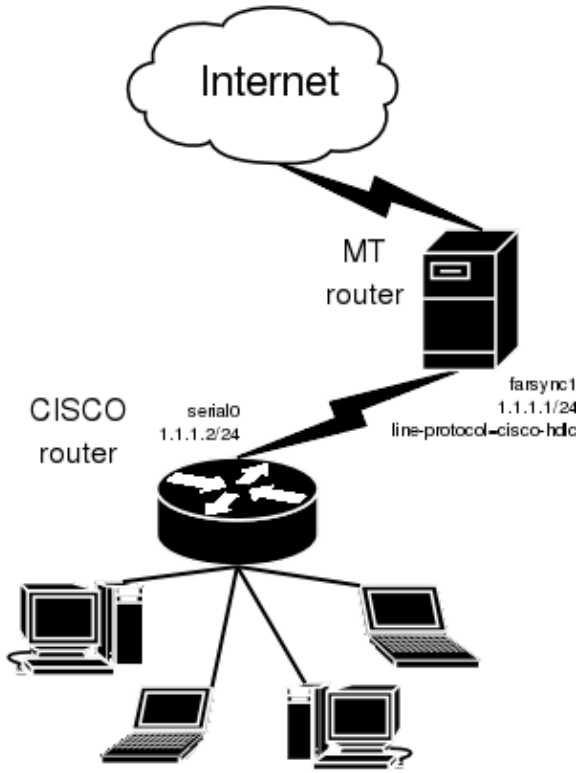
The default value of the property **clock−source** must be changed to **internal** for one of the cards. Both cards must have **media−type** property set to **X21**.

IP addresses configuration on both routers is as follows (by convention, the routers are named **hq** and **office** respectively):

```
[admin@hq] ip address> pri
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   192.168.0.1/24     192.168.0.0     192.168.0.255   ether1
  1   1.1.1.1/32         1.1.1.2         1.1.1.2         farsync1

[admin@hq] ip address>

[admin@office] ip address>
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.112/24      10.0.0.0        10.0.0.255      ether1
  1   1.1.1.2/32         1.1.1.1         1.1.1.1         farsync1

[admin@office] ip address>
```

MikroTik router to Cisco ruter using X21 line

Assume we have the following configuration:



The configuration of MT router is as follows:

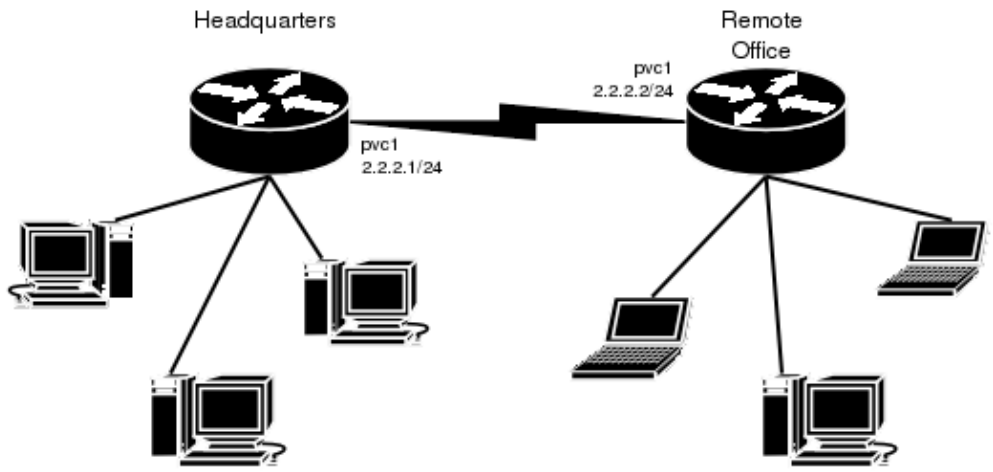
```
[admin@MikroTik] interface farsync> set farsync1 line-protocol=cisco-hdlc \
\... media-type=X21 clock-source=internal
[admin@MikroTik] interface farsync> enable farsync1
[admin@MikroTik] interface farsync> print
Flags: X - disabled, R - running
  0  R name="farsync1" mtu=1500 line-protocol=cisco-hdlc media-type=X21
       clock-rate=64000 clock-source=internal chdlc-keepalive=10s
       frame-relay-lmi-type=ansi frame-relay-dce=no

  1 X  name="farsync2" mtu=1500 line-protocol=sync-ppp media-type=V35
       clock-rate=64000 clock-source=external chdlc-keepalive=10s
       frame-relay-lmi-type=ansi frame-relay-dce=no

[admin@MikroTik] interface farsync>
[admin@MikroTik] interface farsync> /ip address add address=1.1.1.1/24 \
\... interface=farsync1
```

The essential part of the configuration of Cisco router is provided below:

```
interface Serial0
 ip address 1.1.1.2 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 no fair-queue
!
ip classless
ip route 0.0.0.0 0.0.0.0 1.1.1.1
```

MikroTik router to MikroTik router using Frame Relay

Consider the following example:



The default value of the property **clock−source** must be changed to **internal** for one of the cards. This card also requires the property **frame−relay−dce** set to **yes**. Both cards must have **media−type** property set to **X21** and the **line−protocol** set to **frame−relay**.

Now we need to add pvc interfaces:

```
[admin@hq] interface pvc> add dlci=42 interface=farsync1
[admin@hq] interface pvc> print
Flags: X − disabled, R − running
  #    NAME                                                MTU  DLCI INTERFACE
  0 X  pvc1                                                1500 42   farsync1

[admin@hq] interface pvc>
```

Similar routine has to be done also on **office** router:

```
[admin@office] interface pvc> add dlci=42 interface=farsync1
[admin@office] interface pvc> print
Flags: X − disabled, R − running
  #    NAME                                                MTU  DLCI INTERFACE
  0 X  pvc1                                                1500 42   farsync1

[admin@office] interface pvc>
```

Finally we need to add IP addresses to pvc interfaces and enable them.
on the **hq** router:

```
[admin@hq] interface pvc> /ip addr add address 2.2.2.1/24 interface pvc1
[admin@hq] interface pvc> /ip addr print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.112/24      10.0.0.0        10.0.0.255      ether1
  1   192.168.0.1/24     192.168.0.0     192.168.0.255   ether2
  2   2.2.2.1/24         2.2.2.0         2.2.2.255       pvc1

[admin@hq] interface pvc> enable 0
```

```
[admin@hq] interface pvc>
```

and on the **office** router:

```
[admin@office] interface pvc> /ip addr add address 2.2.2.2/24 interface pvc1
[admin@office] interface pvc> /ip addr print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.112/24      10.0.0.0        10.0.0.255      ether1
  1   2.2.2.2/24         2.2.2.0         2.2.2.255       pvc1

[admin@office] interface pvc> enable 0
[admin@office] interface pvc>
```

Now we can monitor the syncronous link status:

```
[admin@hq] interface pvc> /ping 2.2.2.2
2.2.2.2 64 byte ping: ttl=64 time=20 ms
2.2.2.2 64 byte ping: ttl=64 time=20 ms
2.2.2.2 64 byte ping: ttl=64 time=21 ms
2.2.2.2 64 byte ping: ttl=64 time=21 ms
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 20/20.5/21 ms
[admin@hq] interface pvc> /interface farsync monitor 0
           card-type: T2P FarSync T-Series
               state: running-normally
         firmware-id: 2
    firmware-version: 1.0.1
            physical: X.21
               cable: detected
               clock: detected
      input-signals: CTS
     output-signals: RTS,DTR

[admin@hq] interface pvc>
```

Additional Resources

www.farsite.co.uk

FrameRelay (PVC) Interfaces

Document revision 1.3 (30−Jun−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

Frame Relay is a multiplexed interface to packet switched network and is a simplified form of Packet Switching similar in principle to X.25 in which synchronous frames of data are routed to different destinations depending on header information. Frame Relay uses the synchronous HDLC frame format.

Specifications

Packages required : ***synchronous***
License required : *synchronous*
Home menu level : ***/interface pvc***
Standards and Technologies : *Frame Relay* (RFC1490)
Hardware usage : *not significant*

Description

To use Frame Relay interface you must have already working synchronous interface. You can read how to set up synchronous boards supported by Mikrotik RouterOS:

Cyclades PC300 PCI Adapters
Moxa C101 Synchronous interface
Moxa C502 Dual Port Synchronous interface

Configuring Frame Relay Interface

Submenu level : **/interface pvc**

Description

To configure frame relay, you should first set up the synchronous interface, and then the PVC interface.

Property Description

name (*name*; default: **pvc*N***) − assigned name of the interface
mtu (*integer*; default: **1500**) − Maximum Transmission Unit of an interface
dlci (*integer*; default: **16**) − Data Link Connection Identifier assigned to the PVC interface
interface (*name*) − Frame Relay interface

Notes

A DLCI is a channel number (Data Link Connection Identifier)which is attached to data frames to tell the network how to route the data. Frame Relay is "statistically multiplexed", which means that only one frame can be transmitted at a time but many logical connections can co−exist on a single physical line. The DLCI allows the data to be logically tied to one of the connections so that once it gets to the network it knows where to send it.

Frame Relay Configuration Example with Cyclades Interface

Let us consider the following network setup with MikroTik Router with Cyclades PC300 interface connected to a leased line with baseband modems and a CISCO router at the other end.

```
[admin@MikroTik] ip address> add interface=pvc1 address=1.1.1.1 netmask=255.255.255.0
[admin@MikroTik] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   1.1.1.1/24         1.1.1.0         1.1.1.255       pvc1
[admin@MikroTik] ip address>
```

PVC and Cyclades interface configuration

- Cyclades

```
[admin@MikroTik] interface cyclades> print
Flags: X − disabled, R − running
  0  R name="cyclades1" mtu=1500 line−protocol=frame−relay media−type=V35
       clock−rate=64000 clock−source=external line−code=B8ZS framing−mode=ESF
       line−build−out=0dB rx−sensitivity=short−haul frame−relay−lmi−type=ansi
       frame−relay−dce=no chdlc−keepalive=10s

[admin@MikroTik] interface cyclades>
```

- PVC

```
[admin@MikroTik] interface pvc> print
Flags: X − disabled, R − running
  #    NAME                MTU  DLCI INTERFACE
  0  R pvc1                1500 42   cyclades1
[admin@MikroTik] interface pvc>
```

- CISCO router setup

CISCO# show running−config

Building configuration...

Current configuration...

```
...
!
ip subnet−zero
no ip domain−lookup
frame−relay switching
!
interface Ethernet0
 description connected to EthernetLAN
 ip address 10.0.0.254 255.255.255.0
!
interface Serial0
 description connected to Internet
 no ip address
 encapsulation frame−relay IETF
 serial restart−delay 1
 frame−relay lmi−type ansi
 frame−relay intf−type dce
!
interface Serial0.1 point−to−point
 ip address 1.1.1.2 255.255.255.0
 no arp frame−relay
 frame−relay interface−dlci 42
!
...
end.
```

Send ping to MikroTik router

```
CISCO#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100−byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round−trip min/avg/max = 28/31/32 ms
CISCO#
```

Frame Relay Configuration Example with MOXA Interface

Let us consider the following network setup with MikroTik Router with MOXA C502 synchronous interface connected to a leased line with baseband modems and a CISCO router at the other end.

```
[admin@MikroTik] ip address> add interface=pvc1 address=1.1.1.1 netmask=255.255.255.0
[admin@MikroTik] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   1.1.1.1/24         1.1.1.0         1.1.1.255       pvc1
[admin@MikroTik] ip address>
```

PVC and Moxa interface configuration

- Moxa

```
[admin@MikroTik] interface moxa-c502> print
Flags: X - disabled, R - running
  0  R name="moxa1" mtu=1500 line-protocol=frame-relay clock-rate=64000
       clock-source=external frame-relay-lmi-type=ansi frame-relay-dce=no
       cisco-hdlc-keepalive-interval=10s

  1 X  name="moxa-c502-2" mtu=1500 line-protocol=sync-ppp clock-rate=64000
       clock-source=external frame-relay-lmi-type=ansi frame-relay-dce=no
       cisco-hdlc-keepalive-interval=10s

[admin@MikroTik] interface moxa-c502>
```

- PVC

```
[admin@MikroTik] interface pvc> print
Flags: X - disabled, R - running
  #    NAME                MTU  DLCI INTERFACE
  0  R pvc1                1500 42   moxa1
[admin@MikroTik] interface pvc>
```

CISCO router setup

CISCO# show running-config

Building configuration...

Current configuration...

```
...
!
ip subnet-zero
no ip domain-lookup
frame-relay switching
!
interface Ethernet0
 description connected to EthernetLAN
 ip address 10.0.0.254 255.255.255.0
!
interface Serial0
 description connected to Internet
 no ip address
 encapsulation frame-relay IETF
 serial restart-delay 1
 frame-relay lmi-type ansi
 frame-relay intf-type dce
!
interface Serial0.1 point-to-point
 ip address 1.1.1.2 255.255.255.0
 no arp frame-relay
 frame-relay interface-dlci 42
!
...
end.
```

Send ping to MikroTik router

```
CISCO#ping 1.1.1.1
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms
CISCO#
```

MikroTik Router to MikroTik Router

Let us consider the following example:

In this example we will use two Moxa C101 syncronous cards.

Do not forget to set **line-protocol** for synchronous interfaces to **frame-relay**. To achieve proper result, one of the synchronous interfaces must operate in DCE mode:

```
[admin@r1] interface moxa-c101> set 0 frame-relay-dce=yes
[admin@r1] interface moxa-c101> print
Flags: X - disabled, R - running
  0  R name="moxa-c101-1" mtu=1500 line-protocol=frame-relay clock-rate=64000
       clock-source=external frame-relay-lmi-type=ansi frame-relay-dce=yes
       cisco-hdlc-keepalive-interval=10s ignore-dcd=no

[admin@r1] interface moxa-c101>
```

Then we need to add PVC interfaces and IP addresses.

on the **R1**:

```
[admin@r1] interface pvc> add dlci=42 interface=moxa-c101-1
[admin@r1] interface pvc> print
Flags: X - disabled, R - running
  #    NAME                                                MTU  DLCI INTERFACE
  0 X  pvc1                                                1500 42   moxa-c101-1

[admin@r1] interface pvc> /ip address add address 4.4.4.1/24 interface pvc1
```

on the **R2**:

```
[admin@r2] interface pvc> add dlci=42 interface=moxa-c101-1
[admin@r2] interface pvc> print
Flags: X - disabled, R - running
  #    NAME                                                MTU  DLCI INTERFACE
  0 X  pvc1                                                1500 42   moxa-c101-1

[admin@r2] interface pvc> /ip address add address 4.4.4.2/24 interface pvc1
```

Finally we must enable PVC interfaces:

```
[admin@r1] interface pvc> enable pvc1
[admin@r1] interface pvc>

[admin@r2] interface pvc> enable pvc1
[admin@r2] interface pvc>
```

Frame Relay Troubleshooting

- *I cannot ping through the synchronous frame relay interface between MikroTik router and a Cisco router*
  FrameRelay does not support address resolving and IETF encapsulation should be used. Please check the configuration on the Cisco router.

Additional Resources

Frame Relay forum
www2.rad.com/networks/1994/fram_rel/frame.htm

IP over IP (IPIP) Tunnel Interface

Document revision 1.3 (09−May−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The IPIP tunneling implementation on the MikroTik RouterOS is RFC 2003 compliant. IPIP tunnel is a simple protocol that encapsulates IP packets in IP to make a tunnel between two routers. The IPIP tunnel interface appears as an interface under the interface list. Many routers, including Cisco and Linux based, support this protocol. This protocol makes multiple network schemes possible.

IPIP tunneling protocol adds the following possibilities to a network setups:

- to tunnel Intranets over the Internet
- to use it instead of using source routing

Specifications

Packages required : *None*
License required : *Any*
Home menu level : ***/interface ipip***
Standards and Technologies : *IPIP* (RFC2003)
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
Log Management

IPIP Setup

Submenu level : **/interface ipip**

Description

An IPIP interface should be configured on two routers that have the possibility for an IP level connection and are RFC 2003 compliant. The IPIP tunnel may run over any connection that transports IP. Each IPIP tunnel interface can connect with one remote router that has a corresponding interface configured. An unlimited number of IPIP tunnels may be added to the router. For more details on IPIP tunnels, see RFC 2003.

Property Description

name (*name*; default: **ipip*N***) − interface name for reference
mtu (*integer*; default: **1480**) − Maximum Transmission Unit. Should be set to 1480 bytes to avoid fragmentation of packets. May be set to 1500 bytes if mtu path discovery is not working properly on links.
local−address (*IP address*) − Local address on router which sends IPIP traffic to the remote side.
remote−address (*IP address*) − The IP address of the other side of the IPIP tunnel − may be any RFC 2003 compliant router.

Use **/ip address add** command to assign an IP address to the IPIP interface.

There is no authentication or 'state' for this interface. The bandwidth usage of the interface may be monitored with the **monitor** feature from the **interface** menu.

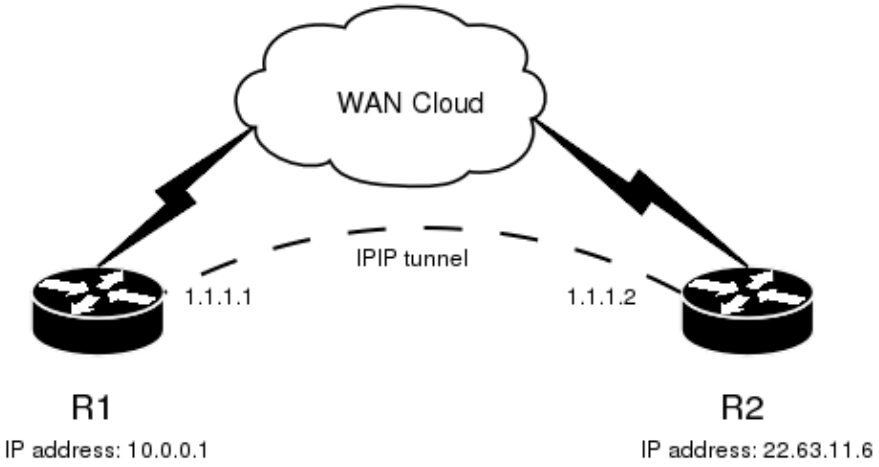
Notes

MikroTik RouterOS IPIP implementation has been tested with Cisco 1005. The sample of the Cisco 1005 configuration is given below:

```
interface Tunnel0
 ip address 10.3.0.1 255.255.255.0
 tunnel source 10.0.0.171
 tunnel destination 10.0.0.204
 tunnel mode ipip
```

IPIP Application Example

Suppose we want to add an IPIP tunnel between routers **R1** and **R2**:



First we need to configure IPIP interfaces and then add IP addresses to them.
The configuration for router **R1** is as follows:

```
[admin@MikroTik] interface ipip> add
local-address: 10.0.0.1
remote-address: 22.63.11.6
[admin@MikroTik] interface ipip> print
Flags: X - disabled, R - running
  #    NAME                                MTU   LOCAL-ADDRESS   REMOTE-ADDRESS
  0 X  ipip1                               1480  10.0.0.1        22.63.11.6

[admin@MikroTik] interface ipip> en 0
[admin@MikroTik] interface ipip> /ip address add address 1.1.1.1/24 interface=ipip1
```

The configuration of the **R2** is shown below:

```
[admin@MikroTik] interface ipip> add local-address=22.63.11.6 remote-address=10.
0.0.1
[admin@MikroTik] interface ipip> print
Flags: X - disabled, R - running
  #    NAME                                MTU   LOCAL-ADDRESS   REMOTE-ADDRESS
  0 X  ipip1                               1480  22.63.11.6      10.0.0.1

[admin@MikroTik] interface ipip> enable 0
[admin@MikroTik] interface ipip> /ip address add address 1.1.1.2/24 interface=ipip1
```

Now both routers can ping each other:

```
[admin@MikroTik] interface ipip> /ping 1.1.1.2
1.1.1.2 64 byte ping: ttl=64 time=24 ms
1.1.1.2 64 byte ping: ttl=64 time=19 ms
1.1.1.2 64 byte ping: ttl=64 time=20 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 19/21.0/24 ms
[admin@MikroTik] interface ipip>
```

Additional Resources

Links for IPIP documentation:
http://www.ietf.org/rfc/rfc1853.txt?number=1853

http://www.ietf.org/rfc/rfc2003.txt?number=2003
http://www.ietf.org/rfc/rfc1241.txt?number=1241

ISDN Interface

Document revision 1.3 (20−Mar−2003)
This document applies to MikroTik RouterOS V2.7

Table of Contents

Summary

The MikroTik router can act as an ISDN client for dialing out, or as an ISDN server for accepting incoming calls. The dial−out connections may be set as dial−on−demand or as permanent connections (simulating a leased line). The remote IP address (provided by the ISP) can be used as the default gateway for the router.

Specifications

Packages required : ***isdn, ppp***
License required : *Any*
Home menu level : ***/interface isdn−server, /interface isdn−client***
Protocols utilized : *PPP (RFC1661)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
Log Management

Supported adapters and appropriate driver names

MikroTik Router OS supports passive PCI adapters with Siemens chipset:

- Eicon.Diehl Diva − **diva**
- Sedlbauer Speed − **sedlbauer**
- ELSA Quickstep 1000 − **quickstep**
- NETjet − **netjet**
- Teles − **teles**
- Dr. Neuhaus Niccy − **niccy**
- AVM − **avm**
- Gazel − **gazel**
- HFC 2BDS0 based adapters − **hfc**
- W6692 based adapters − **w6692**

For example, for the HFC based PCI card, it is enough to use **/driver add name=hfc** command to get the driver loaded.

Notes

ISA ISDN adapters are **not** supported.

ISDN Hardware and Software Installation

Please install the ISDN adapter into the PC accordingly the instructions provided by the adapter manufacturer.

Appropriate packages have to be downloaded from MikroTik's web page www.mikrotik.com. After all, the ISDN driver should be loaded using the **/driver add** command.

Property Description

name (*name*) − Name of the driver.
isdn−protocol (euro | german, default: **euro**) − Data channel protocol

ISDN Channels

ISDN channels are added to the system automatically when the ISDN card driver is loaded. Each channel corresponds to one physical 64K ISDN data channel.

The list of available ISDN channels can be viewed using the **/isdn−channels print** command. The channels are named **channel1**, **channel2**, and so on. E.g., if you have two ISDN channels, and one of them currently used by an ISDN interface, but the other available, the output should look like this:

```
[admin@MikroTik] isdn-channels> print
Flags: X - disabled, E - exclusive
  #    NAME                    CHANNEL    DIR.. TYPE  PHONE
  0    channel1                0
  1    channel2                1
[admin@MikroTik] isdn-channels>
```

ISDN channels are very similar to PPP serial ports. Any number of ISDN interfaces can be configured on a single channel, but only one interface can be enabled for that channel at a time. It means that every ISDN channel is either available or used by an ISDN interface.

MSN and EAZ numbers

In Euro-ISDN a subscriber can assign more than one ISDN number to an ISDN line. For example, an ISDN line could have the numbers 1234067 and 1234068. Each of these numbers can be used to dial the ISDN line. These numbers are referred to as Multiple Subscriber Numbers (MSN).

A similar, but separate concept is EAZ numbering, which is used in German ISDN networking. EAZ number can be used in addition to dialed phone number to specify the required service.

For dial-out ISDN interfaces, MSN/EAZ number specifies the outgoing phone number (the calling end). For dial-in ISDN interfaces, MSN/EAZ number specifies the phone number that will be answered. If you are unsure about your MSN/EAZ numbers, leave them blank (it is the default).

For example, if your ISDN line has numbers 1234067 and 1234068, you could configure your dial-in server to answer only calls to 1234068, by specifying **1234068** as your MSN number. In a sense, MSN is just your phone number.

ISDN Client Interface Configuration

Submenu Level: **/interface isdn-client**

Description

The ISDN client is used to connect to remote dial-in server (probably ISP) via ISDN. To set up an ISDN dial-out connection, use the ISDN dial-out configuration menu under the submenu.

ISDN client interfaces can be added using the **add** command:

Property Description

name (*name*, default: **isdn-out*X***, where x = 1, 2, ...) - interface name
mtu (*integer*; default: **1500**) - Maximum Transmission Unit
mru (*integer*; default: **1500**) - Maximum Receive Unit
phone (*number*; default: "") - phone number to dial
msn (*number*; default: "") - MSN/EAZ of ISDN line provided by the line operator
dial-on-demand (yes | no; default: **no**) - use dialing on demand
l2-protocol (hdlc | x75i | x75ui | x75bui; default: **hdlc**) - level 2 protocol to be used
user (*name*) - user name that will be provided to the remote server
password (*password*) - password that will be provided to the remote server
add-default-route (yes |no; default: **no**) - add default route to remote host on connect

profile (*name*; default: **default**) − profile to use when connecting to the remote server
use−peer−dns (yes | no; default: **no**) − Use or not peer DNS.
bundle−128K (yes | no; default: **yes**) − use both channels instead of just one

Example

```
[admin@MikroTik] interface isdn−client> add msn="142" user="test" \
\... password="test" phone="144" bundle−128K=no
[admin@MikroTik] interface isdn−client> print
Flags: X − disabled, R − running
  0 X  name="isdn−out1" mtu=1500 mru=1500 msn="142" user="test"
       password="test" profile=default phone="144" l2−protocol=hdlc
       bundle−128K=no dial−on−demand=no add−default−route=no use−peer−dns=no

[admin@MikroTik] interface isdn−client>
```

ISDN Server Interface Configuration

Submenu level : **/interface isdn−client**

Description

ISDN server is used to accept remote dial−in connections from ISDN clients. ISDN server interfaces can be added using the **add** command:

Property Description

name (*name*, default: **isdn−in*X***, where x = 1, 2, ...) − interface name
mtu (*integer*, default: **1500**) − Maximum Transmission Unit
mru (*integer*, default: **1500**) − Maximum Receive Unit
phone (*number*, default: "") − phone number to dial
msn (*number*, default: "") − MSN/EAZ of ISDN line provided by the line operator
l2−protocol (hdlc | x75i | x75ui | x75bui, default: **hdlc**) − level 2 protocol to be used
profile (*name*, default: **default**) − profile to use when connecting to the remote server
bundle−128K (yes | no, default: **yes**) − use both channels instead of just one
authentication (pap | chap | mschap1 | mschap2, default: **mschap2**, **mschap1**,**chap**,**pap**) − Used authentication

Example

A sample printout of ISDN server interface is as follows:

```
[admin@MikroTik] interface isdn−server> add msn="142" bundle−128K=no
[admin@MikroTik] interface isdn−server> print
Flags: X − disabled, R − running
  0 X  name="isdn−in1" mtu=1500 mru=1500 msn="142"
       authentication=mschap2,chap,pap profile=default l2−protocol=x75bui
       bundle−128K=no

[admin@MikroTik] interface isdn−server>
```

Troubleshooting

- *The driver could not be loaded or the client/server don't work.*
  There are some older motherboards, which don't support isdn cards. Try to change the motherboard.
- *The ISDN channels do not show up in the isdn−channel list.*
  Check if you have loaded the driver with the **/driver add** command and if you have the **isdn** and the **ppp** packages installed.
- *The ISDN client does not connect, the isdn server doesn't answer a call.*
  Check if you have specified the **msn** and **phone** correctly.

ISDN Examples

The following examples of ISDN applications are discussed below:

- ISDN Dial−out
- ISDN Dial−in
- ISDN Backup

ISDN Dial−out

Dial−out ISDN connections allow a local router to connect to a remote dial−in server (ISP's) via ISDN.

Let's assume you would like to set up a router that connects your local LAN with your ISP via ISDN line. First you should load the corresponding ISDN card driver. Supposing you have an ISDN card with a **W6692**−based chip:

```
[admin@MikroTik]> /driver add name=w6692
```

Now additional channels should appear. Assuming you have only one ISDN card driver loaded, you should get following:

```
[admin@MikroTik] isdn−channels> print
Flags: X − disabled, E − exclusive
  #    NAME                      CHANNEL    DIR.. TYPE  PHONE
  0    channel1                  0
  1    channel2                  1
[admin@MikroTik] isdn−channels>
```

Suppose you would like to use dial−on−demand to dial your ISP and automatically add a default route to it. Also, you would like to disconnect when there is more than 30s of network inactivity. Your ISP's phone number is 12345678 and the user name for authentication is 'john'. Your ISP assigns IP addresses automatically. Add an outgoing ISDN interface and configure it in the following way:

```
[admin@mikrotik]> /interface isdn−client add name="isdn−isp" phone="12345678"
user="john" password="31337!)" add−default−route=yes dial−on−demand=yes
[admin@MikroTik] > /interface isdn−client print
Flags: X − disabled, R − running
  0 X  name="isdn−isp" mtu=1500 mru=1500 msn="" user="john" password="31337!)"
       profile=default phone="12345678" l2−protocol=hdlc bundle−128K=no
       dial−on−demand=yes add−default−route=yes use−peer−dns=no
```

Configure PPP profile.

```
[admin@MikroTik] ppp profile> print
Flags: * − default
  0 * name="default" local−address=0.0.0.0 remote−address=0.0.0.0
      session−timeout=0s idle−timeout=0s use−compression=no
      use−vj−compression=yes use−encryption=no require−encyrption=no only−one=no
      tx−bit−rate=0 rx−bit−rate=0 incoming−filter="" outgoing−filter=""

[admin@Mikrotik] ppp profile> set default idle−timeout=30s
```

(If you would like to remain connected all the time, i.e., as a leased line, then set the **idle−timeout** to 0s.)

All that remains is to enable the interface:

```
[admin@MikroTik] /interface set isdn−isp disabled=no
```

You can monitor the connection status with the following command:

```
[admin@MikroTik] /interface isdn−client monitor isdn−isp
```

ISDN Dial−in

Dial−in ISDN connections allow remote clients to connect to your router via ISDN.

Let us assume you would like to configure a router for accepting incoming ISDN calls from remote clients. You have an ethernet card connected to the LAN, and an ISDN card connected to the ISDN line. First you should load the corresponding ISDN card driver. Supposing you have an ISDN card with an HFC chip:

```
[admin@MikroTik] /driver add name=hfc
```

Now additional channels should appear. Assuming you have only one ISDN card driver loaded, you should get the following:

```
[admin@MikroTik] isdn−channels> print
Flags: X − disabled, E − exclusive
  #    NAME                      CHANNEL    DIR.. TYPE  PHONE
  0    channel1                  0
  1    channel2                  1
[admin@MikroTik] isdn−channels>
```

Add an incoming ISDN interface and configure it in the following way:

```
[admin@MikroTik] interface isdn−server> add msn="7542159" \
\... authentication=chap,pap bundle−128K=no
[admin@MikroTik] interface isdn−server> print
Flags: X − disabled
  0 X  name="isdn−in1" mtu=1500 mru=1500 msn="7542159" authentication=chap,pap
       profile=default l2−protocol=hldc bundle−128K=no
```

Configure PPP settings and add users to router's database.

```
[admin@MikroTik] ppp profile> print
Flags: * − default
  0 * name="default" local−address=0.0.0.0 remote−address=0.0.0.0
      session−timeout=0s idle−timeout=0s use−compression=no
      use−vj−compression=yes use−encryption=no require−encyrption=no only−one=no
```

```
      tx-bit-rate=0 rx-bit-rate=0 incoming-filter="" outgoing-filter=""
[admin@Mikrotik] ppp profile> set default idle-timeout=5s local-address=10.99.8.1 \
\... remote-address=10.9.88.1
```

Add user 'john' to the router's user database. Assuming that the password is '31337!)':

```
[admin@MikroTik] ppp secret> add name=john password="31337!)" service=isdn
[admin@MikroTik] ppp secret> print
[admin@ISDN] ppp secret> print
Flags: X - disabled
  #   NAME              SERVICE CALLER-ID        PASSWORD         PROFILE
  0   john              isdn                     31337!)          default
[admin@MikroTik] ppp secret>
```

Check the status of the ISDN server interface and wait for the call:

```
[admin@MikroTik] interface isdn-server> monitor isdn-in1

    status: Waiting for call...
```

ISDN Backup

Backup systems are used in specific cases, when you need to maintain a connection, even if a fault occurs. For example, if someone cuts the wires, the router can automatically connect to a different interface to continue its work. Such a backup is based on an utility that monitors the status of the connection − netwatch, and a script, which runs the netwatch.

Description

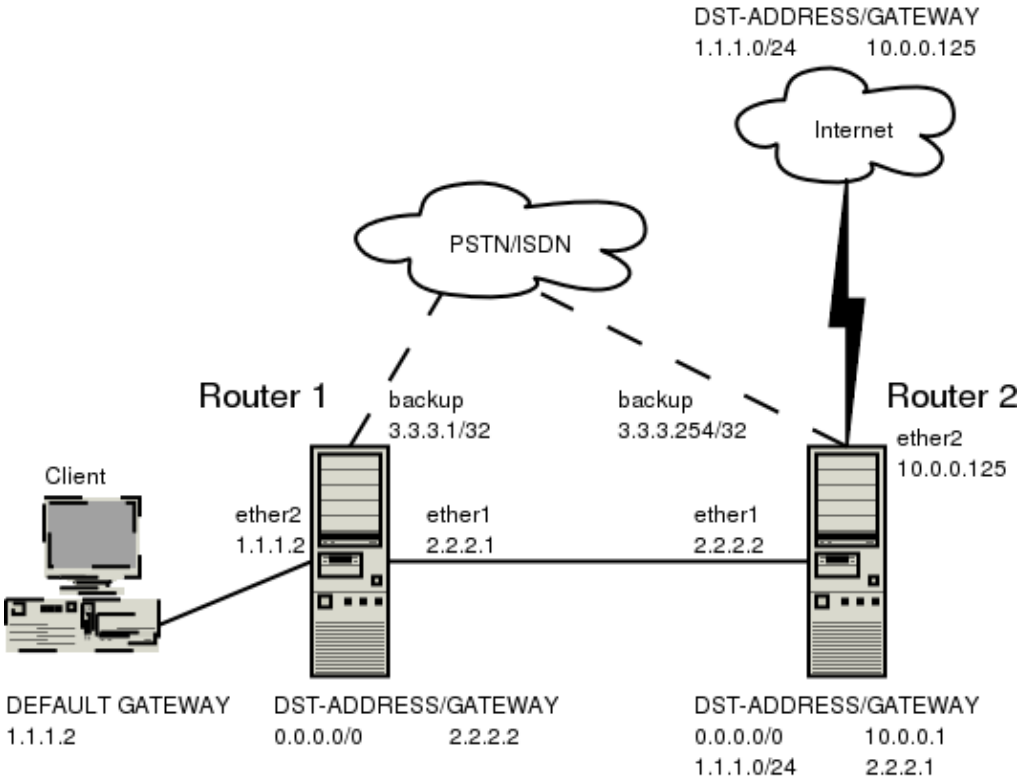
This is an example of how to make simple router backup system. In this example we'll use an ISDN connection for purpose to backup a standard ethernet connection. You can, however, use instead of the ISDN connecion anything you need − PPP, for example. When the ethernet fail (the router nr.1 cannot ping the router nr.2 to 2.2.2.2 (see picture) the router nr.1 will establish an ISDN connection, so−called backup link, to continue comunicating with the nr.2.

Note

You must keep in mind, that in our case there are just two routers, but this system can be extended to support more different networks.

Example

The backup system example is shown in the following picture:



In this case the **backup** interface is an ISDN connection, but in real applications it can be substituted by a particular connection. Follow the instructions below on how to set up the backup link:

- First, you need to set up ISDN connection.

To use ISDN, the ISDN card driver must be loaded:

```
[admin@MikroTik] driver> add name=hfc
```

The PPP connection must have a new user added to the routers one and two:

```
[admin@Mikrotik] ppp secret> add name=backup password=backup service=isdn
```

An ISDN server and PPP profile must be set up on the second router:

```
[admin@MikroTik] ppp profile> set default local-address=3.3.3.254 remote-address=3.3.3.1
[admin@MikroTik] interface isdn-server> add name=backup msn=7801032
```

An ISDN client must be added to the first router:

```
[admin@MikroTik] interface isdn-client>
add name=backup user="backup" password="backup" phone=7801032 msn=7542159
```

- Then, you have to set up Static Routes

Use the **/ip route add** command to add the required static routes and comments to them. Comments are required for references in scrips.

The *first* router:

```
[admin@Mikrotik] ip route> add gateway 2.2.2.2 comment "route1"
```

The *second* router:

```
[admin@Mikrotik] ip route> add gateway 2.2.2.1 comment "route1" dst-address 1.1.1.0/24
```

- And finally, you have to add scripts.

Add scripts in the submenu **system script** using the following commands:

The *first* router:

```
[admin@Mikrotik] system script> add name=connection_down \
\... source={/interface enable backup; /ip route set route1 gateway 3.3.3.254}
[admin@Mikrotik] system script> add name=connection_up \
\... source={/interface disable backup; /ip route set route1 gateway 2.2.2.2}
```

The *second* router:

```
[admin@Mikrotik] system script> add name=connection_down \
\... source={/ip route set route1 gateway 3.3.3.1}
[admin@Mikrotik] system script> add name=connection_up \
\... source={/ip route set route1 gateway 2.2.2.1}
```

- To get all above listed to work, set up Netwatch utility

To use netwatch, you need the advanced tools feature package installed. Please upload it to the router and reboot. When installed, the advanced-tools package should be listed under the **/system package print** list.

Add the following settings to the first router:

```
[admin@Mikrotik] tool netwatch> add host=2.2.2.1 interval=5s \
\... up-script=connection_up down-script=connection_down
```

Add the following settings to the second router:

```
[admin@Mikrotik] tool netwatch> add host=2.2.2.2 interval=5s \
\... up-script=connection_up down-script=connection_down
```

Additional Resources

PPP over ISDN
rfc3057 - ISDN Q.921-User Adaptation Layer
Multiprotocol Interconnect on X.25 and ISDN in the Packet Mode

Layer 2 Tunnel Protocol (L2TP)

Document revision 1.6 (19−May−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

L2TP (Layer 2 Tunnel Protocol) supports encrypted tunnels over IP. The MikroTik RouterOS implementation includes support for L2TP client and server.

General applications of L2TP tunnels:

- For secure router−to−router tunnels over the Internet
- To link (bridge) local Intranets or LANs (when EoIP is also used)
- To extend PPP user connections to a remote location (for example for ISP to authenticate and to provide Internet access separately)
- For mobile or remote clients to remotely access an Intranet/LAN of a company

Each L2TP connection is composed of a server and a client. The MikroTik RouterOS may function as a server or client – or, for various configurations, it may be the server for some connections and client for other connections. For example, the client created below could connect to a Cisco L2TP server, another MikroTik Router, or another router which supports a L2TP server.

Specifications

Packages required : ***ppp***
License required : *Basic (DEMO license is limited to 4 tunnels)*
Home menu level : ***/interface l2tp−server***, ***/interface l2tp−client***
Protocols utilized : *L2TP (RFC2661)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Authentication, Authorization and Accounting
Ethernet over IP (EoIP) Tunnel Interface
IP security (IPsec)

Description

L2TP is a secure tunnel for transporting IP traffic using PPP. L2TP encapsulates PPP in virtual lines that run over IP, FrameRelay and other protocols (that are not currently supported by Mikrotik RouterOS). L2TP incorporates PPP and MPPE (Microsoft Point to Point Encryption) to make encrypted links. The purpose of this protocol is to allow the Layer 2 and PPP endpoints to reside on different devices interconnected by a packet−switched network. With L2TP, a user has an Layer 2 connection to an access concentrator (e.g., modem bank, ADSL DSLAM, etc.), and the concentrator then tunnels individual PPP frames to the Network Access Server . This allows the actual processing of PPP packets to be divorced from the termination of the Layer 2 circuit. From the user's perspective, there is no functional difference between having the L2 circuit terminate in a NAS directly or using L2TP.

It may also be useful to use L2TP just as any other tunneling protocol with or without encryption. The L2TP standard says that the most secure way to encrypt data is using L2TP over IPsec (**Note** that it is default mode for Microsoft L2TP client) as all L2TP control and data packets for a particular tunnel appear as homogeneous UDP/IP data packets to the IPsec system.

L2TP includes PPP authentication and accounting for each L2TP connection. Full authentication and accounting of each connection may be done through a RADIUS client or locally.

MPPE 40bit RC4 and MPPE 128bit RC4 encryption are supported.

L2TP traffic uses UDP protocol for both control and data packets. UDP port 1701 is used only for link establishment, further traffic is using any available UDP port (which may or may not be 1701). This means that L2TP can be used with most firewalls and routers (even with NAT) by enabling UDP traffic to be routed through the firewall or router.

L2TP Client Setup

Submenu level : **/interface l2tp−client**

Property Description

name (*name*; default: **l2tp−out1**) − interface name for reference
mtu (*integer*; default: **1460**) − Maximum Transmit Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 40 (so, for 1500−byte ethernet link, set the MTU to 1460 to avoid fragmentation of packets)
mru (*integer*; default: **1460**) − Maximum Receive Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 40 (so, for 1500−byte ethernet link, set the MRU to 1460 to avoid fragmentation of packets)
connect−to (*IP address*)− the IP address of the L2TP server to connect to
user (*string*)− user name to use when logging on to the remote server
password (*string*; default: "")− user password to use when logging to the remote server
profile (*name*; default: **default**) − profile to use when connecting to the remote server
add−default−route (yes | no; default: **no**) − whether to use the server which this client is connected to as its default router (gateway)

Example

To set up L2TP client named **test2** using username **john** with password **john** to connect to the **10.1.1.12** L2TP server and use it as the default gateway:

```
[admin@MikroTik] interface l2tp−client> add name=test2 connect−to=10.1.1.12 \
\... user=john add−default−route=yes password=john
[admin@MikroTik] interface l2tp−client> print
Flags: X − disabled, R − running
  0 X  name="test2" mtu=1460 mru=1460 connect−to=10.1.1.12 user="john"
       password="john" profile=default add−default−route=yes


[admin@MikroTik] interface l2tp−client> enable 0
```

Monitoring L2TP Client

Command name : **/interface l2tp−client monitor**

Property Description

Statistics:

status (*string*) − status of the client:
- **Dialing** – attempting to make a connection
- **Verifying password...** − connection has been established to the server, password verification in progress
- **Connected** – self−explanatory
- **Terminated** – interface is not enabled or the other side will not establish a connection **uptime** (*time*) − connection time displayed in days, hours, minutes, and seconds
  encoding (*string*) − encryption and encoding (if asymmetric, separated with '/') being used in this connection

Example

Example of an established connection:

```
[admin@MikroTik] interface l2tp-client> monitor test2
      status: "connected"
      uptime: 4m27s
    encoding: "MPPE128 stateless"
[admin@MikroTik] interface l2tp-client>
```

L2TP Server Setup

Submenu level : **/interface l2tp-server server**

```
[admin@MikroTik] interface l2tp-server server> print
            enabled: no
                mtu: 1460
                mru: 1460
     authentication: mschap2
    default-profile: default
[admin@MikroTik] interface l2tp-server server>
```

Description

The L2TP server supports unlimited connections from clients. For each current connection, a dynamic interface is created.

Property Description

enabled (yes | no; default: **no**) - defines whether L2TP server is enabled or not
mtu (*integer*; default: **1460**) - Maximum Transmit Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 40 (so, for 1500-byte ethernet link, set the MTU to 1460 to avoid fragmentation of packets)
mru (*integer*; default: **1460**) - Maximum Receive Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 40 (so, for 1500-byte ethernet link, set the MTU to 1460 to avoid fragmentation of packets)
authentication (*multiple choice*: pap | chap | mschap1 | mschap2; default: **mschap2**) - authentication algorithm
default-profile - default profile to use

Example

To enable L2TP server:

```
[admin@MikroTik] interface l2tp-server server> set enabled=yes
[admin@MikroTik] interface l2tp-server server> print
            enabled: yes
                mtu: 1460
                mru: 1460
     authentication: mschap2
    default-profile: default
[admin@MikroTik] interface l2tp-server server>
```

L2TP Server Users

Submenu level : **/interface l2tp−server**

Description

There are two types of items in L2TP server configuration − static users and dynamic connections. A dynamic connection can be established if the user database or the **default−profile** has its **local−address** and **remote−address** set correctly. When static users are added, the default profile may be left with its default values and only P2P user (in **/ppp secret**) should be configured. **Note** that in both cases P2P users must be configured properly.

Property Description

name − interface name
user − the name of the user that is configured statically or added dynamically

Statistics:

mtu − shows (cannot be set here) client's MTU
client−address − shows (cannot be set here) the IP of the connected client
uptime − shows how long the client is connected
encoding (*string*) − encryption and encoding (if asymmetric, separated with '/') being used in this connection

Example

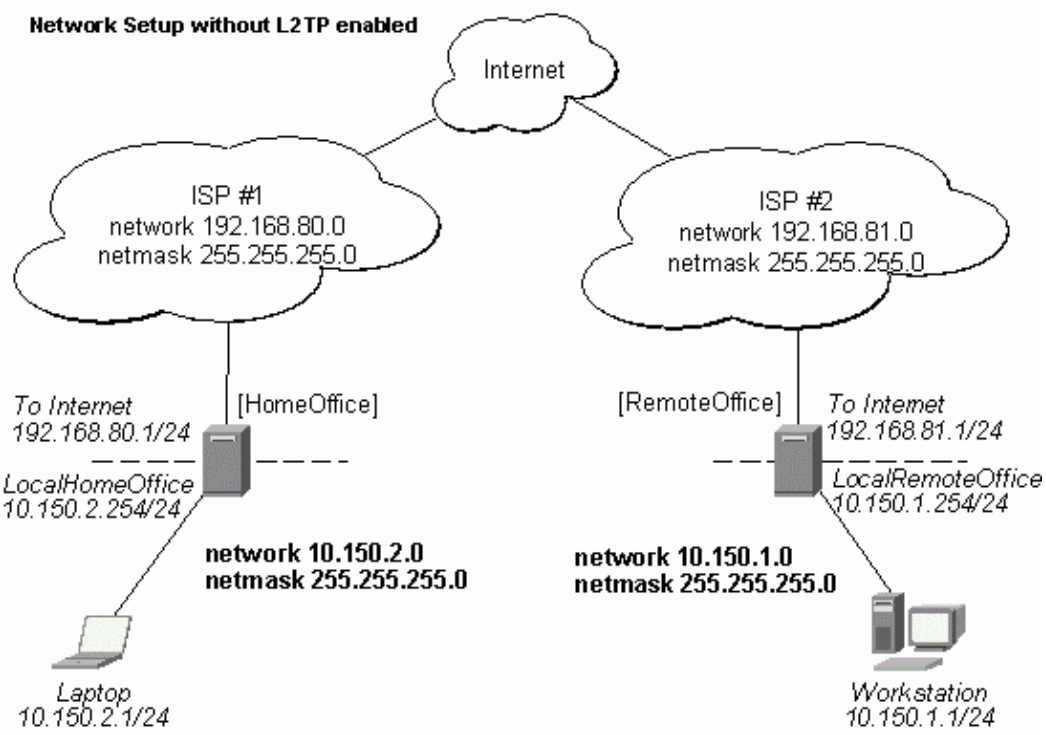
To add a static entry for **ex1** user:

```
[admin@MikroTik] interface l2tp−server> add user=ex1
[admin@MikroTik] interface l2tp−server> print
Flags: X − disabled, D − dynamic, R − running
  #     NAME                USER         MTU   CLIENT−ADDRESS  UPTIME   ENC...
  0  DR <l2tp−ex>           ex           1460  10.0.0.202      6m32s    none
  1     l2tp−in1            ex1
[admin@MikroTik] interface l2tp−server>
```

In this example an already connected user **ex** is shown besides the one we just added.

L2TP Router−to−Router Secure Tunnel Example

The following is an example of connecting two Intranets using an encrypted L2TP tunnel over the Internet.



There are two routers in this example:

- [HomeOffice]
  Interface LocalHomeOffice 10.150.2.254/24
  Interface ToInternet 192.168.80.1/24
- [RemoteOffice]
  Interface ToInternet 192.168.81.1/24
  Interface LocalRemoteOffice 10.150.1.254/24

Each router is connected to a different ISP. One router can access another router through the Internet.

On the L2TP server a user must be set up for the client:

```
[admin@HomeOffice] ppp secret> add name=ex service=l2tp password=lkjrht
local-address=10.0.103.1 remote-address=10.0.103.2
[admin@HomeOffice] ppp secret> print detail
Flags: X - disabled
  0   name="ex" service=l2tp caller-id="" password="lkjrht" profile=default
      local-address=10.0.103.1 remote-address=10.0.103.2 routes==""

[admin@HomeOffice] ppp secret>
```

Then the user should be added in the L2TP server list:

```
[admin@HomeOffice] interface l2tp-server> add user=ex
[admin@HomeOffice] interface l2tp-server> print
Flags: X - disabled, D - dynamic, R - running
  #     NAME                USER         MTU   CLIENT-ADDRESS  UPTIME   ENC...
  0     l2tp-in1            ex
[admin@HomeOffice] interface l2tp-server>
```

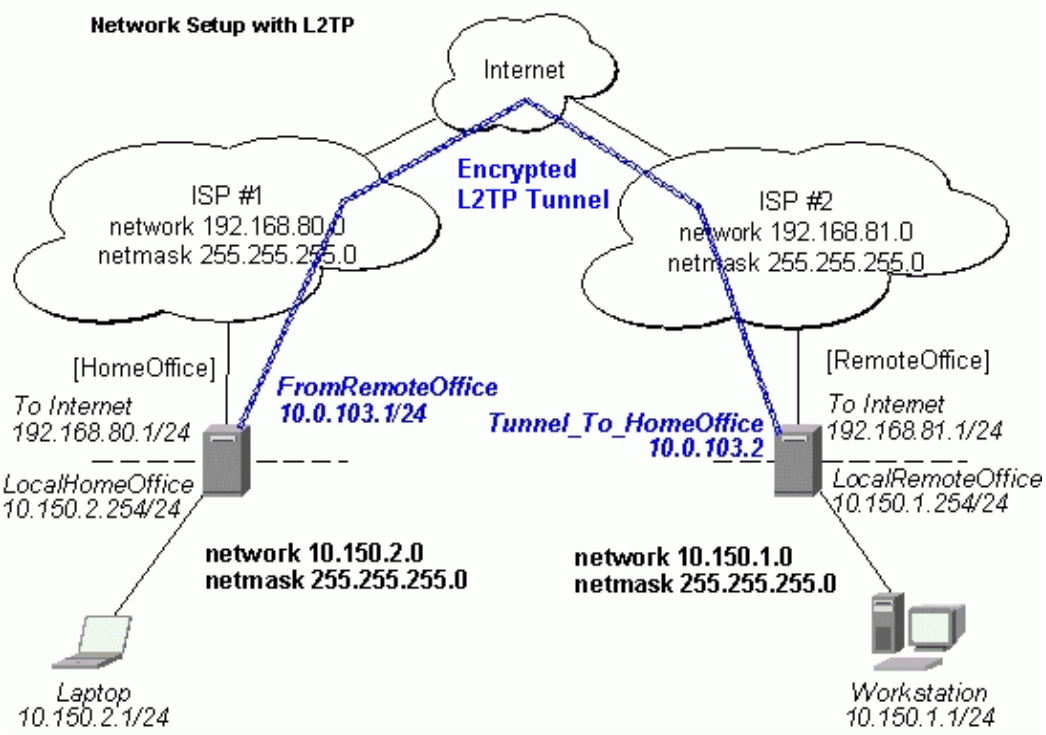
And finally, the server must be enabled:

```
[admin@HomeOffice] interface l2tp-server server> set enabled=yes
[admin@HomeOffice] interface l2tp-server server> print
            enabled: yes
                mtu: 1460
                mru: 1460
     authentication: mschap2
    default-profile: default
[admin@HomeOffice] interface l2tp-server server>
```

Add a L2TP client to the RemoteOffice router:

```
[admin@RemoteOffice] interface l2tp-client> add connect-to=192.168.80.1 user=ex \
\... password=lkjrht disabled=no
[admin@RemoteOffice] interface l2tp-client> print
Flags: X - disabled, R - running
  0  R name="l2tp-out1" mtu=1460 mru=1460 connect-to=192.168.80.1 user="ex"
       password="lkjrht" profile=default add-default-route=no


[admin@RemoteOffice] interface l2tp-client>
```

Thus, a L2TP tunnel is created between the routers. This tunnel is like an Ethernet point-to-point connection between the routers with IP addresses 10.0.103.1 and 10.0.103.2 at each router. It enables 'direct' communication between the routers over third party networks.



To route the local Intranets over the L2TP tunnel – add these routes:

```
[admin@HomeOffice] > ip route add dst-address 10.150.1.0/24 gateway 10.0.103.2
[admin@RemoteOffice] > ip route add dst-address 10.150.2.0/24 gateway 10.0.103.1
```

On the L2TP server it can alternatively be done using **routes** parameter of the user configuration:

```
[admin@HomeOffice] ppp secret> print detail
Flags: X - disabled
  0   name="ex" service=l2tp caller-id="" password="lkjrht" profile=default
      local-address=10.0.103.1 remote-address=10.0.103.2 routes==""

[admin@HomeOffice] ppp secret> set 0 routes="10.150.1.0/24 10.0.103.2 1"
[admin@HomeOffice] ppp secret> print detail
Flags: X - disabled
  0   name="ex" service=l2tp caller-id="" password="lkjrht" profile=default
      local-address=10.0.103.1 remote-address=10.0.103.2
      routes="10.150.1.0/24 10.0.103.2 1"

[admin@HomeOffice] ppp secret>
```

Test the L2TP tunnel connection:

```
[admin@RemoteOffice]> /ping 10.0.103.1
10.0.103.1 pong: ttl=255 time=3 ms
10.0.103.1 pong: ttl=255 time=3 ms
10.0.103.1 pong: ttl=255 time=3 ms
ping interrupted
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3/3.0/3 ms
```

Test the connection through the L2TP tunnel to the LocalHomeOffice interface:

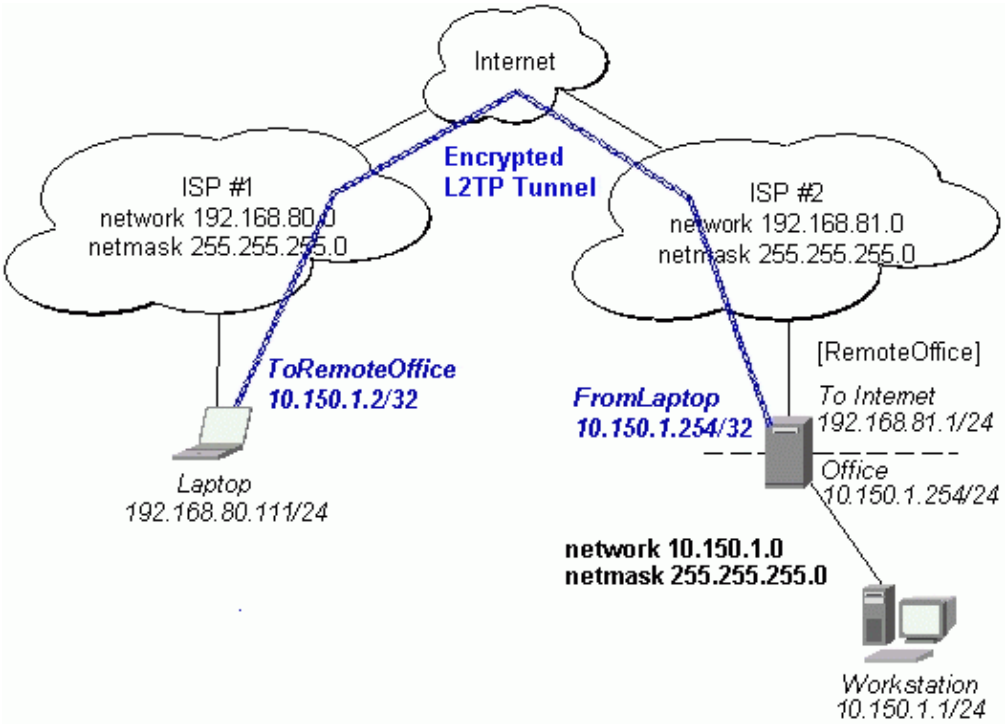
```
[admin@RemoteOffice]> /ping 10.150.2.254
10.150.2.254 pong: ttl=255 time=3 ms
10.150.2.254 pong: ttl=255 time=3 ms
10.150.2.254 pong: ttl=255 time=3 ms
ping interrupted
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3/3.0/3 ms
```

To bridge a LAN over this secure tunnel, please see the example in the 'EoIP' section of the manual. To set the maximum speed for traffic over this tunnel, please consult the 'Queues' section.

Connecting a Remote Client via L2TP Tunnel

The following example shows how to connect a computer to a remote office network over L2TP encrypted tunnel giving that computer an IP address from the same network as the remote office has (without need of bridging over eoip tunnels)

Please, consult the respective manual on how to set up a L2TP client with the software You are using.



The router in this example:

- [RemoteOffice]
  Interface ToInternet 192.168.81.1/24
  Interface Office 10.150.1.254/24

The client computer can access the router through the Internet.

On the L2TP server a user must be set up for the client:

```
[admin@RemoteOffice] ppp secret> add name=ex service=l2tp password=lkjrht
local-address=10.150.1.254 remote-address=10.150.1.2
[admin@RemoteOffice] ppp secret> print detail
Flags: X - disabled
  0   name="ex" service=l2tp caller-id="" password="lkjrht" profile=default
      local-address=10.150.1.254 remote-address=10.150.1.2 routes==""

[admin@RemoteOffice] ppp secret>
```

Then the user should be added in the L2TP server list:

```
[admin@RemoteOffice] interface l2tp-server> add name=FromLaptop user=ex
[admin@RemoteOffice] interface l2tp-server> print
Flags: X - disabled, D - dynamic, R - running
  #     NAME                USER         MTU   CLIENT-ADDRESS  UPTIME   ENC...
  0     FromLaptop          ex
[admin@RemoteOffice] interface l2tp-server>
```

And the server must be enabled:

```
[admin@RemoteOffice] interface l2tp-server server> set enabled=yes
[admin@RemoteOffice] interface l2tp-server server> print
```

```
            enabled: yes
                mtu: 1460
                mru: 1460
     authentication: mschap2
    default-profile: default
[admin@RemoteOffice] interface l2tp-server server>
```

Finally, the proxy APR must be enabled on the 'Office' interface:

```
[admin@RemoteOffice] interface ethernet> set Office arp=proxy-arp
[admin@RemoteOffice] interface ethernet> print
Flags: X - disabled, R - running
  #    NAME                MTU   MAC-ADDRESS       ARP
  0  R ToInternet          1500  00:30:4F:0B:7B:C1 enabled
  1  R Office              1500  00:30:4F:06:62:12 proxy-arp
[admin@RemoteOffice] interface ethernet>
```

L2TP Setup for Windows

Microsoft provides L2TP client support for Windows XP, 2000, NT4, ME and 98. Windows 2000 and XP include support in the Windows setup or automatically install L2TP. For 98, NT and ME, installation requires a download from Microsoft (L2TP/IPSec VPN Client)

For more information, see:
Microsoft L2TP/IPSec VPN Client

On Windows 2000, L2TP setup without IPsec requires editting registry:
Disabling IPSec for the Windows 2000 Client
Disabling IPSEC Policy Used with L2TP

Troubleshooting

- *I use firewall and I cannot establish L2TP connection*
  Make sure UDP connections can pass through both directions between your sites.
- *My Windows L2TP/IPSec VPN Client fails to connect to L2TP server with "Error 789" or "Error 781"*
  The error messages 789 and 781 occur when IPsec is not configured properly on both ends. See the respective documentation on how to configure IPsec in the Microsoft L2TP/IPSec VPN Client and in the MikroTik RouterOS. If you do not want to use IPsec, it can be easily switched off on the client side.
  Note: if you are using Windows 2000, you need to edit system registry using regedt32.exe or regedit.exe. Add the following registry value to **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Rasman\Parameters**:

  ```
  Value Name: ProhibitIpSec
  Data Type: REG_DWORD
  Value: 1
  ```

  You must restart the Windows 2000 for the changes to take effect

For more information on configuring Windows 2000, see:
Configuring Cisco IOS and Windows 2000 Clients for L2TP Using Microsoft IAS
Disabling IPSEC Policy Used with L2TP
How to Configure a L2TP/IPSec Connection Using Pre−shared Key Authentication

---

MOXA C101 Synchronous Interface

Document revision 1.6 (19−Aug−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The MikroTik RouterOS supports the MOXA C101 Synchronous 4Mb/s Adapter hardware. The V.35 synchronous interface is the standard for VSAT and other satellite modems. However, you must check with the satellite system supplier for the modem interface type.

Specifications

Packages required : ***synchronous***
License required : *synchronous*
Home menu level : ***/interface moxa−c101***
Protocols utilized :
- *CISCO/HDLC−X.25 (RFC1356)*
- *Frame Relay (RFC1490)*
- *PPP (RFC−1661, RFC−1662)*

Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)

Log Management

Installation

You can install up to four MOXA C101 synchronous cards in one PC box, if you have so many slots and IRQs available. Assuming you have all necessary packages and licences installed, in most cases it should to be done nothing at that point (all drivers are loaded automatically). However, if you have a non Plug−and−Play ISA card, the corresponding driver requires to be loaded.

MOXA C101 PCI variant cabling

The MOXA C101 PCI requires different from MOXA C101 ISA cable. It can be made using the following table:

DB25f	Signal	Direction	V.35m
4	RTS	OUT	C
5	CTS	IN	D
6	DSR	IN	E
7	GND	−	B
8	DCD	IN	F
10	TxDB	OUT	S
11	TxDA	OUT	P
12	RxDB	IN	T
13	RxDA	IN	R
14	TxCB	IN	AA
16	TxCA	IN	Y
20	DTR	OUT	H
22	RxCB	IN	X
23	RxCA	IN	V
short 9 and 25 pin			

Synchronous Interface Configuration

Submenu level : **/interface moxa−c101**

Description

Moxa c101 synchronous interface is shown under the interfaces list with the name moxa−c101−*N*.

Property Description

cisco−hdlc−keepalive−interval (*time*; default: **10s**) − Keepalive period in seconds
clock−rate (*integer*; default: **64000**) − speed of internal clock
clock−source (external | internal | tx−from−rx | tx−internal; default: **external**) − clock source

frame−relay−dce (yes | no; default: **no**) − operate or not in DCE mode
frame−relay−lmi−type (ansi | ccitt; default: **ansi**) − Frame−relay Local Management Interface type:

- **ansi** − set LMI type to ANSI−617d (also known as Annex D)
- **ccitt** − set LMI type to CCITT Q933a (also known as Annex A)

ignore−dcd (yes | no; default: **no**) − Ignore or not DCD
line−protocol (cisco−hdlc | frame−relay | sync−ppp; default: **sync−ppp**) − line protocol name
mtu (*integer*; default: **1500**) − Maximum Transmit Unit
name (*name*; default: **moxa−c101−*N***) − interface name

Notes

If you purchased the MOXA C101 Synchronous card from MikroTik, you have received a V.35 cable with it. This cable should work for all standard modems, which have V.35 connections. For synchronous modems, which have a DB−25 connection, you should use a standard DB−25 cable.

The MikroTik driver for the MOXA C101 Synchronous adapter allows you to unplug the V.35 cable from one modem and plug it into another modem with a different clock speed, and you do not need to restart the interface or router.

Example

```
[admin@MikroTik] interface> moxa-c101
[admin@MikroTik] interface moxa-c101> print
Flags: X - disabled, R - running
  0  R name="moxa-c101-1" mtu=1500 line-protocol=sync-ppp clock-rate=64000
       clock-source=external frame-relay-lmi-type=ansi frame-relay-dce=no
       cisco-hdlc-keepalive-interval=10s ignore-dcd=no

[admin@MikroTik] interface moxa-c101>
```

You can monitor the status of the synchronous interface:

```
[admin@MikroTik] interface moxa-c101> monitor 0
    dtr: yes
    rts: yes
    cts: no
    dsr: no
    dcd: no

[admin@MikroTik] interface moxa-c101>
```

Connect a communication device, e.g., a baseband modem, to the V.35 port and turn it on. If the link is working properly the status of the interface is:

```
[admin@MikroTik] interface moxa-c101> monitor 0
    dtr: yes
    rts: yes
    cts: yes
    dsr: yes
    dcd: yes

[admin@MikroTik] interface moxa-c101>
```

Troubleshooting

- *The synchronous interface does not show up under the interfaces list*
  Obtain the required license for synchronous feature.
- *The synchronous link does not work*
  Check the V.35 cabling and the line between the modems. Read the modem manual.

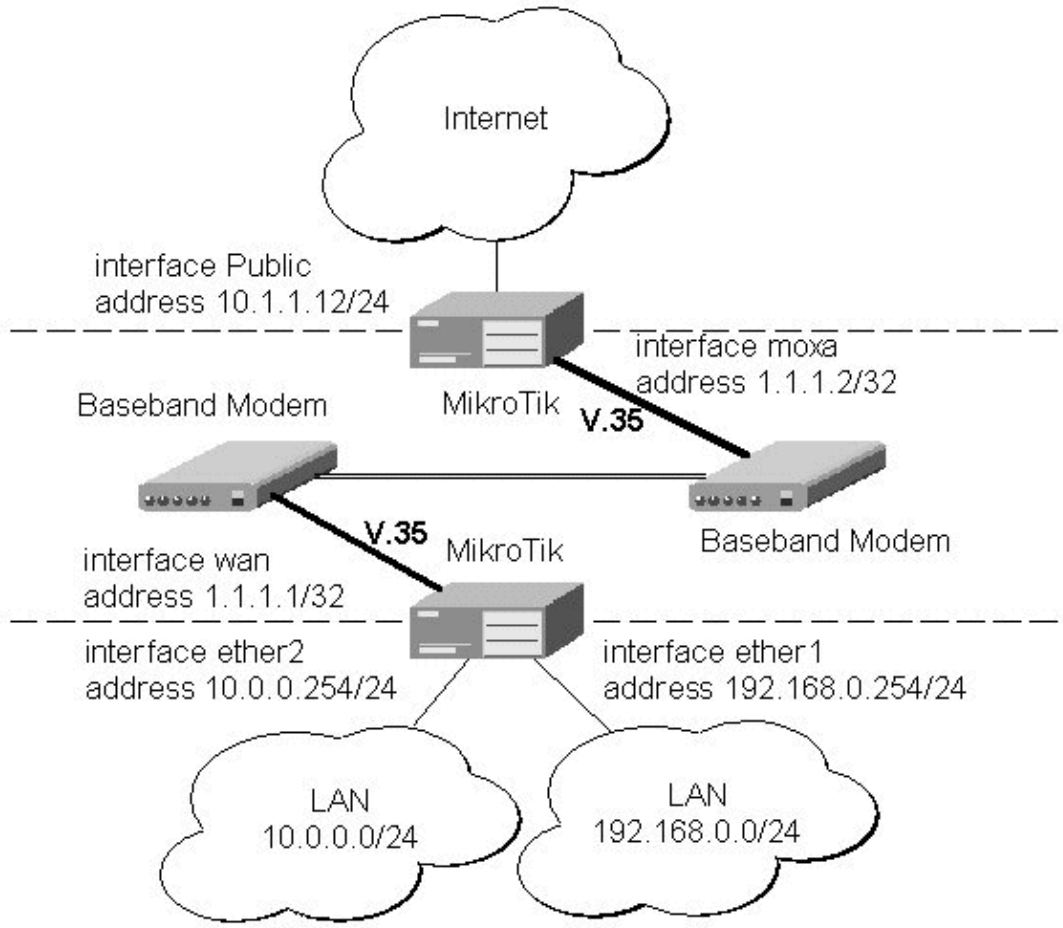
Synchronous Link Applications

Two possible synchronous line configurations are discussed in the following examples:

- MikroTik Router to MikroTik Router
- MikroTik Router to CISCO Router

MikroTik Router to MikroTik Router

Let us consider the following network setup with two MikroTik Routers connected to a leased line with baseband modems:



The driver for MOXA C101 card should be loaded and the interface should be enabled according to the instructions given above. The IP addresses assigned to the synchronous interface should be as follows:

```
[admin@MikroTik] ip address> add address 1.1.1.1/32 interface wan \
```

```
\... network 1.1.1.2 broadcast 255.255.255.255

[admin@MikroTik] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.254/24      10.0.0.254      10.0.0.255      ether2
  1   192.168.0.254/24   192.168.0.254   192.168.0.255   ether1
  2   1.1.1.1/32         1.1.1.2         255.255.255.255 wan
[admin@MikroTik] ip address> /ping 1.1.1.2
1.1.1.2 64 byte pong: ttl=255 time=31 ms
1.1.1.2 64 byte pong: ttl=255 time=26 ms
1.1.1.2 64 byte pong: ttl=255 time=26 ms
3 packets transmitted, 3 packets received, 0% packet loss
round−trip min/avg/max = 26/27.6/31 ms
[admin@MikroTik] ip address>
```

The default route should be set to the gateway router 1.1.1.2:

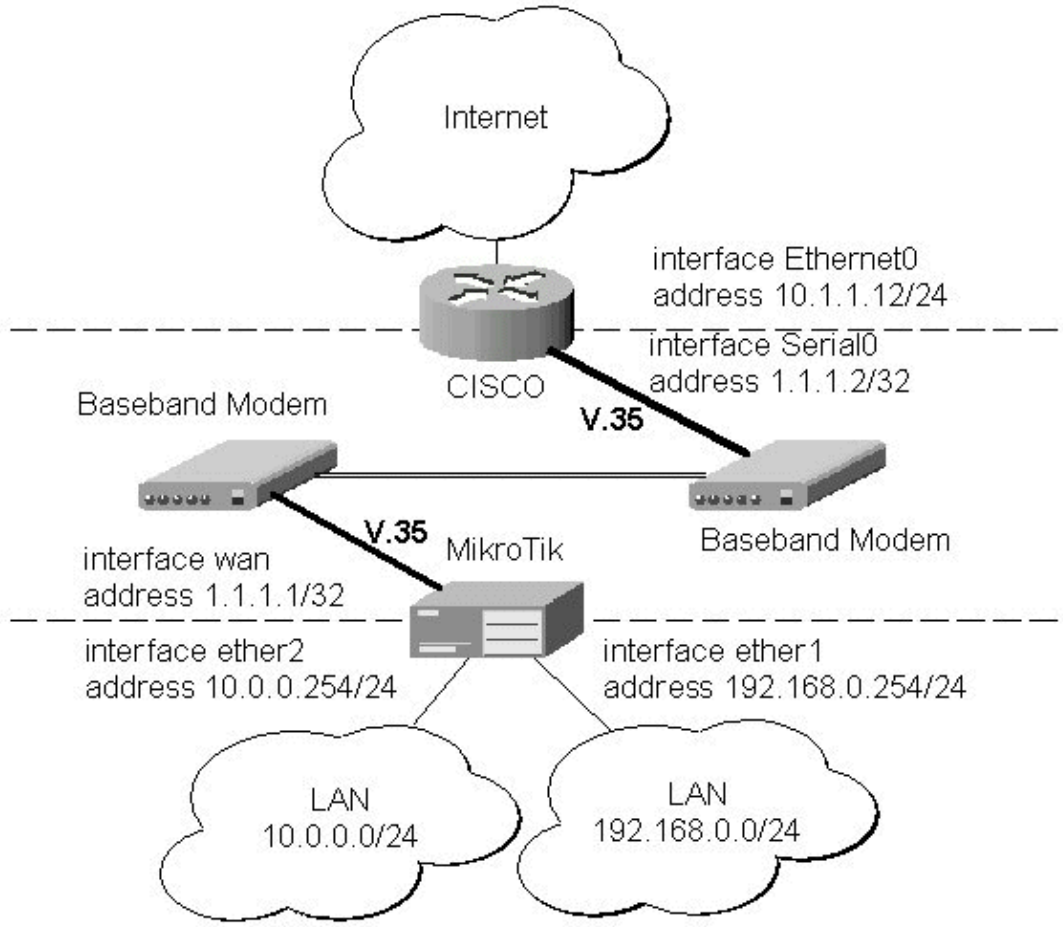
```
[admin@MikroTik] ip route> add gateway 1.1.1.2
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 1.1.1.2         1        wan
    1 DC 10.0.0.0/24        r 10.0.0.254      1        ether2
    2 DC 192.168.0.0/24     r 192.168.0.254   0        ether1
    3 DC 1.1.1.2/32         r 0.0.0.0         0        wan

[admin@MikroTik] ip route>
```

The configuration of the Mikrotik router at the other end is similar:

```
[admin@MikroTik] ip address> add address 1.1.1.2/32 interface moxa \
\... network 1.1.1.1 broadcast 255.255.255.255
[admin@MikroTik] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.1.1.12/24       10.1.1.12       10.1.1.255      Public
  1   1.1.1.2/32         1.1.1.1         255.255.255.255 moxa
[admin@MikroTik] ip address> /ping 1.1.1.1
1.1.1.1 64 byte pong: ttl=255 time=31 ms
1.1.1.1 64 byte pong: ttl=255 time=26 ms
1.1.1.1 64 byte pong: ttl=255 time=26 ms
3 packets transmitted, 3 packets received, 0% packet loss
round−trip min/avg/max = 26/27.6/31 ms
[admin@MikroTik] ip address>
```

MikroTik Router to CISCO Router

Let us consider the following network setup with MikroTik Router connected to a leased line with baseband modems and a CISCO router at the other end:



The driver for MOXA C101 card should be loaded and the interface should be enabled according to the instructions given above. The IP addresses assigned to the synchronous interface should be as follows:

```
[admin@MikroTik] ip address> add address 1.1.1.1/32 interface wan \
\... network 1.1.1.2 broadcast 255.255.255.255
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.254/24      10.0.0.254      10.0.0.255      ether2
  1   192.168.0.254/24   192.168.0.254   192.168.0.255   ether1
  2   1.1.1.1/32         1.1.1.2         255.255.255.255 wan
[admin@MikroTik] ip address> /ping 1.1.1.2
1.1.1.2 64 byte pong: ttl=255 time=31 ms
1.1.1.2 64 byte pong: ttl=255 time=26 ms
1.1.1.2 64 byte pong: ttl=255 time=26 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 26/27.6/31 ms
[admin@MikroTik] ip address>
```

The default route should be set to the gateway router 1.1.1.2:

```
[admin@MikroTik] ip route> add gateway 1.1.1.2
[admin@MikroTik] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
```

```
    0  S 0.0.0.0/0          r 1.1.1.2         1        wan
    1 DC 10.0.0.0/24        r 10.0.0.254      0        ether2
    2 DC 192.168.0.0/24     r 192.168.0.254   0        ether1
    3 DC 1.1.1.2/32         r 1.1.1.1         0        wan

[admin@MikroTik] ip route>
```

The configuration of the CISCO router at the other end (part of the configuration) is:

```
CISCO#show running-config
Building configuration...

Current configuration:
...
!
interface Ethernet0
 description connected to EthernetLAN
 ip address 10.1.1.12 255.255.255.0
!
interface Serial0
 description connected to MikroTik
 ip address 1.1.1.2 255.255.255.252
 serial restart-delay 1
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.254
!
...
end

CISCO#
```

Send ping packets to the MikroTik router:

```
CISCO#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/32/40 ms
CISCO#
```

Notes

Keep in mind, that for the point−to−point link the network mask is set to **32** bits, the argument **network** is set to the IP address of the other end, and the broadcast address is set to **255.255.255.255**.

Additional Resources

For more information about the MOXA C101 Synchronous 4Mb/s Adapter hardware please see the relevant documentation:

- http://www.moxa.com/product/sync/C101.htm − The product on−line documentation
- C101 SuperSync Board User's Manual − The User's Manual in .pdf format

MOXA C502 Synchronous Interface

Document revision 1.3 (30−Jun−2002)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The MikroTik RouterOS supports the MOXA C502 PCI Dual−port Synchronous 8Mb/s Adapter hardware. The V.35 synchronous interface is the standard for VSAT and other satellite modems. However, you must check with the satellite system supplier for the modem interface type.

Specifications

Packages required : ***synchronous***
License required : *synchronous*
Home menu level : ***/interface moxa−c502***
Protocols utilized :
- *CISCO/HDLC−X.25 (RFC1356)*
- *Frame Relay (RFC1490)*
- *PPP (RFC−1661, RFC−1662)*

Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
Log Management

Installation

You can install up to four MOXA C502 synchronous cards in one PC box, if you have so many PCI slots available. Assuming you have all necessary packages and licences installed, in most cases it should to be done nothing at that point (all drivers are loaded automatically).

Synchronous Interface Configuration

Submenu level : **/interface moxa−c502**

Description

Moxa C502 synchronous interface should be listed under the interfaces list with the name moxa−c502−*N*, where *N* is 0,1,2,... .

Property Description

cisco−hdlc−keepalive−interval (*time*; default: **10s**) − Keepalive period in seconds
clock−rate (*integer*; default: **64000**) − speed of internal clock
clock−source (external | internal | tx−from−rx | tx−internal; default: **external**) − clock source
frame−relay−dce (yes | no; default: **no**) − operate or not in DCE mode
frame−relay−lmi−type (ansi | ccitt; default: **ansi**) − Frame−relay Local Management Interface type:

- **ansi** − set LMI type to ANSI−617d (also known as Annex D)
- **ccitt** − set LMI type to CCITT Q933a (also known as Annex A)

ignore−dcd (yes | no; default: **no**) − Ignore or not DCD
line−protocol (cisco−hdlc | frame−relay | sync−ppp; default: **sync−ppp**) − line protocol name
mtu (*integer*; default: **1500**) − Maximum Transmit Unit
name (*name*; default: **moxa−c502−*N***) − interface name

Notes

There will be TWO interfaces for each MOXA C502 card since the card has TWO ports.

The MikroTik driver for the MOXA C502 Dual Synchronous adapter allows you to unplug the V.35 cable from one modem and plug it into another modem with a different clock speed, and you do not need to restart the interface or router.

Example

```
[admin@MikroTik] interface> moxa−c502
[admin@MikroTik] interface moxa−c502> print
Flags: X − disabled, R − running
  0  R name="moxa−c502−1" mtu=1500 line−protocol=sync−ppp clock−rate=64000
       clock−source=external frame−relay−lmi−type=ansi frame−relay−dce=no
       cisco−hdlc−keepalive−interval=10s
  1  R name="moxa−c502−2" mtu=1500 line−protocol=sync−ppp clock−rate=64000
       clock−source=external frame−relay−lmi−type=ansi frame−relay−dce=no
       cisco−hdlc−keepalive−interval=10s

[admin@MikroTik] interface moxa−c502>
```

You can monitor the status of the synchronous interface:

```
[admin@MikroTik] interface moxa-c502> monitor 0
    dtr: yes
    rts: yes
    cts: no
    dsr: no
    dcd: no

[admin@MikroTik] interface moxa-c502>
```

Connect a communication device, e.g., a baseband modem, to the V.35 port and turn it on. If the link is working properly the status of the interface is:

```
[admin@MikroTik] interface moxa-c502> monitor 0
    dtr: yes
    rts: yes
    cts: yes
    dsr: yes
    dcd: yes

[admin@MikroTik] interface moxa-c502>
```

Troubleshooting

- *The synchronous interface does not show up under the interfaces list*
  Obtain the required license for synchronous feature.
- *The synchronous link does not work*
  Check the V.35 cabling and the line between the modems. Read the modem manual.

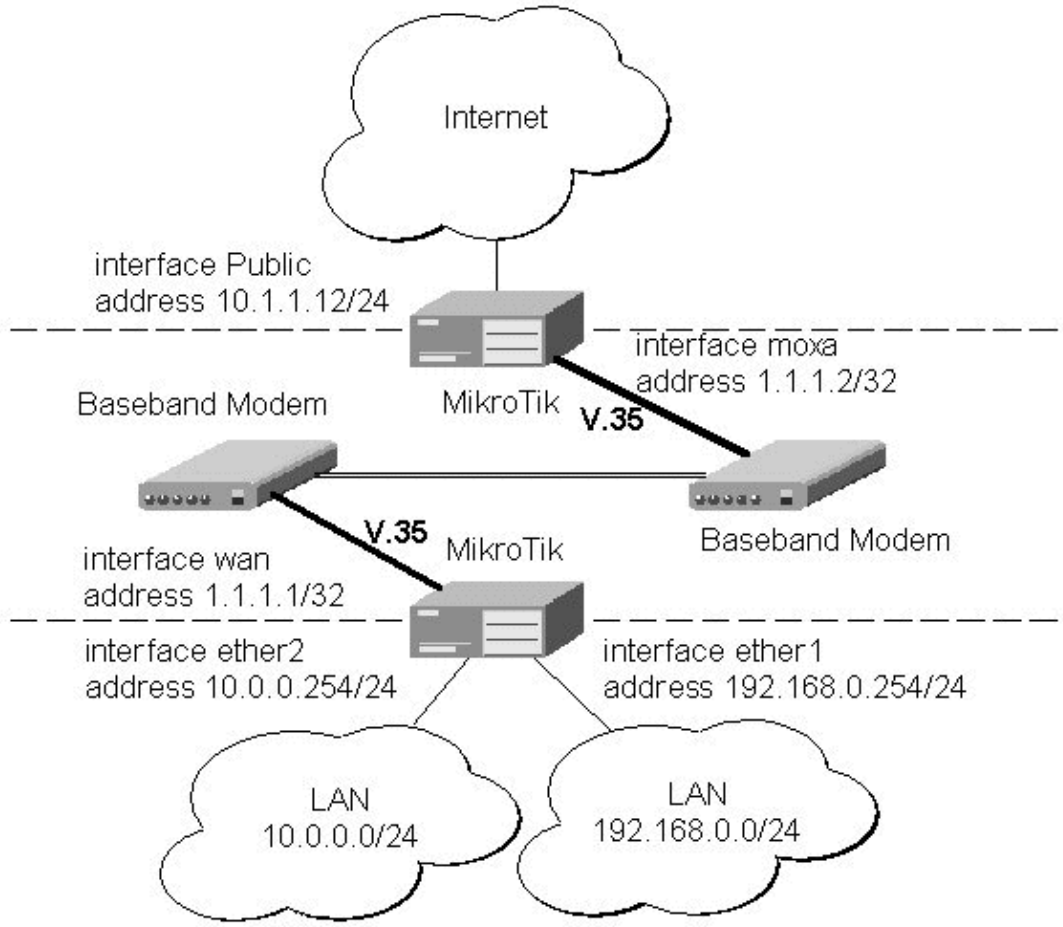
Synchronous Link Applications

Two possible synchronous line configurations are discussed in the following examples:

- MikroTik Router to MikroTik Router
- MikroTik Router to CISCO Router

MikroTik Router to MikroTik Router

Let us consider the following network setup with two MikroTik Routers connected to a leased line with baseband modems:



The driver for MOXA C101 card should be loaded and the interface should be enabled according to the instructions given above. The IP addresses assigned to the synchronous interface should be as follows:

```
[admin@MikroTik] ip address> add address 1.1.1.1/32 interface wan \
\... network 1.1.1.2 broadcast 255.255.255.255

[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.254/24      10.0.0.254      10.0.0.255      ether2
  1   192.168.0.254/24   192.168.0.254   192.168.0.255   ether1
  2   1.1.1.1/32         1.1.1.2         255.255.255.255 wan
[admin@MikroTik] ip address> /ping 1.1.1.2
1.1.1.2 64 byte pong: ttl=255 time=31 ms
1.1.1.2 64 byte pong: ttl=255 time=26 ms
1.1.1.2 64 byte pong: ttl=255 time=26 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 26/27.6/31 ms
[admin@MikroTik] ip address>
```

The default route should be set to the gateway router 1.1.1.2:

```
[admin@MikroTik] ip route> add gateway 1.1.1.2 interface wan
[admin@MikroTik] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
```

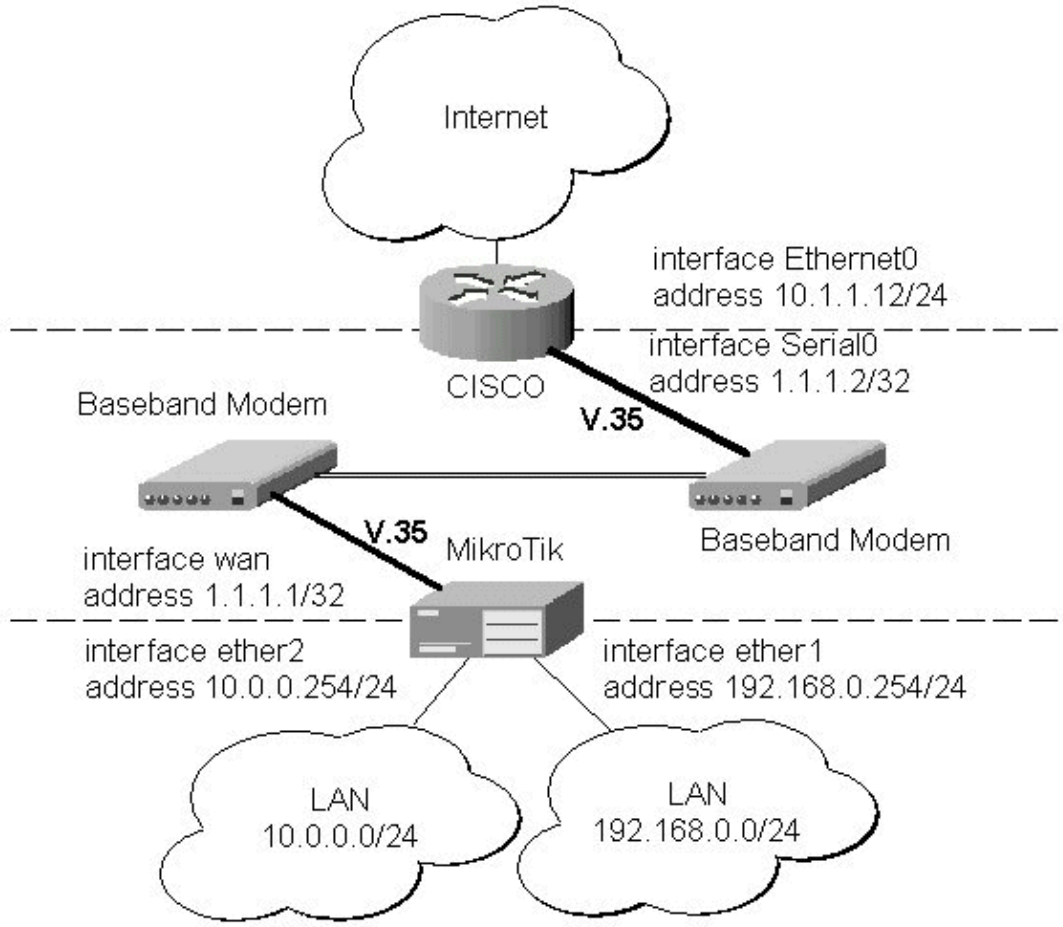
```
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 1.1.1.2         1        wan
    1 DC 10.0.0.0/24        r 10.0.0.254      1        ether2
    2 DC 192.168.0.0/24     r 192.168.0.254   0        ether1
    3 DC 1.1.1.2/32         r 0.0.0.0         0        wan

[admin@MikroTik] ip route>
```

The configuration of the Mikrotik router at the other end is similar:

```
[admin@MikroTik] ip address> add address 1.1.1.2/32 interface moxa \
\... network 1.1.1.1 broadcast 255.255.255.255
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.1.1.12/24       10.1.1.12       10.1.1.255      Public
  1   1.1.1.2/32         1.1.1.1         255.255.255.255 moxa
[admin@MikroTik] ip address> /ping 1.1.1.1
1.1.1.1 64 byte pong: ttl=255 time=31 ms
1.1.1.1 64 byte pong: ttl=255 time=26 ms
1.1.1.1 64 byte pong: ttl=255 time=26 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 26/27.6/31 ms
[admin@MikroTik] ip address>
```

MikroTik Router to CISCO Router

Let us consider the following network setup with MikroTik Router connected to a leased line with baseband modems and a CISCO router at the other end:



The driver for MOXA C101 card should be loaded and the interface should be enabled according to the instructions given above. The IP addresses assigned to the synchronous interface should be as follows:

```
[admin@MikroTik] ip address> add address 1.1.1.1/32 interface wan \
\... network 1.1.1.2 broadcast 255.255.255.255
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.254/24      10.0.0.254      10.0.0.255      ether2
  1   192.168.0.254/24   192.168.0.254   192.168.0.255   ether1
  2   1.1.1.1/32         1.1.1.2         255.255.255.255 wan
[admin@MikroTik] ip address> /ping 1.1.1.2
1.1.1.2 64 byte pong: ttl=255 time=31 ms
1.1.1.2 64 byte pong: ttl=255 time=26 ms
1.1.1.2 64 byte pong: ttl=255 time=26 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 26/27.6/31 ms
[admin@MikroTik] ip address>
```

The default route should be set to the gateway router 1.1.1.2:

```
[admin@MikroTik] ip route> add gateway 1.1.1.2 interface wan
[admin@MikroTik] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
```

```
    0  S 0.0.0.0/0          r 1.1.1.2         1        wan
    1 DC 10.0.0.0/24        r 10.0.0.254      0        ether2
    2 DC 192.168.0.0/24     r 192.168.0.254   0        ether1
    3 DC 1.1.1.2/32         r 1.1.1.1         0        wan

[admin@MikroTik] ip route>
```

The configuration of the CISCO router at the other end (part of the configuration) is:

```
CISCO#show running-config
Building configuration...

Current configuration:
...
!
interface Ethernet0
 description connected to EthernetLAN
 ip address 10.1.1.12 255.255.255.0
!
interface Serial0
 description connected to MikroTik
 ip address 1.1.1.2 255.255.255.252
 serial restart-delay 1
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.254
!
...
end

CISCO#
```

Send ping packets to the MikroTik router:

```
CISCO#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/32/40 ms
CISCO#
```

Notes

Keep in mind, that for the point−to−point link the network mask is set to **32** bits, the argument **network** is set to the IP address of the other end, and the broadcast address is set to **255.255.255.255**.

Additional Resources

For more information about the MOXA C502 Dual−port Synchronous 8Mb/s Adapter hardware please see the relevant documentation:

- http://www.moxa.com/product/sync/C502.htm − The product on−line documentation
- C502 Dual Port Sync Board User's Manual − The User's Manual in .pdf format

Point to Point Protocol (PPP) and Asynchronous Interfaces

Document revision 1.4 (01−Jun−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

PPP (or Point−to−Point Protocol) provides a method for transmitting datagrams over serial point−to−point links. Physically, it relies on **com1** and **com2** ports from standard PC hardware configurations. These appear as **serial0** and **serial1** automatically. You can add more serial ports to use the router for a modem pool using these adapters:

- MOXA (www.moxa.com) Smartio C104H 4−port PCI multiport asynchronous board with maximum of 16 ports (4 cards)
- MOXA (www.moxa.com) Smartio C168H 8−port PCI multiport asynchronous board with maximum of 32 ports (4 cards)
- Cyclades (www.cyclades.com) Cyclom−Y Series PCI multiport asynchronous (serial) cards
- Cyclades (www.cyclades.com) Cyclades−Z Series PCI multiport asynchronous (serial) cards
- TCL (www.thetcl.com) DataBooster 4 or 8 port High Speed Buffered PCI Communication Controllers

Specifications

Packages required : ***ppp***
License required : *Any*
Home menu level : ***/interface ppp−client***, ***/interface ppp−server***
Standards and Technologies : *PPP (RFC−1661)*

Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
Log Management
Authentication, Authorization and Accounting

Serial Port Configuration

Submenu level : **/port**

Property Description

name (*name*) − port name
used−by (read−only: *target*) − shows the user of the port. Only free ports can be used in PPP setup
baud−rate (*integer*; default: **9600**) − maximal data rate of the port
data−bits (7 | 8; default: **8**) − number of bits per character transmitted
parity (none | even | odd; default: **none**) − character parity check method
stop−bits (1 | 2; default: **1**) − number of stop bits after each character transmitted
flow−control (none | hardware | xon−xoff; default: **hardware**) − flow control method

Notes

Keep in mind that **baud−rate**, **data−bits**, **parity**, **stop−bits** and **flow−control** parameters must be the same for both communicating sides.

Example

```
[admin@MikroTik] > /port print
  # NAME                            USED-BY                         BAUD-RATE
  0 serial0                         Serial Console                  9600
  1 databooster1                                                    9600
  2 databooster2                                                    9600
  3 databooster3                                                    9600
  4 databooster4                                                    9600
  5 databooster5                                                    9600
  6 databooster6                                                    9600
  7 databooster7                                                    9600
  8 databooster8                                                    9600
  9 cycladesA1                                                      9600
 10 cycladesA2                                                      9600
 11 cycladesA3                                                      9600
 12 cycladesA4                                                      9600
 13 cycladesA5                                                      9600
 14 cycladesA6                                                      9600
 15 cycladesA7                                                      9600
 16 cycladesA8                                                      9600
[admin@MikroTik] > set 9 baud-rate=38400
[admin@MikroTik] >
```

PPP Server Setup

Submenu level : **/interface ppp−server**

Description

PPP server provide a remote connection service for users. When dialing in, the users can be authenticated locally using the local user database in the **/user** menu, or at the RADIUS server specified in the **/ip ppp** settings.

Property Description

port (*name*; default: **(unknown)**) − serial port
authentication (multiple choice: mschap2,mschap1,chap,pap; default: **mschap2,mschap1,chap,pap**) − authentication protocol
profile (*name*; default: **default**) − profile name used for the link
mtu (*integer*; default: **1500**) − Maximum Transmit Unit. Maximum packet size to be transmitted
mru (*integer*; default: **1500**) − Maximum Receive Unit
null−modem (no | yes; default: **no**) − enable/disable null−modem mode (when enabled, no modem initialization strings are sent)
modem−init (*text*; default: "") − modem initialization string. You may use **"s11=40"** to improve dialling speed
ring−count (*integer*; default: **1**) − number of rings to wait before answering phone
name (*text*; default: **ppp−in*N***) − interface name for reference

Example

You can add a PPP server using the **add** command:

```
[admin@MikroTik] interface ppp−server> add name=test port=serial1
[admin@MikroTik] interface ppp−server> print
Flags: X − disabled, R − running
  0 X  name="test" mtu=1500 mru=1500 port=serial1
       authentication=mschap2,chap,pap profile=default modem−init=""
       ring−count=1 null−modem=no

[admin@MikroTik] interface ppp−server> enable 0
[admin@MikroTik] interface ppp−server> monitor test
            status: "waiting for call..."

[admin@MikroTik] interface ppp−server>
```

PPP Client Setup

Submenu level : **/interface ppp−client**

Description

This section describes PPP clients configuration routines.

Property Description

name (*text*; default: **ppp-out*N***) - new interface name
port (*name*; default: **(unknown)**) - serial port
user (*text*; default: "") - P2P user name on the remote server to use for dialout
password (*text*; default: "") - P2P user password on the remote server to use for dialout
profile (*name*; default: **default**) - local profile to use for dialout
phone (*integer*; default: "") - phone number for dialout
tone-dial (yes | no; default: **yes**) - defines whether use tone dial or pulse dial
mtu (*integer*; default: **1500**) - Maximum Transmit Unit. Maximum packet size to be transmitted
mru (*integer*; default: **1500**) - Maximum Receive Unit
null-modem (no | yes; default: **no**) - enable/disable null-modem mode (when enabled, no modem initialization strings are sent)
modem-init (*text*; default: "") - modem initialization string. You may use **"s11=40"** to improve dialling speed
dial-on-demand (yes | no; default: **no**) - enable/disable dial on demand
add-default-route (yes | no; default: **no**) - add PPP remote address as a default route
use-peer-dns (yes | no; default: **no**) - use DNS server settings from the remote server

Notes

- Additional client profiles must be configured on the server side for clients to accomplish logon procedure. For more information see **Related Documents** section.
- PPP client profiles must match at least partially (**local-address** and values related to encryption should match) with corresponding remote server values.

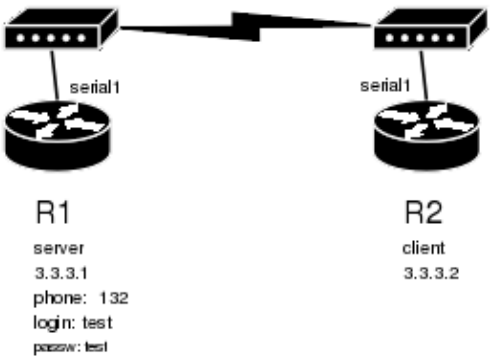
Example

You can add a PPP client using the **add** command:

```
[admin@MikroTik] interface ppp-client> add name=test user=test port=serial1 \
\... add-default-route=yes
[admin@MikroTik] interface ppp-client> print
Flags: X - disabled, R - running
  0 X  name="test" mtu=1500 mru=1500 port=serial1 user="test" password=""
       profile=default phone="" tone-dial=yes modem-init="" null-modem=no
       dial-on-demand=no add-default-route=yes use-peer-dns=no

[admin@MikroTik] interface ppp-client> enable 0
[admin@MikroTik] interface ppp-client> monitor test
[admin@MikroTik] interface ppp-client> monitor 0
           status: "dialing out..."

[admin@MikroTik] interface ppp-client>
```

PPP Application Example

In this example we will consider the following network setup:



For a tupical server setup we need to add one user to the **R1** and configure the PPP server.

```
[admin@MikroTik] ppp secret> add name=test password=test local-address=3.3.3.1 \
\... remote-address=3.3.3.2
[admin@MikroTik] ppp secret> print
Flags: X - disabled
  0   name="test" service=any caller-id="" password="test" profile=default
      local-address=3.3.3.1 remote-address=3.3.3.2 routes=""

[admin@MikroTik] ppp secret> /int ppp-server
[admin@MikroTik] interface ppp-server> add port=serial1 disabled=no
[admin@MikroTik] interface ppp-server> print
Flags: X - disabled, R - running
  0    name="ppp-in1" mtu=1500 mru=1500 port=serial1
       authentication=mschap2,mschap1,chap,pap profile=default modem-init=""
       ring-count=1 null-modem=no

[admin@MikroTik] interface ppp-server>
```

Now we need to setup the client to connect to the server:

```
[admin@MikroTik] interface ppp-client> add port=serial1 user=test password=test \
\... phone=132
[admin@MikroTik] interface ppp-client> print
Flags: X - disabled, R - running
  0 X  name="ppp-out1" mtu=1500 mru=1500 port=serial1 user="test"
       password="test" profile=default phone="132" tone-dial=yes
       modem-init="" null-modem=no dial-on-demand=no add-default-route=no
       use-peer-dns=no

[admin@MikroTik] interface ppp-client> enable 0
```

After a short duration of time the routers will be able to ping each other:

```
[admin@MikroTik] interface ppp-client> /ping 3.3.3.1
3.3.3.1 64 byte ping: ttl=64 time=43 ms
3.3.3.1 64 byte ping: ttl=64 time=11 ms
3.3.3.1 64 byte ping: ttl=64 time=12 ms
3.3.3.1 64 byte ping: ttl=64 time=11 ms
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 11/19.2/43 ms
[admin@MikroTik] interface ppp-client>
```

Additional Resources

http://www.ietf.org/rfc/rfc2138.txt?number=2138
http://www.ietf.org/rfc/rfc2138.txt?number=2139

Point to Point Protocol over Ethernet (PPPoE)

Document revision 1.4 (29−Dec−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

The PPPoE (Point to Point Protocol over Ethernet) protocol provides extensive user management, network management and accounting benefits to ISPs and network administrators. Currently, PPPoE is used mainly by ISPs to control client connections for xDSL and cable modems. PPPoE is an extension of the standard dial−up and synchronous protocol PPP. The transport is Ethernet, as opposed to modem transport.

Generally speaking, the PPPoE is used to hand out IP addresses to clients based on the user (and workstation, if desired) authentication as opposed to workstation only authentication, when static IP addresses or DHCP is used. Do not use static IP addresses or DHCP on interfaces, on which the PPPoE is used for security reasons.

A PPPoE connection is composed of a client and an access concentrator (server). The client may be a Windows computer that has the PPPoE client protocol installed. The MikroTik RouterOS supports both the client and access concentrator implementations of PPPoE. The PPPoE client and server work over any Ethernet level interface on the router: wireless IEEE802.11 (Aironet, Cisco, WaveLAN, Prism, Atheros), 10/100/1000 Mb/s Ethernet, RadioLAN, and EoIP (Ethernet over IP tunnel). No encryption, MPPE 40bit RSA, and MPPE 128bit RSA encryption are supported.

Supported connections:

- MikroTik RouterOS PPPoE client to any PPPoE server (access concentrator)
- MikroTik RouterOS server (access concentrator) to multiple PPPoE clients (clients are available for almost all OSs and some routers)

Specifications

Packages required : ***ppp***
License required : *Basic (DEMO license is limited to 4 tunnels)*
Home menu level : ***/interface pppoe−server***, ***/interface pppoe−client***
Protocols utilized : *PPPoE (RFC2516)*
Hardware usage: *PPPoE server may require additional RAM (uses approx. 200KB for each connection) and CPU power, supports maximum of 10000 connections*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
AAA (Authentication, Authorization and Accounting)

PPPoE Client Setup

Submenu level : **/interface pppoe−client**

Description

The PPPoE client supports high−speed connections. It is fully compatible with the MikroTik PPPoE server (access concentrator).

Note for Windows: Some connection instructions may use the form where the 'phone number' is 'MikroTik_AC\mt1' to indicate that 'MikroTik_AC' is the access concentrator name and 'mt1' is the service name.

Property Description

name (*name*; default: **pppoe−out1**) − name of the PPPoE interface
interface (*name*) − interface the PPPoE server can be connected through
mtu (*integer*; default: **1480**) − Maximum Transmit Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 20 (so, for 1500−byte ethernet link, set the MTU to 1480 to avoid fragmentation of packets)
mru (*integer*; default: **1480**) − Maximum Receive Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 20 (so, for 1500−byte ethernet link, set the MTU to 1480 to avoid fragmentation of packets)
user (*string*; default: "") − a user name that is present on the PPPoE server
password (*string*; default: "") − a user password used to connect the PPPoE server
profile (*name*) − default profile for the connection
service−name (*string*; default: "") − the service name set on the access concentrator. Many ISPs give user−name and address in the form of **user−name@service−name**

ac−name (*string*; default: "") − this may be left blank and the client will connect to any access concentrator that offers the **service−name** selected
add−default−route (yes | no; default: **no**) − whether to add a default route automatically
dial−on−demand (yes | no; default: **no**) − connects to AC only when outbound traffic is generated and disconnects when there is no traffic for the period set in the idle−timeout value
use−peer−dns − whether to set the router default DNS to the PPP peer DNS (i.e. whether to get DNS settings from the peer)

Notes

If there is a default route, **add−default−route** will not create a new one.

Example

To add and enable PPPoE client on the **gig** interface connecting to the AC that provides **testSN** service using username **john** with the password **password**:

```
[admin@RemoteOffice] interface pppoe−client> add interface=gig \
\... service−name=testSN user=john password=password disabled=no
[admin@RemoteOffice] interface pppoe−client> print
Flags: X − disabled, R − running
  0  R name="pppoe−out1" mtu=1480 mru=1480 interface=gig user="john"
       password="password" profile=default service−name="testSN" ac−name=""
       add−default−route=no dial−on−demand=no use−peer−dns=no
```

Monitoring PPPoE Client

Command name : **/interface pppoe−client monitor**

Property Description

Statistics:

status (*string*) − status of the client:
- **Dialing** − attempting to make a connection
- **Verifying password...** − connection has been established to the server, password verification in progress
- **Connected** − self−explanatory
- **Terminated** − interface is not enabled or the other side will not establish a connection

uptime (*time*) − connection time displayed in days, hours, minutes, and seconds
encoding (*string*) − encryption and encoding (if asymmetric, separated with '/') being used in this connection
service−name (*string*) − name of the serice the client is connected to
ac−name (*string*) − name of the AC the client is connected to
ac−mac (*MAC address*) − MAC address of the AC the client is connected to

Example

To monitor the **pppoe−out1** connection:

```
[admin@MikroTik] interface pppoe−client> monitor pppoe−out1
          status: "connected"
```

```
          uptime: 10s
        encoding: "none"
    service-name: "testSN"
         ac-name: "10.0.0.1"
          ac-mac: 00:C0:DF:07:5E:E6

[admin@MikroTik] interface pppoe-client>
```

PPPoE Server Setup (Access Concentrator)

Submenu level : **/interface pppoe-server server**

Description

The PPPoE server (access concentrator) supports multiple servers for each interface with differing service names. Currently the throughput of the PPPoE server has been tested to 160Mb/s on a Celeron 600 CPU. Using higher speed CPUs should increase the throughput proportionately.

The **access concentrator name** and PPPoE **service name** are used by clients to identify the access concentrator to register with. The **access concentrator name** is the same as the **identity** of the router displayed before the command prompt. The identity may be set within the **/system identity** submenu.

Property Description

service-name (*string*) - the PPPoE service name
mtu (*integer*; default: **1480**) - Maximum Transmit Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 20 (so, for 1500-byte ethernet link, set the MTU to 1480 to avoid fragmentation of packets)
mru (*integer*; default: **1480**) - Maximum Receive Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 20 (so, for 1500-byte ethernet link, set the MTU to 1480 to avoid fragmentation of packets)
authentication (*multiple choice*: pap | chap | mschap1 | mschap2; default: **mschap2, mschap, chap, pap**) - authentication algorithm
keepalive-timeout - defines the time period (in seconds) after which the router is starting to send keepalive packets every second. If no traffic and no keepalive responses has came for that period of time (i.e. 2 * **keepalive-timeout**), not responding client is proclaimed disconnected
one-session-per-host (yes | no; default: **no**) - allow only one session per host (determined by MAC address). If a host will try to establish a new session, the old one will be closed
default-profile (*name*; default: **default**) - default profile to use

Notes

The default **keepalive-timeout** value of **10** is OK in most cases. If you set it to **0**, the router will not disconnect clients until they log out or router is restarted. To resolve this problem, the **one-session-per-host** property can be used.

Security issue: do not assign an IP address to the interface you will be receiving the PPPoE requests on.

And also note that if service name is not specified in Windows XP, it will use only service with no name. So if you want to serve Windows XP clients, leave your **service-name** empty.

Example

To add PPPoE server on **ether1** interface providing **ex** service and allowing only one connection per host:

```
[admin@MikroTik] interface pppoe-server server> add interface=ether1 \
\... service-name=ex one-session-per-host=yes
[admin@MikroTik] interface pppoe-server server> print
Flags: X - disabled
  0 X service-name="ex" interface=ether1 mtu=1480 mru=1480
      authentication=mschap2,mschap,chap,pap keepalive-timeout=10
      one-session-per-host=yes default-profile=default

[admin@MikroTik] interface pppoe-server server>
```

PPPoE Server Users

Submenu level : **/interface pppoe-server**

Property Description

Statistics:

name (*name*) - interface name
service-name (*name*) - name of the service the user is connected to
remote-address (*MAC address*) - MAC address of the connected client
user (*name*) - the name of the connected user
encoding (*string*) - encryption and encoding (if asymmetric, separated with '/') being used in this connection
uptime - shows how long the client is connected

Example

To view the currently connected users:

```
[admin@MikroTik] interface pppoe-server> print
Flags: R - running
  #   NAME       SERVICE REMOTE-ADDRESS    USER    ENCO... UPTIME
  0 R <pppoe-ex> ex      00:C0:CA:16:16:A5 ex              12s

[admin@MikroTik] interface pppoe-server>
```

To disconnect the user **ex**:

```
[admin@MikroTik] interface pppoe-server> remove [find user=ex]
[admin@MikroTik] interface pppoe-server> print

[admin@MikroTik] interface pppoe-server>
```

PPPoE Troubleshooting

- ***The PPPoE server shows more than one active user entry for one client, when the clients disconnect, they are still shown and active***

Set the **keepalive−timeout** parameter (in the PPPoE server configuration) to **10** if You want clients to be considered logged off if they do not respond for 10 seconds.
Note that if the **keepalive−timeout** parameter is set to **0** and the **only−one** parameter (in PPP profile settings) is set to **yes** then the clients might be able to connect only once. To resolve this problem **one−session−per−host** parameter in PPPoE server configuration should be set to **yes**

- ***I can get through the PPPoE link only small packets (eg. pings)***
  You need to change MSS of all the packets passing through the PPPoE link to the value of PPPoE link's MTU−40 at least on one of the peers. So for PPPoE link with MTU of 1480:

```
[admin@MikroTik] ip firewall mangle> add protocol=tcp tcp-options=syn-only \
\.. action=passthrough tcp-mss=1440
[admin@MikroTik] ip firewall mangle> print
Flags: X - disabled, I - invalid
  0   src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 protocol=tcp tcp-options=syn-only
      icmp-options=any:any flow="" src-mac-address=00:00:00:00:00:00
      limit-count=0 limit-burst=0 limit-time=0s action=passthrough
      mark-flow="" tcp-mss=1440

[admin@MikroTik] ip firewall mangle>
```

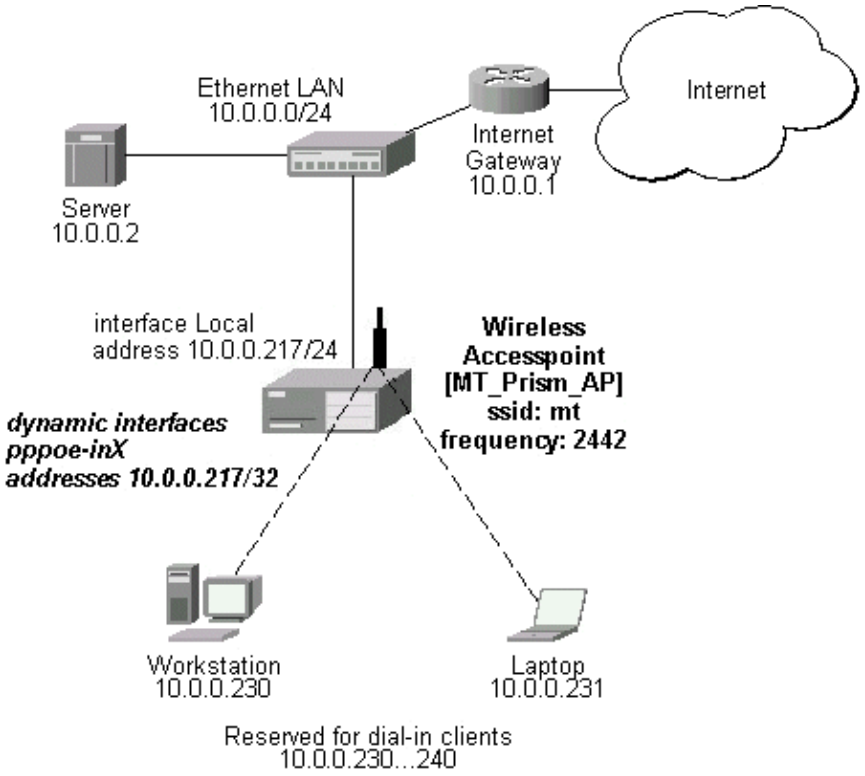
- ***My windows PPPoE client obtains IP address and default gateway from the MikroTik PPPoE server, but it cannot ping beyond the PPPoE server and use the Internet.***
  PPPoE server is not bridging the clients. Configure masquerading for the PPPoE client addresses, or make sure you have proper routing for the address space used by the clients, or you enable Proxy−ARP on the Ethernet interface (See the IP Addresses and Address Resolution Protocol (ARP) Manual).
- ***My Windows XP client cannot connect to the PPPoE server.***
  You have to specify the "Service Name" in the properties of the XP PPPoE client. If the service name is not set, or it does not match the service name of the MikroTik PPPoE server, you get the "line is busy" errors, or the system shows "verifying password − unknown error".
- ***I want to have logs for PPPoE connection establishment***
  Configure the logging feature under the **/system logging facility** and enable the PPP type logs.

Application Examples

PPPoE in a multipoint wireless 802.11 network

In a wireless network, the PPPoE server may be attached to an Access Point (as well as to a regular station of wireless infrastructure). Either our RouterOS client or Windows PPPoE clients may connect to the Access Point for PPPoE authentication. Further, for RouterOS clients, the radio interface may be set to MTU 1600 so that the PPPoE interface may be set to MTU 1500. This optimizes the transmission of 1500 byte packets and avoids any problems associated with MTUs lower than 1500. It has not been determined how to change the MTU of the Windows wireless interface at this moment.

Let us consider the following setup where the MikroTik Wireless AP offers wireless clients transparent access to the local network with authentication:



Note that you should have Basic + Wireless + Wireless AP licenses for this setup.

First of all, the Prism interface should be configured:

```
[admin@MT_Prism_AP] interface prism> set 0 mode=ap-bridge frequency=2442MHz \
\... ssid=mt disabled=no
[admin@MT_Prism_AP] interface prism> print
Flags: X - disabled, R - running
  0  R name="prism1" mtu=1500 mac-address=00:90:4B:02:17:E2 arp=enabled
       mode=ap-bridge root-ap=00:00:00:00:00:00 frequency=2442MHz ssid="mt"
       default-authentication=yes default-forwarding=yes max-clients=2007
       card-type=generic tx-power=auto supported-rates=1-11 basic-rates=1
       hide-ssid=no

[admin@MT_Prism_AP] interface prism> /ip address
```

Now, the Ethernet interface and IP address are to be set:

```
[admin@MT_Prism_AP] ip address> add address=10.0.0.217/24 interface=Local
[admin@MT_Prism_AP] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.217/24      10.0.0.0        10.0.0.255      Local

[admin@MT_Prism_AP] ip address> /ip route
[admin@MT_Prism_AP] ip route> add gateway=10.0.0.1
[admin@MT_Prism_AP] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        Local
    1 DC 10.0.0.0/24        r 0.0.0.0         0        Local
```

```
[admin@MT_Prism_AP] ip route> /interface ethernet
[admin@MT_Prism_AP] interface ethernet> set Local arp=proxy-arp
[admin@MT_Prism_AP] interface ethernet> print
Flags: X - disabled, R - running
  #    NAME                MTU   MAC-ADDRESS       ARP
  0  R Local               1500  00:50:08:00:00:F5 proxy-arp

[admin@MT_Prism_AP] interface ethernet>
```

We should add PPPoE server to the Prism interface:

```
[admin@MT_Prism_AP] interface pppoe-server server> add interface=prism1 \
\... service-name=mt one-session-per-host=yes disabled=no
[admin@MT_Prism_AP] interface pppoe-server server> print
Flags: X - disabled
  0   service-name="mt" interface=prism1 mtu=1480 mru=1480
      authentication=mschap2,mschap,chap,pap keepalive-timeout=10
      one-session-per-host=yes default-profile=default

[admin@MT_Prism_AP] interface pppoe-server server>
```

MSS should be changed for the packets flowing through the PPPoE link:

```
[admin@MT_Prism_AP] ip firewall mangle> add protocol=tcp tcp-options=syn-only \
\.. action=passthrough tcp-mss=1440
[admin@MT_Prism_AP] ip firewall mangle> print
Flags: X - disabled, I - invalid
  0   src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 protocol=tcp tcp-options=syn-only
      icmp-options=any:any flow="" src-mac-address=00:00:00:00:00:00
      limit-count=0 limit-burst=0 limit-time=0s action=passthrough
      mark-flow="" tcp-mss=1440

[admin@MT_Prism_AP] ip firewall mangle>
```

And finally, we can set up PPPoE clients:

```
[admin@MT_Prism_AP] ip pool> add name=pppoe ranges=10.0.0.230-10.0.0.240
[admin@MT_Prism_AP] ip pool> print
  # NAME                                      RANGES
  0 pppoe                                     10.0.0.230-10.0.0.240

[admin@MT_Prism_AP] ip pool> /ppp profile
[admin@MT_Prism_AP] ppp profile> set default use-encryption=yes \
\... local-address=10.0.0.217 remote-address=pppoe
[admin@MT_Prism_AP] ppp profile> print
Flags: * - default
  0 * name="default" local-address=10.0.0.217 remote-address=pppoe
      session-timeout=0s idle-timeout=0s use-compression=no
      use-vj-compression=no use-encryption=yes require-encryption=no
      only-one=no tx-bit-rate=0 rx-bit-rate=0 incoming-filter=""
      outgoing-filter=""

[admin@MT_Prism_AP] ppp profile> .. secret
[admin@MT_Prism_AP] ppp secret> add name=w password=wkst service=pppoe
[admin@MT_Prism_AP] ppp secret> add name=l password=ltp service=pppoe
[admin@MT_Prism_AP] ppp secret> print
Flags: X - disabled
```

```
  #   NAME              SERVICE CALLER-ID       PASSWORD        PROFILE
  0   w                 pppoe                   wkst            default
  1   l                 pppoe                   ltp             default
[admin@MT_Prism_AP] ppp secret> print
```

Thus we have completed the configuration and added two users: **w** and **l** who are able to connect using PPPoE client software.

Note that Windows XP built−in client supports encryption, but RASPPPOE does not. So, if it is planned not to support Windows clients older than Windows XP, it is recommended to switch **require−encryption** to **yes** value in the **default** profile configuration. In other case, the server will accept clients that do not encrypt data.

Additional Resources

Links for PPPoE documentation:

- http://www.ietf.org/rfc/rfc2516.txt
- http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120dc/120dc3/pppoe.
- http://www.carricksolutions.com

PPPoE Clients:

- RASPPPoE for Windows 95, 98, 98SE, ME, NT4, 2000, XP, .NET
  http://user.cs.tu−berlin.de/~normanb/

Point to Point Tunnel Protocol (PPTP)

Document revision 1.8 (27−Mar−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

PPTP (Point to Point Tunnel Protocol) supports encrypted tunnels over IP. The MikroTik RouterOS implementation includes support fot PPTP client and server.

General applications of PPTP tunnels:

- For secure router−to−router tunnels over the Internet
- To link (bridge) local Intranets or LANs (when EoIP is also used)
- For mobile or remote clients to remotely access an Intranet/LAN of a company (see PPTP setup for Windows for more information)

Each PPTP connection is composed of a server and a client. The MikroTik RouterOS may function as a server or client – or, for various configurations, it may be the server for some connections and client for other connections. For example, the client created below could connect to a Windows 2000 server, another MikroTik Router, or another router which supports a PPTP server.

Specifications

Packages required : ***ppp***
License required : *Basic (DEMO license is limited to 4 tunnels)*
Home menu level : ***/interface pptp−server***, ***/interface pptp−client***
Protocols utilized : *PPTP (RFC2637)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Authentication, Authorization and Accounting
Ethernet over IP (EoIP) Tunnel Interface

Description

PPTP is a secure tunnel for transporting IP traffic using PPP. PPTP encapsulates PPP in virtual lines that run over IP. PPTP incorporates PPP and MPPE (Microsoft Point to Point Encryption) to make encrypted links. The purpose of this protocol is to make well−managed secure connections between routers as well as between routers and PPTP clients (clients are available for and/or included in almost all OSs including Windows).

PPTP includes PPP authentication and accounting for each PPTP connection. Full authentication and accounting of each connection may be done through a RADIUS client or locally.

MPPE 40bit RC4 and MPPE 128bit RC4 encryption are supported.

PPTP traffic uses TCP port 1723 and IP protocol GRE (Generic Routing Encapsulation, IP protocol ID 47), as assigned by the Internet Assigned Numbers Authority (IANA). PPTP can be used with most firewalls and routers by enabling traffic destined for TCP port 1723 and protocol 47 traffic to be routed through the firewall or router.

PPTP connections may be limited or impossible to setup though a masqueraded/NAT IP connection. Please see the Microsoft and RFC links at the end of this section for more information.

PPTP Client Setup

Submenu level : **/interface pptp−client**

Property Description

name (*name*; default: **pptp−out1**) − interface name for reference
mtu (*integer*; default: **1460**) − Maximum Transmit Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 40 (so, for 1500−byte ethernet link, set the MTU to 1460 to avoid fragmentation of packets)
mru (*integer*; default: **1460**) − Maximum Receive Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 40 (so, for 1500−byte ethernet link, set the MRU to 1460 to avoid fragmentation of packets)
connect−to (*IP address*)− the IP address of the PPTP server to connect to

user (*string*)− user name to use when logging on to the remote server
password (*string*; default: "")− user password to use when logging to the remote server
profile (*name*; default: **default**) − profile to use when connecting to the remote server
add−default−route (yes | no; default: **no**) − whether to use the server which this client is connected to as its default router (gateway)

Example

To set up PPTP client named **test2** using username **john** with password **john** to connect to the **10.1.1.12** PPTP server and use it as the default gateway:

```
[admin@MikroTik] interface pptp−client> add name=test2 connect−to=10.1.1.12 \
\... user=john add−default−route=yes password=john
[admin@MikroTik] interface pptp−client> print
Flags: X − disabled, R − running
  0 X  name="test2" mtu=1460 mru=1460 connect−to=10.1.1.12 user="john"
       password="john" profile=default add−default−route=yes


[admin@MikroTik] interface pptp−client> enable 0
```

Monitoring PPTP Client

Command name : **/interface pptp−client monitor**

Property Description

Statistics:

uptime (*time*) − connection time displayed in days, hours, minutes, and seconds
encoding (*string*) − encryption and encoding (if asymmetric, separated with '/') being used in this connection
status (*string*) − status of the client:

- **Dialing** – attempting to make a connection
- **Verifying password...** − connection has been established to the server, password verification in progress
- **Connected** – self−explanatory
- **Terminated** – interface is not enabled or the other side will not establish a connection

Example

Example of an established connection:

```
[admin@MikroTik] interface pptp−client> monitor test2
      uptime: 4h35s
    encoding: MPPE 128 bit, stateless
      status: Connected
[admin@MikroTik] interface pptp−client>
```

PPTP Server Setup

Submenu level : **/interface pptp−server server**

```
[admin@MikroTik] interface pptp−server server> print
```

```
            enabled: no
                mtu: 1460
                mru: 1460
     authentication: mschap2
    default-profile: default
[admin@MikroTik] interface pptp-server server>
```

Description

The PPTP server supports unlimited connections from clients. For each current connection, a dynamic interface is created.

Property Description

enabled (yes | no; default: **no**) − defines whether PPTP server is enabled or not
mtu (*integer*; default: **1460**) − Maximum Transmit Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 40 (so, for 1500−byte ethernet link, set the MTU to 1460 to avoid fragmentation of packets)
mru (*integer*; default: **1460**) − Maximum Receive Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 40 (so, for 1500−byte ethernet link, set the MTU to 1460 to avoid fragmentation of packets)
authentication (*multiple choice*: pap | chap | mschap1 | mschap2; default: **mschap2**) − authentication algorithm
default−profile (*name*; default: **default**) − default profile to use

Example

To enable PPTP server:

```
[admin@MikroTik] interface pptp-server server> set enabled=yes
[admin@MikroTik] interface pptp-server server> print
            enabled: yes
                mtu: 1460
                mru: 1460
     authentication: mschap2
    default-profile: default
[admin@MikroTik] interface pptp-server server>
```

PPTP Server Users

Submenu level : **/interface pptp−server**

Description

There are two types of items in PPTP server configuration − static users and dynamic connections. A dynamic connection can be established if the user database or the **default−profile** has its **local−address** and **remote−address** set correctly. When static users are added, the default profile may be left with its default values and only P2P user (in **/ppp secret**) should be configured. **Note** that in both cases P2P users must be configured properly.

Property Description

name − interface name
user − the name of the user that is configured statically or added dynamically

Statistics:

mtu − shows (cannot be set here) client's MTU
client−address − shows (cannot be set here) the IP of the connected client
uptime − shows how long the client is connected
encoding (*string*) − encryption and encoding (if asymmetric, separated with '/') being used in this connection

Example

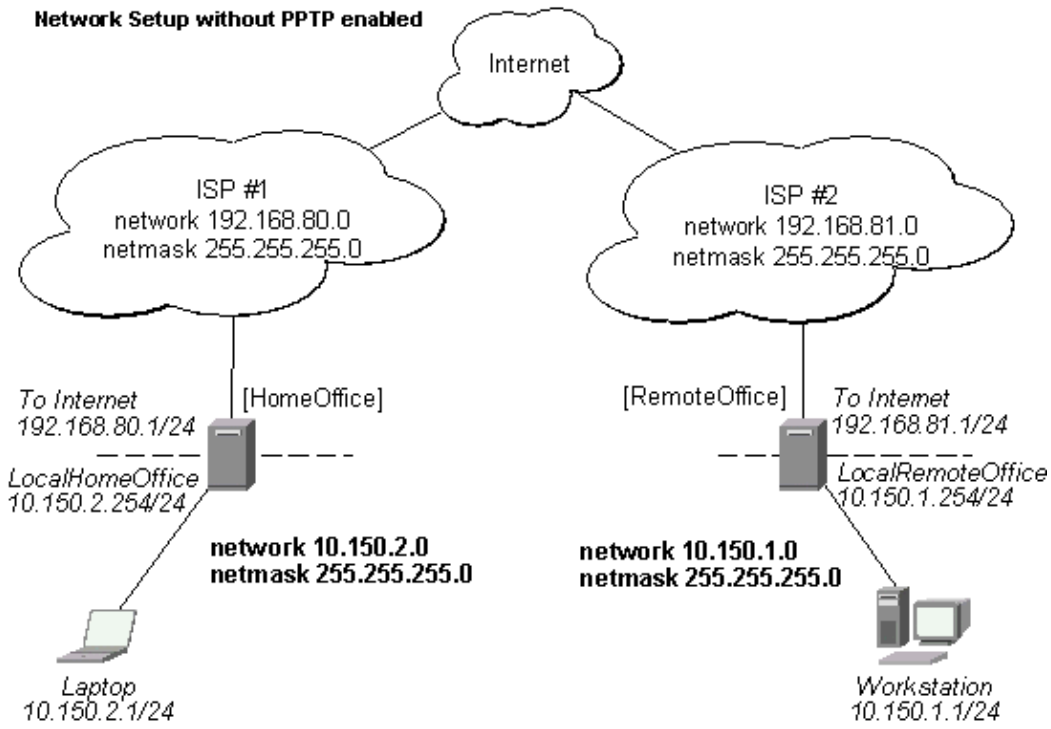
To add a static entry for **ex1** user:

```
[admin@MikroTik] interface pptp−server> add user=ex1
[admin@MikroTik] interface pptp−server> print
Flags: X − disabled, D − dynamic, R − running
  #     NAME                USER         MTU   CLIENT−ADDRESS  UPTIME   ENC...
  0  DR <pptp−ex>           ex           1460  10.0.0.202      6m32s    none
  1     pptp−in1            ex1
[admin@MikroTik] interface pptp−server>
```

In this example an already connected user **ex** is shown besides the one we just added.

PPTP Router−to−Router Secure Tunnel Example

The following is an example of connecting two Intranets using an encrypted PPTP tunnel over the Internet.



There are two routers in this example:

- [HomeOffice]
  Interface LocalHomeOffice 10.150.2.254/24
  Interface ToInternet 192.168.80.1/24
- [RemoteOffice]
  Interface ToInternet 192.168.81.1/24
  Interface LocalRemoteOffice 10.150.1.254/24

Each router is connected to a different ISP. One router can access another router through the Internet.

On the PPTP server a user must be set up for the client:

```
[admin@HomeOffice] ppp secret> add name=ex service=pptp password=lkjrht
local-address=10.0.103.1 remote-address=10.0.103.2
[admin@HomeOffice] ppp secret> print detail
Flags: X - disabled
  0   name="ex" service=pptp caller-id="" password="lkjrht" profile=default
      local-address=10.0.103.1 remote-address=10.0.103.2 routes==""

[admin@HomeOffice] ppp secret>
```

Then the user should be added in the PPTP server list:

```
[admin@HomeOffice] interface pptp-server> add user=ex
[admin@HomeOffice] interface pptp-server> print
Flags: X - disabled, D - dynamic, R - running
  #     NAME                USER         MTU   CLIENT-ADDRESS  UPTIME   ENC...
  0     pptp-in1            ex
[admin@HomeOffice] interface pptp-server>
```

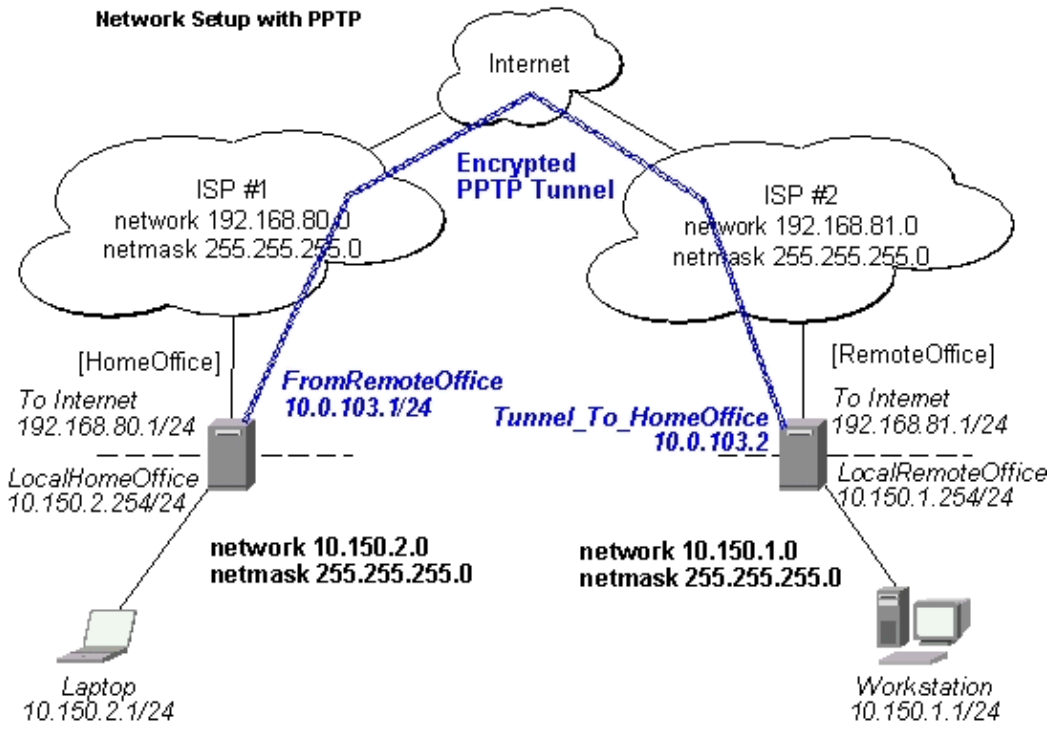
And finally, the server must be enabled:

```
[admin@HomeOffice] interface pptp-server server> set enabled=yes
[admin@HomeOffice] interface pptp-server server> print
            enabled: yes
                mtu: 1460
                mru: 1460
     authentication: mschap2
    default-profile: default
[admin@HomeOffice] interface pptp-server server>
```

Add a PPTP client to the RemoteOffice router:

```
[admin@RemoteOffice] interface pptp-client> add connect-to=192.168.80.1 user=ex \
\... password=lkjrht disabled=no
[admin@RemoteOffice] interface pptp-client> print
Flags: X - disabled, R - running
  0  R name="pptp-out1" mtu=1460 mru=1460 connect-to=192.168.80.1 user="ex"
       password="lkjrht" profile=default add-default-route=no


[admin@RemoteOffice] interface pptp-client>
```

Thus, a PPTP tunnel is created between the routers. This tunnel is like an Ethernet point-to-point connection between the routers with IP addresses 10.0.103.1 and 10.0.103.2 at each router. It enables 'direct' communication between the routers over third party networks.



To route the local Intranets over the PPTP tunnel – add these routes:

```
[admin@HomeOffice] > ip route add dst-address 10.150.1.0/24 gateway 10.0.103.2
[admin@RemoteOffice] > ip route add dst-address 10.150.2.0/24 gateway 10.0.103.1
```

On the PPTP server it can alternatively be done using **routes** parameter of the user configuration:

```
[admin@HomeOffice] ppp secret> print detail
Flags: X − disabled
  0   name="ex" service=pptp caller−id="" password="lkjrht" profile=default
      local−address=10.0.103.1 remote−address=10.0.103.2 routes==""

[admin@HomeOffice] ppp secret> set 0 routes="10.150.1.0/24 10.0.103.2 1"
[admin@HomeOffice] ppp secret> print detail
Flags: X − disabled
  0   name="ex" service=pptp caller−id="" password="lkjrht" profile=default
      local−address=10.0.103.1 remote−address=10.0.103.2
      routes="10.150.1.0/24 10.0.103.2 1"

[admin@HomeOffice] ppp secret>
```

Test the PPTP tunnel connection:

```
[admin@RemoteOffice]> /ping 10.0.103.1
10.0.103.1 pong: ttl=255 time=3 ms
10.0.103.1 pong: ttl=255 time=3 ms
10.0.103.1 pong: ttl=255 time=3 ms
ping interrupted
3 packets transmitted, 3 packets received, 0% packet loss
round−trip min/avg/max = 3/3.0/3 ms
```

Test the connection through the PPTP tunnel to the LocalHomeOffice interface:

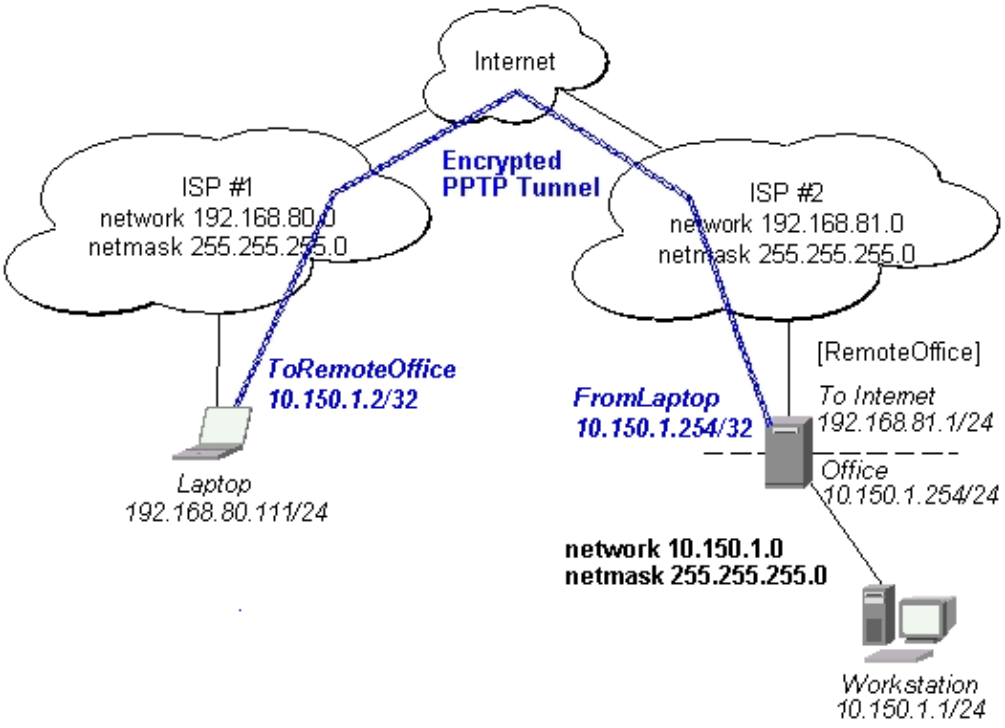
```
[admin@RemoteOffice]> /ping 10.150.2.254
10.150.2.254 pong: ttl=255 time=3 ms
10.150.2.254 pong: ttl=255 time=3 ms
10.150.2.254 pong: ttl=255 time=3 ms
ping interrupted
3 packets transmitted, 3 packets received, 0% packet loss
round−trip min/avg/max = 3/3.0/3 ms
```

To bridge a LAN over this secure tunnel, please see the example in the 'EoIP' section of the manual. To set the maximum speed for traffic over this tunnel, please consult the 'Queues' section.

Connecting a Remote Client via PPTP Tunnel

The following example shows how to connect a computer to a remote office network over PPTP encrypted tunnel giving that computer an IP address from the same network as the remote office has (without need of bridging over eoip tunnels)

Please, consult the respective manual on how to set up a PPTP client with the software You are using.



The router in this example:

- [RemoteOffice]
  Interface ToInternet 192.168.81.1/24
  Interface Office 10.150.1.254/24

The client computer can access the router through the Internet.

On the PPTP server a user must be set up for the client:

```
[admin@RemoteOffice] ppp secret> add name=ex service=pptp password=lkjrht
local-address=10.150.1.254 remote-address=10.150.1.2
[admin@RemoteOffice] ppp secret> print detail
Flags: X - disabled
  0   name="ex" service=pptp caller-id="" password="lkjrht" profile=default
      local-address=10.150.1.254 remote-address=10.150.1.2 routes==""

[admin@RemoteOffice] ppp secret>
```

Then the user should be added in the PPTP server list:

```
[admin@RemoteOffice] interface pptp-server> add name=FromLaptop user=ex
[admin@RemoteOffice] interface pptp-server> print
Flags: X - disabled, D - dynamic, R - running
  #     NAME                USER         MTU   CLIENT-ADDRESS  UPTIME   ENC...
  0     FromLaptop          ex
[admin@RemoteOffice] interface pptp-server>
```

And the server must be enabled:

```
[admin@RemoteOffice] interface pptp-server server> set enabled=yes
[admin@RemoteOffice] interface pptp-server server> print
```

```
            enabled: yes
                mtu: 1460
                mru: 1460
     authentication: mschap2
    default-profile: default
[admin@RemoteOffice] interface pptp-server server>
```

Finally, the proxy APR must be enabled on the 'Office' interface:

```
[admin@RemoteOffice] interface ethernet> set Office arp=proxy-arp
[admin@RemoteOffice] interface ethernet> print
Flags: X - disabled, R - running
  #    NAME                MTU   MAC-ADDRESS       ARP
  0  R ToInternet          1500  00:30:4F:0B:7B:C1 enabled
  1  R Office              1500  00:30:4F:06:62:12 proxy-arp
[admin@RemoteOffice] interface ethernet>
```

PPTP Setup for Windows

Microsoft provides PPTP client support for Windows NT, 2000, ME, 98se, and 98. Windows 98se, 2000, and ME include support in the Windows setup or automatically install PPTP. For 95, NT, and 98, installation requires a download from Microsoft. Many ISPs have made help pages to assist clients with Windows PPTP installation.

http://www.real-time.com/Customer_Support/PPTP_Config/pptp_config.html
http://www.microsoft.com/windows95/downloads/contents/WUAdminTools/S_WUNetworkingTools/W95Winsock

Sample instructions for PPTP (VPN) installation and client setup – Windows 98se

If the VPN (PPTP) support is installed, select 'Dial-up Networking' and 'Create a new connection'. The option to create a 'VPN' should be selected. If there is no 'VPN' options, then follow the installation instructions below. When asked for the 'Host name or IP address of the VPN server', type the IP address of the router. Double-click on the 'new' icon and type the correct user name and password (must also be in the user database on the router or RADIUS server used for authentication).

The setup of the connections takes nine seconds after selection the 'connect' button. It is suggested that the connection properties be edited so that 'NetBEUI', 'IPX/SPX compatible', and 'Log on to network' are unselected. The setup time for the connection will then be two seconds after the 'connect' button is selected.

To install the 'Virtual Private Networking' support for Windows 98se, go to the 'Setting' menu from the main 'Start' menu. Select 'Control Panel', select 'Add/Remove Program', select the 'Windows setup' tab, select the 'Communications' software for installation and 'Details'. Go to the bottom of the list of software and select 'Virtual Private Networking' to be installed.

Troubleshooting

- *I use firewall and I cannot establish PPTP connection*
  Make sure the TCP connections to port 1723 can pass through both directions between your sites. Also, IP protocol 47 should be passed through.

Additional Resources

Links for PPTP documentation:

http://msdn.microsoft.com/library/backgrnd/html/understanding_pptp.htm
http://support.microsoft.com/support/kb/articles/q162/8/47.asp
http://www.ietf.org/rfc/rfc2637.txt?number=2637
http://www.ietf.org/rfc/rfc3078.txt?number=3078
http://www.ietf.org/rfc/rfc3079.txt?number=3079

PrismII Wireless Client and Wireless Access Point Manual

Document revision 1.5 (11−Aug−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The MikroTik RouterOS supports the PrismII chipset based wireless adapter cards for working both as wireless clients (**station** mode) and wireless access points (**ap−bridge** or **bridge** mode).

Supports event logging.

Specifications

Packages required : ***wireless***
License required : *2.4GHz Wireless, 2.4GHz Wireless Access Point (optional)*
Home menu level : ***/interface prism***
Protocols utilized : *IEEE802.11b (IEEE802.11b)*
Hardware usage: not significant

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
Log Management

Description

Prism−based cards can be used in the following network roles:

- **Wireless Client** − IEEE 802.11b wireless client (station) associating with an access point. The station mode has been tested with MikroTik RouterOS PrismII based Access Points and CISCO/Aironet Wireless Ethernet Bridges and Access Points.
- **Wireless Access Point** − IEEE 802.11b wireless access point (requires the 2.4GHz AP Feature License). The access point can register wireless clients. The access point mode has been tested with PrismII, CISCO/Aironet and ORiNOCO/WaveLAN clients. The PrismII Access Point interface can also register other access points. Thus, it is possible to bridge networks over wireless links.
- **Wireless Bridge** − limited version of the Access Point mode which allows only one client to be registered but does not require the 2.4GHz Wireless AP Feature License, only the 2.4GHz Wireless License. Thus, it is possible to create point−to−point links and bridge networks over wireless links.

> ***Important!*** Prism 2.5 200mW cards require firmware 1.3.6 or lower, if you want to bridge networks. Please see the troubleshooting section on how to check the firmware version and downgrade it

The MikroTik RouterOS supports as many Prism chipset based cards as many free adapter slots are there on your system. One license is valid for all cards on your system. **Note** that maximal number of PCMCIA sockets is 8.

Wireless Interface Configuration

Submenu level : **/interface prism**

Property Description

name (*name*; default: **prism1**)− interface name
mtu (*integer*; default: **1500**) − maximum transfer unit
mac−address (*MAC address*) − MAC address of card. In AP mode this will also be BSSID of BSS
arp (disabled | enabled | proxy−arp | reply−only; default: **enable**) − Address Resolution Protocol:

- **disabled** − the interface will not use ARP protocol
- **enabled** − the interface will use ARP protocol
- **proxy−arp** − the interface will be an ARP proxy (see corresponding manual)
- **reply−only** − the interface will only reply to the requests originated to its own IP addresses, but neighbour MAC addresses will be gathered from **/ip arp** statically set table only

mode (station | bridge | ap−bridge; default: **station**) − mode of the interface:

- **station** − card works as station (client) for the wireless infrastructure
- **bridge** − card works as access point, but can register only one client or access point
- **ap−bridge** − card works as access point, i.e., it creates wireless infrastructure

root−ap (*MAC address*; default: **00:00:00:00:00:00**) − MAC address of the root access point to register to
frequency (2412MHz, 2417MHz ... 2472MHz; default: **2412MHz**) − frequency that AP will use to create BSS
ssid (*string*; default: **MikroTik**) − Service Set Identifier:

- in station mode − SSID to connect to
- in AP and P2P mode − SSID to use when creating BSS (can not be left blank)

default−authentication (yes | no; default: **yes**) − what to do with client that wants to associate, but it is not in the access−list
default−forwarding (yes | no; default:**yes**) − what to do with client that wants to send packets to other wireless clients, but it is not in the access−list
max−clients (*integer*:1...2007; default: **2007**) − maximum number of clients (including other access points), that is allowed to associate with this access point
card−type (100mW | 200mW | 30mW | generic; default: **generic**)− card type used for power settings
tx−power (0dBm−1mW...23dBm−200mW | auto; default: **auto**)− transmit power level. Has no effect if card type is **generic**

- **auto** − default setting of the card

supported−rates (1 | 1−11 | 1−2 | 1−5.5 | 11 | 2 | 2−11 | 2−5.5 | 5.5 | 5.5−11; default: **1−11**) − rates at which this node will work
basic−rates (1 | 1−11 | 1−2 | 1−5.5 | 11 | 2 | 2−11 | 2−5.5 | 5.5 | 5.5−11; default: **1**) − rates that every client that plans to connect to this AP should be able to work at. It is recommended to set it to **1**, since not all clients might support rates **1−11**
hide−ssid (yes | no; default: **no**) − whether to hide SSID (do not broadcast SSID over the wireless infrastructure)

Notes

root−ap, **default−authentication**, **default−forwarding**, **max−clients**, **basic−rates** properties are used in **bridge** and **ap−bridge** modes only.

Example

To enable **prism1** interface to connect to the wireless infrastructure **test**:

```
[admin@MikroTik] interface prism> set prism1 disabled=no ssid=test
[admin@MikroTik] interface prism> print
Flags: X - disabled, R - running
  0  R name="prism1" mtu=1500 mac-address=00:02:6F:01:D2:7D arp=enabled
       mode=station root-ap=00:00:00:00:00:00 frequency=2412MHz ssid="test"
       default-authentication=yes default-forwarding=yes max-clients=2007
       card-type=generic tx-power=auto supported-rates=1-11 basic-rates=1
       hide-ssid=no

[admin@MikroTik] interface prism>
```

Monitoring the Interface Status

Command name:**/interface prism monitor**

Property Description

Statistics:

status (searching−for−network | connected−to−ess | out−of−range) − status of the interface:

- **searching−for−network** − the card has not registered to an AP and is searching for one to register to
- **connected−to−ess** − the card has registered to an AP
- **out−of−range** − the card has registered to an AP, but lost the connection to it

data−rate (1Mbps | 2Mbps | 5.5Mbps | 11Mbps) − the actual data rate of the connection
ssid (*string*)− the Service Set Identifier.
bssid (*MAC address*) − the Basic Service Set Identifier (actually, the MAC address of the access point)
signal−quality (0...92) − the signal quality
signal−level (27...154)− the average signal level
noise−level (−100...0) − the average noise level

Notes

The monitor command does not work, if the interface is disabled, or the mode is **ap−bridge** or **bridge**.

Example

To monitor **prism1** interface:

```
[admin@MikroTik] interface prism> monitor prism1
                     status: connected-to-ess
                  data-rate: 11Mbps
                       ssid: "test"
                      bssid: 00:03:2F:04:25:10
              signal-quality: 92
                signal-level: 54
                 noise-level: -99

[admin@MikroTik] interface prism>
```

Registration Table

Submenu level : /**interface prism registration−table**

Property Description

Statistics: **interface** (*name*) − interface that client is registered to
mac−address (*MAC address*) − mac address of the registered client
type (client | local | ap | forward | parent−ap)− type of the client:

- **client** − client registered to the interface
- **local** − client learned from bridged interface
- **ap** − client is an access point
- **forward** − client is forwarded from another access point
- **parent−ap** − the access point this interface is connected to

parent (*MAC address*) − parent access point's MAC address, if forwarded from another access point
packets (*integer*) − number of received and sent packets
bytes (*integer*) − number of received and sent bytes
signal−level (*min/average/max*: 0...92) − min/average/max signal level
noise−level (*min/average/max*: −100...0) − min/average/max noise level
data−rate (*min/average/max*: (1 | 2 | 5.5 | 11)) − min/average/max receive data rate
tx−rate (1 | 2 | 5.5 | 11) − transmit data rate
last−update (*time*) − time since the last update
uptime (*time*) − time the client is associated with the access point

Example

To see registration table showing all clients currently associated with the access point:

```
[admin@MikroTik] interface prism> registration−table print
  # INTERFACE                     MAC−ADDRESS       TYPE      PARENT
  0 prism1                        00:07:EB:30:E7:DA client
  1 prism1                        00:40:96:29:2F:80 client
[admin@MikroTik] interface prism>
```

To get additional statistics:

```
[admin@MikroTik] interface prism> registration−table print stats
  0 interface=prism1 mac−address=00:07:EB:30:E7:DA type=client packets=0,19
    bytes=0,482 signal−level=69/75/138 noise−level=0/0/0 data−rate=10/110/110
    tx−rate=10 last−update=00:00:00.840 uptime=00:02:59.180

  1 interface=prism1 mac−address=00:40:96:29:2F:80 type=client packets=0,14
    bytes=0,196 signal−level=66/72/84 noise−level=0/0/0 data−rate=10/10/10
    tx−rate=10 last−update=00:00:08.380 uptime=00:02:42.220


[admin@MikroTik] interface prism>
```

Access List

Submenu level : /**interface prism access−list**

Description

The access list is used by the access point to restrict authentications (associations) of clients. This list contains MAC address of client and associated action to take when client attempts to connect. Also, the forwarding of frames sent by the client is controlled.

The association procedure is as follows: when a new client wants to associate to the AP that is configured on interface prismX, entry with client's MAC address and interface prismX is looked up in the access−list. If such entry is found, action specified in it is taken. Otherwise **default−authentication** and **default−forwarding** of interface prismX is taken.

Property Description

mac−address (*MAC address*; default: **00:00:00:00:00:00**) − MAC address of the client
interface (*name*) − AP interface
authentication (yes | no; default: **yes**) − accept this client when it tries to connect or not
forwarding (yes | no; default: **yes**) − forward the client's frames to other wireless clients or not

Notes

If you have default authentication action for the interface set to **yes**, you can disallow this node to register at the AP's interface **prism1** by setting **authentication=no** for it. Thus, all nodes except this one will be able to register to the interface **prism1**.

If you have default authentication action for the interface set to **no**, you can allow this node to register at the AP's interface **prism1** by setting **authentication=yes** for it. Thus, only the specified nodes will be able to register to the interface **prism1**.

Example

To allow authentication and forwarding for the client **00:40:96:37:A3:39** from the **prism1** interface:

```
[admin@MikroTik] interface prism access−list> add mac−address=00:40:96:37:A3:39
interface=prism1
[admin@MikroTik] interface prism access−list> print
Flags: X − disabled, I − invalid
  0   mac−address=00:40:96:37:A3:39 interface=prism1 authentication=yes
      forwarding=yes

[admin@MikroTik] interface prism access−list>
```

Network Scan

Command name:**/interface prism scan**

Description

The prism interface has feature that allows scanning for available networks. While scanning, the card unregisters itself from the access point (in **station** mode), or unregisters all clients (in **bridge** or **ap−bridge** mode). Thus, network connections are lost while scanning.

The result of scanning contains a list of discovered access points along with their MAC addresses, channel frequencies, service set identificators, and the measured signal level.

Property Description

(*name*) − interface name to use for scanning
frequencies (*string*; default: all frequencies) − list of frequencies to scan for, e.g., **2412MHz,2427MHz**
time (*time*; default:) − time to scan for one frequency. The total time used for scanning is multiplier of this value and the number of frequencies to scan

Example

To scan the wireless network from **prism1** interface:

```
[admin@MikroTik] interface prism> scan prism1
00:02:6f:01:5d:fe frequency=2412MHz ssid=waubonsie_low_ap1 signal−level=132
00:02:6f:01:63:0b frequency=2427MHz ssid=john signal−level=114
00:02:6f:01:62:ee frequency=2462MHz ssid=sales signal−level=0
[admin@MikroTik] interface prism>
```

Troubleshooting

- *The prism interface does not show up under the interfaces list*
  Obtain the required license for 2.4GHz wireless feature.
- *The access−list has entries restricting the registration, but the node is still registered.*
  Set some parameter of the prism interface to get all nodes re−register.
- *The AP to AP bridge does not work. Both Access Points are shown as clients in the registration table.*
  Prism 2.5 200mW cards require firmware 1.3.6 or lower, if you want to bridge networks. Firmware can be downgraded by installing the card in a Windows computer, and running the firmware upgrade utility FRMUPDATE25.EXE from 2511cd_frm306.zip
- *The wireless card does not register to the AP*
  Check the cabling and antenna alignment.

Basic Configuration Examples

Station Mode Configuration

Description

To set the wireless interface for working with an IEEE 802.11b access point (register to the AP), you should set the following parameters:

- The **Service Set Identifier (ssid)**. It should match the ssid of the AP.
- The **Operation Mode (mode)** of the card should be set to **station**.
- The **Supported Rate (supppрted−rates)** of the card should match the basic rates of the AP. For example, if the AP has **basic−rate=1**, the client can have **supported−rate=1−11**. If the AP has **basic−rate=1−11**, then all clients MUST have the **supported−rate=1−11** Thus, it is okay to leave the **supported−rate=1−11** for the client. All other parameters can be left as default.

Example

To configure the wireless interface **prism1** for registering to an AP with ssid **testing**:

```
[admin@MikroTik] interface prism> set prism1 ssid=testing
[admin@MikroTik] interface prism> enable prism1
[admin@MikroTik] interface prism> print
Flags: X - disabled, R - running
  0  R name="prism1" mtu=1500 mac-address=00:90:4B:02:17:E2 arp=enabled
       mode=station root-ap=00:00:00:00:00:00 frequency=2412MHz ssid="testing"
       default-authentication=yes default-forwarding=yes max-clients=2007
       card-type=generic tx-power=auto supported-rates=1-11 basic-rates=1
       hide-ssid=no

[admin@MikroTik] interface prism>
```

Access Point Mode Configuration

Description

To set the wireless interface for working as an IEEE 802.11b access point (register clients), you need both the 2.4GHz Wireless Feature License and the Prism AP Feature Licenses. You should set the following parameters:

- The **Service Set Identifier (ssid)**. It should be unique for your system.
- The **Operation Mode (mode)** of the card should be set to **ap-bridge** or **bridge** In **bridge** mode, only one client can be registered.
- The **Frequency** of the card.

All other parameters can be left as default. However, you should make sure, that all clients support the basic rate of your access point, i.e., the **supported-rates** of the client should cove the **basic-rates** of the access point.

Example

To configure the wireless interface **prism1** for working as an access point with ssid **testing** and use the frequency **2442MHz**:

```
[admin@MikroTik] interface prism> set prism1 mode=ap-bridge frequency=2442MHz \
\... ssid=testing
[admin@MikroTik] interface prism> print
Flags: X - disabled, R - running
  0  R name="prism1" mtu=1500 mac-address=00:90:4B:02:17:E2 arp=enabled
       mode=ap-bridge root-ap=00:00:00:00:00:00 frequency=2442MHz ssid="testing"
       default-authentication=yes default-forwarding=yes max-clients=2007
       card-type=generic tx-power=auto supported-rates=1-11 basic-rates=1
       hide-ssid=no

[admin@MikroTik] interface prism>
```

Registering the Access Point to another Access Point

Description

You can configure the access point to registering to another (root) access point by specifying the MAC address of the root access point.

The 'non−root' access point will register the clients only if it is registered to the 'root' access point.

Having one access point registered to another one enables bridging the networks, if bridging mode between prism and ethernet interfaces is used. Note, that in the station mode, bridging cannot be used between prism and ethernet interfaces.

> ***Important!*** Prism 2.5 200mW cards require firmware 1.3.6 or lower, if you want to bridge networks. Please see the troubleshooting section on how to check the firmware version and downgrade it.

Example

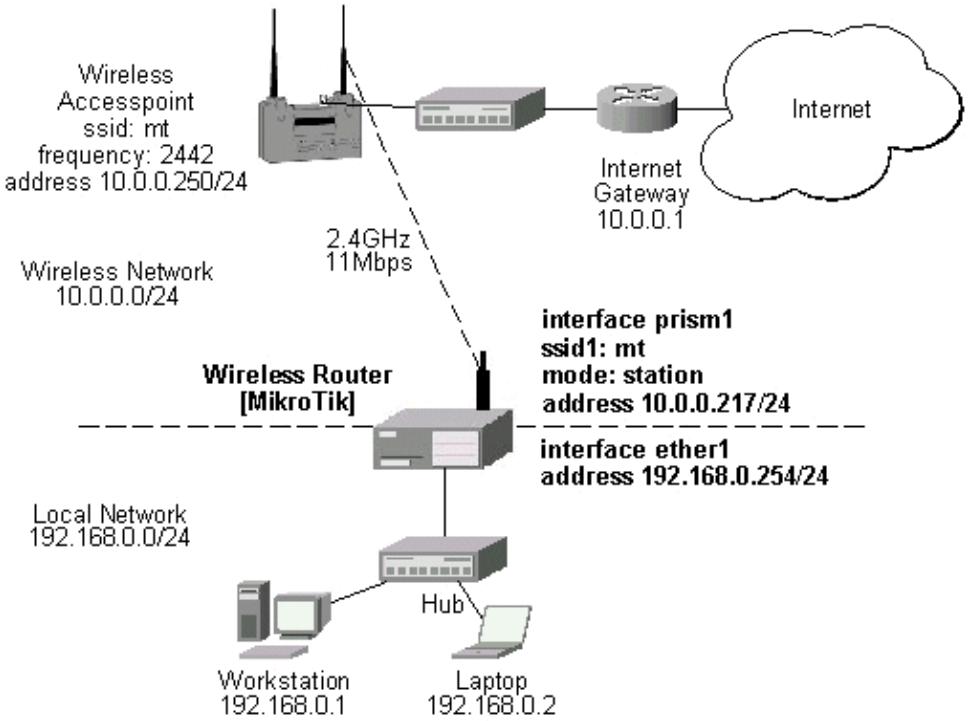
To configure the wireless interface **prism1** to register to the **00:90:4B:02:17:E2** root access point:

```
[admin@MikroTik] interface prism> set prism1 root−ap=00:90:4B:03:F1:71
[admin@MikroTik] interface prism> print
Flags: X − disabled, R − running
  0  R name="prism1" mtu=1500 mac−address=00:90:4B:02:17:E2 arp=enabled
       mode=ap−bridge root−ap=00:90:4B:03:F1:71 frequency=2442MHz ssid="testing"
       default−authentication=yes default−forwarding=yes max−clients=2007
       card−type=generic tx−power=auto supported−rates=1−11 basic−rates=1
       hide−ssid=no

[admin@MikroTik] interface prism>
```

Wireless Network Applications

Wireless Client

Let us consider the following point−to−multipoint network setup with CISCO/Aironet Wireless Access Point as a base station and MikroTik Wireless Router as a client:



3rd Party Wireless AP Configuration

The access point is connected to the wired network's HUB and has IP address from the network 10.0.0.0/24. The minimum configuration required for the AP is:

1. Setting the Service Set Identifier (up to 32 alphanumeric characters). In our case we use ssid "mt".
2. Setting the allowed data rates at 1−11Mbps, and the basic rate at 1Mbps.
3. Choosing the frequency, in our case we use 2442MHz.
4. Setting the identity parameters: ip address/mask and gateway. These are required if you want to access the AP remotely using telnet or http.
5. If you use CISCO/Aironet Wireless Ethernet Bridge or Access Point, you should set the Configuration/Radio/I80211/Extended (Allow proprietary extensions) to **off**, and the Configuration/Radio/I80211/Extended/Encapsulation (Default encapsulation method) to **RFC1042**. If left to the default **on** and **802.1H**, respectively, you won't be able to pass traffic through the bridge.

Note that the AP is not a router! It has just one network address, and is just like any host on the network. It resembles a wireless−to−Ethernet HUB or bridge. The AP does not route the IP traffic!

MikroTik RouterOS Wireless Client Configuration

The minimum configuration for the MikroTik router's prism wireless interface is:

1. Setting the Service Set Identifier to that of the AP, i.e., "mt"
2. The Operation Mode should be **station**.

```
[admin@MikroTik] interface prism> set 0 ssid=mt
[admin@MikroTik] interface prism> print
Flags: X − disabled, R − running
  0  R name="prism1" mtu=1500 mac−address=00:02:6F:01:D2:7D arp=enabled
```

```
       mode=station root-ap=00:00:00:00:00:00 frequency=2412MHz ssid="mt"
       default-authentication=yes default-forwarding=yes max-clients=2007
       card-type=generic tx-power=auto supported-rates=1-11 basic-rates=1
       hide-ssid=no

[admin@MikroTik] interface prism> monitor 0
                     status: connected-to-ess
                  data-rate: 11Mbps
                       ssid: "mt"
                      bssid: 00:40:96:56:E2:AD
              signal-quality: 78
                signal-level: 125
                 noise-level: -99

[admin@MikroTik] interface prism>
```

The IP addresses assigned to the wireless interface should be from the network 10.0.0.0/24, e.g.:

```
[admin@MikroTik] ip address> add address=10.0.0.217/24 interface=prism1
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.217/24      10.0.0.0        10.0.0.255      prism1
  1   192.168.0.254/24   192.168.0.254   192.168.0.254   ether1
[MikroTik] ip address>
```

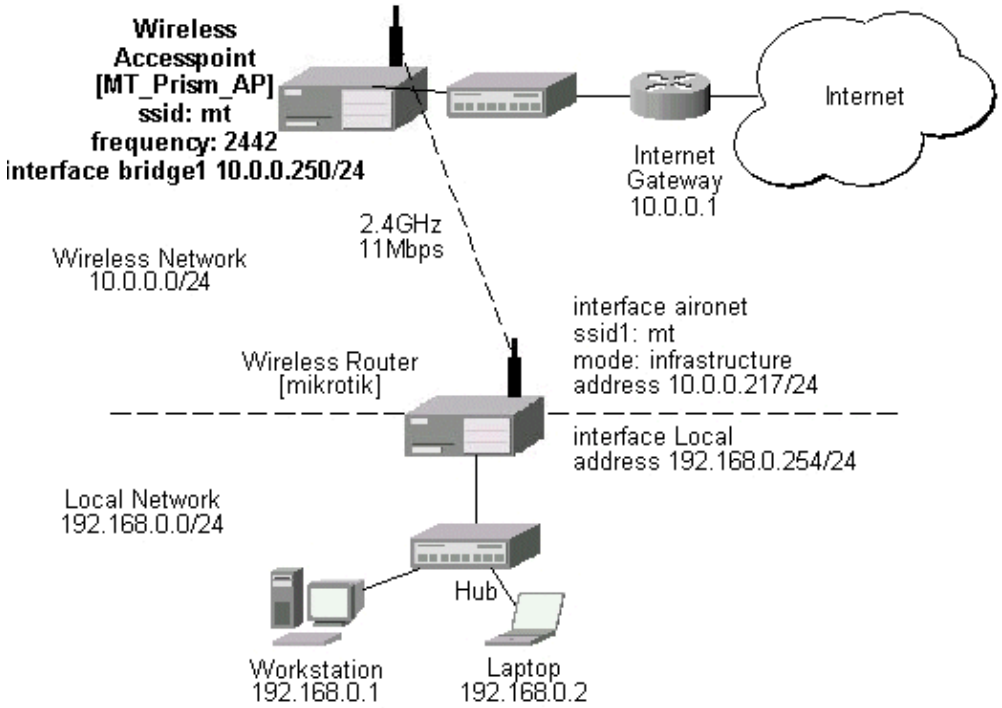
The default route should be set to the gateway router 10.0.0.1 (not to the AP 10.1.1.250 !):

```
[admin@MikroTik] ip route> add gateway=10.0.0.1
[admin@MikroTik] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        prism1
    1 DC 10.0.0.0/24        r 0.0.0.0         0        prism1
    2 DC 192.168.0.0/24     r 0.0.0.0         0        ether1
[admin@MikroTik] interface prism>
```

Note! You cannot use the bridging function between the prism and ethernet interfaces, if the prism interface is in the station mode. The bridge does not work in this case!

Wireless Access Point

Let us consider the following point-to-point wireless network setup with two MikroTik Wireless Routers:



You need both the 2.4GHz Wireless and the Prism AP Feature Licenses to enable the AP mode. To make the MikroTik router work as an access point, the configuration of the prism wireless interface should be as follows:

- A unique Service Set Identificator should be chosen, say "mt"
- A frequency should be selected for the link, say 2442MHz
- The operation mode should be set to **ap−bridge** or **bridge**.

The following command should be issued to change the settings for the prism interface:

```
[admin@MT_Prism_AP] interface prism> set 0 mode=ap-bridge frequency=2442MHz \
\... ssid=mt
[admin@MT_Prism_AP] interface prism> print
Flags: X - disabled, R - running
  0  R name="prism1" mtu=1500 mac-address=00:90:4B:02:17:E2 arp=enabled
       mode=ap-bridge root-ap=00:00:00:00:00:00 frequency=2442MHz ssid="mt"
       default-authentication=yes default-forwarding=yes max-clients=2007
       card-type=generic tx-power=auto supported-rates=1-11 basic-rates=1
       hide-ssid=no

[admin@MT_Prism_AP] interface prism> monitor 0
           current-sta-count: 2
            current-ap-count: 0
         current-local-count: 0
    current-forwarding-count: 0

[admin@MT_Prism_AP] interface prism>
```

The list of registered clients looks like follows:

```
[admin@MT_Prism_AP] interface prism> registration-table print
  # INTERFACE                   MAC-ADDRESS       TYPE      PARENT
  0 prism1                      00:07:EB:30:E7:DA client
```

```
  1 prism1                        00:02:6F:01:5D:FE client
[admin@MT_Prism_AP] interface prism>
```

There are two possible ways of implementing the wireless access point feature:

- Use it as a pure access point with bridging function enabled between the ethernet and prism interfaces. The IP address can be assigned to the bridge interface.
- Use it as a wireless access point router with routing functionality between the ethernet and prism interfaces. It requires different IP addresses assigned to both the Ethernet and prism interfaces. The addresses should be from different networks as well!

To enable bridging between the ethernet and prism interfaces, do the following:

1. Add bridge interface with the desired forwarded protocols:

```
[admin@MT_Prism_AP] interface bridge> add forward-protocols=ip,arp,other
[admin@MT_Prism_AP] interface bridge> print
Flags: X - disabled, R - running
  0 X  name="bridge1" mtu=1500 arp=enabled mac-address=00:00:00:00:00:00
       forward-protocols=ip,arp,other priority=1

[admin@MT_Prism_AP] interface bridge>
```

2. Add the desired interfaces to the bridge interface:

```
[admin@MT_Prism_AP] interface bridge port> set "ether1,prism1" bridge=bridge1
[admin@MT_Prism_AP] interface bridge port> print
Flags: X - disabled
  #   INTERFACE                           BRIDGE
  0   ether1                              bridge1
  1   prism1                              bridge1
[admin@MT_Prism_AP] interface bridge port>
```

3. Enable the bridge interface:

```
[admin@MT_Prism_AP] interface> print
Flags: X - disabled, D - dynamic, R - running
  #    NAME                TYPE             MTU
  0  R ether1               ether            1500
  1  R prism1               prism            1500
  2 X  bridge1              bridge           1500
[admin@MT_Prism_AP] interface> enable bridge1
[admin@MT_Prism_AP] interface> print
Flags: X - disabled, D - dynamic, R - running
  #    NAME                TYPE             MTU
  0  R ether1               ether            1500
  1  R prism1               prism            1500
  2  R  bridge1              bridge           1500
[admin@MT_Prism_AP] interface>
```

4. Assign an IP address to the bridge interface and specify the default gateway for the access point:

```
[admin@MT_Prism_AP] ip address> add address=10.0.0.250/24 interface=bridge1
[admin@MT_Prism_AP] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.250/24      10.0.0.0        10.0.0.255      bridge1
```

```
[admin@MT_Prism_AP] ip address> .. route add gateway=10.0.0.1
[admin@MT_Prism_AP] ip address> .. route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        bridge1
    1 DC 10.0.0.0/24        r 0.0.0.0         0        bridge1
[admin@MT_Prism_AP] ip address>
```

The client router requires the System Service Identificator set to "mt". The IP addresses assigned to the interfaces should be from networks 10.0.0.0/24 and 192.168.0.0./24:

```
[admin@mikrotik] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.217/24      10.0.0.0        10.0.0.255      aironet
  1   192.168.0.254/24   192.168.0.0     192.168.0.255   Local
[admin@mikrotik] ip address>
```

The default route should be set to gateway 10.0.0.1 for the router [mikrotik]:

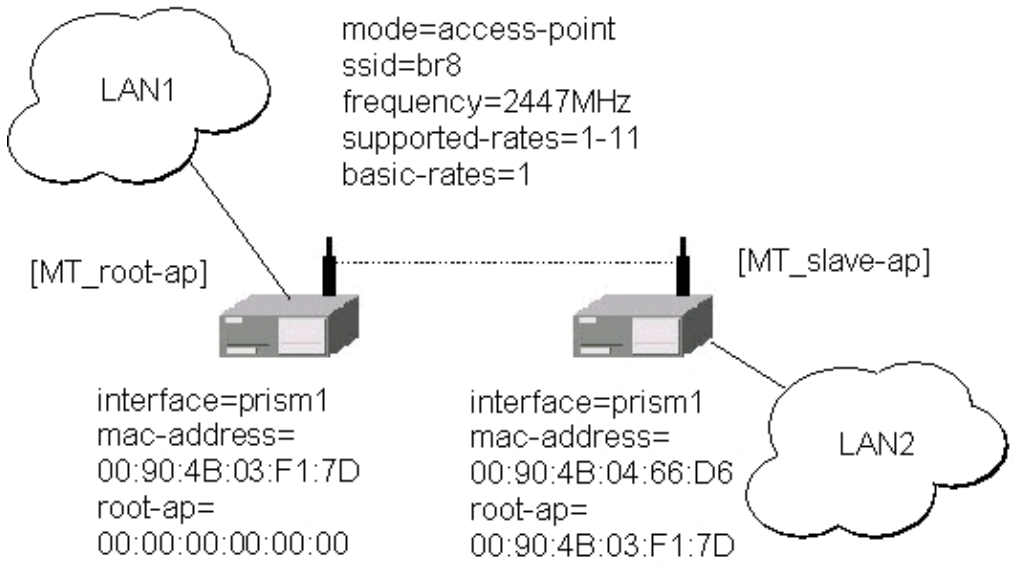
```
[admin@mikrotik] ip route> add gateway=10.0.0.1
[admin@mikrotik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        aironet
    1 DC 10.0.0.0/24        r 0.0.0.0         0        aironet
    2 DC 192.168.0.254/24   r 0.0.0.0         0        Local
[admin@mikrotik] ip route>
```

Wireless Bridge

To set up a wireless bridge between two networks, you need to have a "wireless 2.4GHz" or "AP" license. Configure one MikroTik RouterOS Prism AP to register to another MikroTik RouterOS Prism AP for point−to−point operation.

> ***Important!*** Prism 2.5 200mW cards require firmware 1.3.6 or lower, if you want to bridge networks. Please see the troubleshooting section on how to check the firmware version and downgrade it.

The basic setup is as follows:



Below are step−by−step configurations for both units. The system identities are set to [MT−parent] and [MT−child], respectively.

[MT−parent] Configuration

Assume you have interfaces ether1 and prism1 under **/interface** list.

1. Enable the Ethernet interface ether1:

```
/interface enable ether1
```

2. Configure prism1 interface.
   Set mode=bridge, ssid=br8, frequency=2447MHz, and enable prism1 interface (you can use mode=ap−bridge, if you have Prism AP License):

```
/interface prism set prism1 mode=bridge ssid=br8 frequency=2447 disabled=no
```

3. Add bridge interface and specify forwarded protocol list:

```
/interface bridge add forward−protocols=ip,arp,other disabled=no
```

4. Specify ports prism1 and ether1 that belong to bridge1:

```
/interface bridge port set ether1,prism1 bridge=bridge1
```

5. Assign IP address 10.0.0.217/24 to the bridge1 interface:

```
/ip address add address=10.0.0.217/24 interface=bridge1
```

6. Set default route to 10.0.0.1:

```
/ip route add gw=10.0.0.1
```

[MT−child] Configuration

Assume you have interfaces ether1 and prism1 under **/interface** list.

1. Enable the Ethernet interface ether1:

```
/interface enable ether1
```

2. Configure prism1 interface.
   Here, you have to specify root−ap MAC address, so the Prism radio registers to the root AP. Set mode=bridge, ssid=br8, frequency=2447MHz, root−ap=xx:xx:xx:xx:xx:xx, and enable prism1 interface (you can use mode=ap−bridge, if you have Prism AP License):

```
/interface prism set prism1 mode=bridge ssid=br8 frequency=2447 \
root−ap=xx:xx:xx:xx:xx:xx disabled=no
```

   Here, substitute the xx:xx:xx:xx:xx:xx with MAC address of [MT−parent] prism interface.
3. Check your setup and see, if you have successfully registered to the root AP. Its MAC address should be listed as parent−ap in the registration table of prism interface, for example:

```
[admin@MT−child] interface prism> registration−table print
  # INTERFACE                     MAC−ADDRESS       TYPE      PARENT
  0 prism1                        00:02:6F:01:CE:2A parent−ap
[admin@MT−child] interface prism>
```

4. Add bridge interface and specify forwarded protocol list:

```
/interface bridge add forward−protocols=ip,arp,other disabled=no
```

5. Specify ports prism1 and ether1 that belong to bridge1:

```
/interface bridge port set ether1,prism1 bridge=bridge1
```

6. Assign IP address 10.0.0.218/24 to the bridge1 interface:

```
/ip address add address=10.0.0.218/24 interface=bridge1
```

7. Set default route to 10.0.0.1:

```
/ip route add gw=10.0.0.1
```

Note, that both LANs should use IP addresses from the same network 10.0.0.0/24. Both MikroTik routers belong to the same network too. You should be able to ping through the wireless bridge from one LAN to other and to gateway 10.0.0.1.

RadioLAN 5.8GHz Wireless Interface

Document revision 1.1 (29−Apr−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The MikroTik RouterOS supports the following RadioLAN 5.8GHz Wireless Adapter hardware:

- RadioLAN ISA card (Model 101)
- RadioLAN PCMCIA card

For more information about the RadioLAN adapter hardware please see the relevant User's Guides and Technical Reference Manuals.

Specifications

Packages required : ***radiolan***
License required : ***2.4/5GHz Wireless Client***
Home menu level : ***/interface radiolan***
Protocols utilized : *10BaseRadio*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
Log Management

Installing the Wireless Adapter

These installation instructions apply to non−Plug−and−Play ISA cards. If You have a Plug−and−Play compliant system AND **PnP OS Installed** option in system BIOS is set to **Yes** AND you have a Plug−and−Play compliant ISA or PCI card (using PCMCIA or CardBus card with Plug−and−Play compliant adapter), the driver should be loaded automatically. If it is not, these instructions may also apply to your system

The basic installation steps of the wireless adapter should be as follows:

1. Check the system BIOS settings for peripheral devices, like, Parallel or Serial communication ports. Disable them, if you plan to use IRQ's assigned to them by the BIOS.
2. Use the RLProg.exe to set the IRQ and Base Port address of the RadioLAN ISA card (Model 101). RLProg must not be run from a DOS window. Use a separate computer or a bootable floppy to run the RLProg utility and set the hardware parameters. The factory default values of I/O 0x300 and IRQ 10 might conflict with other devices.

Please note, that not all combinations of I/O base addresses and IRQ's may work on your motherboard. As it has been observed, the IRQ 5 and I/O 0x300 work in most cases.

Wireless Interface Configuration

Submenu level : **/interface radiolan**

Description

To set the wireless interface for working with another wireless card in a point−to−point link, you should set the following parameters:

- The **Service Set Identifier**. It should match the sid of the other card.
- The **Distance** should be set to that of the link. For example, if you have 6km link, use distance 4.7km−6.6km.

All other parameters can be left as default. You can monitor the list of neighbours having the same sid and being within the radio range.

Property Description

name (*name*; default: **radiolan*N***) − assigned interface name
mtu (*integer*; default: **1500**) − Maximum Transmit Unit
mac−address (read−only: *MAC address*) − MAC address
distance (0−150m | 10.2km−13.0km | 2.0km−2.9km | 4.7km−6.6km | 1.1km−2.0km | 150m−1.1km | 2.9km−4.7km | 6.6km−10.2km; default: **0−150m**) − distance setting for the link
rx−diversity (enabled | disabled; default: **disabled**) − receive diversity
tx−diversity (enabled | disabled; default: **disabled**) − transmit diversity
default−destination (ap | as−specified | first−ap | first−client | no−destination; default: **first−client**) − default destination. It sets the destination where to send the packet if it is not for a client in the radio network.
default−address (*MAC address*; default: **00:00:00:00:00:00**) − MAC address of a host in the radio network where to send the packet, if it is for none of the radio clients.
max−retries (*integer*; default: **1500**) − maximum retries before dropping the packet

sid (*text*) − Service Identifier
card−name (*text*) − card name
arp (disabled | enabled | proxy−arp | reply−only; default: **enabled**) − Address Resolution Protocol, one of the:

- **disabled** − the interface will not use ARP protocol
- **enabled** − the interface will use ARP protocol
- **proxy−arp** − the interface will be an ARP proxy (see corresponding manual)
- **reply−only** − the interface will only reply to the requests originated to its own IP addresses, but neighbour MAC addresses will be gathered from **/ip arp** statically set table only.

Example

```
[admin@MikroTik] interface radiolan> print
Flags: X − disabled, R − running
  0  R name="radiolan1" mtu=1500 mac−address=00:A0:D4:20:4B:E7 arp=enabled
       card−name="00A0D4204BE7" sid="bbbb" default−destination=first−client
       default−address=00:00:00:00:00:00 distance=0−150m max−retries=15
       tx−diversity=disabled rx−diversity=disabled


[admin@MikroTik] interface radiolan>
```

You can monitor the status of the wireless interface:

```
[admin@MikroTik] interface radiolan> monitor radiolan1
    default: 00:00:00:00:00:00
      valid: no

[admin@MikroTik] interface radiolan>
```

Here, the wireless interface card has not found any neighbour.

```
[admin@MikroTik] interface radiolan> set 0 sid ba72 distance 4.7km−6.6km
[admin@MikroTik] interface radiolan> print
Flags: X − disabled, R − running
  0  R name="radiolan1" mtu=1500 mac−address=00:A0:D4:20:4B:E7 arp=enabled
       card−name="00A0D4204BE7" sid="ba72" default−destination=first−client
       default−address=00:00:00:00:00:00 distance=4.7km−6.6km max−retries=15
       tx−diversity=disabled rx−diversity=disabled

[admin@MikroTik] interface radiolan> monitor 0
    default: 00:A0:D4:20:3B:7F
      valid: yes

[admin@MikroTik] interface radiolan>
```

Now we'll monitor other cards with the same **sid** within range:

```
[admin@MikroTik] interface radiolan> neighbor radiolan1 print
Flags: A − access−point, R − registered, U − registered−to−us,
D − our−default−destination
      NAME                ADDRESS           ACCESS−POINT
    D 00A0D4203B7F        00:A0:D4:20:3B:7F
[admin@MikroTik] interface radiolan>
```

You can test the link by pinging the neighbour by its MAC address:

```
[admin@MikroTik] interface radiolan> ping 00:a0:d4:20:3b:7f radiolan1 \
\... size=1500 count=50
                 sent: 1
    successfully-sent: 1
          max-retries: 0
      average-retries: 0
          min-retries: 0

                 sent: 11
    successfully-sent: 11
          max-retries: 0
      average-retries: 0
          min-retries: 0

                 sent: 21
    successfully-sent: 21
          max-retries: 0
      average-retries: 0
          min-retries: 0

                 sent: 31
    successfully-sent: 31
          max-retries: 0
      average-retries: 0
          min-retries: 0

                 sent: 41
    successfully-sent: 41
          max-retries: 0
      average-retries: 0
          min-retries: 0

                 sent: 50
    successfully-sent: 50
          max-retries: 0
      average-retries: 0
          min-retries: 0

[admin@MikroTik] interface radiolan>
```

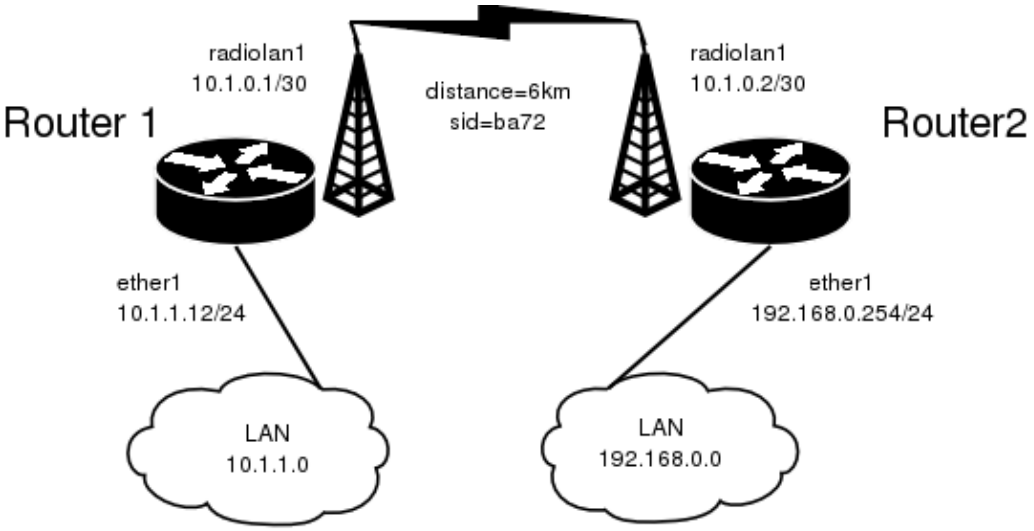
Wireless Troubleshooting

- *The radiolan interface does not show up under the interfaces list*
  Obtain the required license for RadioLAN 5.8GHz wireless feature.
- *The wireless card does not obtain the MAC address of the default destination*
  Check the cabling and antenna alignment.

Wireless Network Applications

Point-to-Point Setup with Routing

Let us consider the following network setup:



The minimum configuration required for the RadioLAN interfaces of both routers is:

1. Setting the Service Set Identifier (up to alphanumeric characters). In our case we use ssid "ba72".
2. Setting the distance parameter, in our case we have 6km link.

The IP addresses assigned to the wireless interface of Router#1 should be from the network 10.1.0.0/30, e.g.:

```
[admin@MikroTik] ip address> add address=10.1.0.1/30 interface=radiolan1
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.1.1.12/24       10.1.1.0        10.1.1.255      ether1
  1   10.1.0.1/30        10.1.0.0        10.1.0.3        radiolan1
[admin@MikroTik] ip address>
```

The default route should be set to the gateway router 10.1.1.254. A static route should be added for the network 192.168.0.0/24:

```
[admin@MikroTik] ip route> add gateway=10.1.1.254
comment  copy-from  disabled  distance  dst-address  netmask  preferred-source
[admin@MikroTik] ip route> add gateway=10.1.1.254 preferred-source=10.1.0.1
[admin@MikroTik] ip route> add dst-address=192.168.0.0/24 gateway=10.1.0.2 \
\... preferred-source=10.1.0.1
[admin@MikroTik] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          u 10.1.1.254      1        radiolan1
    1  S 192.168.0.0/24     r 10.1.0.2        1        radiolan1
    2 DC 10.1.0.0/30        r 0.0.0.0         0        radiolan1
    3 DC 10.1.1.0/24        r 0.0.0.0         0        ether1
[admin@MikroTik] ip route>
```

The Router#2 should have addresses 10.1.0.2/30 and 192.168.0.254/24 assigned to the radiolan and Ethernet interfaces respectively. The default route should be set to 10.1.0.1

Virtual LAN (VLAN) Interface

Document revision 1.3 (06−Mar−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

VLAN is an implementation of the 802.1Q VLAN protocol for MikroTik RouterOS 2.7. It allows you to have multiple Virtual LANs on a single ethernet cable, giving the ability to segregate LANs efficiently. It supports up to 4094 vlan interfaces per ethernet device. Many routers, including Cisco and Linux based, and many Layer 2 switches also support it.

A VLAN is a logical grouping that allows end users to communicate as if they were physically connected to a single isolated LAN, independent of the physical configuration of the network. VLAN support adds a new dimension of security and cost savings permitting the sharing of a physical network while logically maintaining separation among unrelated users.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/interface vlan***
Protocols utilized : *VLAN (IEEE802.1Q)*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)

Description

VLANs are simply a way of grouping a set of switch ports together so that they form a logical network, separate from any other such group. Within a single switch this is straightforward local configuration. When the VLAN extends over more than one switch, the inter−switch links have to become trunks, on which packets are tagged to indicate which VLAN they belong to.

You can use MikroTik RouterOS (as well as Cisco IOS and Linux) to mark these packets as well as to accept and route marked ones.

As VLAN works on OSI Layer 2, it can be used just as any other network interface without any restrictions. And VLAN successfully passes through ethernet bridges (for MikroTik RouterOS bridges you should set **forward−protocols** to **ip**, **arp** and **other**; for other bridges there should be analogical settings)

VLAN Setup

Submenu level : /**interface vlan**

Property Description

name (*name*)− Interface name for reference
mtu (*integer*; default:**1500**)− Maximum Transmit Unit
interface (*name*) − physical interface to the network where are VLANs
arp (disabled | enabled | proxy−arp | replay−only; default:**enabled**) − Address Resolution Protocol:

- **disabled** − the interface will not use ARP protocol
- **enabled** − the interface will use ARP protocol
- **proxy−arp** − the interface will be an ARP proxy
- **reply−only** − the interface will only reply to the requests originated to its own IP addresses, but neighbour MAC addresses will be gathered from **/ip arp** statically set table only

vlan−id (*integer*; default:**1**) − Virtual LAN identificator or tag that is used to distinguish VLANs. Must be equal for all computers in one VLAN

Notes

MTU should be set to 1500 bytes as on Ethernet interfaces. But this may not work with some Ethernet cards that do not support receiving/transmitting of full size ethernet packets with VLAN header added (1500 bytes data + 4 bytes VLAN header + 14 bytes ethernet header). In this situation MTU 1496 can be used, but note that this will cause packet fragmentation if larger packets have to be sent over interface. At the same time remember that MTU 1496 may cause problems if path MTU discovery is not working properly between source and destination.

Example

To add and enable a VLAN interface names **test** with VLAN ID **1** on interface **ether1**:

```
[admin@MikroTik] interface vlan> add name=test vlan−id=1 interface=ether1
[admin@MikroTik] interface vlan> print
Flags: X − disabled, R − running
```

```
  #    NAME                MTU  ARP       VLAN-ID INTERFACE
  0 X  test                1500 enabled   1       ether1
[admin@MikroTik] interface vlan> enable 0
[admin@MikroTik] interface vlan> print
Flags: X - disabled, R - running
  #    NAME                MTU  ARP       VLAN-ID INTERFACE
  0  R test                1500 enabled   1       ether1
[admin@MikroTik] interface vlan>
```

Application Example

Lets assume that we have two or more MikroTik RouterOS routers connected with a hub. Interfaces to the physical network, where VLAN is to be created is **ether1** for all of them (it is needed only for example simplification, it is NOT a must)

To connect computers through VLAN they must be connected physically and unique IP addresses should be assigned them so that they could **ping** each other. Then on each of them the VLAN interface should be created:

```
[admin@MikroTik] interface vlan> add name=test vlan-id=32 interface=ether1
[admin@MikroTik] interface vlan> print
Flags: X - disabled, R - running
  #    NAME                MTU  ARP       VLAN-ID INTERFACE
  0  R test                1500 enabled   32      ether1
[admin@MikroTik] interface vlan>
```

If the interfaces were successfully created, both of them will be **running**. If computers are connected incorrectly (through network device that does not retransmit or forward VLAN packets), either both or one of the interfaces will not be **running**.

When the interface is running, IP addresses can be assigned to the VLAN interfaces.

On the Router 1:

```
[admin@MikroTik] ip address> add address=10.10.10.1/24 interface=test
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.204/24      10.0.0.0        10.0.0.255      ether1
  1   10.20.0.1/24       10.20.0.0       10.20.0.255     pc1
  2   10.10.10.1/24      10.10.10.0      10.10.10.255    test
[admin@MikroTik] ip address>
```

On the Router 2:

```
[admin@MikroTik] ip address> add address=10.10.10.2/24 interface=test
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.201/24      10.0.0.0        10.0.0.255      ether1
  1   10.10.10.2/24      10.10.10.0      10.10.10.255    test
[admin@MikroTik] ip address>
```

If it set up correctly, then it is possible to **ping** Router 2 from Router 1 and vice versa:

```
[admin@MikroTik] ip address> /ping 10.10.10.1
10.10.10.1 64 byte pong: ttl=255 time=3 ms
10.10.10.1 64 byte pong: ttl=255 time=4 ms
10.10.10.1 64 byte pong: ttl=255 time=10 ms
10.10.10.1 64 byte pong: ttl=255 time=5 ms
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 3/10.5/10 ms
[admin@MikroTik] ip address> /ping 10.10.10.2
10.10.10.2 64 byte pong: ttl=255 time=10 ms
10.10.10.2 64 byte pong: ttl=255 time=11 ms
10.10.10.2 64 byte pong: ttl=255 time=10 ms
10.10.10.2 64 byte pong: ttl=255 time=13 ms
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 10/11/13 ms
[admin@MikroTik] ip address>
```

Additional Resources

Links for VLAN documentation:

http://www.csd.uwo.ca/courses/CS457a/reports/handin/jpbojtos/A2/trunking.htm
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t3/dtbridge.htm#xtocid114533
http://www.cisco.com/warp/public/473/27.html#tagging
http://www.cisco.com/warp/public/538/7.html
http://www.nwfusion.com/news/tech/2001/0305tech.html
http://www.intel.com/network/connectivity/resources/doc_library/tech_brief/virtual_lans.htm

Currently Supported Interfaces

This is a list of network interfaces on which VLAN was tested and worked:

- Realtek 8139
- Intel PRO/100
- Intel PRO1000 server adapter

This is a list of network interfaces on which VLAN was tested and worked, but **WITHOUT LARGE PACKET (>1496 bytes) SUPPORT**:

- 3Com 3c59x PCI
- DEC 21140 (tulip)

Xpeed SDSL (Single−line Digital Subscriber Line) Interface

Document revision 1.4 (09−Apr−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The MikroTik RouterOS supports the Xpeed 300 SDSL PCI Adapter hardware with speeds up to 2.32Mbps. This device can operate either using Frame Relay or PPP type of connection. SDSL (Single−line Digital Subscriber Line or Symmetric Digital Subscriber Line) stands for the type of DSL that uses only one of the two cable pairs for transmission. SDSL allows residential or small office users to share the same telephone for data transmission and voice or fax telephony.

Specifications

Packages required : *synchronous*
License required : *synchronous*
Home menu level : ***/interface xpeed***
Protocols utilized : *PPP (RFC1661), Frame Relay (RFC1490)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
Device Driver Management
IP Addresses and Address Resolution Protocol (ARP)
FrameRelay (PVC, Private Virtual Circuit) Interface

Xpeed Interface Configuration

Submenu level : **/interface xpeed**

Property Description

name (*name*) − interface name
mtu (*integer*; default: **1500**) − Maximum Transmit Unit
mac−address (*MAC address*) − MAC address of the card
arp (disabled | enabled | proxy−arp | reply−only; default: **enabled**) − Address Resolution Protocol settings, one of the following:

- **disabled** − the interface will not use ARP protocol
- **enabled** − the interface will use ARP protocol
- **proxy−arp** − the interface will be an ARP proxy (see corresponding manual)
- **reply−only** − the interface will only reply to the requests originated to its own IP addresses, but neighbour MAC addresses will be gathered from **/ip arp** statically set table only

mode (network−termination | line termination; default: **line−termination**) − interface mode, either line termination (LT) or network termination (NT)
sdsl−speed (*integer*; default: 2320) − SDSL connection speed
sdsl−invert (yes | no; default: **no**) − whether the clock is phase inverted with respect to the Transmitted Data interchange circuit. This configuration option is useful when long cable lengths between the Termination Unit and the DTE are causing data errors
sdsl−swap (yes | no; default: **no**) − whether or not the Xpeed 300 SDSL Adapter performs bit swapping. Bit swapping can maximize error performance by attempting to maintain an acceptable margin for each bin by equalizing the margin across all bins through bit reallocation
bridged−ethernet (yes | no; default: **yes**) − if the adapter operates in briged ethernet mode
dlci (*integer*; default: **16**) − defines the DLCI to be used for the local interface. The DLCI field identifies which logical circuit the data travels over
lmi−mode (*mode*; default: **off**) − defines how the card will perform LMI protocol negotiation:

- **off** − no LMI will be used
- **line−termination** − LMI will operate in LT (Line Termination) mode
- **network−termination** − LMI will operate in NT (Network Termination) mode
- **network−termination−bidirectional** − LMI will operation in bidirectional NT mode

cr (0 | 2; default: **0**) − a special mask value to be used when speaking with certain buggy vendor equipment. Can be 0 or 2

Example

To enable interface:

```
[admin@r1] interface> print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                                                TYPE            MTU
  0  R outer                                               ether           1500
  1  R inner                                               ether           1500
  2 X  xpeed1                                              xpeed           1500

[admin@r1] interface> enable 2
[admin@r1] interface> print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                                                TYPE            MTU
  0  R outer                                               ether           1500
```

```
  1  R inner                                                ether            1500
  2  R xpeed1                                               xpeed            1500

[admin@r1] interface>
```

Frame Relay Configuration Examples

MikroTik Router to MikroTik Router

Consider the following network setup with MikroTik router connected via SDSL line using Xpeed interface to another MikroTik router with Xpeed 300 SDSL adapter. *SDSL line* can refer a common patch cable included with the Xpeed 300 SDSL adapter (such a connection is called Back−to−Back). Lets name the first router r1 and the second r2.

Router r1 setup

The following setup is identical to one in first example:

```
[admin@r1] ip address> add inter=xpeed1 address 1.1.1.1/24
[admin@r1] ip address> pri
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   1.1.1.1/24         1.1.1.0         1.1.1.255       xpeed1

[admin@r1] interface xpeed> print
Flags: X − disabled
  0   name="xpeed1" mtu=1500 mac−address=00:05:7A:00:00:08 arp=enabled
      mode=network−termination sdsl−speed=2320 sdsl−invert=no sdsl−swap=no
      bridged−ethernet=yes dlci=16 lmi−mode=off cr=0
[admin@r1] interface xpeed>
```

Router r2 setup

First, we need to add a suitable IP address.

```
[admin@r2] ip address> add inter=xpeed1 address 1.1.1.2/24
[admin@r2] ip address> pri
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   1.1.1.2/24         1.1.1.0         1.1.1.255       xpeed1
```

Then, some changes in **xpeed** interface configuration shoud to be done

```
[admin@r2] interface xpeed> print
Flags: X − disabled
  0   name="xpeed1" mtu=1500 mac−address=00:05:7A:00:00:08 arp=enabled
      mode=network−termination sdsl−speed=2320 sdsl−invert=no sdsl−swap=no
      bridged−ethernet=yes dlci=16 lmi−mode=off cr=0
[admin@r2] interface xpeed> set 0 mode=line−termination
[admin@r2] interface xpeed>
```

Now r1 and r2 can ping each other.

MikroTik Router to CISCO Router

Let us consider the following network setup with MikroTik Router with Xpeed interface connected to a leased line with a CISCO router at the other end.

MikroTik router setup

```
[admin@r1] ip address> add inter=xpeed1 address 1.1.1.1/24
[admin@r1] ip address> pri
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   1.1.1.1/24         1.1.1.0         1.1.1.255       xpeed1

[admin@r1] interface xpeed> print
Flags: X − disabled
  0   name="xpeed1" mtu=1500 mac−address=00:05:7A:00:00:08 arp=enabled
      mode=network−termination sdsl−speed=2320 sdsl−invert=no sdsl−swap=no
      bridged−ethernet=yes dlci=42 lmi−mode=off cr=0
[admin@r1] interface xpeed>
```

CISCO router setup

```
CISCO# show running−config
Building configuration...
Current configuration...

...
!
ip subnet−zero
no ip domain−lookup
frame−relay switching
!
interface Ethernet0
 description connected to EthernetLAN
 ip address 10.0.0.254 255.255.255.0
!
interface Serial0
 description connected to Internet
 no ip address
 encapsulation frame−relay IETF
 serial restart−delay 1
 frame−relay lmi−type ansi
 frame−relay intf−type dce
!
interface Serial0.1 point−to−point
 ip address 1.1.1.2 255.255.255.0
 no arp frame−relay
 frame−relay interface−dlci 42
!
...
end.

Send ping to MikroTik router

CISCO#ping 1.1.1.1
Type escape sequence to abort.
Sending 5, 100−byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round−trip min/avg/max = 28/31/32 ms
```

```
CISCO#
```

Troubleshooting

- I tried to connect two routers as shown in MT−to−MT, but nothing happens.
  The *link* indicators on both cards must be on. If it's not, check the cable or interface configuration. One adapter should use LT mode and the other NT mode. You can also change **sdsl−swap** and **sdsl−invert** parameters on the router running LT mode if you have a very long line.

Additional Resources

Xpeed homepage

WaveLAN/ORiNOCO 2.4GHz 11Mbps Wireless Interface

Document revision 1.0 (19−May−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The MikroTik RouterOS supports the following WaveLAN/ORiNOCO 2.4GHz 11Mbps Wireless Adapter hardware:

- ORiNOCO 2.4GHz 11Mbps PC Card (Silver/Gold), firmware versions 4.xx...7.52.
- ORiNOCO ISA and PCI adapters for using the PC card in desktop computers.

Specifications

Packages required : ***wireless***
License required : *2.4/5GHz Wireless Client*
Home menu level : ***/interface wavelan***
Standards and Technologies : *IEEE802.11b (IEEE802.11b)*
Hardware usage : *not significant*

Wireless Interface Configuration

Submenu level : **/interface wavelan**

Description

WaveLAN/ORiNOCO 2.4GHz 11Mbps Wireless cards operate in 2.4 GHz band providing connection speed up to 11 Mbit/s.

Property Description

name (*name*; default: **wavelan*N***) − assigned interface name
mtu (*integer: 256..2296*; default: **1500**) − Maximum Transmit Unit
mac−address (*read−only: MAC address*) − MAC address of the card
frequency (2412MHz | 2422MHz | 2432MHz | 2442MHz | 2452MHz | 2462MHz | 2472MHz | 2417MHz | 2427MHz | 2437MHz | 2447MHz | 2457MHz | 2467MHz | 2484MHz; default: **2412MHz**) − channel frequency
data−rate (11Mbit/s | 1Mbit/s | 2Mbit/s | 5.5Mbit/s | auto; default: **11Mbit/s**) − data rate
mode (infrastructure | ad−hoc; default: **ad−hoc**) − operation mode of the card
ssid (*text: 0..32 chars*; default: "") − Service Set Identifier
client−name (*text: 0..32 chars*; default: "") − client name
key1 (*text*; default: "") − encryption key #1
key2 (*text*; default: "") − encryption key #2
key3 (*text*; default: "") − encryption key #3
key4 (*text*; default: "") − encryption key #4
tx−key (key1 | key2 | key3 | key4; default: **key1**) − transmit key
encryption (no | yes; default: **no**) − specifies whether to use the encryption
arp (enabled | disabled | reply−only | proxy−arp; default: **enabled**) − ARP setting

Example

```
[admin@MikroTik] interface> print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE            MTU
  0  R Public              1500  ether
  1  R Local               1500  ether
  2 X  wavelan1            1500  wavelan
[MikroTik] interface> enable 2
[admin@MikroTik] interface> print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE            MTU
  0  R Public              1500  ether
  1  R Local               1500  ether
  2  R wavelan1            1500  wavelan
[admin@MikroTik] interface>
```

More configuration and statistics parameters can be found under the **/interface wavelan** menu:

```
[admin@MikroTik] interface> wavelan
[admin@MikroTik] interface wavelan> print
Flags: X − disabled, R − running
  0  R name=wavelan1 mtu=1500 mac−address=00:02:2D:07:D8:44 arp=enabled
       frequency=2412MHz data−rate=11Mbit/s mode=ad−hoc ssid="" client−name=""
       key1="" key2="" key3="" key4="" tx−key=key1 encryption=no

[admin@MikroTik] interface wavelan>
```

You can monitor the status of the wireless interface:

```
[admin@MikroTik] interface wavelan> moitor 0
             bssid: 44:44:44:44:44:44
         frequency: 2422MHz
         data-rate: 11Mbit/s
              ssid: tsunami
    signal-quality: 0
      signal-level: 0
             noise: 0

[admin@MikroTik] interface wavelan>
```

To set the wireless interface for working with an IEEE 802.11b access point (register to the AP), you should set the following parameters:

- The **Service Set Identifier**. It should match the ssid of the AP.
- The **Operation Mode** of the card should be set to **infrastructure**.
- The **Data Rate** of the card should match one of the supported data rates of the AP. Data rate **auto** should work for most of the cases.

All other parameters can be left as default. To configure the wireless interface for registering to an AP with ssid "MT_w_AP", it is enough to change the argument value of ssid to "MT_w_AP":

```
[admin@MikroTik] interface wavelan> set 0 ssid MT_w_AP mode infrastructure
[admin@MikroTik] interface wavelan> monitor wavelan1
             bssid: 00:40:96:42:0C:9C
         frequency: 2437MHz
         data-rate: 11Mbit/s
              ssid: MT_w_AP
    signal-quality: 65
      signal-level: 228
             noise: 163

[admin@MikroTik] interface wavelan>
```

Wireless Troubleshooting

- *The wavelan interface does not show up under the interfaces list*
  Obtain the required license for 2.4GHz wireless feature.
- *The wireless card does not register to the AP*
  Check the cabling and antenna alignment.
- *I get the wireless interface working and registering to the AP, but there is no data transmitted, I cannot ping the AP*
  There is an IRQ conflict. You can try to use different motherboard or PCMCIA adapter.

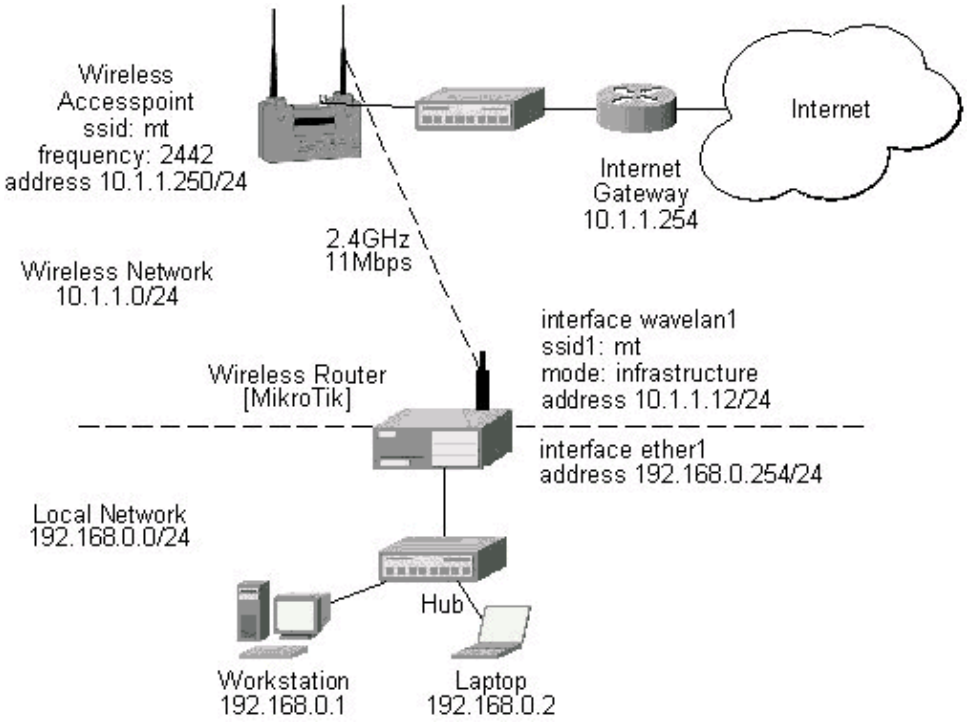
Application Example

Two possible wireless network configurations are discussed in the following examples:

- Point−to−Multipoint (Wireless Infrastructure)
- Point−to−Point with MikroTik Client (Peer−to−Peer, or Ad−Hoc Wireless LAN)
- Point−to−Point with Windows Client (Peer−to−Peer, or Ad−Hoc Wireless LAN)

Point−to−Multipoint Wireless LAN

Let us consider the following network setup with WaveLAN / ORiNOCO or CISCO/Aironet Wireless Access Point or MikroTik router configured as Access Point as a base station and MikroTik Wireless Router as a client:



The access point is connected to the wired network's HUB and has IP address from the network 10.1.1.0/24. The minimum configuration required for the AP is:

1. Setting the Service Set Identifier (up to 32 alphanumeric characters). In our case we use ssid "mt".
2. Setting the allowed data rates at 1−11Mbps, and the basic rate at 1Mbps.
3. Choosing the frequency, in our case we use 2452MHz.
4. Setting the identity parameters: ip address/mask and gateway. These are required if you want to access the AP remotely.

The minimum configuration for the MikroTik router's wavelan wireless interface is:

1. Setting the Service Set Identifier to that of the AP, i.e., "mt"
2. Setting the Operation Mode to **infrastructure**

```
[admin@MikroTik] interface wavelan> set wavelan1 ssid mt mode infrastructure
[admin@MikroTik] interface wavelan>
             bssid: 00:40:96:42:0C:9C
         frequency: 2437MHz
         data-rate: 11Mbit/s
              ssid: mt
    signal-quality: 64
      signal-level: 228
             noise: 163

[admin@MikroTik] interface wavelan>
```

The channel frequency argument does not have any meaning, since the frequency of the AP is used.

IP Network Configuration

The IP addresses assigned to the wireless interface should be from the network 10.1.1.0/24, e.g.:

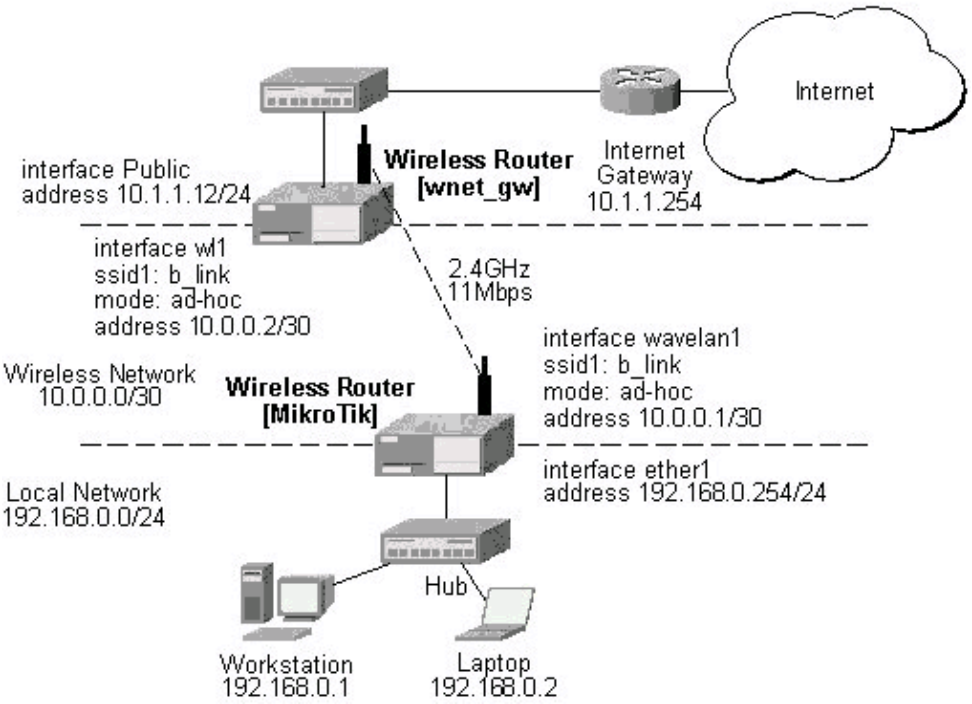
```
[admin@MikroTik] ip address> add address 10.1.1.12/24 interface wavelan1
[admin@MikroTik] ip address> add address 192.168.0.254/24 interface ether1
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   192.168.0.254/24   192.168.0.0     192.168.0.255   ether1
  1   10.1.1.12/24       10.1.1.0        10.1.1.255      wavelan1
[admin@MikroTik] ip address>
```

The default route should be set to the gateway router 10.1.1.254 (not the AP 10.1.1.250 !):

```
[admin@MikroTik] ip route> add gateway 10.1.1.254
[admin@MikroTik] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.1.1.254      1        wavelan1
    1 DC 192.168.0.0/24     r 0.0.0.0         0        ether1
    2 DC 10.1.1.0/24        r 0.0.0.0         0        wavelan1
[admin@MikroTik] ip route>
```

Point-to-Point Wireless LAN

Let us consider the following point-to-point wireless network setup with two MikroTik Wireless Routers:



To establish a point−to−point link, the configuration of the wireless interface should be as follows:

- A unique Service Set Identificator should be chosen for both ends, say "b_link"
- A channel frequency should be selected for the link, say 2412MHz
- The operation mode should be set to **ad−hoc**

The following command should be issued to change the settings for the wavelan interface:

```
[admin@MikroTik] interface wavelan> set 0 ssid b_link mode ad−hoc frewency 2412MHz
[admin@MikroTik] interface wavelan> monitor wavelan1
             bssid: 00:02:2D:07:17:23
         frequency: 2412MHz
         data−rate: 11Mbit/s
              ssid: b_link
    signal−quality: 0
      signal−level: 154
             noise: 154
[admin@MikroTik] interface wavelan>
```

The other router of the point−to−point link requires the same parameters to be set:

```
[admin@wnet_gw] interface wavelan> set 0 ssid b_link mode ad−hoc frequency 2412MHz
[admin@wnet_gw] interface wavelan> enable 0
[admin@wnet_gw] interface wavelan> monitor 0
             bssid: 00:02:2D:07:17:23
         frequency: 2412MHz
         data−rate: 11Mbit/s
              ssid: b_link
    signal−quality: 0
      signal−level: 154
             noise: 154
[admin@wnet_gw] interface wavelan>
```

As we see, the MAC address under the 'bssid' parameter is the same as generated on the first router.

IP Network Configuration

If desired, IP addresses can be assigned to the wireless interfaces of the pint−to−point link routers using a smaller subnet, say 30−bit one:

```
[admin@MikroTik] ip address> add address 10.0.0.1/30 interface wavelan1
[admin@MikroTik] ip address> add address 192.168.0.254/24 interface ether1
[admin@MikroTik] ip address> print
  # ADDRESS         NETMASK         NETWORK         BROADCAST       INTERFACE
  0 10.0.0.1        255.255.255.252 10.0.0.1        10.0.0.3        wavelan1
  1 192.168.0.254   255.255.255.0   192.168.0.254   192.168.0.255   ether1
[admin@MikroTik] ip address> /ip route add gateway 10.0.0.2
[admin@MikroTik] > ip route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.2        1        wavelan1
    1 DC 10.0.0.0/30        r 0.0.0.0         0        wavelan1
    2 DC 192.168.0.0/24     r 0.0.0.0         0        ether1
[admin@MikroTik] ip address>
```

The second router will have address 10.0.0.2, the default route to 10.1.1.254, and a static route for network 192.168.0.0/24 to 10.0.0.1:

```
[admin@wnet_gw] ip address> add address 10.0.0.2/30 interface wl1
[admin@wnet_gw] ip address> add address 10.1.1.12/24 interface Public
[admin@wnet_gw] ip address> print
  # ADDRESS         NETMASK         NETWORK         BROADCAST       INTERFACE
  0 10.0.0.2        255.255.255.252 10.0.0.2        10.0.0.3        wl1
  1 10.1.1.12       255.255.255.0   10.1.1.12       10.1.1.255      Public
[admin@wnet_gw] ip address> /ip route
[admin@wnet_gw] ip route> add gateway 10.1.1.254 interface Public
[admin@wnet_gw] ip route> add gateway 10.0.0.1 interface wl1 \
\... dst-address 192.168.0.0/24
[admin@MikroTik] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0 0.0.0.0/0             r 10.1.1.254      1        Public
    1 192.168.0.0/24        r 10.0.0.1        1        wl1
    2 10.0.0.0/30           r 0.0.0.0         0        wl1
    3 10.1.1.0/24           r 0.0.0.0         0        Public
[admin@wnet_gw] ip route>
```

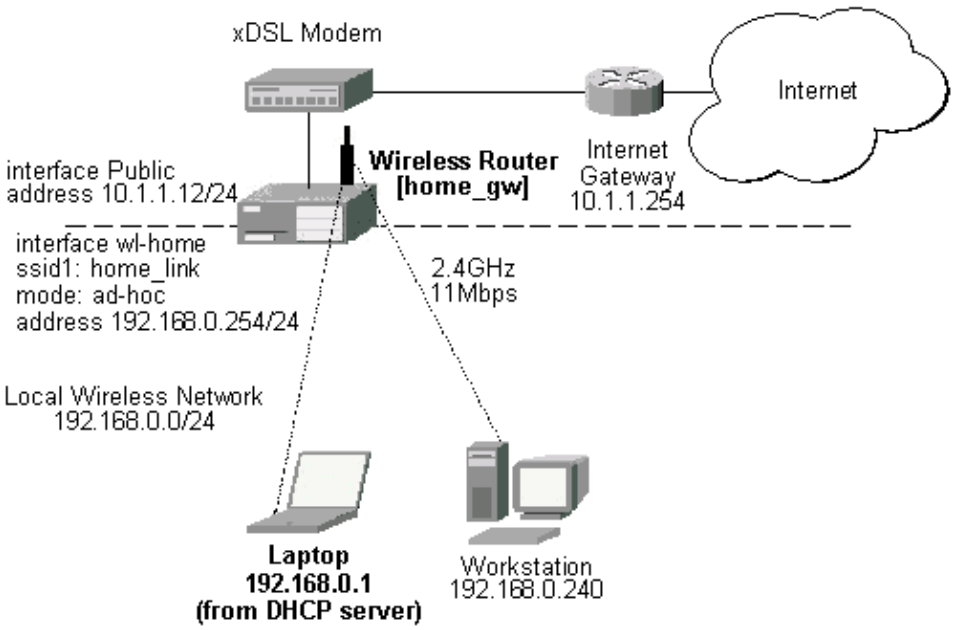
Testing the Network Connectivity

The network connectivity can be tested by using ping:

```
[admin@MikroTik]> ping 10.0.0.2
10.0.0.2 pong: ttl=255 time=2 ms
10.0.0.2 pong: ttl=255 time=2 ms
10.0.0.2 pong: ttl=255 time=2 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 2/2.0/2 ms
[admin@MikroTik]>
```

Point-to-Point Wireless LAN with Windows Client

Let us consider the following point-to-point wireless network setup with one MikroTik Wireless Router and a laptop computer with Wavelan card:



It is very important, that the MikroTik Router is configured prior turning on and configuring the wireless client. The MikroTik router should be up and running, so the client could join its network.

The configuration of the wireless interface of the MikroTik Router should be as follows:

- A unique Service Set Identificator should be chosen, say "home_link"
- A channel frequency should be selected for the link, say 2447MHz
- The operation mode should be set to **ad−hoc**

The following command should be issued to change the settings for the wavelan interface:

```
[admin@home_gw] interface wavelan> set wl−home frequency 2447MHz \
/... mode ad−hoc ssid home_link
[admin@home_gw] interface wavelan> enable wl−home
[admin@home_gw] interface wavelan> print
[admin@MikroTik] interface wavelan> print
Flags: X − disabled, R − running
  0  R name=wl−home mtu=1500 mac−address=00:02:2D:07:D8:44 arp=enabled
       frequency=2447MHz data−rate=11Mbit/s mode=ad−hoc ssid="home_link"
       client−name="" key1="" key2="" key3="" key4="" tx−key=key1 encryption=no

[admin@home_gw] interface wavelan> monitor 0
             bssid: 02:02:2D:07:D8:44
         frequency: 2447MHz
         data−rate: 11Mbit/s
              ssid: home_link
    signal−quality: 0
      signal−level: 154
             noise: 154
[admin@home_gw] interface wavelan>
```

Configure the laptop computer with the Wavelan card following the manufacturer's instructions.

> ***Note!*** In Ad−Hoc (Peer−to−Peer) mode the V1.76 ORiNOCO Client Manager program allows setting only the Network Name (ssid) parameter. The channel (frequency) parameter is

chosen that of the other peer. Therefore, the MikroTik Router should be configured for the ad−hoc mode operation prior turning on the laptop Wavelan client.

If the laptop Wavelan client has established the wireless link with the MikroTik router, it should report the same parameters as set on the MikroTik router's wavelan interface:

Here, we see the channel #8, which has 2447MHz frequency.

IP Network Configuration

The IP addresses assigned to the wireless interface of the MikroTik Router should be from the network 192.168.0.0/24:

```
[admin@home_gw] ip address> add interface Public address 10.1.1.12/24
[admin@home_gw] ip address> add interface wl−home address 192.168.0.254/24
[admin@home_gw] ip address> print
  # ADDRESS          NETMASK          NETWORK          BROADCAST        INTERFACE
  0 10.1.1.12        255.255.255.0    10.1.1.12        10.1.1.255       Public
  1 192.168.0.254    255.255.255.0    192.168.0.254    192.168.0.255    wl−home
[admin@home_gw] ip address> /ip route
[admin@home_gw] ip route> add gateway 10.1.1.254
[admin#home_gw] ip route> print
[admin@MikroTik] > ip route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.1.1.254      1        Public
    1 DC 192.168.0.0/24     r 0.0.0.0         0        wl−home
    2 DC 10.1.1.0/24        r 0.0.0.0         0        Public
[admin@MikroTik] ip route>
```

Testing the Network Connectivity

Use the ping command to test the connectivity from the router:

```
[admin@home_gw] > ping 192.168.0.1
192.168.0.1 pong: ttl=32 time=3 ms
192.168.0.1 pong: ttl=32 time=2 ms
192.168.0.1 pong: ttl=32 time=2 ms
3 packets transmitted, 3 packets received, 0% packet loss
round−trip min/avg/max = 2/2.3/3 ms
```

```
[admin@home_gw] >
```

Additional Resources

For more information about the WaveLAN / ORiNOCO adapter hardware please see the relevant User's Guides and Technical Reference Manuals in .pdf format from the manufacturer:

- gsg_pc.pdf ORiNOCO PC Card Getting Started Guide
- ug_pc.pdf ORiNOCO PC Card User's Guide
- GSG_ISA.pdf ORiNOCO ISA Adapter Getting Started Guide
- GSG_PCI.pdf ORiNOCO PCI Adapter Getting Started Guide

Information about configuring the ORiNOCO wireless access point can be found there:

- GSAP1000.pdf ORiNOCO Access Point 1000 (AP−1000) Getting Started Guide
- ug_OM.pdf ORiNOCO Manager Suite User's Guide

DHCP Client and Server

Document revision 1.6 (05−May−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

DHCP (Dynamic Host Configuration Protocol) supports easy distribution of IP addresses for a network. The MikroTik RouterOS implementation includes both server and client modes and is compliant with RFC2131.

General usage of DHCP:

- IP assignment in LAN, cable−modem, and wireless systems
- Obtaining IP settings on cable−modem systems

IP addresses can be bound to MAC addresses using static lease feature.

DHCP server can be used with MikroTik RouterOS HotSpot feature to authenticate and account for DHCP clients. See the HotSpot Manual for more details.

Specifications

Packages required : ***dhcp***
License required : *Any*
Home menu level : ***/ip dhcp−client, /ip dhcp−server***

Protocols utilized : *DHCP (RFC2131)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
IP Pools
HotSpot Gateway

Description

The DHCP protocol gives and allocates IP addresses to IP clients. DHCP is basically insecure and should only be used on secure networks. DHCP server listens on UDP 67 port, DHCP client − on UDP 68 port.

DHCP Client Setup

Submenu level : **/ip dhcp−client**

Description

The MikroTik RouterOS DHCP client may be enabled on one Ethernet−like interface. The client will accept an address, netmask, default gateway, and two dns server addresses. The IP address will be added to the interface with the netmask. The default gateway will be added to the routing table as a dynamic entry. When the DHCP client is disabled, the dynamic default route will be removed. If there is already a default route installed prior the DHCP client obtains one, the route obtained by the DHCP client would be shown as invalid.

The DNS−server from the DHCP server will be used as the router's default DNS if the router's DNS is set to **0.0.0.0** under the **/ip dns** settings.

Property Description

enabled (yes | no, default: **no**) − whether the DHCP client is enabled
interface (*name*) − any Ethernet−like interface (this includes wireless and EoIP tunnels)
host−name (*string*; default: "") − (optional) the host name of the client
client−id (*string*; default: "") − (optional) corresponds to the settings suggested by the network administrator or ISP
add−default−route (yes | no, default: **yes**) − whether to add the default route to the gateway specified by DHCP server
use−peer−dns (yes | no, default: **yes**) − whether to accept the DNS settings advertized by DHCP server (they will appear in **/ip dns** settings)

Notes

If **host−name** property is not specified, client's system identity will be sent in the respective field of DHCP request.

If **client−id** property is not specified, client's MAC address will be sent in the respective field of DHCP request.

To renew current leases, use the **renew** command. If the renew operation was not successful, client tries to reinitialize lease (i.e. it starts lease request procedure as it has not received an IP address yet).

Example

To enable DHCP client on **ether1** interface:

```
[admin@MikroTik] ip dhcp−client> set enabled=yes interface=ether1
[admin@MikroTik] ip dhcp−client> print
             enabled: yes
           interface: ether1
           host−name: ""
           client−id: ""
   add−default−route: yes
        use−peer−dns: yes
```

To show obtained leases:

```
[admin@MikroTik] ip dhcp−client> lease print
         address: 80.232.241.15/21
         expires: oct/20/2002 09:43:50
         gateway: 80.232.240.1
     primary−dns: 195.13.160.52
   secondary−dns: 195.122.1.59
[admin@MikroTik] ip dhcp−client>
```

DHCP Server Setup

Submenu level : **/ip dhcp−server**

Description

The router supports an individual server for each Ethernet like interface. The MikroTik RouterOS DHCP server supports the basic functions of giving each requesting client an IP address/netmask lease, default gateway, domain name, DNS−server(s) and WINS−server(s) (for Windows clients) information.

Property Description

name (*name*; default: "")− descriptive name for server
interface (*name*) − Ethernet−like interface name
lease−time (*time*; default: **72h**) − the time that a client may use an address. The client will try to renew this address after a half of this time and will request a new address after time limit expires
address−pool (*name* | static−only; default: **static−only**) − IP pool, from which to take IP addresses for clients

- **static−only** − allow only the clients that have a static lease (i.e. no dynamic addresses will be given to clients, only the ones added in **lease** submenu)
  netmask (*integer*; default: **0**) − the netmask to be used by DHCP client
  gateway (*IP address*; default: **0.0.0.0**) − the default gateway to be used by DHCP client
  src−address (*IP address*; default: **0.0.0.0**) − the address which the DHCP client must use to renew an IP address lease. If there is only one static address on the DHCP server interface and the source−address is left

as **0.0.0.0**, then the static address will be used. If there are multiple addresses on the interface, an address in the same subnet as the range of given addresses should be used
dns−server (*string*; default: "") − the DHCP client will use this as the default DNS server. Two comma−separated DNS servers can be specified to be used by DHCP client as primary and secondary DNS servers
domain (*string*; default: "") − the DHCP client will use this as the 'DNS domain' setting for the network adapter
wins−server (*string*; default: "") − the Windows DHCP client will use this as the default WINS server. Two comma−separated WINS servers can be specified to be used by DHCP client as primary and secondary WINS servers
add−arp (yes | no; default: **no**) − whether to add dynamic ARP entry

- **no** − either ARP mode should be **enabled** on that interface or static ARP entries should be defined in **/ip arp** submenu

Notes

Winbox does not have option for specifying two DNS or WINS servers, you should use terminal console instead.

If using both Universal Client and DHCP Server on the same interface, client will only receive a DHCP lease in case it is directly reachable by its MAC address through that interface (some wireless bridges may change client's MAC address).

Example

To use MikroTik RouterOS DHCP server feature:

1. Specify address pool to be used for DHCP clients.
   Address pools are added/managed under the **/ip pool** menu, for example:

```
[admin@MikroTik] ip pool> add name=our−dhcp−clients ranges=10.0.0.2−10.0.1.254
[admin@MikroTik] ip pool> print
  # NAME                                      RANGES
  0 our−dhcp−clients                          10.0.0.2−10.0.1.254

[admin@MikroTik] ip pool>
```

   Do not inlude the DHCP server's (interface's) address into the pool range! See IP Pool Manual for more details!
2. Add a DHCP server to the interface, for example:

```
[admin@MikroTik] ip dhcp−server> add name=dhcp−office \
\... address−pool=our−dhcp−clients interface=ether1 lease−time=72h netmask=24 \
\... gateway=10.0.0.1 dns−server=10.0.0.1,159.148.60.2 domain=mt.lv
[admin@MikroTik] ip dhcp−server> enable dhcp−office
[admin@MikroTik] ip dhcp−server> print
Flags: X − disabled, I − invalid
  0   name="dhcp−office" interface=ether1 lease−time=72h
      address−pool=our−dhcp−clients netmask=24 gateway=10.0.0.1
      src−address=10.0.0.1 dns−server=10.0.0.1,159.148.60.2 domain="mt.lv"
      wins−server="" add−arp=yes
[admin@MikroTik] ip dhcp−server>
```

DHCP Server Leases

Submenu level : **/ip dhcp−server lease**

Description

DHCP server lease submenu is used to monitor and manage server's leases. You can also add static leases to issue the definite client (determined by MAC address) with the specified IP address.

Property Description

address (*IP address*; default: **0.0.0.0**) − leased IP address for the client
mac−address (*MAC address*; default: **00:00:00:00:00:00**) − MAC address of the client. It is base for static lease assignment
lease−time (*time*; default: **0**) − dictates the time that a client may use an address

- **never** (the same as **0**) − lease will never expire

netmask (*integer*; default: **0**) − the netmask to be given with the IP address coming from the range of addresses that can be given out
gateway (*IP address*; default: "") − the default gateway to be used by the DHCP client

Statistics:

server (*name*) − server name which serves this client
expires−after (*time*) − time until lease expires
status (waiting | testing | busy | offeres | bound) − lease status:

- **waiting** − not used static lease
- **testing** − testing whether this address is used or not
- **busy** − this address is used in the network, so it can not be leased
- **offered** − server has offered this lease to a client, but did not receive client confirmation
- **bound** − server has received client confirmation that it accepts offered address and is using it now

Notes

Blank default values for some properties meand that property will be taken from the server's default values.

Even though client address may be changed (with adding a new item) in **lease print** list, it will not change for the client. It is true for any changes in in the DHCP server configuration because of DHCP protocol. Client tries to renew assigned IP address only when half a lease time is past (it tries to renew several times). Only when full lease time is past and IP address was not renewed, new lease is asked (rebind operation).

Example

To assign **10.5.2.100** static IP address for the existing DHCP client (shown in the lease table as item #0):

```
[admin@MikroTik] ip dhcp−server lease> print
Flags: X − disabled, D − dynamic, H − hotspot
  #    ADDRESS         MAC−ADDRESS       EXPIRES−A... SERVER                STATUS
  0 D  10.5.2.90       00:04:EA:C6:0E:40 1h48m59s     switch                bound
  1 D  10.5.2.91       00:04:EA:99:63:C0 1h42m51s     switch                bound
[admin@MikroTik] ip dhcp−server lease> add copy−from=0 address=10.5.2.100
[admin@MikroTik] ip dhcp−server lease> print
```

```
Flags: X - disabled, D - dynamic, H - hotspot
  #    ADDRESS         MAC-ADDRESS       EXPIRES-A... SERVER                STATUS
  1 D  10.5.2.91       00:04:EA:99:63:C0 1h42m18s     switch                bound
  2    10.5.2.100      00:04:EA:C6:0E:40 1h48m26s     switch                bound
[admin@MikroTik] ip dhcp-server lease>
```

Additional DHCP Resources

Links for DHCP documentation:

http://www.ietf.org/rfc/rfc2131.txt?number=2131
http://www.isc.org/products/DHCP/
http://www.linuxdoc.org/HOWTO/mini/DHCP/
http://arsinfo.cit.buffalo.edu/FAQ/faq.cgi?pkg=ISC%20DHCP

DNS Client and Cache

Document revision 1.4 (21−Jul−2003)
This document applies to the MikroTik RouterOS V2.7

Table Of Contents

Summary

DNS cache is used to minimize DNS requests to an external DNS server as well as to minimize DNS resolution time. This is a simple recursive DNS server with local items.

Specifications

Packages required : ***dns−cache***
License required : *Any*
Home menu level : ***/ip***
Standards and Technologies : *DNS* (RFC1035)
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
AAA (Authentication, Authorization and Accounting)
HotSpot Gateway

Description

The MikroTik router with DNS cache feature enabled can be set as a primary DNS server for any DNS−compliant clients. Moreover, MikroTik router can be specified as a primary DNS server under its dhcp−server settings. When the DNS cache is enabled, the MikroTik router responds to DNS TCP and UDP requests on port 53.

DNS Client Configuration

Submenu level : **/ip dns**

Description

DNS client is used to provide domain name resolution for router itself as well as for the P2P clients connected to the router.

Property Description

resolve−mode (*read−only*: remote−dns | local−dns−cache) − the type of domain name resolution
- **remote−dns** − names will be resolved by asking remote DNS servers
- **local−dns−cache** − names will be resolved using local DNS cache

primary−dns (*IP address*; default: **0.0.0.0**) − primary DNS server

secondary−dns (*IP address*; default: **0.0.0.0**) − secondary DNS server

Notes

resolve−mode automatically changes to **local−dns−cache** when **dns−cache** is enabled.

When using DHCP Server and Client if the parameter **use−peer−dns** under **ip dhcp−client** is set to **yes** then **primary−dns** under **/ip dns** will change to DNS address given by DHCP Server.

If **resolve−mode** is **remote−dns** then DHCP, PPP, PPTP, L2TP, ISDN and PPPoE servers as DNS server address will specify the values listed under **/ip dns**, otherwise − server's own address.

Example

We will set the primary DNS server to 159.148.60.2:

```
[admin@MikroTik] ip dns> set primary−dns=159.148.60.2
[admin@MikroTik] ip dns> print
     resolve−mode: remote−dns
      primary−dns: 159.148.60.2
    secondary−dns: 0.0.0.0
[admin@MikroTik] ip dns> .. dns−cache set enabled=yes
[admin@MikroTik] ip dns> print
     resolve−mode: local−dns−cache
      primary−dns: 159.148.60.2
    secondary−dns: 0.0.0.0
[admin@MikroTik] ip dns>
```

You can see that **resolve−mode** has changed its value to **local−dns−cache**. It means that from this moment domain names will be resolved using local DNS cache.

DNS Cache Setup

Submenu level : **/ip dns−cache**

```
[admin@MikroTik] ip dns−cache> print
             enabled: no
                size: 256
      primary−server: 0.0.0.0
    secondary−server: 0.0.0.0
             running: no
[admin@MikroTik] ip dns−cache>
```

Property Description

enabled (yes | no; default: **no**) − defines whether DNS cache (TCP and UDP port 53) is enabled
size (*integer*; default: **256**) − size of cache in kilobytes
primary−server (*IP address*; default: **0.0.0.0**) − primary DNS server
secondary−server (*IP address*; default: **0.0.0.0**) − secondary DNS server
running (read only: yes | no) − whether the DNS cache is currently running
usage (read only: *percentage*) − percent of cache used
entries (read only: *integer*) − number of entries in cache

Notes

usage and **entries** are showed only when the DNS cache is running.

DNS servers are queried by DNS cache in the following order (the ones which are **0.0.0.0** are skipped):

1. ip dns−cache primary−server
2. ip dns−cache secondary−server
3. ip dns primary−server
4. ip dns secondary−server

Example

To enable DNS cache using 159.148.60.2 as the router's primary DNS server:

```
[admin@MikroTik] ip dns−cache> set enabled=yes primary−server=159.148.60.2
[admin@MikroTik] ip dns−cache> print
             enabled: yes
                size: 256
      primary−server: 159.148.60.2
    secondary−server: 0.0.0.0
             running: yes
               usage: 0 %
             entries: 0
[admin@MikroTik] ip dns−cache>
```

Adding Static DNS Entries

Submenu level : **/ip dns−cache hosts**

Description

The MikroTik RouterOS has an embedded DNS server feature in DNS cache. It allows you to link the particular domain names with the respective IP addresses and advertize this link to the DNS clients using the router as their DNS server.

Property Description

address (*IP address*) − IP address to link the domain name with
name (*text*) − the name to be resolved to the given IP address

Example

To add a static DNS entry for **admin.home.pc** domain name to be resolved to **10.0.0.10** IP address:

```
[admin@MikroTik] ip dns−cache hosts> add name=admin.home.pc address=10.0.0.10
[admin@MikroTik] ip dns−cache hosts> print
  # ADDRESS         NAME
  0 10.0.0.10       admin.home.pc

[admin@MikroTik] ip dns−cache hosts>
```

Flushing DNS cache

Command name : **/ip dns−cache flush**

Description

DNS cache can be flushed using this command when it is disabled and not running.

Example

To flush DNS cache:

```
[admin@MikroTik] ip dns−cache> print
             enabled: yes
                size: 256
      primary−server: 159.148.60.2
    secondary−server: 0.0.0.0
             running: yes
               usage: 32 %
             entries: 358
[admin@MikroTik] ip dns−cache> set enabled=no
[admin@MikroTik] ip dns−cache> print
             enabled: no
                size: 256
      primary−server: 159.148.60.2
    secondary−server: 0.0.0.0
```

```
             running: no
[admin@MikroTik] ip dns-cache> flush
[admin@MikroTik] ip dns-cache> set enabled=yes
[admin@MikroTik] ip dns-cache> print
             enabled: yes
                size: 256
      primary-server: 159.148.60.2
    secondary-server: 0.0.0.0
             running: yes
               usage: 0 %
             entries: 0
[admin@MikroTik] ip dns-cache>
```

Additional Resources

Below are the links to DNS documentation:

http://www.freesoft.org/CIE/Course/Section2/3.htm
http://www.networksorcery.com/enp/protocol/dns.htm
http://www.ietf.org/rfc/rfc1035.txt?number=1035

HotSpot Gateway

Document revision 1.31 (08−Oct−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

The MikroTik HotSpot Gateway enables provision of public network access for clients using wireless or wired network connections.

HotSpot Gateway features:

- authentication of clients using local client database, or RADIUS server
- accounting using local log database, or RADIUS server
- HotSpot Gateway can provide access for authorized clients using two different methods
  - **dhcp−pool** method uses DHCP server to assign temporary (not valid in outer networks) IP addresses to clients prior to authentication. After successful authentication the DHCP server assigns address to client from different pool. This method may be used to assign a fixed IP address to each user (i.e. no matter which computed does the user use, he/she will always use the same IP address)
  - **enabled−address** method enables traffic for authorized clients without IP address change
- traffic and connection time accounting
- clients can be limitated by
  - download/upload speed (tx/rx bitrate)
  - connection time
  - downloaded/uploaded traffic (bytes)

Universal Clinet feature may be used with HotSpot **enabled−address** method to provide IP network services regardless of client computers' IP network settings.

Specifications

Packages required : ***hotspot***, ***dhcp****(optional, required by* ***dhcp−pool*** *method)*, **web−proxy** (optional)
License required : ***Basic*** *plus any additional (limited to 4 active users otherwise)*

Home menu level : ***/ip hotspot***
Protocols utilized : *ICMP (RFC792), DHCP (RFC2131)*
Hardware usage: *not significant*

Related Documents

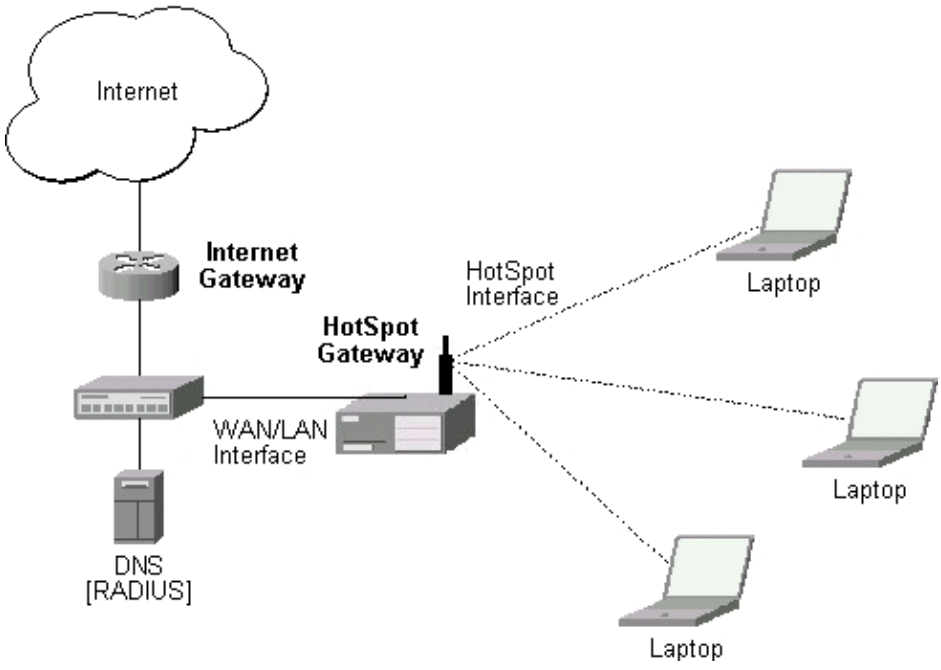
Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
IP Pools
DHCP Client and DHCP Server
General Point to Point Settings
Firewall Filters and Network Address Translation (NAT)
Log Management
Authentication, Authorization and Accounting
Certificate Management

Description

MikroTik HotSpot Gateway should have at least two network interfaces:

1. HotSpot interface, which is used to connect HotSpot clients;
2. LAN/WAN interface, which is used to access network resources. For example, DNS and RADIUS server(s) should be accessible.

The diagram below shows sample HotSpot setup.



The HotSpot interface should have an IP address assigned to it. To use **dhcp−pool** method, there should be two IP addresses: one as gateway for the temporary address pool prior to authentication, and second as gateway for the permanent address pool used for authenticated clients. **Note**, that you have to provide routing for these address pools, unless you plan to use masquerading (source NAT).

Only in **dhcp−pool** case, the arp feature should be set to **reply−only** on HotSpot interface to prevent network access using static IP addresses (the DHCP server will add static ARP entries for each DHCP client).

Physical network connection has to be established between the HotSpot user's computer and the gateway. It can be wireless (the wireless card should be registed to AP), or wired (the NIC card should be connected to a HUB or a switch).

Note that Universal Client feature cannot be used with **dhcp−pool** method.

The Initial Contact

One of two methods may be used for each client individually (you may choose one or allow it to be done automatically). The **enabled−address** method is preferred, so if it is configured correctly and the client has a proper IP address (that matches the one set in the user database), this method is to be used.

If the **enabled−address** method is not enabled or the client's IP address should be changed, the HotSpot Gateway tries to use **dhcp−pool** method. In that case, MikroTik HotSpot Gateway's DHCP server assigns IP addresses from the temporary address pool with a very short lease time (approx. 14s), so the address can be changed after authentication.

If user tries to access network resources using web browser, the destination NAT rule redirects all TCP connection requests to HotSpot service (port 8088 by default). This brings up the HotSpot Welcome/Login page.

It may be useful to have port 80 for HotSpot service because the users might want to see status and log out pages. If this is impossible, you may assign virtual IP address for hotspot service. It is done by redirecting all requests going to that virtual IP to hotspot service.

Note that you may want to have DNS traffic enabled (or redirected to the router's DNS cache) so that the client could be logged in connecting any valid web−page (using it's DNS name). Enabling ICMP ping might be useful as well, since it shows network connectivity. Other traffic should be dropped.

The Servlet

If user is not logged in, login page will be shown (where username and password has to be entered), but if user is logged in, status page will be shown (status: username, IP address, session time, bytes and packets transferred, ...). There are 6 HTML pages that can be easily modified by creating HTML template pages and uploading them to the hotspot folder on MikroTik router. These pages are called 'servlet' in this document and are described in detailes later on.

Authentication

Going to any HTTP address with web browser will be redirected to HotSpot authentication page prompting for username and password. Password together with HotSpot generated challenge string is hashed using MD5 algorithm (which in this case is implemented using JavaScript) and is executed on client's computer by web browser. After that, the hash result together with username is sent over Ethernet network to HotSpot service. So, password is never sent in plain text over ip network.

Note that password encryption is the reason why web browsers, that do not support JavaScript (for example, Internet Explorer 2.0), will not be able to authenticate users. It is possible to allow unencrypted passwords to

be accepted, but it is not recommended to use this feature.

HotSpot can authenticate users using local user database or a RADIUS server (local database if consupled first, then − RADIUS server) If authentication is done locally, profile corresponding to that user is used, otherwise (in case of RADIUS) default profile is used to set default values for parameters, which are not set in RADIUS access−accept message.

If authentication by http cookie is enabled, then after each successful login cookie is sent to web browser and the same cookie is added to active HTTP cookie list. Next time user will try to log in, web browser will send http cookie. This cookie will be compared to the one on HotSpot and only if there is the same source MAC address and the same randomly generated ID, user is automatically logged in. New cookie with different random ID is sent to web browser. Old cookie is removed from local HotSpot active cookie list. New one with new expire time is added.

Address Assignment with dhcp−pool Method

When user is successfully authenticated, HotSpot assigns another IP address for client (static or from some IP pool). On next clients DHCP request, the new IP address will be given by DHCP server to this client. How much time this IP address change requires, depends on DHCP lease time for non authenticated users. HotSpot **login−delay** parameter should be set accordingly to this DHCP server lease time. If lease time is 10s, then real login−delay will be about 1..7 seconds. So, it is quite safe to set **login−delay** to 8s in this case.

While IP address is changed, user sees after−login (**alogin.html**) page. This page will automatically forward user to original destination address (or status page, if there was no original dst address) after login−delay time will pass.

Logging Out

User can log out using status page. There is a link to http://*HotSpot−IP*/logout Going to this page will logout user. After that logout page (**logout.html**) will be shown to user.

HotSpot Gateway Setup

Submenu level : **/ip hotspot**

```
[admin@MikroTik] ip hotspot> print
                         use-ssl: no
                hotspot-address: 0.0.0.0
                        dns-name: ""
              status-autorefresh: 1m
                 universal-proxy: no
                        auth-mac: no
               auth-mac-password: no
                auth-http-cookie: no
            http-cookie-lifetime: 1d
    allow-unencrypted-passwords: no
                split-user-domain: no
[admin@MikroTik] ip hotspot>
```

Property Description

use−ssl (yes | no, default: **no**) − whether the servlet allows only HTTPS:
- **yes** − the regintration may only occur using the Secure HTTP (HTTPS) protocol
- **no** − the registration may be accomplished using both HTTP and HTTPS protocols

hotspot−address (*IP address*, default: **0.0.0.0**) − IP address for HotSpot service (used for www access)
status−autorefresh (*time*, default: **1m**) − WWW status page autorefresh time
universal−proxy (yes | no; default: **no**) − whether to intercept the requests to HTTP proxy servers
auth−mac (yes | no, default: **no**) − defines whether authentication by ethernet MAC address is enabled
auth−mac−password (yes | no, default: **no**) − uses MAC address as password if MAC authorization is enabled
auth−http−cookie (yes | no, default: **no**) − defines whether HTTP authentication by cookie is enabled
http−cookie−lifetime (*time*, default: **1d**) − validity time of HTTP cookies
allow−unencrypted−passwords (yes | no; default: **no**) − whether to authenticate user if plain−text password is received
split−user−domain (yes | no; default: **no**) − whether to split username from domain name when the username is given in "user@domain" or in "domain\user" format

Notes

If **dns−name** property is not specified, **hotspot−address** is used instead. If **hotspot−address** is also absent, then both are to be detected automatically.

If **auth−mac** is enabled, then client is not prompted for username and password if the MAC address of this computer is in the user database (either local or on RADIUS). Nevertheless this method does not excuse clients from the common login procedure, just from filling out the registration form (i.e. regardless of whether MAC authorization is applicable for a client, he/she should open the Login page in order to get registred)

universal−proxy requires **web−proxy** software package. This feature automatically creates DST−NAT rules to redirect requests of each particular user to a proxy server he/she is using (it may be set in his/her settings to use an unknown to us proxy server) to the local proxy server. To get it work you shuold have web proxy server up and running. This feature may be used in combination with Universal Client feature to provide Internet access for users regardless of their network settings.

If you are using a parent proxy with universal proxy feature, you should add a rule to the Direct Access list for each IP address HotSpot is running on allowing the requests destined to the local HotSpot server to be resolved directly by the local web proxy. For example, if the HotSpot server is running on **10.0.0.1** address:

```
/ip web−proxy direct add dst−address=10.0.0.1/32 action=allow
```

allow−unencrypted−passwords property makes it possible to authenticate with the browsers not supporting JavaScript. It is also possible to log−in using telnet connection, just requesting the page **/login?user=*username*word**. An another use of this property is the possibility of hard−coded authentication information in the servlet's login page simply creating the appropriate link.

The domain get from the username (enabling **split−user−domain** property) is used later in AAA process (for example, to specify different RADIUS servers for authentication depending on domain name)

Example

To enable cookie support:

```
[admin@MikroTik] ip hotspot> set auth-http-cookie=yes
[admin@MikroTik] ip hotspot> print
                        use-ssl: no
                hotspot-address: 0.0.0.0
                       dns-name: ""
             status-autorefresh: 1m
                universal-proxy: no
                       auth-mac: no
              auth-mac-password: no
               auth-http-cookie: yes
           http-cookie-lifetime: 1d
    allow-unencrypted-passwords: no
              split-user-domain: no
```

HotSpot Server Settings

Submenu level : **/ip hotspot server**

Property Description

name (*name*, default: "") − server profile name
dhcp-server (*name*) − DHCP server with which to use this profile
lease-time (*time*, default: **1m**) − DHCP lease time for logged in user
login-delay (*time*, default: **10s**) − Time required to log in user
address-pool (*name*) − IP pool name, from which HotSpot client will get IP address if it is not given some static already
netmask (*integer*, default: **0**)− network mask
gateway (*IP address*, default: **0.0.0.0**) − default gateway

Notes

This configuration is ignored by **enabled-address** method.

There can be added one server for each DHCP server. Which server profile to apply will depend on DHCP server which gave DHCP lease to that client. Actually it means that if user will log in from different interfaces, then different server profiles will be used. It allows assigning different IP addresses on different ethernet interfaces.

Example

To add hotspot server named **dhcp1** to the DHCP server **hotspot-dhcp** giving IP addresses from the **hotspot** pool with network mask **24** and gateway address **10.0.0.1**:

```
[admin@MikroTik] ip hotspot server> add name=dhcp1 dhcp-server=hotspot-dhcp \
\... address-pool=hotspot netmask=24 gateway=10.0.0.1
[admin@MikroTik] ip hotspot server> print
  0 name="dhcp1" dhcp-server=hotspot-dhcp lease-time=1m login-delay=10s
    address-pool=hotspot netmask=24 gateway=10.0.0.1
```

```
[admin@MikroTik] ip hotspot server>
```

HotSpot AAA

HotSpot User Profiles

Submenu level : **/ip hotspot profile**

Description

User profiles are used for common user settings. Profiles are like user groupes, they are grouping users with the same limits

Property Descriptions

name (*name*) − profile name
session−timeout (*time*, default: **0s**) − session timeout (maximal session time) for client
idle−timeout (*time*, default: **0s**) − idle timeout (maximal period of inactivity) for client
only−one (yes | no, default: **yes**) − only one simultaneous login per user
tx−bit−rate (*integer*, default: **0**) − transmit bitrate

- **0** − no limitation

rx−bit−rate (*integer*, default: **0**) − receive bitrate

- **0** − no limitation

incoming−filter (*name*, default: "") − firewall chain name for incoming packets
outgoing−filter (*name*, default: "") − firewall chain name for outgoing packets
mark−flow (*name*, default: "") − traffic from logged in users will be marked by firewall mangle with this flow name
login−method (dhcp−pool | enabled−address | smart, default: **smart**) − the login method user will be using:

- **dhcp−pool** − login by changing IP address via DHCP server
- **enabled−address** − login by enabling access for client's existing IP address
- **smart** − choose best login method for each case

keepalive−timeout (*time*, default: **2m**) − keepalive timeout for client

Notes

To use **enabled−address** method, **mark−flow** should be set

To use **dhcp−pool** method, **dhcp** software package should be installed

idle−timeout is used to detect, that client is not using outer networks (e.g. Internet), i.e., there is NO TRAFFIC coming from that client and going through the router. **keepalive−timeout** is used to detect, that the computer of the client is still alive and reachable. Server will check client's reachability. If check will fail during this period, client will be logged out.

Example

To use **enabled−address** method that uses **logged−in** mark and logs a client off if he disappears for more then a minute:

```
[admin@MikroTik] ip hotspot profile> set default login−method=enabled−address \
\... mark−flow=logged−in keepalive−timeout=1m
```

```
[admin@MikroTik] ip hotspot profile> print
Flags: * − default
  0 * name="default" session−timeout=0s idle−timeout=0s only−one=yes
      tx−bit−rate=0 rx−bit−rate=0 incoming−filter="" outgoing−filter=""
      mark−flow="logged−in" login−method=enabled−address keepalive−timeout=1m



[admin@MikroTik] ip hotspot profile>
```

To define an additional profile that also limits download speed to 64 kilobyte/s and upload data rate to 32 kilobyte/s, and call it **limited**:

```
[admin@MikroTik] ip hotspot profile> add copy−from=default tx−bit−rate=65536 rx−
bit−rate=32768 name=limited
[admin@MikroTik] ip hotspot profile> print
Flags: * − default
  0 * name="default" session−timeout=0s idle−timeout=0s only−one=yes
      tx−bit−rate=0 rx−bit−rate=0 incoming−filter="" outgoing−filter=""
      mark−flow="logged−in" login−method=enabled−address keepalive−timeout=1m


  1   name="limited" session−timeout=0s idle−timeout=0s only−one=yes
      tx−bit−rate=65536 rx−bit−rate=32768 incoming−filter=""
      outgoing−filter="" mark−flow="logged−in" login−method=enabled−address
      keepalive−timeout=1m


[admin@MikroTik] ip hotspot profile>
```

HotSpot Users

Submenu level : **/ip hotspot user**

Property Description

name (*name*) − user name
password (*string*, default: "") − user password
address (*IP address*, default: **0.0.0.0**) − static IP address. If not **0.0.0.0**, client will get always the same IP address. It implies, that only one simultaneous login for that user is allowed
mac−address (*MAC address*, default: **00:00:00:00:00:00**) − static MAC address. If not **00:00:00:00:00:00**, client is allowed to login only from that MAC address
profile (*name*, default: **default**) − user profile
routes (*string*, default: "") − routes that appear on the server when the client is connected. The route format is: "dst−address gateway metric" (for example, "10.1.0.0/ 24 10.0.0.1 1"). Several routes may be specified separated with commas
limit−uptime (*time*, default: **0s**) − total uptime limit for user (pre−paid time)
limit−bytes−in (*integer*, default: **0**) − maximum amount of bytes user can transmit
limit−bytes−out (*integer*, default: **0**)− maximum amount of bytes user can receive

Notes

MAC address should be written in CAPITAL letters

If **auth−mac** parameter is enabled, clients' MAC addresses (written with CAPITAL letters) can be used as usernames. If **auth−mac−password** is set to **no**, there should be no password for that users. In the other case, the password should be equal to the username. When client is connecting, it's MAC address is checked first. If there is a user with that MAC address, the client is authenticated as this user. If there is no match, client is asked for username and password.

The **address** property is used only for **dhcp−pool** login method to tell it DHCP server. If a user already has a permanent IP address (as it is happening when **enabled−address** method is used), this property will just be ignored.

The byte limits are total limits for each user (not for each session as at **/ip hotspot active**). So, if user has already downloaded something, then session limit will be total limit − (minus) already downloaded. For example, if download limit for user is 100MB and user has already downloaded 30MB, then session download limit after login at **/ip hotspot active** will be 100MB − 30MB = 70MB.

If user will reach his limits (bytes−in >= **limit−bytes−in** or bytes−out >= **limit−bytes−out**), he will not be able to log on anymore.

Example

To add user **Ex** with password **Ex** that is allowed to log in only with **01:23:45:67:89:AB** MAC address and is limited to 1 hour of work:

```
[admin@MikroTik] ip hotspot user> add name=Ex password=Ex \
\... mac−address=01:23:45:67:89:AB limit−uptime=1h
[admin@MikroTik] ip hotspot user> print
Flags: X − disabled
  #   NAME        ADDRESS         MAC−ADDRESS       PROFILE     UPTIME
  0   Ex          0.0.0.0         01:23:45:67:89:AB default     0s
[admin@MikroTik] ip hotspot user> print detail
Flags: X − disabled
  0   name="Ex" password="Ex" address=0.0.0.0 mac−address=01:23:45:67:89:AB
      profile=default routes="" limit−uptime=1h limit−bytes−in=0
      limit−bytes−out=0 uptime=0s bytes−in=0 bytes−out=0 packets−in=0
      packets−out=0


[admin@MikroTik] ip hotspot user>
```

HotSpot Active Users

Submenu level : **/ip hotspot active**

Description

The active user list shows the list of currently logged in users. Nothing can be changed here, except user can be logged out with the **remove** command

Property Description

Statistics:

user (*name*) − name of user logged in
domain (*string*) − domain of logged−in user (if split from username)
address (*IP address*) − IP address of logged in user
uptime (*time*) − current session time (logged in time) for this IP address
session−timeout (*time*) − how much time it is left for IP address until it will be automatically logged out
idle−timeout (*time*) − how much idle time it is left for IP address until it will be automatically logged out

Example

To get the list of active users:

```
[admin@MikroTik] ip hotspot active> print
Flags: R − radius, H − DHCP
  #    USER        ADDRESS         UPTIME          SESSION−TIMEOUT IDLE−TIMEOUT
  0    Ex          10.0.0.144      4m17s           55m43s
[admin@MikroTik] ip hotspot active>
```

HotSpot User Statistics

Command name : **/ip hotspot user print stats**

Property Description

Statistics:

uptime (*time*)− total time user has been logged in
bytes−in (*integer*) − total bytes received from user
bytes−out (*integer*) − total bytes sent to user
packets−in (*integer*) − total packets received from user
packets−out (*integer*) − total packets sent to user

Notes

These stats are updated if user is authenticated via local user databese each time user logs out. It means, that if user is currently logged in, then these stats will not show current total values. Use **/ip hotspot active print stats** to produce statistics on current user sessions.

Example

To get the list of active users:

```
[admin@MikroTik] ip hotspot user> print stats
Flags: X − disabled
  #   NAME         UPTIME          BYTES−IN   BYTES−OUT  PACKETS−IN PACKETS−OUT
  0   Ex           5m5s            0          0          0          0
[admin@MikroTik] ip hotspot user>
```

HotSpot Remote AAA

Submenu level : **/ip hotspot aaa**

```
[admin@MikroTik] ip hotspot aaa> print
```

```
        use-radius: no
        accounting: yes
    interim-update: 0s
[admin@MikroTik] ip hotspot aaa>
```

Property Description

use-radius (yes | no, default: **no**) - whether user database in a RADIUS server should be consulted
accounting (yes | no, default: **yes**) - whether RADIUS accounting is used
interim-update (*time*, default: **0s**) - Interim-Update time interval

Notes

RADIUS user database is consulted only if the required username is not found in local user database

Example

To enable RADIUS AAA:

```
[admin@MikroTik] ip hotspot aaa> set use-radius=yes
[admin@MikroTik] ip hotspot aaa> print
        use-radius: yes
        accounting: yes
    interim-update: 0s
[admin@MikroTik] ip hotspot aaa>
```

HotSpot Cookies

Submenu level : **/ip hotspot cookie**

Property Description

Statistics:

user (*name*) - username
domain (*string*) - domain name (if split from username)
mac-address (*MAC address*) - client's MAC address
expires-in (*time*) - how long the cookie is valid

Example

To get the list of valid cookies:

```
[admin@MikroTik] ip hotspot cookie> print
  # USER               DOMAIN            MAC-ADDRESS       EXPIRES-IN
  0 Ex                                   01:23:45:67:89:AB 23h54m16s
[admin@MikroTik] ip hotspot cookie>
```

Customizing Hotspot Servlet

Description

There are 6 HTML pages to interact with hotspot client:

- **login.html** − login page
- **status.html** − status page for logged in user
- **logout.html** − after−logged−out page
- **error.html** − various error messages
- **redirect.html** − redirecting web browser to another url
- **alogin.html** − page, which is shown after successful login while client gets new IP address from DHCP server (for 10 seconds or so)

There are many possibilities to customize what the hotspot authentication pages look like:

- The pages are easily modifiable. They are stored on the router's FTP server in **hotspot** directory.
- By changing the variables, which client sends to the HotSpot servlet, it is possible to reduce keyword count to one (username or password; the client's MAC address may be used as the other value) or even to zero (License Agreement; some predefined values general for all users or client's MAC address may be used as username and password)
- Registration may occur on a different server. Client's MAC address may be passed to it, so that this information need not be written in manually. After the registration, the server may change RADIUS database enabling client to log in.

Variable Description

All of the pages use variables to show user specific values. For each variable there is an example included in brackets.

Common variables (available in all pages):

- **hostname** − IP address for hotspot www access ("10.5.50.1")
- **link−logout** − link to logout page ("http://10.5.50.1/logout")
- **link−login** − link to login page ("http://10.5.50.1/login?dst=http://www.mt.lv/")
- **link−status** − link to status page ("http://10.5.50.1/status")
- **link−orig** − link to original destination page ("http://www.mt.lv/")
- **session−id** − value of 'session−id' parameter in last request
- **var** − value of 'var' parameter in last request

Page specific variables:

- redirect.html:
  - **link−redirect** − page to which redirect has to be done (for example, "http://www.mt.lv/")
- login.html:
  - **mac** − MAC address ("01:23:45:67:89:AB")
  - **error** − error message, if previous login failed ("invalid username or password")
  - **input−user** − name and value of username input field ("name=user value=john")
  - **input−password** − name of password input field ("name=password")
  - **input−popup** − name and value of popup input field ("name=popup checked")
  - **form−input** − name of input form and login JavaScript for password encoding ("name=login onSubmit=...")

- ♦ **main** − MD5 encryption JavaScript and form for encrypted password
- ♦ **user** − value of username input field ("john")
- ♦ **domain** − value of domain ("mikrotik")
- ♦ **popup** − value of pop−up checkbox ("true")
- ♦ **chap−id** − value of chap ID ("\371")
- ♦ **chap−challenge** − value of chap challenge ("\357\015\330\013\021\234\145\245\303\253\142\246\133\175\375\316")

Note that it is required login page to use use **main** variable. And it is strongly suggested to place it BEFORE **form−input** input form. Otherwise situation can happen, that user already has entered his username/password, but MD5 encryption JavaScript is not yet loaded. It may result in password being sent over network in plain text. And of course, that login will fail in this case, too (if **allow−unencrypted−password** property is not set to **yes**).

Note that the resulting password to be sent to the HotSpot gateway is formed MD5−hashing the concatenation of the following: **chap−id**, the password of the user and **chap−challange** (in the given order).

- alogin.html:
  - ♦ **link−redirect** − page to which redirect has to be done (for example, "http://www.mt.lv/")
  - ♦ **login−time** − time in seconds after which redirect has to be done ("9")
  - ♦ **popup** − **true** if alogin.html should pop−up status page in new window, **false** − otherwise
- status.html, logout.html: information on logged in user
  - ♦ **username** − name ("john")
  - ♦ **ip** − IP address ("192.168.0.222")
  - ♦ **mac** − MAC address ("01:23:45:67:89:AB")
  - ♦ **uptime** − session uptime ("10h2m33s")
  - ♦ **session−timeout** − session timeout left for user ("5h" or "−−−" if none)
  - ♦ **session−valid−till** − date and time when session will expire ("Sep/21/2002 16:12:33" or "−−−" if there is no session−timeout)
  - ♦ **idle−timeout** − idle timeout ("20m" or "−−−" if none)
  - ♦ **bytes−in** − number of bytes received from client ("15423")
  - ♦ **bytes−out** − number of bytes sent to client ("11352")
  - ♦ **packets−in** − number of packets received from client ("251")
  - ♦ **packets−out** − number of packets sent to client ("211")
- status.html:
  - ♦ **refresh−time** − time in seconds after which to automatically refresh status page
  - ♦ **refresh−time−str** − more friendly representation of **refresh−time**
- error.html:
  - ♦ **error** − error message ("DHCP lease not found")

To insert variable in some place in HTML file, variable name surrounded by **%** symbols is used. For example, to show link to login page, following construction can be used:

```
<a href="%link-login%">login</a>
```

It can be used in any hotspot HTML file.

Note, that to insert **%** symbol as a text (not as a part of variable construction), "%%" has to be used (if there is only one **%** symbol on a page or string between it and next **%** symbol is not a valid variable name, **%** may be used with the same result).

Examples

With basic HTML language knowledge and the information below it should be easy to implement the ideas described above

1. To provide predefined value as username, in login.html change:

   ```
   <input type="text" %input-user%>
   ```

   to this line:

   ```
   <input type="hidden" name="user" value="hsuser">
   ```

   (where **hsuser** is the username you are providing)
2. To provide predefined value as password, in login.html change:

   ```
   <input type="password" %input-password%>
   ```

   to this line:

   ```
   <input type="hidden" name="password" value="hspass">
   ```

   (where **hspass** is the password you are providing)
3. To send client's MAC address to a registration server in form of:

   ```
   https://www.server.serv/register.html?mac=XX:XX:XX:XX:XX:XX
   ```

   change the Login button link in login.html to:

   ```
   https://www.server.serv/register.html?mac=%mac%
   ```

   (you should correct the link to point to your server)
4. To show a banner after user login, in alogin.html after

   ```
   if ('%popup%' == 'true') newWindow();
   ```

   add the folowing line:

   ```
   open('http://your.web.server/your-banner-page.html', 'my-banner-name','');
   ```

   (you should correct the link to point to the page you want to show)
5. To choose different page shown after login, in login.html change:

   ```
   <input type="hidden" name="dst" value="%link_orig%">
   ```

   to this line:

   ```
   <input type="hidden" name="dst" value="http://your.web.server">
   ```

   (you should correct the link to point to your server)

Resetting Hotspot Servlet customizations

Command name : **/ip hotspot reset−html**

Description

The command overwrites the existing hotspot servlet with the original HTML files. It is used if you have changed the servlet and it is not working after that.

Example

To reset hotspot servlet html pages:

```
[admin@MikroTik] ip hotspot> reset−html
Current hotspot html pages will be lost! Reset anyway? [y/N]: y
[admin@MikroTik] ip hotspot>
```

QuestionSetup

Command name : **/ip hotspot setup**

Questions

hotspot interface (*name*) − interface to run HotSpot on
interface already configured (yes | no) − whether to add hotspot authentication for existing interface setup or interface setup should be configured from the scrach
enable universal client (yes | no; default: **no**) − whether to enable Universal Clinet on HotSpot interface
login method (dhcp−pool | enabled−address | smart; default: **enabled− address**) − login method to use
local address of temporary network (*IP address/mask*; default: **192.168.0.1/24**) − temporary HotSpot address for interface (for **dhcp−pool** method)
masquerade temporary network (yes | no; default: **yes**) − whether to masquerade temporary network
address pool of temporary network (*name*) − pool for temporary HotSpot addresses
local address of hotspot network (*IP address/mask*; default: **10.5.50.1/24**) − HotSpot address for interface
masquerade hotspot network (yes | no; default: **yes**) − whether to masquerade HotSpot network
address pool of hotspot network (*name*) − pool for HotSpot addresses
use ssl (yes | no; default: **no**) −whether to use secure SSL authentication
import and setup certificate (yes | no; default: **yes**) − if the setup should try to import and set up a certificate
passphrase (*text*) − the **passphrase** of the certificate
select certificate (*name*) − which cetrificate to use
ip address of smtp server (*IP address*) − IP address of the SMTP server to redirect SMTP requests (TCP port 25) to

- **0.0.0.0** − no redirect

use local dns cache (yes | no) − whether to redirect all DNS requests (UDP port 53) to the local DNS cache
dns−server (*IP address*, *IP address*) − DNS servers for HotSpot clients
dns name (*test*) − DNS domain name of the HotSpot gateway
name of local hotspot user (*string*; default: **admin**) − username of one automatically created user
password for the user (*string*; default: "") − password for the automatically created user
another port for service (*integer*; default: **8081**) − another port for **www** service (so that **hotspot** service

could be put on port **80**
use transparent web proxy (yes | no; default: **no**) − whether to use transparent web proxy for hotspot clients

Notes

Depending on current settings and answers to the previous questions, default values of following questions may be different. Some questions may disappear if they become redundant (for example, there is no use of 'temporary network' when login method is **enabled−address**)

If Universal Client is enabled, and DNS cache is not used, DNS requests are redirected to the first DNS server configured.

Example

To configure HotSpot on **ether1** interface (which is already configured), enabling transparent web proxy and adding user **admin** with password **rubbish**:

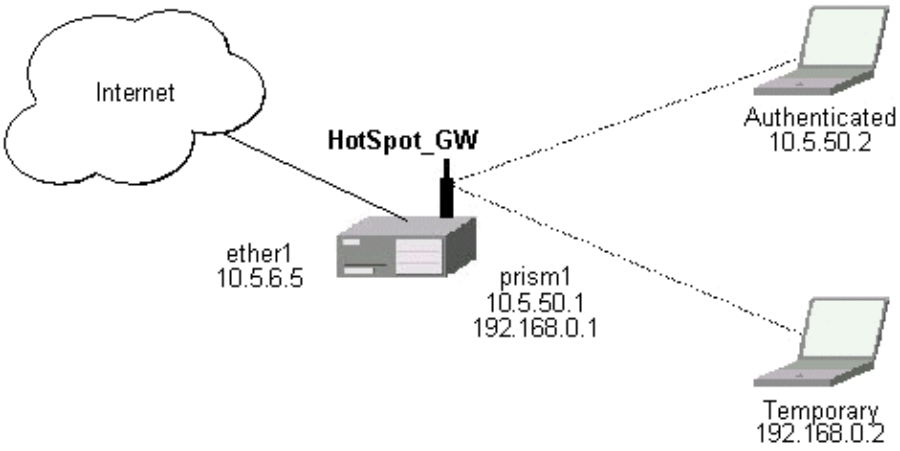
```
[admin@MikroTik] ip hotspot> setup
Select interface to run HotSpot on

hotspot interface: ether1
Use SSL authentication?

use ssl: no
Add hotspot authentication for existing interface setup?

interface already configured: yes
Create local hotspot user

name of local hotspot user: admin
password for the user: rubbish
Use transparent web proxy for hotspot clients?

use transparent web proxy: yes
[admin@MikroTik] ip hotspot>
```

HotSpot Step−by−Step User Guide

dhcp−pool Method

Planning the Configuration

Let us consider following example HotSpot setup:



There will be 2 hotspot IP address ranges used for clients on prism1 interface. You are free to choose the address ranges, just make sure you use masquerading for not routed ones. In our example, we are using

- temporary addresses which must be masqueraded:
  network: 192.168.0.0/24
  gateway: 192.168.0.1
  pool: 192.168.0.2−192.168.0.254
- real addresses which require routing:
  network: 10.5.50.0/24
  gateway: 10.5.50.1
  pool: 10.5.50.2−10.5.50.254

Temporary addresses are given out by DHCP server (configured within **/ip dhcp−server**), but real addresses are given out by hotspot server configuration.

For hotspot client accounting, hotspot will add dynamic firewall rules in firewall hotspot chain. This chain has to be created manually. And all network packets (to/from hotspot clients) have to pass this chain.

Setup Example

Follow the steps below:

1. Your ether1 interface is configured with IP address 10.5.6.5/24 and the default route points to gateway 10.5.6.1
2. Your prism1 interface is configured for AP mode and can register IEEE 802.11b wireless clients. See the Prism Interface Manual for more details.
3. ARP should be set to **reply−only** on prism interface, so no dynamic entries are added to the ARP table. DHCP server will add entries only for clients which have obtained DHCP leases.

   ```
   /interface prism set prism1 arp=reply-only
   ```
4. Add two IP addresses to prism1 interface:

   ```
   /ip address add address=192.168.0.1/24 interface=prism1 \
       comment="hotspot temporary network"
   /ip address add address=10.5.50.1/24 interface=prism1 \
       comment="hotspot real network"
   ```
5. add 2 IP pools:

```
/ip pool add name=hs-pool-temp ranges=192.168.0.2-192.168.0.254
/ip pool add name=hs-pool-real ranges=10.5.50.2-10.5.50.254
```

6. add masquerading rule for temporary IP pool, which is not routed:

```
/ip firewall src-nat add src-address=192.168.0.0/24 action=masquerade \
    comment="masquereade hotspot temporary network"
```

   Make sure you have routing for authenticated address space. Try to ping 10.5.50.1 from your internet gateway 10.5.6.1, for example. See the Basic Setup Guide on how to set up routing.
7. Add dhcp server (for temporary IP addresses):

```
/ip dhcp-server add name="hs-dhcp-server" interface=prism1 lease-time=14s \
    address-pool=hs-pool-temp netmask=24 gateway=192.168.0.1 \
    dns-server=159.148.60.2,159.148.108.1 domain="mt.lv" add-arp=yes disabled=no
```

8. Add hotspot server setup (for real IP addresses):

```
/ip hotspot server add name=hs-server dhcp-server=hs-dhcp-server \
    address-pool=hs-pool-real netmask=24 gateway=10.5.50.1
```

9. Add local hotspot user:

```
/ip hotspot user add name=Ex password=Ex
```

10. Setup hotspot service to run on port 80 (www service has to be assigned another port, e.g., 8081):

```
/ip service set www port=8081
/ip service set hotspot port=80
```

    Note! Changing www service to other port than 80 requires thet you specify the new port when connecting to MikroTik router using WinBox, e.g., use 10.5.50.1:8081 in this case.
11. Redirect all TCP requests from temporary IP addresses to hotspot service:

```
/ip firewall dst-nat add src-address=192.168.0.0/24 dst-port=443 protocol=tcp \
    action=redirect to-dst-port=443\
    comment="redirect unauthorized hotspot clients to hotspot service"
/ip firewall dst-nat add src-address=192.168.0.0/24 protocol=tcp \
    action=redirect to-dst-port=80 \
    comment="redirect unauthorized hotspot clients to hotspot service"
```

12. Allow DNS requests and ICMP ping from temporary addresses and reject everything else:

```
/ip firewall add name=hotspot-temp comment="limit unauthorized hotspot clients"

/ip firewall rule forward add src-address=192.168.0.0/24 action=jump \
jump-target=hotspot-temp comment="limit access for unauthorized hotspot clients"

/ip firewall rule input add src-address=192.168.0.0/24 dst-port=80 \
protocol=tcp action=accept comment="accept requests for hotspot servlet"
/ip firewall rule input add src-address=192.168.0.0/24 dst-port=443 \
protocol=tcp action=accept comment="accept request for hotspot servlet"
/ip firewall rule input add src-address=192.168.0.0/24 dst-port=67 \
protocol=udp action=accept comment="accept requests for local DHCP server"
/ip firewall rule input add src-address=192.168.0.0/24 action=jump \
jump-target=hotspot-temp comment="limit access for unauthorized hotspot clients"

/ip firewall rule hotspot-temp add protocol=icmp action=return \
    comment="allow ping requests"
/ip firewall rule hotspot-temp add protocol=udp dst-port=53 action=return \
    comment="allow dns requests"
/ip firewall rule hotspot-temp add action=reject \
```

```
    comment="reject access for unauthorized hotspot clients"
```

13. Add hotspot chain:

```
/ip firewall add name=hotspot comment="account authorized hotspot clients"
```

14. Pass all through going traffic to hotspot chain:

```
/ip firewall rule forward add action=jump jump-target=hotspot \
    comment="account traffic for authorized hotspot clients"
```

Note that in order to use SSL authentication, you should install an SSL certificate. This topic is not covered by this manual section. Please see the respective manual section on how to install certificates in MikroTik RouterOS

If client has obtained temporary address, its lease is shown as:

```
[admin@HotSpot-GW] > ip dhcp-server lease print
Flags: X - disabled, H - hotspot, D - dynamic
  #   ADDRESS         MAC-ADDRESS       EXPIRES-A... SERVER         STATUS
  0 D 192.168.0.254   00:40:96:13:B3:47 8s           hs-dhcp-server bound
[admin@HotSpot-GW] >
```

After successful authorization its DHCP address is changed, and it is listed under active hotspot users:

```
[admin@HotSpot-GW] > ip dhcp-server lease print
Flags: X - disabled, H - hotspot, D - dynamic
  #   ADDRESS         MAC-ADDRESS       EXPIRES-A... SERVER         STATUS
  0 H 10.5.50.2       00:40:96:13:B3:47 56s          hs-dhcp-server bound
[admin@HotSpot-GW] > ip hotspot active print
Flags: R - radius, H - DHCP
  #    USER        ADDRESS         UPTIME          SESSION-TIMEOUT IDLE-TIMEOUT
  0 R  Ex          10.5.50.2       2m25s
[admin@HotSpot-GW] > /ip hotspot active print stats
Flags: R - radius, H - DHCP
  #    USER        UPTIME          BYTES-IN   BYTES-OUT  PACKETS-IN PACKETS-OUT
  0 R  Ex          13m26s          145268     264282     475        494
[admin@HotSpot-GW] >
```

User statistics show accumulated values prior to current session.

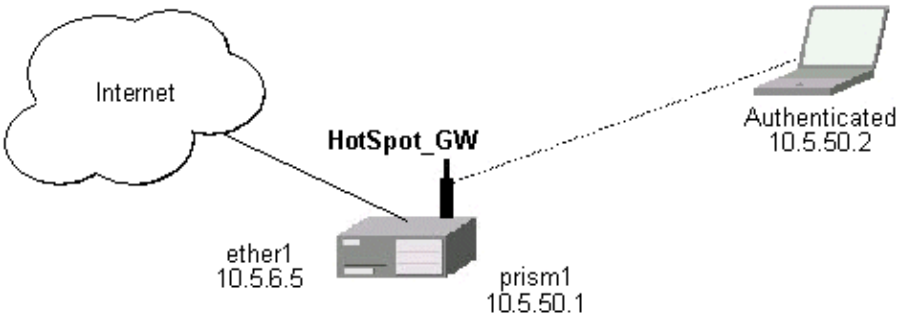
```
[admin@HotSpot-GW] > ip hotspot user print stats
Flags: X - disabled
  #   NAME         UPTIME          BYTES-IN   BYTES-OUT  PACKETS-IN PACKETS-OUT
  0   Ex           6m29s           9896       31156      80         77
[admin@HotSpot-GW] >
```

User statistics values are updated after current session is closed. Values can be reset to '0' using the **reset-counters** command.

enabled-address Method

Planning the Configuration

Let us consider following example HotSpot setup:



There are clients at prism1 interface, which are able to use Internet already. You want all clients at prism1 interface to authenticate before they are able to use Internet.

For hotspot client accounting, hotspot will add dynamic firewall rules in firewall hotspot chain. This chain has to be created manually. And all network packets (to/from hotspot clients) have to pass this chain.

Setup Example

Follow the steps below:

1. Setup hotspot service to run on port 80 (www service has to be assigned another port, e.g., 8081):

   ```
   /ip service set www port=8081
   /ip service set hotspot port=80
   ```

   Note! Changing www service to other port than 80 requires thet you specify the new port when connecting to MikroTik router using WinBox, e.g., use 10.5.50.1:8081 in this case.
2. Setup hotspot profile to mark authenticated users with flow name "hs-auth":

   ```
   /ip hotspot profile set default mark-flow="hs-auth" login-method=enabled-address
   ```
3. Add local hotspot user:

   ```
   /ip hotspot user add name=Ex password=Ex
   ```
4. Redirect all TCP requests from unauthorized clients to hotspot service:

   ```
   /ip firewall dst-nat add in-interface="prism1" flow="!hs-auth" protocol=tcp \
       dst-port=443 action=redirect to-dst-port=443 \
       comment="redirect unauthorized hotspot clients to hotspot service"
   /ip firewall dst-nat add in-interface="prism1" flow="!hs-auth" protocol=tcp \
       action=redirect to-dst-port=80 \
       comment="redirect unauthorized clients to hotspot service"
   ```
5. Allow DNS requests and ICMP ping from temporary addresses and reject everything else:

   ```
   /ip firewall add name=hotspot-temp comment="limit unauthorized hotspot clients"

   /ip firewall rule forward add in-interface=prism1 action=jump \
   jump-target=hotspot-temp comment="limit access for unauthorized hotspot clients"

   /ip firewall rule input add in-interface=prism1 dst-port=80 protocol=tcp \
   action=accept comment="accept requests for hotspot servlet"
   /ip firewall rule input add in-interface=prism1 dst-port=443 protocol=tcp \
   action=accept comment="accept request for hotspot servlet"
   /ip firewall rule input add in-interface=prism1 dst-port=67 protocol=udp \
   protocol=udp action=accept comment="accept requests for local DHCP server"
   ```

```
/ip firewall rule input add in-interface=prism1 action=jump \
jump-target=hotspot-temp comment="limit access for unauthorized hotspot clients"

/ip firewall rule hotspot-temp add flow="hs-auth" action=return \
    comment="return if connection is authorized"
/ip firewall rule hotspot-temp add protocol=icmp action=return \
    comment="allow ping requests"
/ip firewall rule hotspot-temp add protocol=udp dst-port=53 action=return \
    comment="allow dns requests"
/ip firewall rule hotspot-temp add action=reject \
    comment="reject access for unauthorized clients"
```

6. Create hotspot chain for authorized hotspot clients:

```
/ip firewall add name=hotspot comment="account authorized hotspot clients"
```

7. Pass all through going traffic to hotspot chain:

```
/ip firewall rule forward add action=jump jump-target=hotspot \
    comment="account traffic for authorized hotspot clients"
```

Note that in order to use SSL authentication, you should install an SSL certificate. This topic is not covered by this manual section. Please see the respective manual section on how to install certificates in MikroTik RouterOS

As we see from example, only hotspot interface is used − we don't care what IP addresses are there.

It is possible to add hotspot authentication for one more interface (prism2) by adding only 2 additional firewall rules:

1. Setup dst−nat to redirect unauthorized clients to hotspot service:

```
/ip firewall dst-nat add in-interface="prism2" flow="!hs-auth" protocol=tcp \
action=redirect to-dst-port=80 \
comment="redirect unauthorized prism2 clients to hotspot service"
```

2. Limit access for unauthorized prism2 interface clients:

```
/ip firewall rule forward add in-interface=prism2 action=jump \
jump-target=hotspot-temp comment="limit access for unauthorized prism2 clients"
/ip firewall rule input add in-interface=prism2 action=jump \
jump-target=hotspot-temp comment="limit access for unauthorized prism2 clients"
```

After successful authorization client is listed under active hotspot users:

```
[admin@HotSpot-GW] > ip hotspot active print
Flags: R - radius, H - DHCP
  #    USER        ADDRESS         UPTIME          SESSION-TIMEOUT IDLE-TIMEOUT
  0 R  Ex          10.5.50.2       2m25s
[admin@HotSpot-GW] > /ip hotspot active print stats
Flags: R - radius, H - DHCP
  #    USER        UPTIME          BYTES-IN   BYTES-OUT  PACKETS-IN PACKETS-OUT
  0 R  Ex          13m26s          145268     264282     475        494
[admin@HotSpot-GW] >
```

User statistics show accumulated values prior to current session.

```
[admin@HotSpot-GW] > ip hotspot user print stats
Flags: X - disabled
```

```
  #   NAME          UPTIME           BYTES-IN   BYTES-OUT  PACKETS-IN PACKETS-OUT
  0   Ex            6m29s            9896       31156      80         77
[admin@HotSpot-GW] >
```

User statistics values are updated after current session is closed. Values can be reset to '0' using the **reset-counters** command.

Optional Settings

1. You may want to use same address space both for your LAN and HotSpot networks. Please consult the IP Address and ARP Manual for proxy-arp feature.
2. You may want to translate the destination address of all TCP port 25 connections (SMTP) from HotSpot users to your mail sever for mail relaying. Thus, users can retain their mail client setup and use your mail server for outgoing mail without reconfiguring their mail clients. If 10.5.6.100 is your mail server accepting connections from network 10.5.50.0/24, then the required destination NAT rule would be:

   ```
   /ip firewall dst-nat add src-address=10.5.50.0/24 dst-port=25 protocol=tcp \
       to-dst-address=10.5.6.100 action=nat \
       comment="Translate SMTP TCP 25 port to our mail server"
   ```
3. Another option is to allow access certain pages without authentication. This is useful, for example, to give access to some general information about HotSpot service provider or billing options. Include firewall rules into the hotspot-temp chain allowing access to certain IP addresses prior the rule that rejects all other traffic from temporary addresses. Also, add rules excluding destination NAT for these addresses. For example:

   1) in dst-nat: don't redirect requests going to your web server (x.x.x.x:80) (this rule has to be before "redirect to hotspot service" rule!)

   ```
   /ip firewall dst-nat add dst-address=x.x.x.x/32 dst-port=80 protocol=tcp\
       action=accept
   ```

   2) in hotspot-temp chain: accept requests going to your web server (this rule has to be before "reject access for unauthorized hotspot clients" rule!)

   ```
   /ip firewal rule hotspot-temp add dst-address=x.x.x.x/32 dst-port=80 \
       protocol=tcp action=return
   ```
4. For HotSpot clients to use transparent web-proxy on the same router, following configuration can be used:

   1) make sure, web-proxy package is installed;
   2) it is assumed, that HotSpot is set up and successfully running. Hotspot clients are connected on interface named 'prism1'.
   3) set up web-proxy to run on port 3128 using transparent mode:

   ```
   /ip web-proxy set enabled=yes address=0.0.0.0:3128 transparent-proxy=yes
   ```

   4) set up HotSpot to use one of router's local IP addresses (10.5.50.1):

   ```
   /ip hotspot set hotspot-address=10.5.50.1
   ```

   5) redirect all requests from hotspot interface to port 80 (except to 10.5.50.1), to web-proxy:

```
/ip firewall dst-nat add in-interface=prism1 dst-address=!10.5.50.1/32 \
    dst-port=80 protocol=tcp action=redirect
    to-dst-port=3128 comment="transparent proxy"
```

Now, everything should be working. Only traffic of redirected requests to web-proxy will not be accounted. It's because this traffic will not pass through the forward chain.

6) to enable accounting for hotspot user traffic to/from transparent web-proxy, additional firewall rules should be added:

```
/ip firewall rule input add in-interface=prism1 dst-port=3128\
    protocol=tcp action=jump jump-target=hotspot\
    comment="account traffic from hotspot client to local web-proxy"
/ip firewall rule output add src-port=3128 protocol=tcp\
    out-interface=prism1 action=jump jump-target=hotspot\
        comment="account traffic from local web-proxy to hotspot client"
```

5. You may want to allow multiple logins using the same username/password. Set the argument value of **only-one** to **no** in hotspot profile, for example:

```
/ip hotspot profile set default only-one=no
```

6. If you have dns-cache package installed, setup local DNS cache and specify HotSpot gateway's address as primary DNS server for DHCP clients, for example:

```
/ip dns-cache set primary-server=159.148.60.2 enabled=yes
/ip dhcp-server set hs-dhcp-server dns-server=10.5.50.1,159.148.108.1
```

Troubleshooting

- **User cannot log in because of "NO CHAP" error**
  "NO CHAP" means, that hotspot server does not have a challenge for this IP address. It can be caused by:
  - ♦ logging in too slowly, i.e., client opens login page, but does login only after more than an hour;
  - ♦ web browser gets login page from its own cache, without asking for it to hotspot server

  So, in case of "NO CHAP" error, please reload login page from hotspot server (generally [F5] button in web browser).
  Don't use **Back** button of web browser to enter login page! That "old" login page has already used challenge value, which is not valid on hotspot server anymore.
- **User cannot log in, although username and password are proven correct**
  Web browsers, that do not support JavaScript (for example, Internet Explorer 2.0), are not be able to authenticate users because of password encryption. In this case you may either update the browser or enable **allow-unencrypted-passwords** property in HotSpot Server's general settings, allowing plain-text passwords to travel in your network:

  ```
  /ip hotspot set allow-unencrypted-passwords=yes
  ```
- **User cannot log in with Netscape 4.7x, bacause of "INVALID USER" error**
  It is caused by uninitialized domain value, which has value of "++" for those Netscape browsers. It will be fixed in RouterOS version 2.7.4. Now this can be fixed by changing hotspot login.html page. You will have to add line

  ```
  <input type="hidden" name="domain" value="">
  ```

  within form

```
<form %form-input%>
...
</form>
```

IP Addresses and Address Resolution Protocol (ARP)

Document revision 1.4 (29−Dec−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The following Manual discusses managing IP addresses and the Address Resolution Protocol (ARP). IP addresses serve as identification when communicating with other network devices using the TCP/IP protocol. In turn, communication between devices in one physical network proceeds with the help of Address Resolution Protocol and ARP addresses.

Specifications

Packages required : *None*
License required : *Any*
Home menu level : ***/ip address***, ***/ip arp***
Protocols utilized : *IP (RFC791), ARP (RFC826)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading

IP Addressing

Submenu level : **/ip address**

Description

IP addresses serve for a general host identification purposes in IP networks. Tupical (IPv4) address consists of four octets. For correct addressing the router also needs the network mask value, *id est* which bits of the complete IP address refer to the address of the host, and which − to the address of the network. The network address value is calculated by binary AND operation from network mask and IP address values. It's also possible to specify IP address followed by slash "/" and amount of bits assigned to a network mask.

In most cases, it is enough to specify the address, the netmask, and the interface arguments. The network prefix and the broadcast address are calculated automatically.

It is possible to add multiple IP addresses to an interface or to leave the interface without any addresses assigned to it. Leaving a physical interface without an IP address is a must when the bridging between interfaces is used. In case of bridging, the IP address is assigned to a bridge interface.

MikroTik RouterOS has following types of addresses:

- **Static IP Addresses** are user−assigned addresses to the network interfaces.
- **Dynamic IP Addresses** are assigned automatically when ppp, ppptp, or pppoe connections are established.

Property Description

address (*IP address*) − IP address of the host
broadcast (*IP address*; default: **255.255.255.255**) − broadcasting IP address, by default calculated from an IP address and a network mask
comment (*text*; default: "") − an optional comment for the IP address
disabled (yes | no; default: **no**) − is the address disabled or not
interface (*name*) − the name of the interface IP address assigned to
netmask (*IP address*; default: **0.0.0.0**) − specifies the network address part of an IP address
network (*IP address*; default: **0.0.0.0**) − IP address of the network. For the point−to−point links should be the address of the remote end

Example

```
[admin@MikroTik] ip address> add address=10.10.10.1/24 interface=ether2
[admin@MikroTik] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   2.2.2.1/24         2.2.2.0         2.2.2.255       ether2
  1   10.5.7.244/24      10.5.7.0        10.5.7.255      ether1
  2   10.10.10.1/24      10.10.10.0      10.10.10.255    ether2

[admin@MikroTik] ip address>
```

Address Resolution Protocol

Submenu level : **/ip arp**

Description

Address Resolution Protocol is used to map IP addreses to MAC layer addreses. A router has a table of currently used ARP entries. Normally the table is built dynamically, but to increase network security, static entries can be added.

Property Description

address (*IP address*) − IP address
comment (*text*; default: "") − an optional comment
disabled (yes | no; default: **no**) − is the entry disabled or not
interface (*name*) − the name of the interface
mac−address (*MAC address*; default: **00:00:00:00:00:00**) − MAC address to be mapped to

Notes

Maximal number of ARP entries is 1024.

If arp feature is turned off on interface, i.e., **arp=disabled** is used, ARP requests from clients are not answered by the router. Therefore, static arp entry should be added to the clients as well. For example, the router's IP and MAC addresses should be added to the Windows workstations using the **arp** command:

```
C:\> arp −s 10.5.8.254  00−aa−00−62−c6−09
```

Example

```
[admin@MikroTik] ip arp> add address=10.10.10.10 interface=ether2 mac−address=06 \\
\\... :21:00:56:00:12
[admin@MikroTik] ip arp> print
Flags: X − disabled, I − invalid, H − DHCP, D − dynamic
  #   ADDRESS         MAC−ADDRESS       INTERFACE
  0 D 2.2.2.2         00:30:4F:1B:B3:D9 ether2
  1 D 10.5.7.242      00:A0:24:9D:52:A4 ether1
  2   10.10.10.10     06:21:00:56:00:12 ether2
[admin@MikroTik] ip arp>
```

If static arp entries are used for network security on an interface, you should set arp to 'reply−only' on that interface. Do it under the relevant **/interfaces** menu:

```
[admin@MikroTik] ip arp> /interface ethernet set ether2 arp=reply−only
[admin@MikroTik] ip arp> print
Flags: X − disabled, I − invalid, H − DHCP, D − dynamic
  #   ADDRESS         MAC−ADDRESS       INTERFACE
  0 D 10.5.7.242      00:A0:24:9D:52:A4 ether1
  1   10.10.10.10     06:21:00:56:00:12 ether2

[admin@MikroTik] ip arp>
```

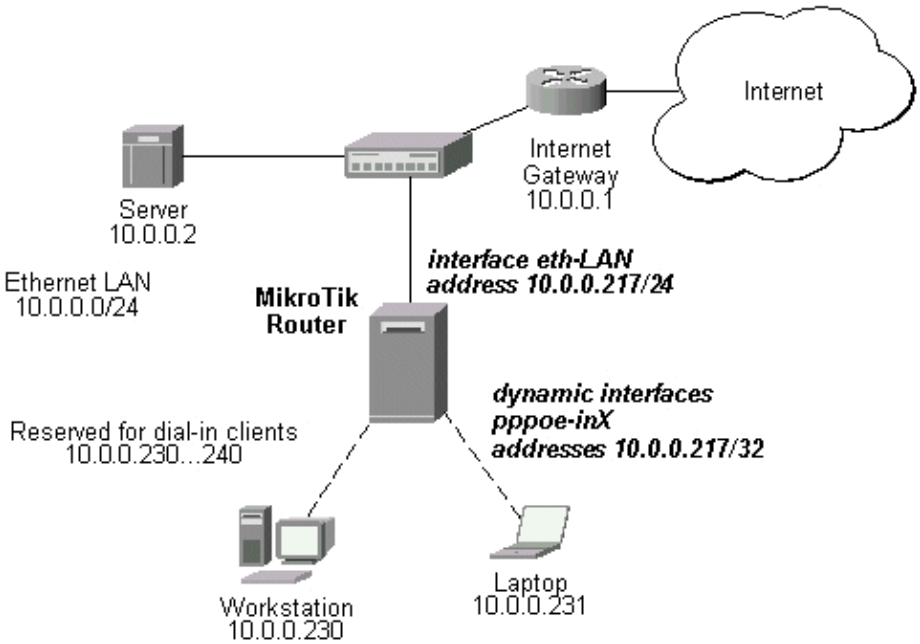
Using the Proxy−ARP Feature

Description

All physical interfaces, like Ethernet, Prism, Aironet (PC), WaveLAN, etc., can be set for using the Address Resolution Protocol or not. By default, the arp feature is **enabled**. However, it can be changed to **proxy−arp**. The Proxy−ARP feature means that the router will be listening to arp requests received at the relevant interface and respond to them with it's own MAC address, if the requests matches any other IP address of the router.

Example

For example, you can assign IP addresses to dial−in (ppp, pppoe, pptp) clients from the same address space as used on the connected LAN, of you enable the **proxy−arp** on the LAN interface. Let us consider the following setup:



The MikroTik router setup is as follows:

```
[admin@MikroTik] ip arp> /interface ethernet print
Flags: X − disabled, R − running
  #    NAME                MTU   MAC−ADDRESS       ARP
  0  R eth−LAN              1500  00:50:08:00:00:F5 proxy−arp
[admin@MikroTik] ip arp> /interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                TYPE             MTU
  0    eth−LAN             ether            1500
  1    prism1              prism            1500
  2 D  pppoe−in25          pppoe−in
  3 D  pppoe−in26          pppoe−in
[admin@MikroTik] ip arp> /ip address print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.217/24      10.0.0.0        10.0.0.255      eth−LAN
```

```
  1 D 10.0.0.217/32      10.0.0.230      0.0.0.0         pppoe-in25
  2 D 10.0.0.217/32      10.0.0.231      0.0.0.0         pppoe-in26
[admin@MikroTik] ip arp> /ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        eth-LAN
    1 DC 10.0.0.0/24        r 0.0.0.0         0        eth-LAN
    2 DC 10.0.0.230/32      r 0.0.0.0         0        pppoe-in25
    3 DC 10.0.0.231/32      r 0.0.0.0         0        pppoe-in26
[admin@MikroTik] ip arp>
```

Using Unnumbered Interfaces

Description

The unnumbered interfaces can be used on serial point-to-point links, e.g., MOXA or Cyclades interfaces. A private address should be put on the interface with the **network** being the same as an address on the router on the other side of the p2p link (there may be no IP on that interface, but there is an ip for that router).

Example

```
[admin@MikroTik] ip address> add address=10.0.0.214/32 network=192.168.0.1 \
\... interface=pppsync
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.214/32      192.168.0.1     192.168.0.1     pppsync
[admin@MikroTik] ip address>
[admin@MikroTik] ip address> .. route print detail
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    0  S dst-address=0.0.0.0/0 preferred-source=0.0.0.0 gateway=192.168.0.1
         gateway-state=reachable distance=1 interface=pppsync

    1 DC dst-address=192.168.0.1/32 preferred-source=10.0.0.214
         gateway=0.0.0.0 gateway-state=reachable distance=0 interface=pppsync

[admin@MikroTik] ip address>
```

Here, you can see, that a dynamic connected route has been automatically added to the routes list. If you want the default gateway be the other router of the p2p link, just add a static route for it. It is shown as #0 in the example above.

Troubleshooting

- *I added IP addresses 10.0.0.1/24 and 10.0.0.2/24 to the interfaces ether1 and ether2, but nothing works.*
  Both addresses are from the same network 10.0.0.0/24, use addresses from different networks on different interfaces, or enable **proxy-arp** on ether1 or ether2.
- *I was going to use static ARP and have my network secured that way. For the first 10 minutes everything is fine, then router becomes totally unavailable.*
  After you turn off ARP on router's interface, the dynamic ARP entries expire on the client computers. You should add the router's IP and MAC addresses to the static ARP entries of the workstations.

Additional Resources

Addressing in Local Area Networks

IP Pool Management

Document revision 1.1 (17−Feb−2003)
This document applies to the MikroTik RouterOS v2.7

Table of Contents

Summary

IP pools are used to define range of IP addresses that is used for DHCP server and Point−to−Point servers

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/ip pool***
Protocols utilized : *None*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Authentication, Authorization and Accounting
Dynamic Host Configuration Protocol (DHCP) Client and Server
HotSpot Gateway
Universal Client Interface
General Point to Point Settings

Description

IP pools simply group IP addresses for further usage. It is a single configuration point for all features that assign IP addresses to clients.

Note that whenever possible, the same ip address is given out to each client (OWNER/INFO pair).

IP Pool Setup

Submenu level : **/ip pool**

Property Description

name (*name*) – name of the pool
ranges (*string*) − IP address list of non−overlapping IP address ranges in form of: **from1−to1,from2−to2,...,fromN−toN**. For example, **10.0.0.1−10.0.0.27,10.0.0.32−10.0.0.47**

Example

To define a pool named **ip−pool** with the 10.0.0.1−10.0.0.125 address range excluding gateway's address 10.0.0.1 and server's address 10.0.0.100, and the other pool **dhcp−pool**, with the **10.0.0.200−10.0.0.250** address pool:

```
[admin@MikroTik] ip pool> add name=ip−pool ranges=10.0.0.2−10.0.0.99,10.0.0.101
10.0.0.126
[admin@MikroTik] ip pool> add name=dhcp−pool ranges=10.0.0.200−10.0.0.250
[admin@MikroTik] ip pool> print
  # NAME                                      RANGES
  0 ip−pool                                   10.0.0.2−10.0.0.99
                                              10.0.0.101−10.0.0.126
  1 dhcp−pool                                 10.0.0.200−10.0.0.250

[admin@MikroTik] ip pool>
```

Monitoring Used IP Addresses

Command name : **/ip pool used print**

Property Description

Statistics:

pool (name) − name of the pool, the address is given from
address (*IP address*) − IP address assigned to the client
owner (*string*) − application name, that has given the address out
info (*string*) − unique client identifier

Example

To see, what addresses are currently used:

```
[admin@MikroTik] ip pool> used print
 POOL                 ADDRESS         OWNER               INFO
 dhcp−pool            10.0.0.250      DHCP                00:e0:c5:6e:23:1d
[admin@MikroTik] ip pool>
```

IPsec

Document revision 1.5 (17−Jun−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

Specifications

Packages required : ***security***
License required : *Any*
Home menu level : ***/ip ipsec***
Protocols utilized : *IPsec (RFC2401)*
Hardware usage: *consumes a lot of CPU time (Intel Pentium MMX or AMD K6 suggested as minimal configuration)*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Firewall Filters and Network Address Translation (NAT)

Description

IPsec (IP Security) supports secure (encrypted) communications over IP networks.

Encryption

After packet is src−natted, but before putting it into interface queue, IPsec policy database is consulted to find out if packet should be encrypted. Security Policy Database (SPD) is a list of rules that have two parts:

- **Packet matching**: Packet source/destination, protocol and ports (for TCP and UDP) are compared to values in policy rules, one after another
- **Action**: If rule matches action specified in rule is performed:

accept − continue with packet as if there was no IPsec
drop − drop packet
encrypt − encrypt packet

Each SPD rule can be associated with several Security Associations (SA) that determine packet encryption parameters (key, algorithm, SPI).

Note that packet can only be encrypted if there is usable SA for policy rule. By setting SPD rule security "level" user can control what happens when there is no valid SA for policy rule:

- **use** − if there is no valid SA, send packet unencrypted (like **accept** rule)
- **acquire** − send packet unencrypted, but ask IKE daemon to establish new SA
- **require** − drop packet, and ask IKE daemon to establish new SA.

If packet can be encrypted, it is encrypted and sent as LOCALLY ORIGINATED packet − i.e. it is processed with "output" firewall, src−nat again and IPsec SPD again (this way one packet can be encrypted several times if encrypted packet has to be sent over encrypted tunnel itself). If packet matches the same SPD rule that it matched before, it is sent out without encrypting (to avoid encryption loops).

Decryption

When encrypted packet is received for local host (after dst−nat and **input** filter), appropriate SA to decrypt it is looked up (using packet source, destination, security protocol and SPI value). If no SA is found, packet is dropped. If SA is found, packet is decrypted. Then decrypted packets fields are compared to policy rule that SA is linked to. If packet does not match policy rule it is dropped. If packet is decrypted fine (or authenticated fine) it is "received once more" − it goes through dst−nat and routing (which finds out what to do − either forward or deliver locally) again.

Note that before **forward** and **input** firewall chains, packet that was not decrypted on local host is compared with SPD reversing its matching rules. If SPD requires encryption (there is valid SA associated with matching SPD rule), packet is dropped. This is called incoming policy check.

Internet Key Exchange

The Internet Key Exchange (IKE) is a protocol that provides authenticated keying material for Internet Security Association and Key Management Protocol (ISAKMP) framework. There are other key exchange schemes that work with ISAKMP, but IKE is the most widely used one. Together they provide means for authentication of hosts and automatic management of security associations (SA).

Most of the time IKE daemon is doing nothing. There are two possible situations when it is activated:

- Some traffic is caught by policy that needs to provide encryption or authentication, but doesn't have any SAs. It notifies IKE daemon about that, and IKE daemon initiates connection to remote host.
- IKE daemon responds to remote connection.

In both cases, peers establish connection and execute 2 phases:

- **Phase 1** − peers agree on algorithms they will use in following IKE messages, authenticate. Also, keying material (used to derive keys for all SAs and to protect following ISAKMP exchanges between hosts) is generated.
- **Phase 2** − peers establish one or several SAs that will be used by IPsec to encrypt data. All SAs established by IKE daemon will have lifetime values (either limiting time, after which SA will become invalid, or amount of data that can be encrypted by this SA, or both).

There are two lifetime values − soft and hard. When SA reaches it's soft lifetime, IKE daemon receives notice about it and starts another phase 2 exchange to replace this SA with fresh one. If SA reaches hard lifetime, it is discarded.

Perfect Forward Secrecy (PFS) that can optionally be provided by IKE is a property of key exchanges, which for IKE means that compromising the long term phase 1 key will not allow to easily gain access to all IPsec data that is protected by SAs established through this phase 1. It means an additional keying material is generated for each phase 2.

Generation of keying material is computationally very expensive. Use of modp8192 group can take several seconds even on very fast computer. It usually takes place once per phase 1 exchange, which happens only once between any host pair and then is kept for long time. PFS adds this expensive operation also to each phase 2 exchange.

Diffie−Hellman MODP Groups

Diffie−Hellman (DH) key exchange protocol allows two parties without any initial shared secret to create one. The following Modular Exponential (MODP) Diffie−Hellman (also known as "Oakley") Groups are supported:

Diffie−Hellman Group	Modulus	Reference
Group 1	**768 bits**	**RFC2409**
Group 2	**1024 bits**	**RFC2409**
Group 5	**1536 bits**	**RFC3526**
Group 14	**2048 bits**	**RFC3526**
Group 15	**3072 bits**	**RFC3526**
Group 16	**4096 bits**	**RFC3526**
Group 18	**8192 bits**	**RFC3526**

IKE Traffic

To avoid problems with IKE packets hit some SPD rule and require to encrypt it with not yet established SA (that this packet perhaps is trying to establish), locally originated packets with UDP source port 500 are not processed with SPD. The same way packets with UDP destination port 500 that are to be delivered locally are not processed in incoming policy check.

Setup Steps

To get IPsec to work with automatic keying you will have to configure **policy**, **peer** and **proposal** (optional) entries.
For manual keying you will have to configure **policy** and **manual−sa** entries.

Policy Settings

Submenu level :**/ip ipsec policy**

Description

Policy table is needed to determine if encryption should be applied to a packet.

Property Description

src−address (*IP address/mask:ports*; default: **0.0.0.0/32:any**) − source IP address
dst−address (*IP address/mask:ports*; default: **0.0.0.0/32:any**) − destination IP address
protocol (*name* | *integer*; default: **all**) − name or number of protocol
action (accept | drop | encrypt; default: **accept**) − what to do with packet that matches policy:

- **accept** − pass the packet
- **drop** − drop the packet
- **encrypt** − apply transormations specified by this policy and it's security

level (acquire | require | use; default: **require**) − what to do if some of the SAs for this policy cannot be found:

- **use** − skip this transform, don't drop packet, don't acquire SA from IKE daemon
- **acquire** − skip this transform, but acquire SA for it from IKE daemon
- **require** − drop packet, acquire SA

ipsec−protocols (*multiple choice*: ah , esp; default: **esp**) − specifies what combination of Authentication Header and Encapsulating Security Payload protocols you want to apply to matched traffic. AH is applied after ESP, and in case of tunnel mode ESP will be applied in tunnel mode and AH − in transport mode
tunnel (yes | no; default: **no**) − whether to use tunnel mode
sa−src−address (*IP address*; default: **0.0.0.0**) − SA source
sa−dst−address (*IP address*; default: **0.0.0.0**) − SA destination
proposal (*name*; default: **default**) − name of proposal info that will be sent by IKE daemon to establish SAs for this policy
manual−sa (*name*; default: **none**) − name of manual−sa template that will be used to create SAs for this policy

- **none** if you don't want to set up any manual keys

dont−fragment (clear | inherit | set; default: **clear**)− The state of the **Don't Fragment** IP header field:

- **clear** − clear (unset) the fiels, so that packets previously marked as **Don't Fragment**, got fragmented
- **inherit** − do not change the field
- **set** − set the fiels, so that each packet matching the rule, will not be fragmented

Statistics:

ph2−state (*string*) − progress of key establishing:

- **expired** − there are some leftovers from previous phase2, it is similar to **no−phase2**
- **no−phase2** − nothing has happened
- **established** − SAs are in place and everything should be working

Anything else falls between these last two states
in−accepted (*integer*) − how many incoming packets were passed through by policy without attempting decryption
in−dropped (*integer*) − how many incoming packets were dropped by policy without attempting decryption
out−accepted (*integer*) − how many outgoing packets were passed through by policy without encryption
out−dropped (*integer*) − how many outgoing packets were dropped by policy without attempting encryption
encrypted (*integer*) − how many outgoing packets were encrypted and passed on successfully
not−encrypted (*integer*) − how many outgoing packets policy attempted to encrypt, but discarded for any reason
decrypted (*integer*) − how many incoming packets policy decrypted and passed on successfully
not−decrypted (*integer*) − how many incoming packets policy tried to decrypt, but discarded for any reason

Notes

In tunnel mode all packets are IPIP encapsulated, and their new IP header src and dst are set to sa−src and sa−dst values of this policy. If you don't use tunnel mode (i.e. you use transport mode), then only packets whose source and destination is the same as sa−src and sa−dst can be processed by this policy. Transport mode can only work with packets that originate at and are destined for IPsec peers (hosts that established security associations). To encrypt traffic between networks (or network and host) you have to use tunnel mode.

It is good to have **dont−fragment** cleared because encrypted packets are always bigger than original and thus they may need fragmentation.

If you are using IKE to establish SAs automatically, then policies on both routers must be exactly matching, i.e. **src−address=1.2.3.0/27** on one router and **dst−address=1.2.3.0/28** on another won't work. **src** values on one router MUST be equal to **dst** values on the other one, and vice versa.

Example

To add policy to encrypt all the traffic between us (**10.0.0.147**) and **10.0.0.148** host:

```
[admin@MikroTik] ip ipsec policy> add sa−src−address=10.0.0.147 \
\... sa−dst−address=10.0.0.148 action=encrypt
[admin@MikroTik] ip ipsec policy> print
Flags: X − disabled, I − invalid
  0   src−address=10.0.0.147/32:any dst−address=10.0.0.148/32:any
      protocol=all action=encrypt level=require ipsec−protocols=esp tunnel=no
      sa−src−address=10.0.0.147 sa−dst−address=10.0.0.148 proposal=default
      manual−sa=none dont−fragment=clear


[admin@MikroTik] ip ipsec policy>
```

To view the policy statistics:

```
[admin@MikroTik] ip ipsec policy> print stats
Flags: X − disabled, I − invalid
  0   src−address=10.0.0.147/32:any dst−address=10.0.0.148/32:any
      protocol=all ph2−state=no−phase2 in−accepted=0 in−dropped=0
      out−accepted=0 out−dropped=0 encrypted=0 not−encrypted=0 decrypted=0
      not−decrypted=0


[admin@MikroTik] ip ipsec policy>
```

Peer

Submenu level : **/ip ipsec peer**

Description

Peer configuration settings are used to establish connections between IKE daemons (phase 1 configuration). This connection then will be used to negotiate keys and algorithms for SAs.

Property Description

address (*IP address/mask:port*; default: **0.0.0.0/32:500**) − address prefix. If remote peer's address matches this prefix, then this peer configuration is used while authenticating and establishing phase 1. If several peer's address matches several configuration entries, the most specific one (i.e. the one with largest netmask) will be used
secret (*string*; default: "") − secret string. If it starts with '0x', it is parsed as a hexadecimal value
generate−policy (yes | no; default: **no**) − allow this peer to establish SA for non−existing policies. Such policies are created dynamically for the lifetime of SA. This way it is possible, for example, to create IPsec secured l2tp tunnels, or any other setup where remote peer's IP address is not known at configuration time
exchange−mode (main | aggressive | base; default: **main**) − see RFC 2408 for an overview of ISAKMP phase 1 exchange modes. Currently only **main** mode is tested
send−initial−contact (yes | no; default: **yes**) − **yes**
proposal−check (claim | exact | obey | strict; default: **strict**) − phase 2 lifetime check logic:

- **claim** − take shortest of proposed and configured lifetimes, notify initiator about it
- **exact** − lifetimes must be the same
- **obey** − accept whatever is sent by initiator
- **strict** − If initiator proposes longer lifetime than default, reject proposal, otherwise accept proposed lifetimes

hash−algorithm (md5 | sha; default: **md5**) − hashing algorithm. SHA (Secure Hash Algorithm) is stronger, but slower
enc−algorithm (des | 3des | aes−128 | aes−192 | aes−256; default: **3des**) − encryption algorithm. Algorithms are named in strength increasing order
dh−group (modp768 | modp1024 | modp1536 | modp2048 | modp3072 | modp4096 | modp8192; default: **modp1024**) − Diffie−Hellman MODP group (cipher strength). First three allowed strengths (768, 1024 and 1536) are standard, others might be incompatible with similarly named groups in other implementations
lifetime (*integer*; default: **1d**) − phase 1 lifetime: how long the SA is valid; it is discarded after this time
lifebytes (*integer*; default: **0**) − phase 1 lifetime: how much bytes can be transferred before SA is discarded

- **0** − SA won't expire based on byte count

Notes

AES (Advanced Encryption Standard) encryption algorithms are much faster than DES, so it is recommended to use this algorithm class whenever possible. But, AES's speed is also its drawback as it potentially can be cracked faster, so use AES−256 when you need security and AES−128 when speed is also important.

Both peers MUST have the same encryption and authentication algorithms, DH group and exchange mode. Some legacy hardware may support only DES and MD5.

You should only set **generate−policy** flag to **yes** for trusted peers, because there is no verification done for the established policy. To protect yourself against possible unwanted events, add poilcies with **action=accept** for all networks you don't want to be encrypted at the top of policy list. Since dynamic policies are added at the bottom of the list, they will not be able to override your configuration.

Example

To define new peer configuration for **10.0.0.147** peer with secret = **gwejimezyfopmekun**:

```
[admin@MikroTik] ip ipsec peer> add address=10.0.0.147/32 secret=gwejimezyfopmekun
[admin@MikroTik] ip ipsec peer> print
```

```
Flags: X - disabled
  0   address=10.0.0.147/32:500 secret="gwejimezyfopmekun" generate-policy=no
      exchange-mode=main send-initial-contact=yes proposal-check=strict
      hash-algorithm=md5 enc-algorithm=3des dh-group=modp1024 lifetime=1d
      lifebytes=0


[admin@MikroTik] ip ipsec peer>
```

Remote Peer Statistics

Submenu level : **/ip ipsec remote-peers**

Description

You can see various statistics about remote peers that curently have phase 1 established with this router. Note that if peer doesn't show up here, it doesn't mean that no IPsec traffic is being exchanged with it. For example, manual SA configurations will not show up here.

Property Description

Statistics:

local-address (*IP address*) - local ISAKMP SA address
remote-address (*IP address*) - remote address of the peer
state (*string*) - state of phase 1 negotiation with this peer

- **established** is the normal working state

side (initiator | responder) - who spoke first:

- **initiator** - phase 1 negotiation was started by this router
- **responder** - phase 1 negotiation was started by peer

established (*sting*) - data and time when phase 1 was established with this peer
ph2-active (*integer*) - how many phase 2 negotiations with this peer are currently taking place
ph2-total (*integer*) - how many phase 2 negotiations with this peer took place

Example

To see currently established SA:

```
[admin@MikroTik] ip ipsec> remote-peers print
  0 local-address=10.0.0.148 remote-address=10.0.0.147 state=established
    side=initiator established=jan/25/2003 03:34:45 ph2-active=0 ph2-total=1


[admin@MikroTik] ip ipsec>
```

Manual SA

Submenu level : **ip ipsec manual-sa**

Property Description

name (*name*; default: **sa1**) − name of item for reference from policies
ah−algorithm (null | md5 | sha1; default: **null**) − Authentication Header encryption algorithm, one of the following:

- **md5** − 128 bit key
- **null** − any key length
- **sha1** − 160 bit key

esp−auth−algorithm (null | md5 | sha1; default: **null**) − Encapsulating Security Payload authentication encryption algorithm, one of the following:

- **md5** − 128 bit key
- **null** − any key length
- **sha1** − 160 bit key

esp−enc−algorithm (null | des | 3des | aes−128 | aes−192 | aes−256; default: **null**) − Encapsulating Security Payload encryption algorithm, one of the following:
ah−key (*string*; default: "") − incoming−authentication−key/outgoing−authentication−key (even−length hexadecimal string)
esp−auth−key (*string*; default: "") − incoming−authentication−key/outgoing−authentication−key (even−length hexadecimal string)
esp−enc−key (*string*; default: "") − incoming−encryption−key/outgoing−encryption−key (even−length hexadecimal string)
ah−spi (*integer* > **255**; default: **0x100**) − incoming−SA−SPI/outgoing−SA−SPI, in hexadecimal. May be equal − in this case only one SPI number is printed
esp−spi (*integer* > **255**; default: **0x100**) − incoming−SA−SPI/outgoing−SA−SPI, in hexadecimal. May be equal − in this case only one SPI number is printed

Notes

Note that incoming SPI numbers on one router must match outgoing SPI numbers on another, and vice versa. Same for keys.

You can reference same manual−sa template from several policies, because actual SAs are inserted based on info in policies (AH, ESP) as well as in this template, as well as in key config. Also, each SA is distinguished by its source (sa−src), destination (sa−dst), protocol (AH or ESP), SPI and direction.

Example

To add manual−sa entry and specify its incomming AH key is **A0** and outgoing AH key is **0A**:

```
[admin@MikroTik] ip ipsec manual−sa> add ah−key=A0/0A
[admin@MikroTik] ip ipsec manual−sa> print
Flags: X − disabled, I − invalid
  0   name="sa1" ah−algorithm=null esp−auth−algorithm=null
      esp−enc−algorithm=null ah−key=A0/0A esp−auth−key="" esp−enc−key=""
      ah−spi=100 esp−spi=100


[admin@MikroTik] ip ipsec manual−sa>
```

Proposal

Submenu level : **/ip ipsec proposal**

Description

Proposal is used to set which algorithms may be used on the actual traffic (phase 2 configuration). It also defines if PFS should be used.

Property Description

name (*name*; default: **proposal1**) − name of proposal for referencing it from policy
auth−algorithms (*multiple choice*: md5, sha1, null; default: **sha1**) − allowed algorithms for authorization:

- **md5** − 128 bit key
- **null** − any key length
- **sha1** − 160 bit key

enc−algorithms (*multiple choice*:des, 3des, aes−128, aes−192, aes−256, null; default: **3des**) − allowed algorithms and key lengths to use for SAs that will be acquired from IKE daemon by policy that references this proposal
lifetime (*time*) − how long to use SA before throwing it out
lifebytes (*integer*; default: **0**) − how many bytes to encrypt using SA before throwing it out and making new one

- **0** − SA won't expire based on byte count

pfs−group (none | modp768 | modp1024 | modp1536 | modp2048 | modp3072 | modp4096 | modp8192; default: **modp1024**) − Diffie−Hellman MODP group (cipher strength) for PFS. First four allowed strengths (none, 768, 1024 and 1536) are standard, others might be incompatible with similarly named groups in other implementations

- **none** − PFS will not be used

Notes

Proposals on both peers must (at least partially) match. The more they match the better.

There is the default proposal already.

Example

To set the **default** proposal to use DES and AES−128 encryption algorithms:

```
[admin@MikroTik] ip ipsec proposal> set default enc−algorithms=des,aes−128
[admin@MikroTik] ip ipsec proposal> print
Flags: X − disabled
  0   name="default" auth−algorithms=sha1 enc−algorithms=des,aes−128
      lifetime=30m lifebytes=0 pfs−group=modp1024


[admin@MikroTik] ip ipsec proposal>
```

Installed SA

Submenu level : **/ip ipsec installed−sa**

Description

Prints a lot of information about each installed SA (including keys)

Property Description

Statistics:

spi (*integer*) − SPI value of SA, in hexadecimal
direction (in | out) − SA direction
src−address (*IP address*) − source of SA from policy configuration
dst−address (*IP address*) − destination of SA from policy configuration
auth−algorithm (none | md5 | sha1) − authentication algorithm
enc−algorithm (none | des | 3des | aes) − encryption algorithm
replay (*integer*) − size of replay window, in bytes
state (larval | mature | dying | dead) − period of SA's life
auth−key (*string*) − authentication key, as hex string
enc−key (*string*) − encryption key, as hex string (only used by ESP SAs)
add−lifetime (*time*/*time*) − soft/hard expiration time counted from installation of SA
use−lifetime (*time*/*time*) − soft/hard expiration time counter from the first use of SA
lifebytes *integer*/*integer*) − soft/hard expiration threshold for amount of processed data
current−addtime (*string*) − time when this SA was installed
current−usetime (*string*) − time when this SA was first used
current−bytes (*integer*) − amount of data processed by this SA's crypto algorithms

Example

```
[admin@MikroTik] ip ipsec> installed−sa print
Flags: A − AH, E − ESP, P − pfs, M − manual
  0 E   spi=E727605 direction=in src−address=10.0.0.148
        dst−address=10.0.0.147 auth−algorithm=sha1 enc−algorithm=3des
        replay=4 state=mature
        auth−key="ecc5f4aee1b297739ec88e324d7cfb8594aa6c35"
        enc−key="d6943b8ea582582e449bde085c9471ab0b209783c9eb4bbd"
        add−lifetime=24m/30m use−lifetime=0s/0s lifebytes=0/0
        current−addtime=jan/28/2003 20:55:12
        current−usetime=jan/28/2003 20:55:23 current−bytes=128

  1 E   spi=E15CEE06 direction=out src−address=10.0.0.147
        dst−address=10.0.0.148 auth−algorithm=sha1 enc−algorithm=3des
        replay=4 state=mature
        auth−key="8ac9dc7ecebfed9cd1030ae3b07b32e8e5cb98af"
        enc−key="8a8073a7afd0f74518c10438a0023e64cc660ed69845ca3c"
        add−lifetime=24m/30m use−lifetime=0s/0s lifebytes=0/0
        current−addtime=jan/28/2003 20:55:12
        current−usetime=jan/28/2003 20:55:12 current−bytes=512


[admin@MikroTik] ip ipsec>
```

Flushing Installed SA table

Command name : **/ip ipsec installed−sa flush**

Description

In some cases when incorrect/incomplete negotiations took place, it is required to manually flush the installed SA table so that SA could be renegotiated.

Property Description

sa−type (ah | all | esp; default: **all**) − which SA types to flush:

- **ah** − delete AH protocol SAs only
- **esp** − delete ESP protocol SAs only
- **all** − delete SAs of both AH and ESP protocols

Example

To flush all the SAs installed:

```
[admin@MikroTik] ip ipsec installed−sa> flush
[admin@MikroTik] ip ipsec installed−sa> print

[admin@MikroTik] ip ipsec installed−sa>
```

Counters

Submenu level : **/ip ipsec counters**

Property Description

Statistics:

out−accept (*integer*) − how many outgoing packets were matched by **accept** policy (including the default "accept all" case)
out−accept−isakmp (*integer*) − how many locally originated UDP packets on source port 500 (which is how ISAKMP packets look) were let through without policy matching
out−drop (*integer*) − how many outgoing packets were matched by **drop** policy (or **encrypt** policy with level=require that doesn't have all SAs)
out−encrypt (*integer*) − how many outgoing packets were encrypted successfully
in−accept (*integer*) − how many incoming packets were matched by **accept** policy
in−accept−isakmp (*integer*) − how many incoming UDP packets on port 500 were let through without policy matching
in−drop (*integer*) − how many incoming packets matched **drop** policy (or **encrypt** policy with level=require that didn't have all SAs)
in−decrypted (*integer*) − how many incoming packets were successfully decrypted
in−drop−encrypted−expected (*integer*) − how many incoming packets were matched by **encrypt** policy and dropped because they were not encrypted

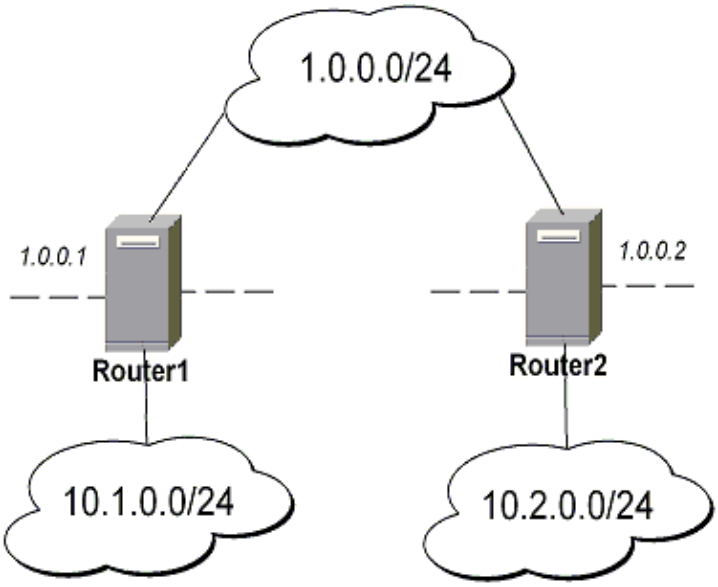
Example

To view current statistics:

```
[admin@MikroTik] ip ipsec> counters print
                     out-accept: 6
              out-accept-isakmp: 0
                       out-drop: 0
                    out-encrypt: 7
                      in-accept: 12
               in-accept-isakmp: 0
                        in-drop: 0
                    in-decrypted: 7
    in-drop-encrypted-expected: 0
[admin@MikroTik] ip ipsec>
```

Application examples

IPsec setup between two RourerOS routers



Minimal config example for transport mode ESP with automatic keying on Router 1:

```
ip ipsec policy add sa-src="IP/1.0.0.1 sa-dst=1.0.0.2 action=encrypt
"ip ipsec peer add address=1.0.0.2 secret="roberkenon"
```

And for Router 2:

```
ip ipsec policy add sa-src="IP/1.0.0.2 sa-dst=1.0.0.1 action=encrypt
"ip ipsec peer add address=1.0.0.1 secret="roberkenon"
```

Minimal config example for transport mode ESP with automatic keying and automatic policy generating on Router 1:

```
ip ipsec peer add address=1.0.0.0/24 secret="roberkenon" generate-policy=yes
```

And with static policy on Router 2:

```
ip ipsec policy add sa-src="IP/1.0.0.2 sa-dst=1.0.0.1 action=encrypt
"ip ipsec peer add address=1.0.0.1 secret="roberkenon"
```

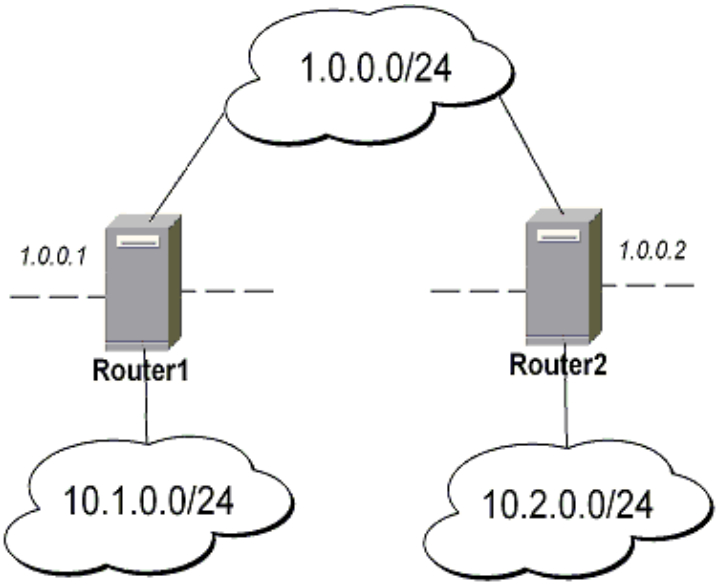
Minimal config example for tunnel mode AH with manual keying on Router 1:

```
ip ipsec manual-sa add name=ah-sa1 ah-spi=0x101/0x100 ah-key=abcfed
ip ipsec policy add src-address=10.1.0.0/24 dst-address=10.2.0.0/24 \
   action=encrypt ipsec-protocols=ah tunnel=yes sa-src="IP/1.0.0.1 sa-dst=1.0.0.2 \
"   manual-sa=ah-sa1
```

And for Router 2:

```
ip ipsec manual-sa add name=ah-sa1 ah-spi=0x100/0x101 ah-key=abcfed
ip ipsec policy add src-address=10.2.0.0/24 dst-address=10.1.0.0/24 \
   action=encrypt ipsec-protocols=ah tunnel=yes sa-src="IP/1.0.0.2 sa-dst=1.0.0.1 \
"   manual-sa=ah-sa1
```

IPsec Setup for Routing Between two Masquerading MikroTik Routers



On Router1:

- Add accept and masquerading rules in SRC-NAT:

```
/ip firewall src-nat add src-address=10.1.0.0/24 dst-address=10.2.0.0/24
/ip firewall src-nat add out-interface=public action=masq
```

- Configure IPsec:

```
/ip ipsec policy add src-address=10.1.0.0/24 dst-address=10.2.0.0/24 \
    action=encrypt tunnel=yes sa-src-address=1.0.0.1 sa-dst-address=1.0.0.2
/ip ipsec peer add address=1.0.0.2 exchange-mode=aggressive secret="sviestapika"
```

On Router2:

- Add accept and masquerading rules in SRC-NAT:

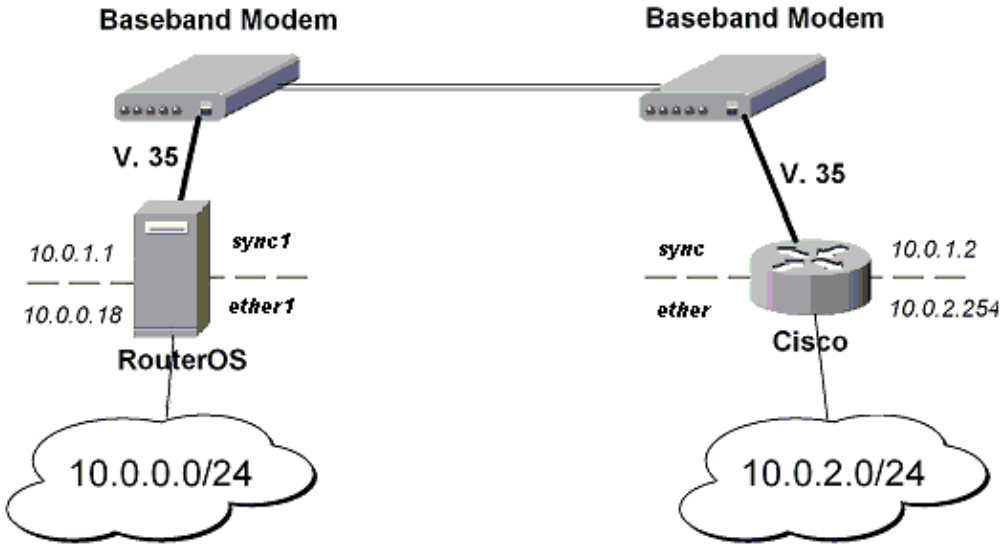
```
/ip firewall src-nat add src-address=10.2.0.0/24 dst-address=10.1.0.0/24
```

```
/ip firewall src-nat add out-interface=public action=masq
```

- Configure IPsec:

```
/ip ipsec policy add src-address=10.2.0.0/24 dst-address=10.1.0.0/24 \
action=encrypt tunnel=yes sa-src-address=1.0.0.2 sa-dst-address=1.0.0.1
/ip ipsec peer add address=1.0.0.1 exchange-mode=aggressive secret="sviestapika"
```

IPsec Setup Between MikroTik and CISCO Routers



Must configure IPsec encryption for traffic between 10.0.0.0/24 and 10.0.2.0/24 subnets.

Configuring RouterOS

Set encryption proposal (phase2 proposal − settings that will be used to encrypt actual data) to use DES to encrypt data:

```
[admin@MikroTik] ip ipsec proposal> set default enc-algorithms=des
```

Add peer (with phase1 configuration parameters), DES and SHA1 will be used to protect IKE traffic:

```
[admin@MikroTik] ip ipsec peer> add address=10.0.1.2 secret=test_key \
\... enc-algorithm=des
```

Add policy rule that matches traffic between subnets and requires encryption with ESP in tunnel mode:

```
[admin@MikroTik] ip ipsec policy> add src-address=10.0.0.0/24 \
\... dst-address=10.0.2.0/24 action=encrypt tunnel=yes sa-src="IP/10.0.1.1 \
"\... sa-dst=10.0.1.2
```

Configuring Cisco

Parts from Cisco configuration with comments follow...

```
configure terminal

! Configure ISAKMP policy (phase1 config, must match configuration
! of "/ip ipsec peer" on RouterOS). Note that DES is default (and only)
```

```
! encryption algorithm on this Cisco. SHA1 is default authentication
! algorithm
crypto isakmp policy 9
  encryption des
  group 2
  hash md5
  exit

! Add preshared key to be used when talking to RouterOS
crypto isakmp key mykey address 10.0.1.1 255.255.255.255

! Create IPsec transform set − transformations that should be applied to
! traffic − ESP encryption with DES and ESP authentication with SHA1
! This must match "/ip ipsec proposal"
crypto ipsec transform−set myset esp−des esp−sha−hmac
  mode tunnel
  exit

! Create access list that matches traffic that should be encrypted
access−list 101 permit ip 10.0.2.0 0.0.0.255 10.0.0.0 0.0.0.255

! Create crypto map that will use transform set "myset", use peer 10.0.1.1
! to establish SAs and encapsulate traffic and use access−list 101 to
! match traffic that should be encrypted
crypto map mymap 10 ipsec−isakmp
  set peer 10.0.1.1
  set transform−set myset
  set pfs group2
  match address 101
  exit

! And finally apply crypto map to serial interface:
interface Serial 0
  crypto map mymap
  exit
```

Testing

After this simply ping from some host in one network to some host in other network − after some time (~10sec) replies should start coming back because SAs are established and data is being encrypted.

On RouterOS we can see installed SAs:

```
[admin@MikroTik] ip ipsec installed−sa> print
Flags: A − AH, E − ESP, P − pfs, M − manual
  0 E   spi=9437482 direction=out src−address=10.0.1.1
        dst−address=10.0.1.2 auth−algorithm=sha1 enc−algorithm=des
        replay=4 state=mature
        auth−key="9cf2123b8b5add950e3e67b9eac79421d406aa09"
        enc−key="ffe7ec65b7a385c3" add−lifetime=24m/30m use−lifetime=0s/0s
        lifebytes=0/0 current−addtime=jul/12/2002 16:13:21
        current−usetime=jul/12/2002 16:13:21 current−bytes=71896

  1 E   spi=319317260 direction=in src−address=10.0.1.2
        dst−address=10.0.1.1 auth−algorithm=sha1 enc−algorithm=des
        replay=4 state=mature
        auth−key="7575f5624914dd312839694db2622a318030bc3b"
        enc−key="633593f809c9d6af" add−lifetime=24m/30m use−lifetime=0s/0s
        lifebytes=0/0 current−addtime=jul/12/2002 16:13:21
        current−usetime=jul/12/2002 16:13:21 current−bytes=0
```

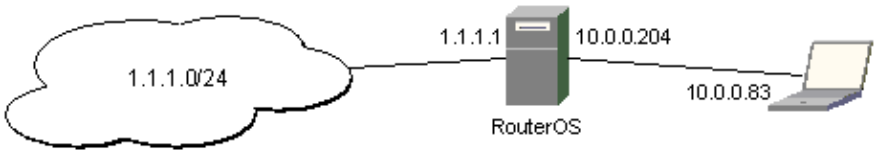
```
[admin@MikroTik] ip ipsec installed-sa>
```

And on Cisco:

```
interface: Serial1
    Crypto map tag: mymap, local addr. 10.0.1.2

   local  ident (addr/mask/prot/port): (10.0.2.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.0/255.255.255.0/0/0)
   current_peer: 10.0.1.1
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 1810, #pkts encrypt: 1810, #pkts digest 1810
    #pkts decaps: 1861, #pkts decrypt: 1861, #pkts verify 1861
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.0.1.2, remote crypto endpt.: 10.0.1.1
     path mtu 1500, media mtu 1500
     current outbound spi: 1308650C

     inbound esp sas:
      spi: 0x90012A(9437482)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2000, flow_id: 1, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4607891/1034)
        IV size: 8 bytes
        replay detection support: Y

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x1308650C(319317260)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2001, flow_id: 2, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4607893/1034)
        IV size: 8 bytes
        replay detection support: Y

     outbound ah sas:

     outbound pcp sas:
```

IPsec setup between RouterOS router and Windows SonicWall Client

IPSec setup of RouterOS router as a Security Gateway for SonicWALL VPN client



Configuring remote access of 1.1.1.0 network through 10.0.0.204 RouterOS router

Configuring RouterOS

1. Add peer configuration. Use Triple−DES and SHA−1 algorithms to protect phase 1 traffic. Set "proposal−check" to "obey" to allow remote client to connect even if lifetime and pfs settings in its proposal don't match ours.

```
/ ip ipsec peer add address=10.0.0.81:500 exchange-mode=main \
send-initial-contact=no proposal-check=obey hash-algorithm=sha \
enc-algorithm=3des dh-group=modp1024 secret="********"
```

2. Add encryption proposal. Use MD5, DES and Diffie−Hellman Group 1 for Perfect Forward Secrecy.

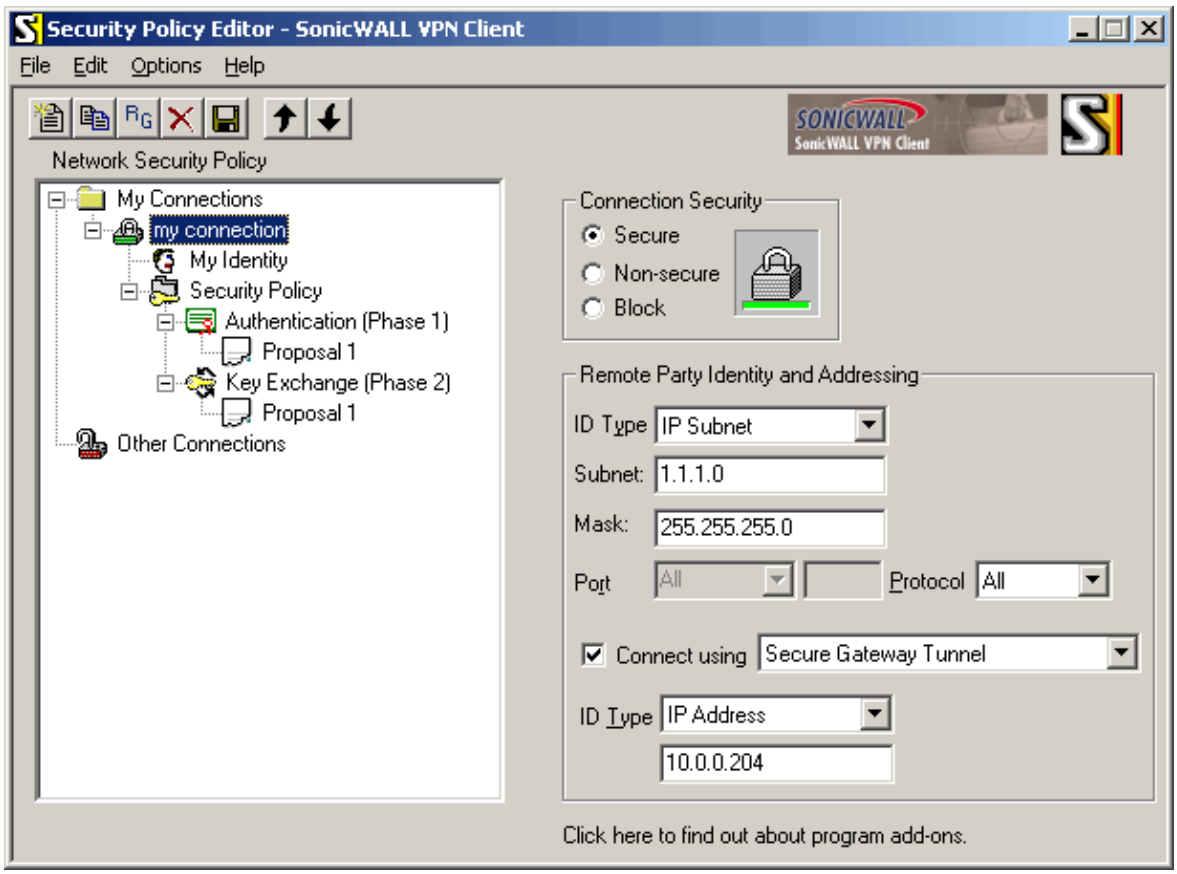
```
/ ip ipsec proposal add name=sw-client auth-algorithms=md5 enc-algorithms=des \
lifetime=30m pfs-group=modp768
```

3. Add policy rule that matches traffic between remote client and 1.1.1.0/24 network, use ESP in tunnel mode to encript all data.

```
/ ip ipsec policy add src-address=1.1.1.0/24 dst-address=10.0.0.81/32 \
action=encrypt ipsec-protocols=esp tunnel=yes sa-src-address=10.0.0.204 \
sa-dst-address=10.0.0.81 proposal=sw-client
```

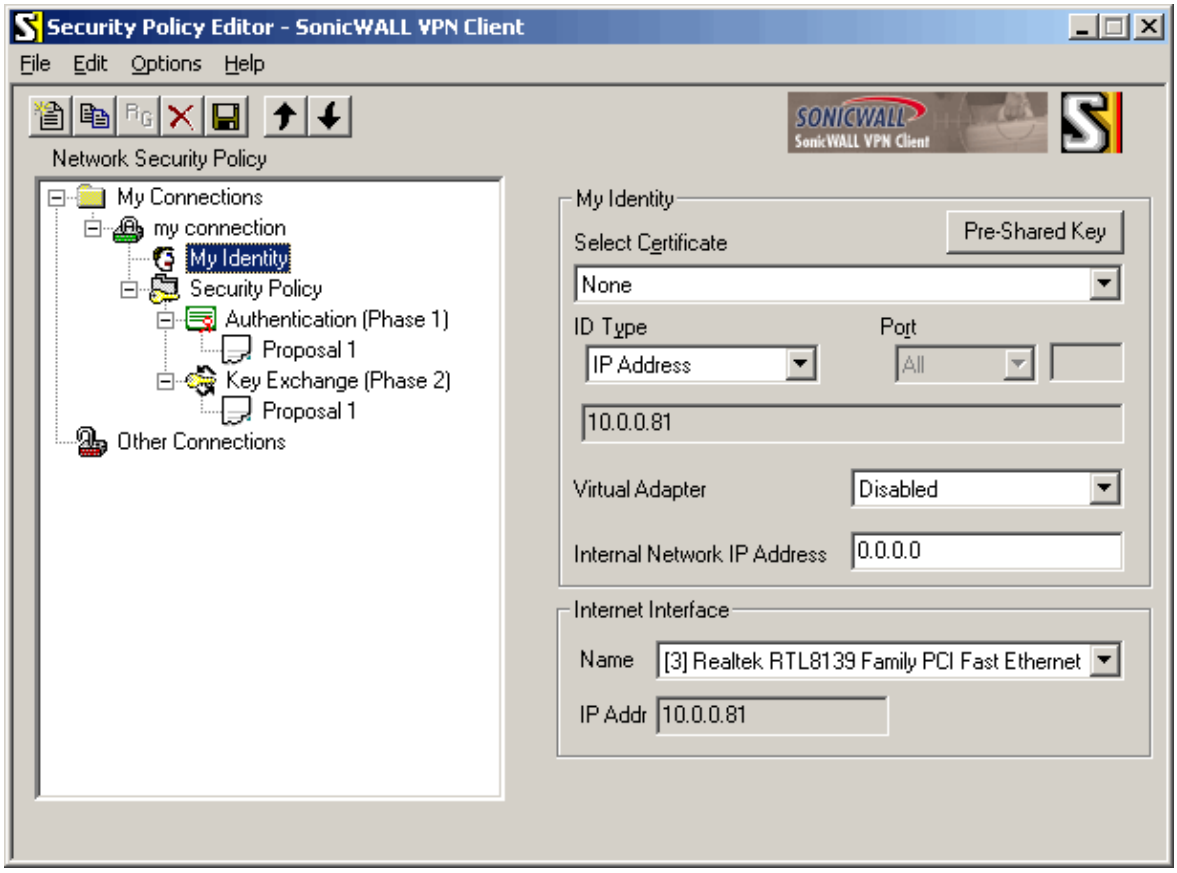
Configuring SonicWALL

Here you create IPSec policy that should match all traffic between 10.0.0.81 host and 1.1.1.0/24 network. You also specify the address of remote IPSec peer:

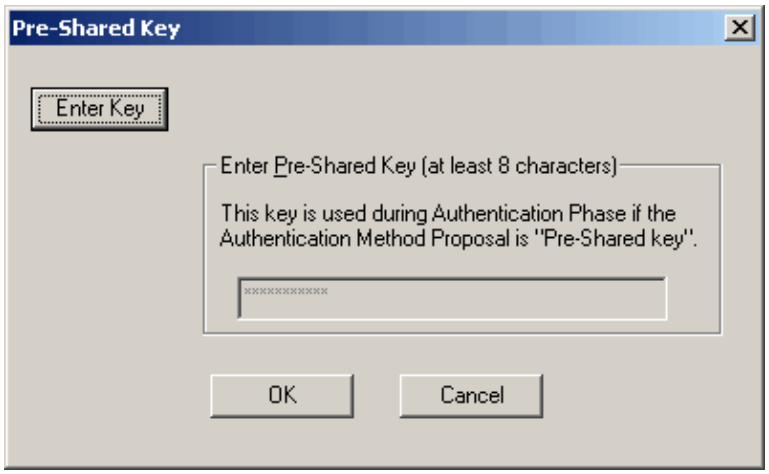


```
"Connection Security": select "Secure"
in "Remote Party Identity And Addressing" box:
  "ID Type": select "IP Subnet"
  "Subnet": enter "1.1.1.0"
  "Mask": enter "255.255.255.0"
  check "Connect using", select "Secure Gateway Tunnel"
  "ID Type": select "IP Address", enter below "10.0.0.204"
```

Configure pre−shared key, select correct interface to connect to 10.0.0.204 router with the proper address 10.0.0.81:

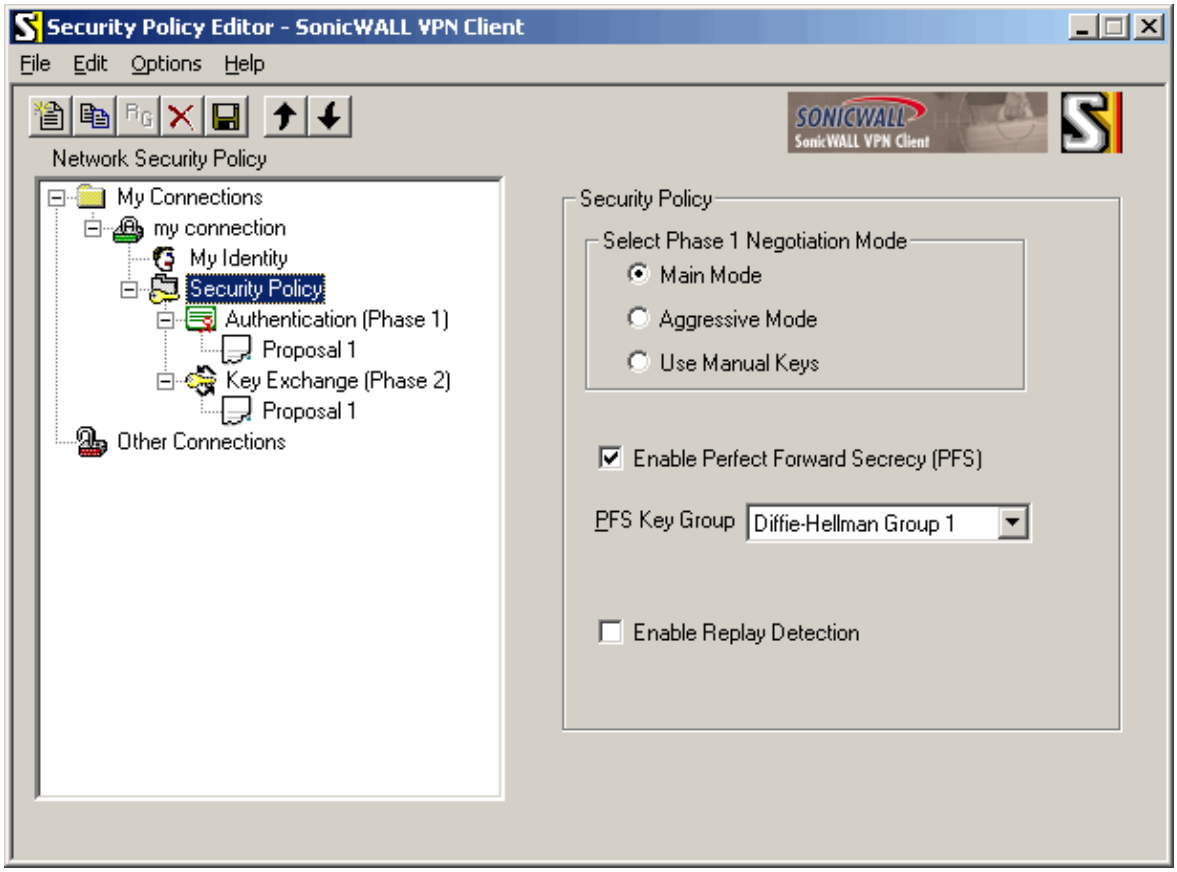



```
in "My Identity" box:
  "Select Certificate": select "None"
  click "Pre−Shared Key"

"Pre−Shared Key" pops up:
```

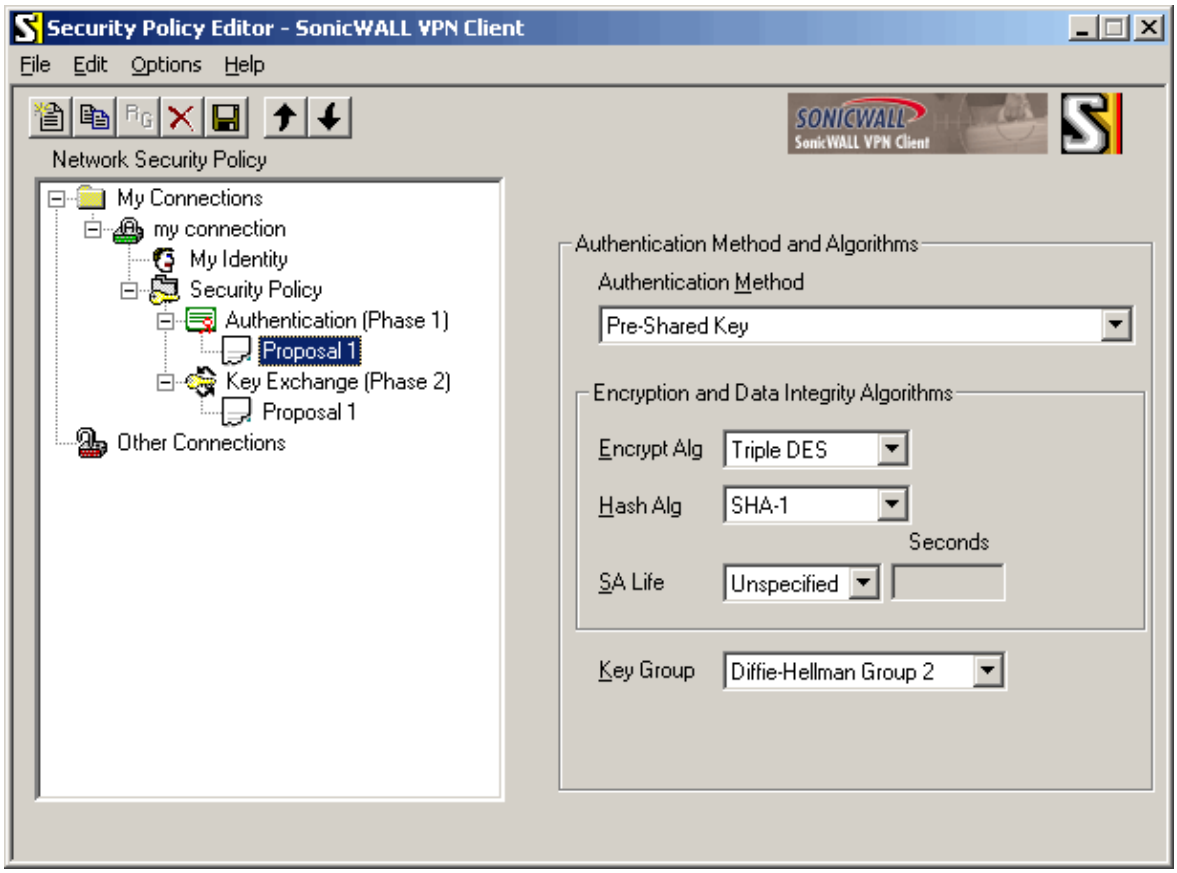



```
    click "Enter Key"
    type ********, click "OK"

in "Internet Interface" box:
  "Name": select interface that is connected to 10.0.0.0/24 network
  "IP Addr": check that it shows 10.0.0.81
```

Configure phase 1 setting to use same algorithms as on RouterOS side:

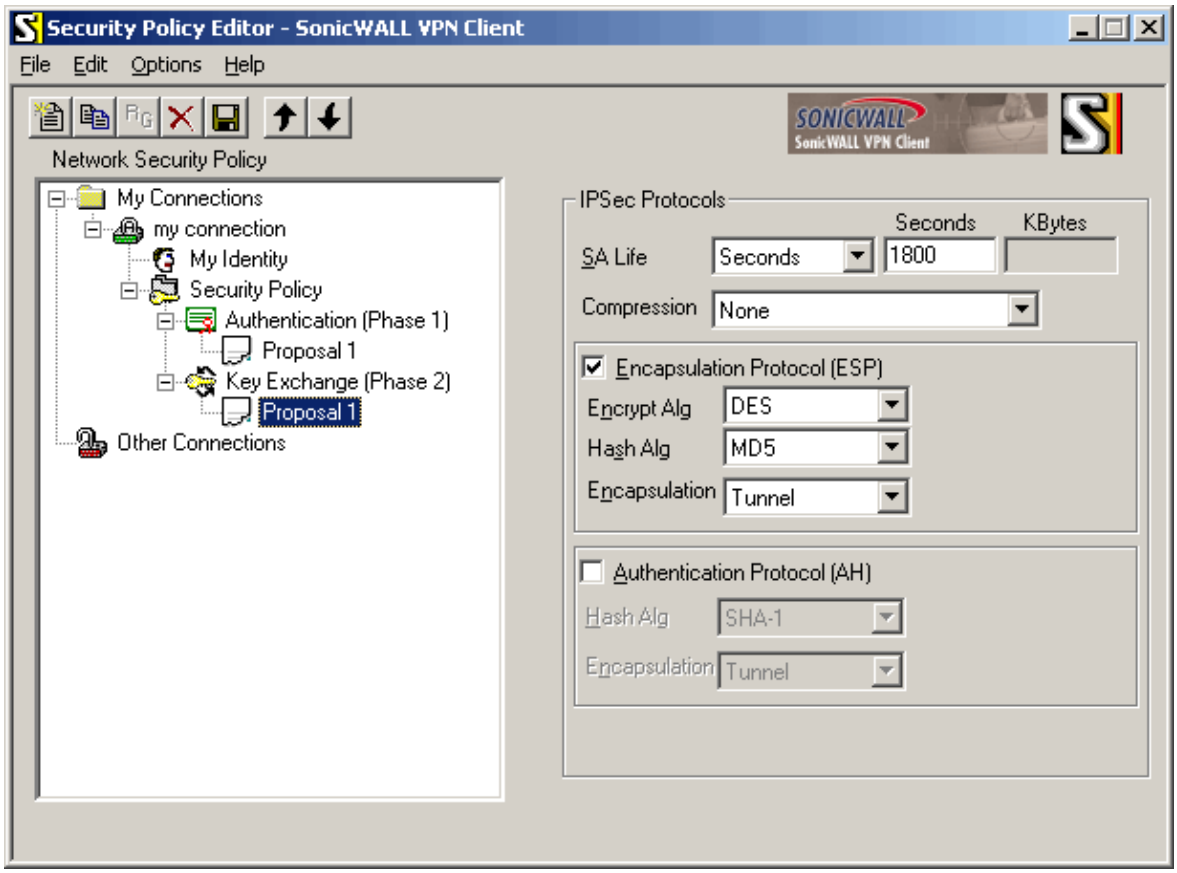



```
"Select Phase 1 Negotiation Mode": select "Main Mode"
check "Enable Perfect Forward Secrecy (PFS)"
"PFS Key Group": select "Diffie-Hellman Group 1"
clear "Enable Replay Detection"
```




```
"Authentication Method": select "Pre-Shared Key"
in "Encryption and Data Integrity Algorithms" box:
  "Encrypt Alg": select "Triple DES"
  "Hash Alg": select "SHA-1"
  "SA Life": select "Unspecified"

"Key Group": select "Diffie-Hellman Group 2" (this is called "modp1024"
in RouterOS)
```

Configure phase 2 settings:




```
in "IPSec Protocols" box:
  "SA Life": select "Seconds", enter "1800" in "Seconds" field
  "Compression": select "None"
  check "Encapsulation Protocol (ESP)"
    "Encrypt Alg": select "DES"
    "Hash Alg": select "MD5"
    "Encapsulation": select "Tunnel"
  clear "Authentication Protocol (AH)"

click "Save" (on the toolbar)
```

Testing

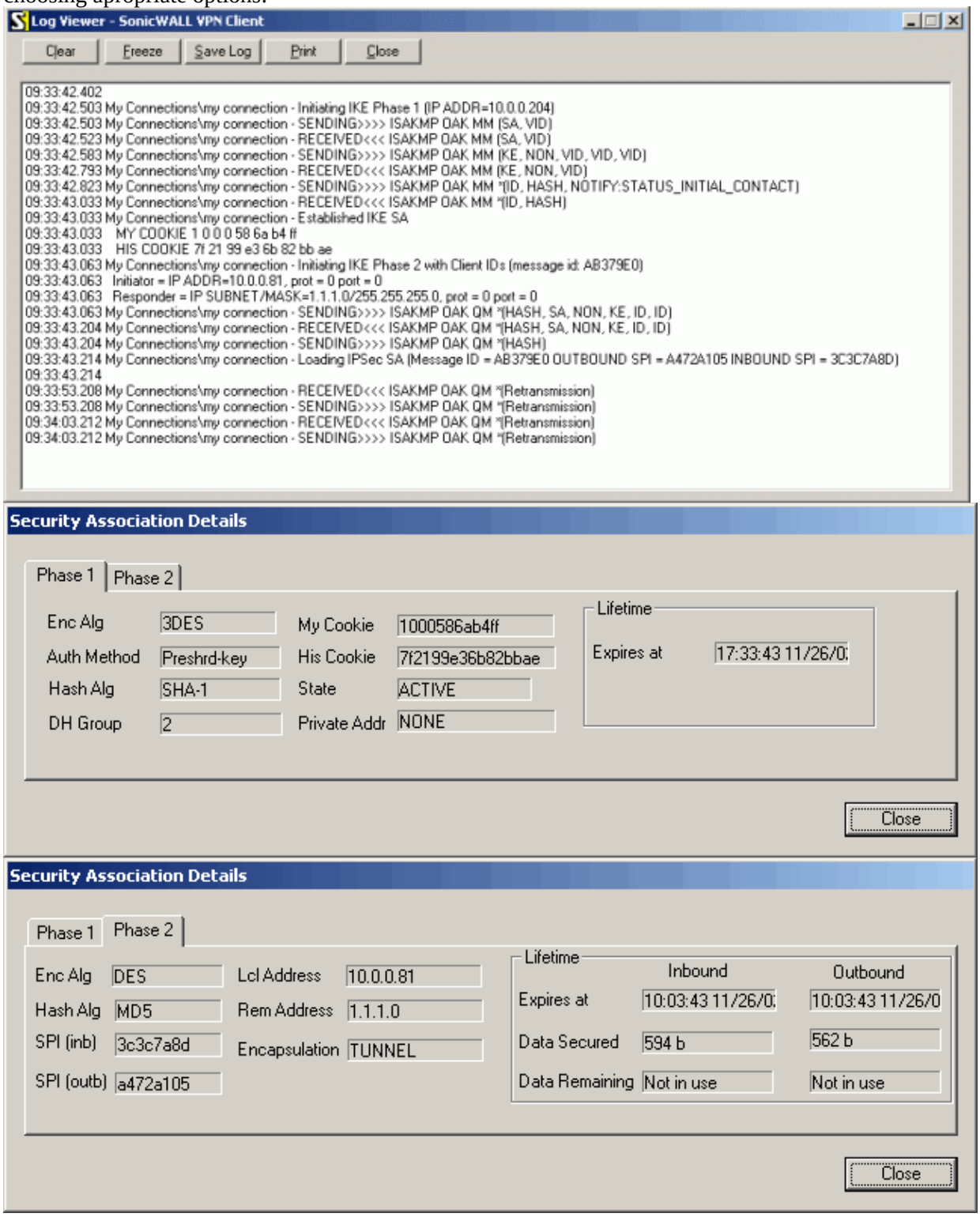
Try accessing some host on 1.1.1.0/24 network from 10.0.0.81 box. After some time IPSec tunnel will be established and data will start to pass through.

On RouterOS side you can see the statistics for established SAs:

```
[admin@xxx] ip ipsec installed-sa> print
Flags: A - AH, E - ESP, P - pfs, M - manual
  0 E   spi=3C3C7A8D direction=out src-address=10.0.0.204
        dst-address=10.0.0.81 auth-algorithm=md5 enc-algorithm=des replay=4
        state=mature auth-key="5697ee9fe98867005ac057e1b62a6c3b"
        enc-key="7b992840ea30b180" add-lifetime=24m/30m use-lifetime=0s/0s
        lifebytes=0/0 current-addtime=nov/26/2002 09:33:47
        current-usetime=nov/26/2002 09:33:53 current-bytes=896

  1 E   spi=A472A105 direction=in src-address=10.0.0.81
        dst-address=10.0.0.204 auth-algorithm=md5 enc-algorithm=des replay=4
```

```
state=mature auth-key="70655b51846308f68ce964d90b5580cd"
enc-key="a3623a16f6bef13d" add-lifetime=24m/30m use-lifetime=0s/0s
lifebytes=0/0 current-addtime=nov/26/2002 09:33:47
current-usetime=nov/26/2002 09:33:53 current-bytes=0
```

On SonicWall side you can view logs and connection statistics by right-clicking SonicWALL tray icon and choosing apropriate options:



Additional Resources

How to Configure a L2TP/IPSec Connection Using Pre−shared Key Authentication

IP Telephony

Document revision 1.5 (11−Aug−2003)
This document applies to the MikroTik RouterOS V2.7

Table Of Contents

Summary

The MikroTik RouterOS IP Telephony feature enables Voice over IP (VoIP) communications using routers equipped with the following voice port hardware:

- Quicknet LineJACK or PhoneJACK analog telephony cards
- ISDN cards
- Voicetronix OpenLine4 (was V4PCI) − 4 analog telephone lines cards
- Zaptel Wildcard X100P IP telephony card − 1 analog telephone line

Specifications

Packages required : ***telephony***
License required : *Any*
Home menu level : ***/ip telephony***
Protocols utilized : *Complete list of VoIP protocols*
Hardware usage: *may require additional RAM (64MB recommended)*

Related Documents

Software Package Installation and Upgrading
ISDN Interface
Authentication, Authorization and Accounting

Description

IP telephony, known as Voice over IP (VoIP), is the transmission of telephone calls over a data network like one of the many networks that make up the Internet. There are four ways that you might talk to someone using VoIP:

- Computer−to−computer − This is certainly the easiest way to use VoIP, and you don't have to pay for long−distance calls.
- Computer−to−telephone − This method allows you to call anyone (who has a phone) from your computer. Like computer−to−computer calling, it requires a software client. The software is typically free, but the calls may have a small per−minute charge.
- Telephone−to−computer − Allows a standard telephone user to initiate a call to a computer user.
- Telephone−to−telephone − Through the use of gateways, you can connect directly with any other standard telephone in the world.

IP Telephony Specifications

Supported Hardware

The MikroTik RouterOS V2.7 supports following telephony cards from Quicknet Technologies, Inc. (www.quicknet.net):

- Internet PhoneJACK (ISA) for connecting an analog telephone,
- Internet LineJACK (ISA) for connecting an analog telephone line or a telephone.

For supported ISDN cards please see the ISDN Interface Manual.

The MikroTik RouterOS V2.7 supports the Voicetronix OpenLine4 card for connecting four (4) analog telephone lines telephony cards from Voicetronix, Inc. (www.voicetronix.com.au)

The MikroTik RouterOS V2.7 supports the Zaptel Wildcard X100P IP telephony card for connecting one analog telephone line from Linux Support Services (www.digium.com)

Supported Standards

- **Standards for VoIP**
  The MikroTik RouterOS supports IP Telephony in compliance with the International Telecommunications Union − Telecommunications (ITU−T) specification H.323v4. H.323 is a specification for transmitting multimedia (voice, video, and data) across an IP network. H.323v4 includes: H.245, H.225, Q.931, H.450.1, RTP(real−time protocol)
- **CODECs**
  The following audio CODECs are supported:
  G.711 − the 64 kbps Pulse code modulation (PCM) voice coding technique. The encoded voice is already in the correct format for digital voice delivery in the PSTN or through PBXs.
  G.723.1 − the 6.3 kbps compression technique that can be used for compressing audio signal at very low bit rate.
  GSM−06.10 − the 13.2 kbps coding
  LPC−10 − the 2.5 kbps coding
  G.729, G.729a − the 8 kbps CS−ACELP software coding

G.728 − 16 kbps coding technique, supported only on Quicknet LineJACK cards

- **RFCs**
  Compliant to the RFC1889(RTP) http://www.ietf.org/rfc/rfc1889.txt?number=1889
- **Regional Standards**
  Quicknet cards are approved in United States, United Kingdom, France, Germany, Australia, Japan. Voicetronix OpenLine4 is approved in Australia, Europe, New Zealand and USA (FCC).

Implementation Options

- **IP Telephony Gateway**
  When connected to a PBX or PSTN telephone line, the MikroTik router can act as a gateway between the telephone network and the VoIP network.
- **IP Telephone System**
  When connecting an analog telephone, the MikroTik router acts as an IP Telephone

The MikroTik IP Telephones and IP Telephony Gateways are interoperable with the following H.323 terminals:

- Microsoft Netmeeting
- Siemens IP phone HiNet LP 5100
- Cisco ATA 186
- Welltech LAN Phone 101
- Most H.323 compatible devices

IP Telephony Hardware Installation

Please install the telephony hardware into the PC accordingly the instructions provided by card manufacturer. Each installed Quicknet card requires IO memory range in the following sequence: the first card occupies addresses 0x300−0x31f, the second card 0x320−0x33f, the third 0x340−0x35f, and so on. Make sure there is no conflict in these ranges with other devices, e.g., network interface cards, etc.

If the MikroTik router will be used as

- an **IP telephone** − connect an analog telephone with tone dialing capability to the PhoneJACK or LineJACK card,
- an **IP telephony gateway** − connect an analog telephone line to the LineJACK, Voicetronix, Zaptel card or ISDN telephone line to ISDN card.

Please consult the ISDN Manual for more information about installing the ISDN adapters.

IP Telephony Configuration

Submenu level : **/ip telephony**

Description

The IP Telephony requires IP network connection and configuration. The basic IP configuration can be done under the **/ip address** and **/ip route** menus.

Telephony Voice Ports

Submenu level : **/ip telephony voice−port**

Description

This submenu is used for managing all IP telephony voice ports (**linejack**, **phonejack**, **isdn**, **voip**, **voicetronix**, **zaptel**).

Property Description

name − assigned name of the voice port
type (read−only: unknown | phonejack | linejack | phonejack−lite | phonejack−pci | voip | isdn | voicetronix | zaptel) − type of the installed telephony voice port:

- **unknown** − unknown card type
- **phonejack** − Quicknet PhoneJACK (ISA)
- **linejack** − Quicknet LineJACK (ISA)
- **phonejack−lite** − Quicknet PhoneJACK Lite Linux Edition (ISA)
- **phonejack−pci** − Quicknet PhoneJACK (PCI)
- **voip** − generic Voice over IP
- **isdn** − ISDN cards
- **voicetronix** − Voicetronix OpenLine4
- **zaptel** − Zaptel Wildcard X100P

autodial (*integer*; default: "") − number to be dialed automatically, if call is coming in from this voice port

Notes

If **autodial** does not exactly match an item in **/ip telephony numbers**, there can be two possibilities:

- if **autodial** is incomplete, rest of the number is asked (local voice port) or incoming call is denied (VoIP)
- if **autodial** is invalid, line is hung up (PSTN line), busy tone is played (POTS) or incoming call is denied (VoIP)

Monitoring the Voice Ports

Property Description

status (read−only: on−hook | off−hook | ring | connection | busy) − current state of the port:

- **on−hook** − the handset is on−hook, no activity
- **off−hook** − the handset is off−hook, the number is being dialed
- **ring** − call in progress, direction of the call is shown by the argument **direction**
- **connection** − the connection has been established
- **busy** − the connection has been terminated, the handset is still **off−hook**

port (*name*) − (only for LineJACK) the active port of the card

- **phone** − telephone connected to the card (POTS)
- **line** − line connected to the linejack card (PSTN)

direction (ip−to−port | port−to−ip) − direction of the call

- **ip−to−port** − call from the IP network to the voice card
- **port−to−ip** − call from the voice card to an IP address

line−status (plugged | unplugged) − (only for LineJACK and Zaptel) state of the PSTN line

- **plugged** − the telephone line is connected to the PSTN port of the card
- **unplugged** − there is no working line connected to the PSTN port of the card

phone−number (*integer*) − the number which is being dialed
remote−party−name (*name, integer*) − name and IP address of the remote party
codec (*name*) − CODEC used for the audio connection
duration (*time*) − duration of the audio call

Notes

Monitoring feature is not available for VoIP ports.

Use the **monitor** command under the corresponding menu to view the current state of the port.

Example

The following example will monitor **linejack** voice port:

```
[admin@MikroTik] ip telephony voice−port linejack> monitor PBX_Line
               status: connection
                 port: phone
            direction: port−to−ip
          line−status: unplugged
         phone−number: 26
    remote−party−name: pbx_20 [10.5.8.12]
                codec: G.723.1−6.3k/hw
             duration: 14s

[admin@MikroTik] ip telephony voice−port linejack>
```

Voice−Port Statistics

Notes

Voice−port statistics are available for all local voice ports (only VoIP voice ports do not provide this ability).

Use the **show−stats** command under the corresponding menu to view the statistics of current audio connection.

Example

The following example will shows statistics of LineJACK card:

```
[admin@MikroTik] ip telephony voice−port linejack> show−stats PBX_Line
        round−trip−delay: 5ms
            packets−sent: 617
              bytes−sent: 148080
```

```
              send-time: 31ms/30ms/29ms
          packets-received: 589
            bytes-received: 141360
              receive-time: 41ms/30ms/19ms
      average-jitter-delay: 59ms
             packets-lost: 0
      packets-out-of-order: 0
          packets-too-late: 2

[admin@MikroTik] ip telephony voice-port linejack>
```

The **average−jitter−delay** shows the approximate delay time till the received voice packet is forwarded to the driver for playback. The value shown is never less than 30ms, although the actual delay time could be less. If the shown value is >40ms, then it is close (+/−1ms) to the real delay time.

The jitter buffer preserves quality of the voice signal against the loss or delay of packets while traveling over the network. The larger the jitter buffer, the larger the total delay, but fewer packets lost due to timeout. If the jitter−buffer=0, then it is adjusted automatically during the conversation to keep lost packet rate under 1%. The **average−jitter−delay** is the approximate average time from the moment of receiving an audio packet from the IP network till it is played back over the telephony voice port.

The total delay from the moment of recording the voice signal till its playback is the sum of following three delay times:

- delay time at the recording point (approx. 38ms),
- delay time of the IP network (1..5ms and up),
- delay time at the playback point (the jitter delay).

A voice call can be terminated using the **clear−call** command (not available for VoIP voice ports). If the *voiceport* has an active connection, the command **clear−call *voiceport*** terminates it. The command is useful in cases, when the termination of connection has not been detected by one of the parties, and there is an "infinite call". It can also be used to terminate someone's call, if it is using up the line required for another call.

Voice Port for Telephony cards

Property Description

name − name given by the user or the default one
type (read−only: phonejack | phonejack−lite | phonejack−pci) − (only for PhoneJACK) type of the card, cannot be changed
autodial (*integer*; default: "") − phone number which will be dialed immediately after the handset has been lifted. If this number is incomplete, then the remaining part has to be dialed on the dial−pad. If the number is incorrect, busy tone is played. If the number is correct, then the appropriate number is dialed. If it is an incoming call from the PSTN line (linejack), then the **directcall** mode is used − the line is picked up only after the remote party answers the call.
playback−volume (*integer*; default: **0**) − playback volume in dB, 0dB means no change, possible values are −48...48dB
record−volume (*integer*; default: **0**) − record volume in dB, 0dB means no change, possible values are −48...48dB.
ring−cadence (**string**) − (only for quicknet cards) a 16−symbol ring cadence for the phone, each symbol is 0.5 seconds, **+** means ringing, **−** means no ringing.

region (australia | estonia | france | germany | japan | latvia | lithuania | mikrotik | uk | us; default: **us**) − regional setting for the voice port. For phonejack, this setting is used for generating the tones. For linejacks, this setting is used for setting the parameters of PSTN line, as well as for detecting and generating the tones.
aec (yes | no; default: **yes**) − echo detection and cancellation.
If the echo cancellation is on, then the following parameters are used:

- **aec−tail−length** (short | medium | long; default: **short**) − size of the buffer of echo detection.
- **aec−nlp−threshold** (off | low | medium | high; default: **low**) − level of cancellation of silent sounds.
- **aec−attenuation−scaling** (*integer*; default: **4**) − factor of additional echo attenuation. Possible values are 0...10.
- **aec−attenuation−boost** (*integer*; default: **0**) − level of additional echo attenuation. Possible values are 0 ... 90dB.
- **software−aec** (yes | no; default: **no**) − software echo canceller (experimental, for most of the cards.

agc−on−playback (yes | no; default: *no*) − automatic gain control on playback (can not be used together with hardware voice codecs)
agc−on record (yes | no; default: *no*) − automatic gain control on record (can not be used together with hardware voice codecs)
detect−cpt (yes | no; default: no) − automatically detect call progress tones

Notes

All commands relating the Quicknet, Voicetronix and Zaptel Wildcard cards are listed under the **/ip telephony voice−port** submenus:

```
[admin@MikroTik] ip telephony voice-port linejack> print
Flags: X - disabled
  0   name="linejack1" autodial="" region=us playback-volume=0
      record-volume=0 ring-cadence="++-++--- ++-++---" agc-on-playback=no
      agc-on-record=no aec=yes aec-tail-length=short aec-nlp-threshold=low
      aec-attenuation-scaling=4 aec-attenuation-boost=0 software-aec=no
      detect-cpt=yes


[admin@MikroTik] ip telephony voice-port linejack>
```

For linejacks, there is a command **blink *voiceport***, which blinks the LEDs of the specified *voiceport* for five seconds after it is invoked. This command can be used to locate the respective card from several linejack cards.

Voice Port for Voicetronix cards

Submenu level : **/ip telephony voice−port voicetronix**

Property Description

Voicetronix telephony cards have some additional properties other cards haven't:

balance−registers (*integer*; default: **199**) − registers which depend on telephone line impedance. Can be adjusted to get best echo cancellation
balance−status (*read−only*: *integer*) − shows quality of hardware echo cancellation in dB
loop−drop−detection (yes | no; default: **yes**) − automatically clear call when loop drop is detected

Notes

balance−status depends on **balance−registers** value. When **balance−registers** are changed, gets status **unknown**. After **test−balance** command execution gets some value in dB − the less, the better. At least −6dB or less is required for echo canceller to do his job.

As some Voicetronix cards fail to detect loop drop correctly, with **loop−drop−detection** you can manage whether loop drop detection feature is enabled.

Voicetronix telephony cards also have some additional commands that other cards haven't:

- **test−balance** − current **balance−registers** value is tested once. Result is placed in **balance−status** parameter. Balance can be tested only when line is **off−hook**. It won't work if line is **on−hook** or there is established connection.
- **find−best−balance** − series of **test−balance** is executed with different **balance−registers** values. During tests **balance−registers** are updated to the best ones.

Some tips for testing balance registers:

- test is sensitive to noise from the phone, so it's recommended to cover mouth peice during it;
- **find−best−balance** can be interrupted by **clean−call** command;
- once best **balance−registers** value is known, it can be set manually to this best value for all voicetronix voice ports, which will use the same telephone line;
- **balance−registers** should be changed only if echo cancellation on voicetronix card does not work good enough. Echo cancellation problems can imply DTMF and **busy−tone** detection failures.
- **balance−registers** value has to be in format **bal1[,bal3[,bal2]]**, where bal1, bal2, bal3 − balance registers. **bal1** has to be in interval 192..248 (0xC0..0xF8). The others should be in interval 0..255 (0x00..0xFF).

Voice Port for ISDN

Submenu level : **/ip telephony voice−port isdn**

Property Description

name − Name given by the user or the default one.
msn (*integer*) − Telephone number of the ISDN voice port (ISDN MSN number).
lmsn (*character*) − msn pattern to listen on. It determines which calls from the ISDN line this voice port should answer. If left empty, **msn** is used. Meaning of special symbols:

- **;** − separates pattern entries (more than one pattern can be specified this way)
- **?** − matches one character
- ***** − matches zero or more characters
- **[ ]** − matches any single character from the set in brackets
- **[^]** − matches any single character not from the set in brackets

autodial (*integer*) − phone number which will be dialed immediately on each incoming ISDN call. If this number contains 'm', then it will be replaced by originally called (ISDN) telephone number. If this number is incomplete, then the remaining part has to be dialed by the caller. If the number is incorrect, call is refused. If the number is correct, then the appropriate number is dialed. For that **directcall** mode is used − the line is picked up only after the remote party answers the call.
playback−volume (*integer*; default: **0**) − playback volume in dB, 0dB means no change, possible values

are −48...48dB.
record−volume (*integer*; default: **0**) − record volume in dB, 0dB means no change, possible values are −48...48dB.
region (australia | estonia | france | germany | japan | latvia | lithuania | mikrotik | uk | us; default: **us**) − regional setting for the voice port (for tone generation only).
aec (yes | no; default: **yes**) − echo detection and cancellation. Possible values are **yes** and **no**. If the echo cancellation is on, then **aec−tail−length** parameter is used.
aec−tail−length (short | medium | long; default: **short**) − size of the buffer of echo detection. Possible values are: **short** (8 ms), **medium** (16 ms), **long** (32 ms).
software−aec (yes | no; default: **no**) − software echo cancellation (experimental)
agc−on−playback (yes | no; default: *no*) − automatic gain control on playback
agc−on−record (yes | no; default: *no*) − automatic gain control on record

Notes

In contrary to the phonejack and linejack voice ports, which are as many as the number of cards installed, the isdn ports can be added as many as desired.

Example

```
[admin@MikroTik] ip telephony voice−port isdn> print
Flags: X − disabled
  0   name="isdn1" autodial="" region=germany msn="140" lmsn=""
      playback−volume=0 record−volume=0 agc−on−playback=no agc−on−record=no
      software−aec=no aec=yes aec−tail−length=short

[admin@MikroTik] ip telephony voice−port isdn>
```

Voice Port for Voice over IP (voip)

Submenu level : **/ip telephony voice−port voip**

Description

The voip voice ports are virtual ports, which designate a voip channel to another host over the IP network. You must have at least one voip voice port to be able to make calls to other H.323 devices over IP network.

Property Description

name − Name given by the user or the default one.
remote−address (*IP address*; default: **0.0.0.0**) − IP address of the remote party (IP telephone or gateway) associated with this voice port. If the call has to be performed through this voice port, then the specified IP address is called. If there is an incoming call from the specified IP address, then the parameters of this voice port are used. If there is an incoming call from an IP address, which is not specified in any of the voip voice port records, then the default record is used. If there is no default record, then default values are used.
autodial (*integer*) − phone number which will be added in front of the telephone number received over the IP network. In most cases it should be blank.
jitter−buffer (*integer*; default: **100ms**) − size of the jitter buffer, 0...1000ms. The jitter buffer preserves quality of the voice signal against the loss or delay of packets while traveling over the network. The larger the jitter buffer, the larger the total delay, but fewer packets lost due to timeout. If the setting is jitter−buffer=0, the size of it is adjusted automatically during the conversation, to keep amount of lost packets under 1%.

silence−detection (yes | no; default: **no**) − if **yes**, then silence is detected and no audio data is sent over the IP network during the silence period.
prefered−codec (none | G.711−ALaw−64k/hw | G.711−ALaw−64k/sw | G.711−uLaw−64k/hw | G.711−uLaw−64k/sw | G.723.1−6.3k/hw | G.723.1−6.3k/sw | G.729−8k/sw | G.729A−8k/sw | GSM−06.10−13.2k/sw | LPC−10−2.5k/sw; default: **none**) − the preferred codec to be used for this voip voice port. If possible, the specified codec will be used.
fast−start (yes | no; default: **yes**) − allow or disallow the fast start. The fast start allows establishing the audio connection in a shorter time. However, not all H.323 endpoints support this feature. Therefore, it should be turned off, if there are problems to establish telephony connection using the fast start mode.

Example

```
[admin@MikroTik] ip telephony voice−port voip> print detail
Flags: X − disabled, D − dynamic, R − registered
  0    name="test" autodial="" remote−address=0.0.0.0 jitter−buffer=100ms
       prefered−codec=none silence−detection=no fast−start=yes


[admin@MikroTik] ip telephony voice−port voip>
```

Numbers

Submenu level : **/ip telephony numbers**

Description

This is the so−called "routing table" for voice calls. This table assigns numbers to the voice ports.The main function of the numbers routing table is to determine:

1. to which voice port route the call, and
2. what number to send over to the remote party.

Property Description

dst−pattern (*integer*) − pattern of the telephone number. Symbol **.** designate any digit, symbol **_** (only as the last one) designate any symbols (i.e. any number of characters can follow, ended with **#** character)
voice−port (*name*) − voice port to be used when calling the specified telephone number.
prefix (*integer*) − prefix, which will be used to substitute the known part of the **destination−pattern**, i.e., the part containing digits. The **dst−pattern** argument is used to determine which voice port to be used, whereas the **prefix** argument designates the number to dial over the voice port (be sent over to the remote party). If the remote party is an IP telephony gateway, then the number will be used for making the call.

Notes

More than one entry can be added with exactly the same dst−pattern. If first one of them is already busy, next one with the same dst−pattern is used. Telephony number entries can be moved, to select desired order.

Example

The example of actual printout:

```
[admin@MikroTik] ip telephony numbers> print
Flags: I - invalid, X - disabled, D - dynamic, R - registered
 #     DST-PATTERN                  VOICE-PORT     PREFIX
  0    26                           VoIP_GW        26

[admin@MikroTik] ip telephony numbers>
```

Let us consider the following example for the number table:

```
[admin@MikroTik] ip telephony numbers> print
Flags: I - invalid, X - disabled, D - dynamic, R - registered
 #     DST-PATTERN                  VOICE-PORT     PREFIX
 0     12345                               XX
 1     1111.                               YY
 2     22...                               ZZ      333
 3     ...                                 QQ      55

[admin@MikroTik] ip telephony numbers>
```

We will analyze the Number Received (nr) − number dialed at the telephone, or received over the line, the Voice Port (vp) − voice port to be used for the call, and the Number to Call (nc) − number to be called over the Voice Port.

If nr=55555, it does not match any of the destination patterns, therefore it is rejected.
If nr=123456, it does not match any of the destination patterns, therefore it is rejected.
If nr=1234, it does not match any of the destination patterns (incomplete for record #0), therefore it is rejected.
If nr=12345, it matches the record #0, therefore number "" is dialed over the voice port XX.
If nr=11111, it matches the record #1, therefore number "1" is dialed over the voice port YY.
If nr=22987, it matches the record #2, therefore number "333987" is dialed over the voice port ZZ.
If nr=22000, it matches the record #2, therefore number "333000" is dialed over the voice port ZZ.
If nr=444, it matches the record #3, therefore number "55444" is dialed over the voice port QQ.

Let us add a few more records:

```
[admin@MikroTik] ip telephony numbers> print
Flags: I - invalid, X - disabled, D - dynamic, R - registered
   #     DST-PATTERN                  VOICE-PORT     PREFIX
 .....
   4     222                            KK           44444
   5     3..                            LL           553

[admin@MikroTik] ip telephony numbers>
```

If nr=222 => the best match is the record # 4=> nc=44444, vp=KK.
The 'best match' means that it has the most coinciding digits between the nr and destination−pattern.
If nr=221 => incomplete record # 2 => call is rejected
If nr=321 => the best match is the record # 5 => nc=55321, vp=LL
If nr=421 => matches the record # 3 => nc=55421, vp=QQ
If nr=335 => the best match is the record # 5 => nc=55321, vp=LL

Let us add a few more records:

```
[admin@MikroTik] ip telephony numbers> print
Flags: I - invalid, X - disabled, D - dynamic, R - registered
```

```
   #     DST-PATTERN                  VOICE-PORT     PREFIX
 .....
   6     33...                        MM              33
   7     11.                          NN              7711

[admin@MikroTik] ip telephony numbers>
```

If nr=335 => incomplete record # 6 => the call is rejected.
Explanation of this case:

> The nr=335 fits perfectly both the record # 3 and # 5. The # 5 is chosen as the 'best match' candidate at the moment. Furthermore, there is record # 6, which has two matching digits (more than for # 3 or # 5). Therefore the # 6 is chosen as the 'best match'. However, the record # 6 requires five digits, but the nr has only three. Two digits are missing, therefore the number is incomplete. Two additional digits would be needed to be entered on the dialpad. If the number is sent over from the network, it is rejected.

If nr=325 => matches the record # 5 => nc=55325, vp=LL
If nr=33123 => matches the record # 6 => nc=33123, vp=MM
If nr=123 => incomplete record # 0 => call is rejected
If nr=111 => incomplete record # 1 => call is rejected
If nr=112 => matches the record # 7 => nc=77112, vp=NN
If nr=121 => matches the record # 3 => nc=55121, vp=QQ

It is impossible to add the following records:

```
[admin@MikroTik] ip telephony numbers> print
Flags: I - invalid, X - disabled, D - dynamic, R - registered
   #     DST-PATTERN             VOICE-PORT
 .....                                              reason:
         11                                 DD      conflict with record # 1
                                                    and # 7
         11..                               DD      conflict with record # 7
         111                                DD      conflict with record # 1
         22.                                DD      conflict with record # 2
         .....                              DD      conflict with record # 3
```

Regional Settings

Submenu level : **/ip telephony region**

Description

Regional settings are used to adjust the voice port properties to the PSTN system or the PBX. For example, to detect hang−up from line, there has to be correct regional setting for the LineJACK card: there must be correct busy−tone−frequency and busy−tone−cadence set for region which this LineJACK card uses. Without that, detect−cpt parameter for LineJACK's voice port has to be set to true.

Property Description

flag − (P) predefined, cannot be changed or removed. Users can add their own regional settings, which can be changed and removed.
name − Name of the regional setting

busy−tone−cadence (*integer*; default: **500,500**) − Busy tone cadence in ms (0 − end of cadence), 0...30000
busy−tone−frequency (*integer* x *integer*; default: **440x0**) − Frequency (20...2000) and volume gain (−24...6) of busy tone Hz x dB.
data−access−arrangement (australia | france | germany | japan | uk | us; default: **us**) − ring voltage, impedance setting for line−jack card
dial−tone−frequency (*integer* x *integer*; default: **440x0**) − Frequency (20...2000) and volume gain (−24...6) of dial tone Hz x dB
dtmf−tone−cadence (*integer*; default: **180,60**) − Dual Tone Multi Frequency tone cadence in ms
dtmf−tone−volume (*integer*; default: **−3,−3**) − Dual Tone Multi Frequency tone volume in dB
ring−tone−cadence (*integer*; default: 1000,2000) − Ring tone cadence in ms (0 − end of cadence), 0...30000
ring−tone−frequency (*integer* x *integer*; default: **440x0**) − Frequency (20...2000) and volume gain (−24...6) of ring tone Hz x dB

Notes

For generating the tone, the frequency and cadence arguments are used. The dialtone always is continuous signal, therefore it does not have the cadence argument. When detecting the dialtone, it should be at least 100ms long.

Example

```
[admin@MikroTik] ip telephony region> print
Flags: P − predefined
  0 P name="us" data−access−arrangement=us dial−tone−frequency=350x0,440x0
      busy−tone−frequency=480x0,620x0 busy−tone−cadence=500,500,500,500
      ring−tone−frequency=480x0,440x0 ring−tone−cadence=2000,4000
      dtmf−tone−volume=−3,−3 dtmf−tone−cadence=180,60

  1 P name="uk" data−access−arrangement=uk dial−tone−frequency=350x0,440x0
      busy−tone−frequency=400x0 busy−tone−cadence=375,375,375,375
      ring−tone−frequency=400x0,450x0 ring−tone−cadence=400,200,400,2000
      dtmf−tone−volume=−3,−3 dtmf−tone−cadence=180,60

  2 P name="france" data−access−arrangement=france dial−tone−frequency=440x0
      busy−tone−frequency=440x0 busy−tone−cadence=250,250,250,250
      ring−tone−frequency=440x0 ring−tone−cadence=1500,3500
      dtmf−tone−volume=−3,−3 dtmf−tone−cadence=180,60

  3 P name="germany" data−access−arrangement=germany
      dial−tone−frequency=425x0 busy−tone−frequency=425x0
      busy−tone−cadence=480,480,480,480 ring−tone−frequency=425x0
      ring−tone−cadence=1000,4000 dtmf−tone−volume=−3,−3
      dtmf−tone−cadence=180,60

  ...
```

Sometimes it is necessary to add an additional regional setting matching the properties of a particular PBX. The following example will show you how with **add** command to add a new regional setting:

```
[admin@MikroTik] ip telephony region> add
Creates new item with specified property values.
        busy−tone−cadence  Busy tone cadence in ms
      busy−tone−frequency  Frequency and volume gain of busy tone Hz x dB
                copy−from  Item number
  data−access−arrangement  Ring voltage, impedance setting for line−jack card
```

```
      dial-tone-frequency  Frequency and volume gain of dial tone Hz x dB
        dtmf-tone-cadence  Dual Tone Multi Frequency tone cadence in ms
         dtmf-tone-volume  Dual Tone Multi Frequency tone volume in dB
                     name  Name of the regional setting
        ring-tone-cadence  Ring tone cadence in ms
      ring-tone-frequency  Frequency and volume gain of ring tone Hz x dB
[admin@MikroTik] ip telephony region>
```

To change, for example, the volume gain of both dial tone frequencies to −6dB for a user defined region **home**, you need to enter the command:

```
[admin@MikroTik] ip telephony region> set home dial-tone-frequency=350x-6,440x-6
```

Audio CODEC

Submenu level : **/ip telephony codec**

Notes

CODECs are listed according to their priority of use. The highest priority is at the top. CODECs can be enabled, disabled and moved within the list. When connecting with other H.323 systems, the protocol will negotiate the CODEC which both of them support according to the priority order.

The hardware codecs (/hw) are built−in CODECs supported by Quicknet cards. If an ISDN card is used, then the hardware CODECs are ignored, only software CODECs (/sw) are used.

The choice of the CODEC type is based on the throughput and speed of the network. Better audio quality can be achieved by using CODEC requiring higher network throughput. The highest audio quality can be achieved by using the G.711−uLaw CODEC requiring 64kb/s throughput for each direction of the call. It is used mostly within a LAN. The G.723.1 CODEC is the most popular one to be used for audio connections over the Internet. It requires only 6.3kb/s throughput for each direction of the call.

Example

```
[admin@MikroTik] ip telephony codec> print
Flags: X - disabled
  #   NAME
  0   G.723.1-6.3k/sw
  1   G.728-16k/hw
  2   G.711-ALaw-64k/hw
  3   G.711-uLaw-64k/hw
  4   G.711-uLaw-64k/sw
  5   G.711-ALaw-64k/sw
  6   G.729A-8k/sw
  7   GSM-06.10-13.2k/sw
  8   LPC-10-2.5k/sw
  9   G.723.1-6.3k/hw
 10   G.729-8k/sw
[admin@MikroTik] ip telephony codec>
```

AAA

Submenu level : **/ip telephony aaa**

Description

AAA (Authentication Authorization Accounting) can be used to configure the RADIUS accounting feature.

```
[admin@MikroTik] ip telephony aaa> print
    use-radius-accounting: no
           interim-update: 0s
[admin@MikroTik] ip telephony aaa>
```

Property Description

use-radius-accounting (yes | no; default: **no**) - defines whether to use radius accounting or not
interim-update (*integer*; default: **0**) - defines time interval between communications with the router. If this time will exceed, RADIUS server will assume that this connection is down. This value is suggested to be not less than 3 minutes. If set to **0s**, no interim-update messages are sent at all

The contents of the CDR (Call Detail Record) are as follows: **NAS-Identifier** - router name (from **/system identity print**)
NAS-IP-Address - router's local IP address which the connection was established to (if exist)
NAS-Port-Type - always **Async**
Event-Timestamp - data and time of the event
Acct-Session-Time - current connection duration (only in INTERIM-UPDATE and STOP records)
Acct-Output-Packets - sent RTP (Real-Time Transport Protocol) packet count (only in INTERIM-UPDATE and STOP records)
Acct-Input-Packets - received RTP (Real-Time Transport Protocol) packet count (only in INTERIM-UPDATE and STOP records)
Acct-Output-Octets - sent byte count (only in INTERIM-UPDATE and STOP records)
Acct-Input-Octets - received byte count (only in INTERIM-UPDATE and STOP records)
Acct-Session-Id - unique session participient ID
h323-disconnect-cause - session disconnect reason (only in STOP records):

- **0** - Local endpoint application cleared call
- **1** - Local endpoint did not accept call
- **2** - Local endpoint declined to answer call
- **3** - Remote endpoint application cleared call
- **4** - Remote endpoint refused call
- **5** - Remote endpoint did not answer in required time
- **6** - Remote endpoint stopped calling
- **7** - Transport error cleared call
- **8** - Transport connection failed to establish call
- **9** - Gatekeeper has cleared call
- **10** - Call failed as could not find user (in GK)
- **11** - Call failed as could not get enough bandwidth
- **12** - Could not find common capabilities
- **13** - Call was forwarded using FACILITY message
- **14** - Call failed a security check and was ended
- **15** - Local endpoint busy
- **16** - Local endpoint congested
- **17** - Remote endpoint busy
- **18** - Remote endpoint congested
- **19** - Could not reach the remote party

- **20** − The remote party is not running an endpoint
- **21** − The remote party host off line
- **22** − The remote failed temporarily app may retry

h323−disconnect−time − session disconnect time (only in INTERIM−UPDATE and STOP records)
h323−connect−time − session establish time (only in INTERIM−UPDATE and STOP records)
h323−gw−id − name of gateway emitting message (should be equal to **NAS−Identifier**)
h323−call−type − call leg type (should be **VoIP**)
h323−call−origin − indicates origin of call relative to gateway (**answer** for calls from IP network, **originate** − to IP network)
h323−setup−time − call setup time
h323−conf−id − unique session ID
h323−remote−address − the remote address of the session
NAS−Port−Id − voice port ID
Acct−Status−Type − record type:

- **START** − session is established
- **STOP** − session is closed
- **INTERIM−UPDATE (ALIVE)** − session is alive. The time between the messages is defined by **interim−update−interval** parameter (if it is set to **0s**, there will be no such messages)

Notes

All the parameters, which names begin with **h323**, are CISCO vendor specific Radius attributes

IP Telephony Gatekeeper

Submenu level : **/ip telephony gatekeeper**

```
[admin@MikroTik] ip telephony gatekeeper> print
         gatekeeper: local
          remote-id: ""
     remote-address: 0.0.0.0
         registered: yes
    registered-with: "tst-2.7@localhost"
```

Property Description

gatekeeper (none | local | remote; default: **none**) − Gatekeeper name to use

- **none** − don't use any gatekeeper at all
- **local** − start and use local gatekeeper
- **remote** − use some other gatekeeper

remote−address (*IP address*; default: **0.0.0.0**) − IP address of remote gatekeeper to use. If set to 0.0.0.0, broadcast gatekeeper discovery is used
remote−id (*name*) − name of remote gatekeeper to use. If left empty, first available gatekeeper will be used. Name of locally started gatekeeper is the same as system identity

Statistics:

registered (yes | no) − shows whether local H.323 endpoint is registered to any gatekeeper
registered−with (*name*) − name of gatekeeper to which local H.323 endpoint is registered

Notes

For each H.323 endpoint gatekeeper stores its telephone numbers. So, gatekeeper knows all telephone numbers for all registered endpoints. And it knows which telephone number is handled by which endpoint. Mapping between endpoints and their telephone numbers is the main functionality of gatekeepers.

If endpoint is registered to endpoint, it does not have to know every single endpoint and every single telephone number, which can be called. Instead, every time some number is dialed, endpoint asks gatekeeper for destination endpoint to call by providing called telephone number to it.

Gatekeeper Configuration

Example

In most simple case with one phonejack card and some remote gatekeeper, configuration can be as follows:

```
[admin@MikroTik] ip telephony voice−port> print
Flags: X − disabled
  #   NAME                        TYPE            AUTODIAL
  0   phonejack1                phonejack
  1   voip1                       voip

[admin@MikroTik] ip telephony voice−port voip> print
Flags: X − disabled, D − dynamic, R − registered
  #    NAME      AUTODIAL REMOTE−ADDRESS  JITTER−BUFFER PREFERED−CODEC  SIL FAS
  0    voip1             0.0.0.0         0s            none            no  yes

[admin@MikroTik] ip telephony numbers> print
Flags: I − invalid, X − disabled, D − dynamic, R − registered
  #     DST−PATTERN             VOICE−PORT              PREFIX
  0     11                      phonejack1
  1     _                       voip1

[admin@MikroTik] ip telephony gatekeeper> print
         gatekeeper: remote
          remote−id: ""
     remote−address: 10.0.0.98
         registered: yes
    registered−with: "MikroTik@10.0.0.98"
```

In this case this endpoint will register to gatkeeper at IP address 10.0.0.98 with telephone number 11. Every call to telephone number 11 will be transfered from gatekeeper to this endpoint. And this endpoint will route this call to phonejack1 voice port. On any other telephone number gatekeeper will be asked for real destination. >From this endpoint it will be possible to call all the endpoints, which are registered to the same gatekeeper. If that gatekeeper has static entries about endpoints, which are not registered to gatekeeper, it still will be possible to call those endpoints by those statically defined telephone numbers at gatekeeper.

Notes

MikroTik IP telephony package includes very simple gatekeeper. This gatekeeper can be activated by setting "gatekeeper" parameter to "local". In this case local endpoint automatically is registered to local gatekeeper. And any other endpoint can register to this gatekeeper, too.

Registered endpoints are added to **/ip telephony voice−port voip** table. Those entries are marked with "D − dynamic". These entries can not be removed and their remote−address can not be changed. If there already was an voip entry with the same IP address, it is marked with "R − registered". Remote−address can not be changed for these entries, too. But registered voip voice ports can be removed − they will stay as dynamic. If there is already dynamic voip voice port and static voip voice port with the same IP address is added, then instead of dynamic entry registered will appear.

Dynamic entries disappear when corresponding endpoint unregisters itself from this gatekeeper. Registered entries are static and will stay even after that endpoint will be unregistered from this gatekeeper.

Registered telephone numbers are added to "/ip telephony numbers" table. Here is exactly the same idea behind dynamic and registered telephone numbers as it is with voip voice ports.

When endpoint registers to gatekeeper, it sends its own telephone numbers (aliases and prefixes) within this registration request. **/ip telephony numbers** entry is registered to endpoint only if voice−port for that entry is local (not voip). If dst−pattern contains '.' or '_', it is sent as prefix, otherwise − as alias. As prefix is sent the known part of the dst−pattern. If there is no known part (dst−pattern is "_" or "...", for example), then this entry is not sent at all.

So, for example, if numbers table is like this:

```
[admin@MikroTik] ip telephony numbers> print
Flags: I − invalid, X − disabled, D − dynamic, R − registered
  #     DST−PATTERN             VOICE−PORT              PREFIX
  0     1.                      phonejack1
  1     128                     voip1                   128
  2     78                      voip2                   78
  3     77                      phonejack1
  4     76                      phonejack1              55
  5     _                       voip1
```

then entries 0, 3 and 4 will be sent, others are voip voice ports and are ignored. Entry **0** will be sent as prefix **1**, entry **3** − as alias **77**, entry **4** − as alias **76**.

If IP address of local endpoint is 10.0.0.100, then gatekeeper voip and numbers tables will look as follows:

```
[admin@MikroTik] ip telephony voice−port voip> print
Flags: X − disabled, D − dynamic, R − registered
  #    NAME      AUTODIAL REMOTE−ADDRESS  JITTER−BUFFER PREFERED−CODEC  SIL FAS
  0    tst−2.5           10.0.0.101      0s            none            no  yes
  1  D local             127.0.0.1       100ms         none            no  yes
  2  D 10.0.0...         10.0.0.100      100ms         none            no  yes

[admin@MikroTik] ip telephony numbers> print
Flags: I − invalid, X − disabled, D − dynamic, R − registered
  #     DST−PATTERN             VOICE−PORT              PREFIX
  0     78                      linejack1
  1     3...                    vctx1
  2     33_                     voip1
```

```
3     5..                   voip1
4  XD 78                    local                78
5  XD 3_                    local                3
6   D 76                    10.0.0.100           76
7   D 77                    10.0.0.100           77
8   D 1_                    10.0.0.100           1
```

Here we can see how aliases and prefixes are added to numbers table. Entries 0..3 are static. Entries 4 and 5 are added by registering local endpoint to local gatekeeper. Entries 6..8 are added by registering endpoint (with IP address 10.0.0.100) to local gatekeeper.

For prefixes, '_' is added at the end of dst−pattern to allow any additional digits to be added at the end.

Local endpoint is registered to local gatekeeper, too. So, local aliases and prefixes are added as dynamic numbers, too. Only, as they are local and corresponding number entries already exists in number table, then these dynamically added entries are disabled by default.

If any registered telephone number will conflict with some existing telephone numbers entry, it will be added as disabled and dynamic.

If in gatekeeper's numbers table there already exists exactly the same dst−pattern as some other endpoint is trying to register, this gatekeeper registration for that endpoint will fail.

IP Telephony Troubleshooting

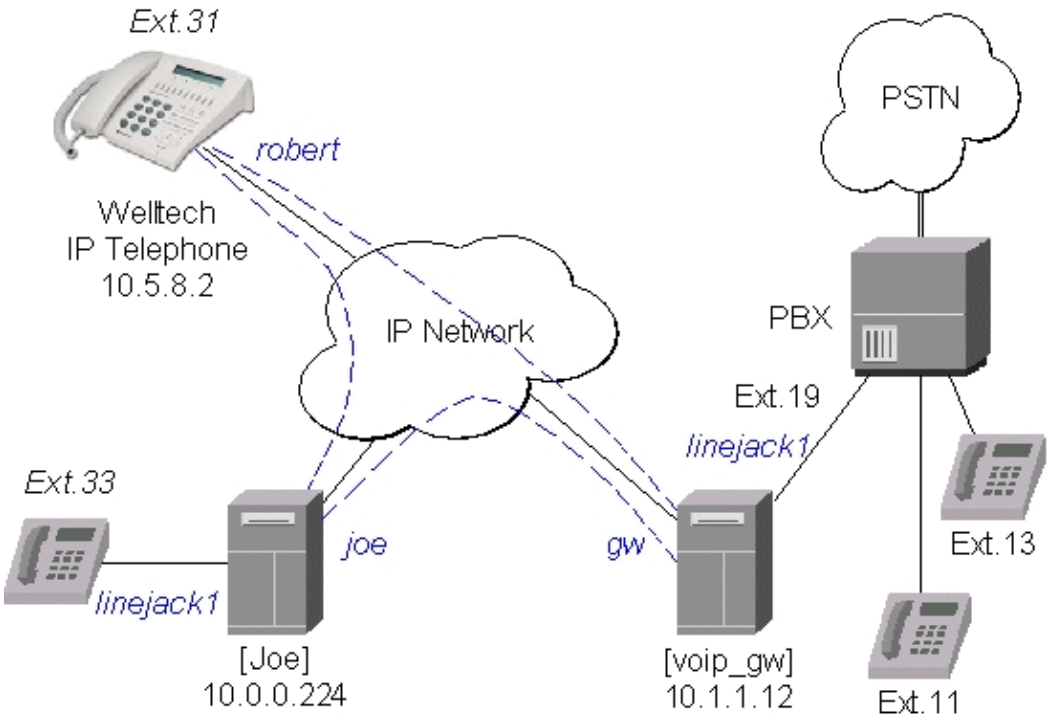
- *The IP Telephony does not work after upgrading from 2.5.x version*
  You need to completely reinstall the router using any installation procedure. You may keep the configuration using either the installation program option or the backup file.
- *The IP Telephony gateway does not detect the drop of the line when connected to some PBXs*
  Different regional setting should be used to match the parameters of the PBX. For example, try using **uk** for Meridian PBX.
- *The IP Telephone does not call the gateway, but gives busy signal*
  Enable the logging of IP telephony events under **/system logging facility**. Use the monitoring function for voice ports to debug your setup while making calls.
- *The IP telephony is working without NAT, but sound goes only in one direction*
  Disable **h323** service port in firewall:

```
/ip firewall service-port set h323 disabled=yes
```

IP Telephony Applications

The following describes examples of some useful IP telephony applications using the MikroTik RouterOS Quicknet telephony cards or ISDN cards.

Let us consider the following example of IP telephony gateway, one MikroTik IP telephone, and one Welltech LAN Phone 101 setup:



Setting up the MikroTik IP Telephone

The QuickNet LineJACK or PhoneJACK card and the MikroTik RouterOS telephony package should be installed in the MikroTik router (IP telephone) 10.0.0.224x. An analog telephone should be connected to the 'phone' port of the QuickNet card. If you pick up the handset, a dialtone should be heard.

The basic telephony configuration should be as follows:

1. Add a voip voice port to the **/ip telephony voice-port voip** for each of the devices you want to call, or want to receive calls from, i.e., (the IP telephony gateway 10.1.1.12 and the Welltech IP telephone 10.5.8.2):

```
[admin@Joe] ip telephony voice-port voip> add name=gw remote-address=10.1.1.12
[admin@Joe] ip telephony voice-port voip> add name=rob remote-address=10.5.8.2
[admin@Joe] ip telephony voice-port voip> print
Flags: X - disabled, D - dynamic, R - registered
  #    NAME      AUTODIAL REMOTE-ADDRESS  JITTER-BUFFER PREFERED-CODEC  SIL FAS
  0    gw                 10.1.1.12       100ms         none            no  yes
  1    rob                10.5.8.2        100ms         none            no  yes
[admin@Joe] ip telephony voice-port voip>
```

   You should have three vioce ports now:

```
[admin@Joe] ip telephony voice-port> print
Flags: X - disabled
  #   NAME                          TYPE           AUTODIAL
  0   linejack1                     linejack
  1   gw                            voip
  2   rob                           voip
[admin@Joe] ip telephony voice-port>
```

2. Add a at least one unique number to the **/ip telephony numbers** for each voice port. This number

will be used to call that port:

```
[admin@Joe] ip telephony numbers> add dst-pattern=31 voice-port=rob
[admin@Joe] ip telephony numbers> add dst-pattern=33 voice-port=linejack1
[admin@Joe] ip telephony numbers> add dst-pattern=1. voice-port=gw prefix=1
[admin@Joe] ip telephony numbers> print
Flags: I - invalid, X - disabled, D - dynamic, R - registered
  #     DST-PATTERN             VOICE-PORT              PREFIX
  0     31                          rob                 31
  1     33                          linejack1
  2     1.                          gw                  1
[admin@Joe] ip telephony numbers>
```

Here, the dst-pattern=31 is to call the Welltech IP Telephone, if the number '31' is dialed on the dialpad.
The dst-pattern=33 is to ring the local telephone, if a call for number '33' is received over the network.
Anything starting with digit '1' would be sent over to the IP Telephony gateway.

Making calls from the IP telephone 10.0.0.224:

- To call the IP telephone 10.5.8.2, it is enough to lift the handset and dial the number "31".
- To call the PBX extension 13, it is enough to lift the handset and dial the number "13".

After establishing the connection with '13', the voice port monitor shows:

```
[admin@Joe] ip telephony voice-port linejack> monitor linejack
               status: connection
                 port: phone
            direction: port-to-ip
          line-status: unplugged
         phone-number: 13
    remote-party-name: PBX_Line [10.1.1.12]
                codec: G.723.1-6.3k/hw
             duration: 16s

[admin@Joe] ip telephony voice-port linejack>
```

Use the telephony logging feature to debug your setup.

Setting up the IP Telephony Gateway

QuickNet LineJACK, Voicetronix, Zaptel Wildcard or ISDN (see the appropriate manual) card and the MikroTik RouterOS telephony package should be installed in the MikroTik router (IP telephony gateway) 10.1.1.12. A PBX line should be connected to the 'line' port of the card. For LineJACK card the LED next to the 'line' port should be green, not red.

The IP telephony gateway [voip_gw] requires the following configuration:

1. Set the regional setting to match our PBX. The **mikrotik** seems to be best suited:

```
[admin@voip_gw] ip telephony voice-port linejack> set linejack1 region=mikrotik
[admin@voip_gw] ip telephony voice-port linejack> print
Flags: X - disabled
  0   name="linejack1" autodial="" region=mikrotik playback-volume=0
```

```
      record-volume=0 ring-cadence="++-++--- ++-++---" agc-on-playback=no
      agc-on-record=no aec=yes aec-tail-length=short aec-nlp-threshold=low
      aec-attenuation-scaling=4 aec-attenuation-boost=0 software-aec=no
      detect-cpt=yes

[admin@voip_gw] ip telephony voice-port linejack>
```

2. Add a voip voice port to the **/ip telephony voice-port voip** for each of the devices you want to call, or want to receive calls from, i.e., (the IP telephone 10.0.0.224 and the Welltech IP telephone 10.5.8.2):

```
[admin@voip_gw] ip telephony voice-port voip> add name=joe \
\... remote-address=10.0.0.224
[admin@voip_gw] ip telephony voice-port voip> add name=rob \
\... remote-address=10.5.8.2 prefered-codec=G.723.1-6.3k/hw
[admin@voip_gw] ip telephony voice-port voip> print
Flags: X - disabled, D - dynamic, R - registered
  #    NAME      AUTODIAL REMOTE-ADDRESS  JITTER-BUFFER PREFERED-CODEC  SIL FAS
  0    joe               10.0.0.224      100ms         none            no yes
  1    rob               10.5.8.2        100ms         G.723.1-6.3k/hw no yes
[admin@voip_gw] ip telephony voice-port voip>
```

3. Add number records to the **/ip telephony numbers**, so you are able to make calls:

```
[admin@voip_gw] ip telephony numbers> add dst-pattern=31 voice-port=rob prefix=31
[admin@voip_gw] ip telephony numbers> add dst-pattern=33 voice-port=joe prefix=33
[admin@voip_gw] ip telephony numbers> add dst-pattern=1. voice-port=linejack1 \
\... prefix=1
[admin@voip_gw] ip telephony numbers> print
Flags: I - invalid, X - disabled, D - dynamic, R - registered
  #     DST-PATTERN             VOICE-PORT              PREFIX
  0     31                          rob                 31
  1     33                          joe                 33
  2     1.                          linejack1           1
[admin@voip_gw] ip telephony numbers>
```

Making calls through the IP telephony gateway:

- To dial the IP telephone 10.0.0.224 from the office PBX line, the extension number 19 should be dialed, and, after the dial tone has been received, the number 33 should be entered. Thus, the telephone [Joe] is ringed.

  After establishing the voice connection with '33' (the call has been answered), the voice port monitor shows:

```
[admin@voip_gw] ip telephony voice-port linejack> monitor linejack1
               status: connection
                 port: line
            direction: port-to-ip
          line-status: plugged
         phone-number: 33
    remote-party-name: linejack1 [10.0.0.224]
                codec: G.723.1-6.3k/hw
             duration: 1m46s

[admin@voip_gw] ip telephony voice-port linejack>
```

- To dial the IP telephone 10.5.8.2 from the office PBX line, the extension number 19 should be dialed, and, after the dial tone has been received, the number 31 should be entered.

Setting up the Welltech IP Telephone

Please follow the documentation from www.welltech.com.tw on how to set up the Welltech LAN Phone 101. Here we give just brief recommendations:

1. We recommend to upgrade the Welltech LAN Phone 101 with the latest application software. Telnet to the phone and check what you have, for example:

```
usr/config$ rom -print

Download Method  :  TFTP
 Server Address  :  10.5.8.1

   Hardware Ver. :  4.0
       Boot Rom  :  nblp-boot.102a
Application Rom  :  wtlp.108h
        DSP App  :  48302ce3.127
     DSP Kernel  :  48302ck.127
  DSP Test Code  :  483cbit.bin
  Ringback Tone  :  wg-ringbacktone.100
      Hold Tone  :  wg-holdtone10s.100
  Ringing Tone1  :  ringlow.bin
  Ringing Tone2  :  ringmid.bin
  Ringing Tone3  :  ringhi.bin

usr/config$
```

2. Check if you have the codecs arranged in the desired order:

```
usr/config$ voice -print
Voice codec setting relate information
    Sending packet size  :
            G.723.1      : 30 ms
            G.711A       : 20 ms
            G.711U       : 20 ms
            G.729A       : 20 ms
            G.729        : 20 ms
    Priority order codec :
            g7231 g711a g711u g729a g729
    Volume levels        :
            voice volume : 54
            input gain   : 26
             dtmf volume : 23
Silence suppression &CNG:
            G.723.1      : Off
    Echo canceller       : On
 JitterBuffer Min Delay  : 90
 JitterBuffer Max Delay  : 150
usr/config$
```

3. Make sure you have set the H.323 operation mode to phone to phone (P2P), not gatekeeper (GK):

```
usr/config$ h323 -print
H.323 stack relate information
    RAS mode              : Non-GK mode
    Registered e164       : 31
    Registered H323 ID    : Rob
    RTP port              : 16384
    H.245 port            : 16640
    Allocated port range  :
```

```
              start port   : 1024
              end port     : 65535
    Response timeOut       : 5
    Connect  timeOut       : 5000
usr/config$
```

4. Add the gateway's address to the phonebook:

```
usr/config$ pbook −add name gw ip 10.1.1.12
usr/config$
This may take a few seconds, please wait....

Commit to flash memory ok!

usr/config$ pbook −print
index   Name                 IP                    E164
=============================================================================
1       gw                   10.1.1.12
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
usr/config$
```

Making calls from the IP telephone 10.5.8.2:

- Just lift the handset and dial '11', or '13' fo the PBX extensions.
- Dial '33' for [Joe]. The call request will be sent to the gateway 10.1.1.12, where it will be forwarded to [Joe]. If you want to call [Joe] directly, add a phonebook record for it:

```
usr/config$ pbook −add name Joe ip 10.0.0.224 e164 33
```

Use the telephony logging feature on the gateway to debug your setup.

Setting up the MikroTik Router and CISCO Router

Here are some hints on how to get working configuration for telephony calls between CISCO and MikroTik router.

Tested on:

- MT: 2.4.1
- CISCO: 1750

Configuration on the **MikroTik** side:

- G.729a codec MUST be disabled (otherwise connections are not possible at all)!!!

```
/ip telephony codec disable G.729A−8k/sw
```

- G.711−ALaw codec should not be used (in some cases there is no sound)

```
/ip telephony codec disable "G.711−ALaw−64k/sw G.711−ALaw−64k/hw"
```

- Fast start has to be used (otherwise no ring−back tone and problems with codec negotiation)

```
/ip telephony voice−port set cisco fast−start=yes
```

- Telephone number we want to call to must be sent to Cisco, for example

```
/ip telephony numbers add destination−pattern=101 voice−port=cisco prefix=101
```

- Telephone number, cisco will call us, must be assigned to some voice port, for example,

```
/ip telephony numbers add destination-pattern=098 voice-port=linejack
```

Configuration on the **CISCO** side:

- IP routing has to be enabled

```
ip routing
```
- Default values for fast start can be used

```
voice service pots
    default h323 call start
    exit
voice service voip
    default h323 call start
    exit
```
- Enable opening of RTP streams

```
voice rtp send-recv
```
- Assign some E.164 number for local telephone, for example, 101 to port 0/0

```
dial-peer voice 1 pots
    destination-pattern 101
    port 0/0
    exit
```
- create preferred codec listing

```
voice class codec codec_class_number
    codec preference 1 g711ulaw
    codec preference 2 g723r63
    exit
```

  NOTE: g723r53 codec can be used, too
- Tell, that some foreign E.164 telephone number can be reached by calling to some IP address, for example, 098 by calling to 10.0.0.98

```
dial-peer voice 11 voip
    destination-pattern 098
    session target ipv4:10.0.0.98
    voice-class codec codec_class_number
    exit
```

  NOTE: instead of codec class, one specified codec could be specified:

```
codec g711ulaw
```

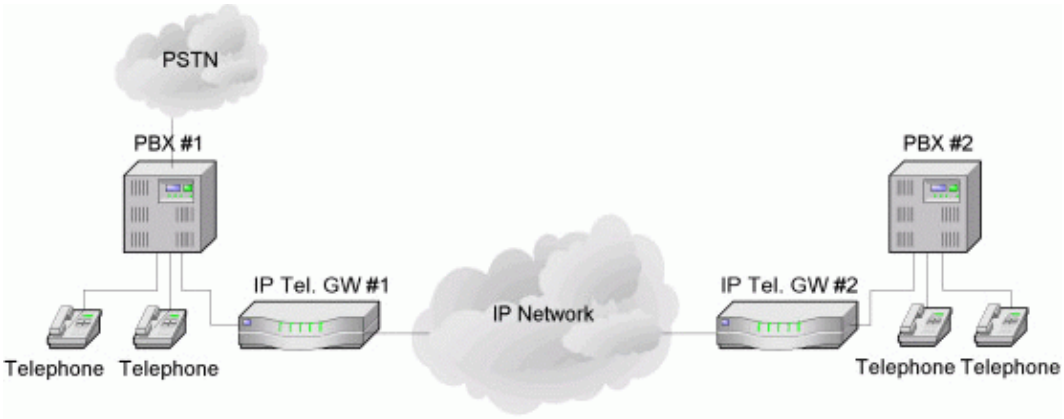
For reference, following is an exported CISCO configuration, that works:

```
!
version 12.1
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
```

```
hostname Router
!
logging rate-limit console 10 except errors
enable secret 5 $1$bTMC$nDGl9/n/pc3OMbtWxADMg1
enable password 123
!
memory-size iomem 25
ip subnet-zero
no ip finger
!
call rsvp-sync
voice rtp send-recv
!
voice class codec 1
 codec preference 1 g711ulaw
 codec preference 2 g723r63
!
interface FastEthernet0
 ip address 10.0.0.101 255.255.255.0
 no ip mroute-cache
 speed auto
 half-duplex
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.0.0.1
no ip http server
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
voice-port 0/0
!
voice-port 0/1
!
voice-port 2/0
!
voice-port 2/1
!
dial-peer voice 1 pots
 destination-pattern 101
 port 0/0
!
dial-peer voice 97 voip
 destination-pattern 097
 session target ipv4:10.0.0.97
 codec g711ulaw
!
dial-peer voice 98 voip
 destination-pattern 098
 voice-class codec 1
 session target ipv4:10.0.0.98
!
!
line con 0
 transport input none
line aux 0
line vty 0 4
 password 123
 login
!
end
```

Setting up PBX to PBX Connection over an IP Network

To interconnect two telephone switchboards (PBX) over an IP network, two IP telephony gateways should be configured. The setup is shown in the following diagram:



We want to be able to use make calls from local telephones of one PBX to local telephones or external lines of the other PBX.

Assume that:

- The IP telephony gateway #1 has IP address 10.0.0.182, and the name of the Voicetronix's first line is 'vctx1'.
- The IP telephony gateway #2 has IP address 10.0.0.183, and the name of the Voicetronix's first line is 'vctx1'.

The IP telephony configuration should be as follows:

- IP telephony gateway #1 should have

```
/ip telephony voice-port voip
add name=gw2 remote-address=10.0.0.183
/ip telephony numbers
add dst-pattern=1.. voice-port=gw2 prefix=2
add dst-pattern=2.. voice-port=vctx1 prefix=1
```

- IP telephony gateway #2 should have

```
/ip telephony voice-port voip
add name=gw1 remote-address=10.0.0.182
/ip telephony numbers
add dst-pattern=2.. voice-port=vctx1 prefix=1
add dst-pattern=1.. voice-port=gw1 prefix=2
```

The system works as follows:

To dial from the main office PBX#1 any extension of the remote office PBX#2, the extension with the connected gateway at PBX#1 should be dialed first. Then, after the dial tone of the gateway#1 is received, the remote extension number should be dialed.

To dial from the main office PBX#2 any extension of the remote office PBX#1, the actions are the same as in first situation.

Additional Resources

IP Telephony Online

IP Packet Packer Protocol (M3P)

Document revision 1.5 (13−Jun−2003)
This document applies to the MikroTik RouterOS v2.7

Table Of Contents

Summary

The MikroTik Packet Packer Protocol (M3P) optimizes the data rate usage of links using protocols that have a high overhead per packet transmitted. The basic purpose of this protocol is to better enable wireless networks to transport VoIP traffic and other traffic that uses small packet sizes of around 100 bytes.

M3P features:

- enabled by a per interface setting
- other routers with MikroTik Discovery Protocol enabled will broadcast M3P settings
- significantly increases bandwidth availability over some wireless links – by approximately four times
- offer configuration settings to customize this feature

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/ip packing***
Protocols utilized : *none*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
MikroTik Neighbor Discovery Protocol (MNDP)

MikroTik Packet Packer Protocol Description

The wireless protocol IEEE 802.11 and, due to a lesser extent, Ethernet protocol have a high overhead per packet as for each packet it is necessary to access the media, check for errors, resend in case of errors occured,

and send network maintenance messages (network maintenance is only for wireless). The MikroTik Packet Packer Protocol improves network performance by aggregating many small packets into a big packet, thereby minimizing the network per packet overhead cost. The M3P is very effective when the average packet size is 50−300 bytes – the common size of VoIP packets.

Specific Properties:

- may work on any Ethernet−like media
- is disabled by default for all interfaces
- when older version on the RouterOS are upgraded from a version without M3P to a version with discovery, current wireless interfaces will not be automatically enabled for M3P
- small packets going to the same MAC level destination (regardless of IP destination) are collected according to the set configuration and aggregated into a large packet according to the set size
- the packet is sent as soon as the maximum aggregated−packet packet size is reached or a maximum time of 15ms (+/−5ms)

MikroTik Packet Packer Protocol Setup

Submenu level : **/ip packing**

Description

IP MikroTik Packet Packer Protocol is working only between MikroTik routers, which are discovered with MikroTik Neighbor Discovery Protocol (MNDP). When M3P is enabled router needs to know which of its neighbouring hosts have enabled M3P. MNDP is used to negotiate unpacking settings of neighbours, therefore it has to be enabled on interfaces you wish to enable M3P. Consult MNDP manual on how to do it.

Property Description

aggregated−size (*integer*; default: **1500**) – the maximum size of the aggregated packet
interface (*name*) − interface to enable M3P on
packing (compress−all | compress−headers | none | simple; default: **simple**) − set the packing mode supported on interface

- **none** − no packing is applied to packets
- **simple** − aggregate many small packets in one big packet, minimizing the network per packet overhead cost
- **compress−headers** − further increase network performace by applying packet header compression (uses more CPU resource)
- **compress−all** − increase network performance even more by using header and data compression (extensive CPU usage)

unpacking (compress−all | compress−headers | none | simple; default: **simple**) −

- **none** − accept only usual packets
- **simple** − accept aggregated packets without compression and usual packets
- **compress−headers** − accept all packets except those with data compression
- **compress−all** − accept all packets

Notes

Level of packet compression increases like this: **none** −> **simple** −> **compress−headers** −> **compress−all**.

When router has to send a packet it choses minimum level of packet compression from what its own packing type is set and what other router's unpacking type is set. Same is with **aggregated−size** setting − minimum value of both endds is actual maximum size of aggregated packet used.

aggregated−size can be bigger than interface MTU if network device allows it to be (i.e., it supports sending and receiving frames bigger than 1514 bytes)

Example

To enable maximal compression on the **ether1** interface:

```
[admin@MikroTik] ip packing> add interface=ether1 packing=compress−all \
\... unpacking=compress−all
[admin@MikroTik] ip packing> print
Flags: X − disabled
  #   INTERFACE PACKING          UNPACKING        AGGREGATED−SIZE
  0   ether1    compress−all     compress−all     1500

[admin@MikroTik] ip packing>
```

MikroTik Neighbor Discovery Protocol (MNDP)

Document revision 1.6 (23−May−2003)
This document applies to the MikroTik RouterOS v2.7

Contents of the Manual

Summary

The MikroTik Neighbor Discovery Protocol (MNDP) eases network configuration and management by enabling each MikroTik router to discover other connected MikroTik routers and learn information about the system along with features which are enabled. The MikroTik routers can then automatically use learned information to set up some features with minimal or no configuration.

MNDP features:

- works on IP level connections
- works on all non−dynamic interfaces
- distributes basic information on the software version
- distributes information on configured features that should interoperate with other MikroTik routers

MikroTik RouterOS is able to discover both MNDP and CDP (Cisco Discovery Protocol) devices.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/ip neighbour***
Standards and Technologies : *MNDP*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
M3P (MikroTik Packet Packer Protocol)

Description

MNDP basic function is to assist with automatic configuration of features that are only available between MikroTik routers. Currently this is used for the 'Packet Packer' feature. The 'Packet Packer' may be enabled on a per interface basis. The MNDP protocol will then keep information about what routers have enabled the 'unpack' feature and the 'Packet Packer' will be used for traffic between these routers.

Specific Properties:

- works on interfaces that support IP protocol and have at least one IP address and on all ethernet−like interfaces even without IP addresses
- is enabled by default for all new Ethernet−like interfaces −− Ethernet, wireless, EoIP, IPIP tunnels, PPTP−static−server
- when older version on the RouterOS are upgraded from a version without discovery to a version with discovery, current Ethernet like interfaces will not be automatically enabled for MNDP
- uses UDP protocol port 5678
- an UDP packet with router info is broadcasted over the interface every 60 seconds
- every 30 seconds, the router checks if some of the neighbor entries are not stale
- if no info is received from a neighbor for more than 180 seconds the neighbor information is discarded

MikroTik Neighbour Discovery Protocol Setup

Submenu level : **/ip neighbor discovery**

Property Description

name (*read−only*: *name*)− interface name for reference
discover (yes | no; default: **yes**) − defines if discover is enabled or disabled

Example

To disable MNDP protocol on **Public** interface:

```
[admin@MikroTik] ip neighbor discovery> set Public discover=no
[admin@MikroTik] ip neighbor discovery> print
  # NAME      DISCOVER
  0 Public    no
  1 Local     yes
```

Listing the Discovered Neighbours

Submenu level : **/ip neighbor**

Property Description

interface (*read−only*: *name*) − local interface the neighbor is connected to
address (*read−only*: *address*) − IP address of the neighbor router

mac−address (*read−only*: *mac−address*) − MAC−address of the neighbor router
identity (*read−only*: *string*) − identity of the neighbour router
version (*read−only*: *string*) − router version of the neighbour router
unpack (*read−only*: none | simple | compress−headers | compress−all) − identifies if the interface of the neighbour router is unpacking 'Packed Packets'

Example

To view the table of discovered neighbours:

```
[admin@MikroTik] ip neighbor> print
  # INTERFACE ADDRESS         MAC−ADDRESS       IDENTITY   VERSION
  0 eth100... 10.5.2.100      00:04:EA:C6:0E:6F HP_10.5... Revisio...
  1 jevg_v... 10.5.1.1        00:40:96:58:20:14 0040965... Cisco 3...
  2 local_... 10.5.5.50       00:40:63:C1:23:C4 10.5.7.1   2.7rc4
  3 local_... 10.5.5.51       00:E0:C5:6E:23:25 GW_10.5... 2.7rc4
[admin@MikroTik] ip neighbor> print detail
  0 interface=eth100−temp address=10.5.2.100 mac−address=00:04:EA:C6:0E:6F
    identity="HP_10.5.2.100 Basement(0004ea−c60e40)" platform="HP 2524"
    version="Revision F.02.11 /sw/code/build/info(f00)" unpack=none age=12s

  1 interface=jevg_vlan2 address=10.5.1.1 mac−address=00:40:96:58:20:14
    identity="004096582014platform="AIR−BR350"
    version="Cisco 350 Series Bridge 11.21" unpack=none age=34s

  2 interface=local_vlan5 address=10.5.5.50 mac−address=00:40:63:C1:23:C4
    identity="10.5.7.1" platform="MikroTik" version="2.7rc4" unpack=none
    age=48s

  3 interface=local_vlan5 address=10.5.5.51 mac−address=00:E0:C5:6E:23:25
    identity="GW_10.5.51.1" platform="MikroTik" version="2.7rc4" unpack=none
    age=45s

[admin@MikroTik] ip neighbor>
```

As you can see, not only MikroTik RouterOS routers were discovered, but HP Procurve 2524 switch and Cisco 350 Series Wireless Bridge

Firewall Filters and Network Address Translation (NAT)

Document revision 1.12 (06−Sep−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The firewall supports filtering and security functions that are used to manage data flows to the router, through the router, and from the router. Along with the Network Address Translation it serve as security tools for preventing unauthorized access to networks.

Specifications

Packages required : ***system***
Licence required : ***Any***
Home menu level : ***/ip firewall***
Protocols utilized : *IP* (RFC791)
Hardware usage : *Increases with rules count*

Related documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
IP Route Management

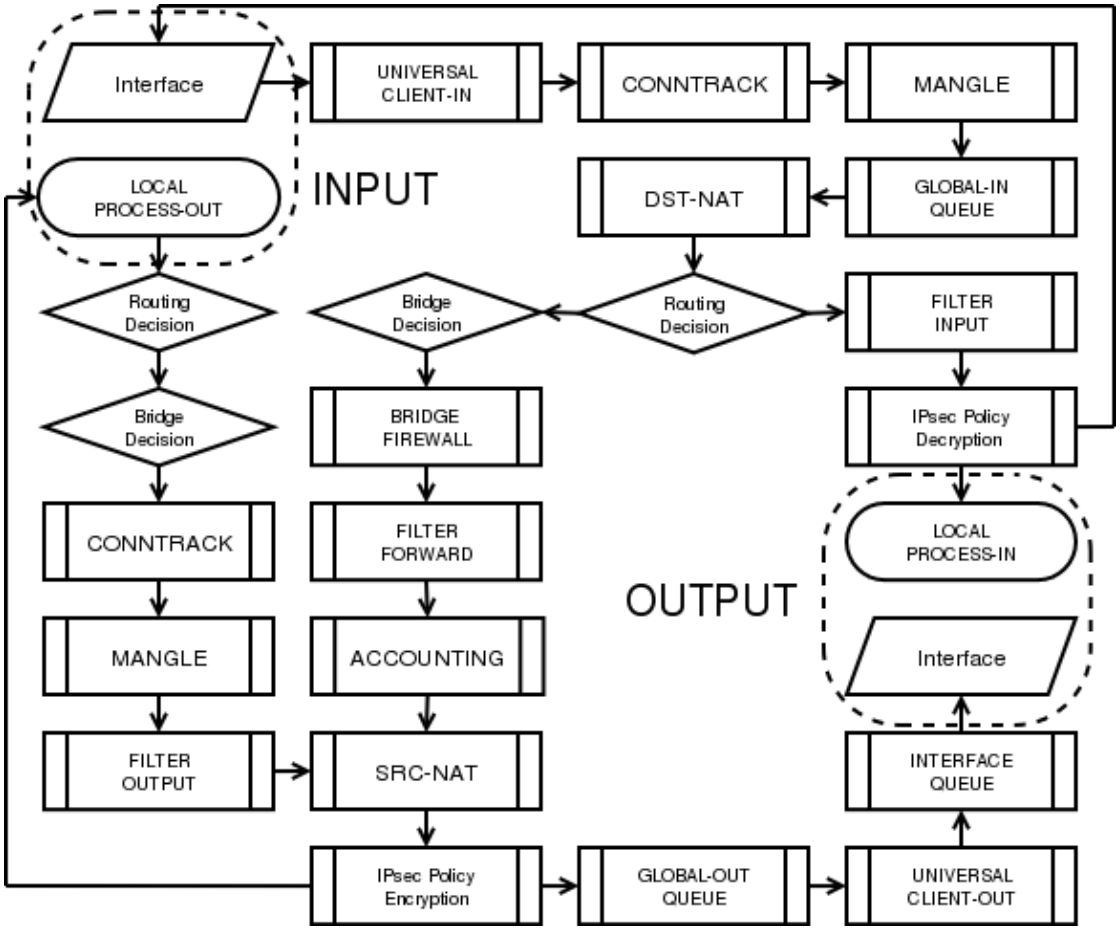
Description

Network firewalls keep outside threats away from sensitive data available inside the network. Whenever different networks are joined together, there is always a threat that someone from outside of your network will break into your LAN. Such break−ins may result in private data being stolen and distributed, valuable data being altered or destroyed, or entire hard drives being erased. Firewalls are used as a means of preventing or minimizing the security risks inherent in connecting to other networks. MikroTik RouterOS implements wide firewalling features as well as masquerading capabilities, which allows you to hide your network infrastructure from outside world.

Packet Flow

Description

MikroTik RouterOS simplifies the creation and deployment of a sophisticated firewall policies. In fact, you can easily create a simple one to filter your traffic or enable source NAT without need to know how packets are processed in the router. But in case you want to create more complicated policies, it is worth to know the underlying process details. IP packet flow through the router is depicted in the following diagram:



As we can see, a packet can enter the conveyer in two ways: whether the packet has come from an interface or whether it has been originated by the router. Analogically, a packet has two ways to leave the conveyer: through an outgoing interface or, in case the packet is locally destined, in the local process.

When the packet arrives to the router's interface, firewall rules are applied in the following order:

- The NAT rules are applied first. The firewall rules of the input chain and routing are applied after the packet has passed the NAT rule set.
- If the packet should be forwarded through the router, the firewall rules of the forward chain are applied next.
- When a packet leaves an interface, firewall rules of the output chain are applied first, then the NAT rules and queuing.

Additional arrows from IPsec boxes shows the processing of encrypted packets (they need to be encrypted / decrypted first and then processed as usual, *id est* from the point an typical packet enters the router).

If the packet is bridged one, the 'Routing Decision' changes to 'Bridge Forwarding Decision'. And in case the bridge is forwarding non−IP packets, all things regarding IP protocol are not applicable ('Universal Client', 'Conntrack', 'Mangle', *et cetera*).

Firewall Setup

Submenu level : **/ip firewall**
Firewall can be managed through the WinBox Console as well. Go to **IP Firewall** and select the desired chain. Press the **List** button to access the rules of the selected chain.

Description

To view the byte and packet counters, use commands **print bytes** and **print packets**, correspondingly. To reset the counters, use the command **reset−counters**.

Firewall Chains

Submenu level: **/ip firewall**

Description

The firewall filtering rules are grouped together in chains. It is very advantageous, if packets can be matched against one common criterion in one chain, and then passed over for processing against some other common criteria to another chain. Let us assume that, for example, packets must be matched against the IP addresses and ports. Then matching against the IP addresses can be done in one chain without specifying the protocol ports. Matching against the protocol ports can be done in a separate chain without specifying the IP addresses.

The chain **input** is used to process packets entering the router through one of the interfaces with the destination of the router. Packets passing through the router are not processed against the rules of the input chain.

The chain **forward** is used to process packets passing through the router.

The chain **output** is used to process originated from the router and leaving it through one of the interfaces. Packets passing through the router are not processed against the rules of the output chain.
These three chains cannot be deleted.

When processing a chain, rules are taken from the chain in the order they are listed there from the top to the bottom. If it matches the criteria of the rule, then the specified action is performed on the packet, and no more rules are processed in that chain. If the packet has not matched any rule within the chain, then the default policy action of the chain is performed.

The available policy actions are:

- **accept** − Accept the packet
- **drop** − Silently drop the packet (without sending the ICMP reject message)
- **none** − N/A

You can change the chain policies by using the **/ip firewall set** command.

Usually packets should be matched against several criteria. More general filtering rules can be grouped together in a separate chain. To process the rules of additional chains, the **jump** action should be used to this chain from another chain.

The policy of user added chains is **none**, and it cannot be changed. Chains cannot be removed, if they contain rules (are not empty).

Notes

- Because the NAT rules are applied first, it is important to hold this in mind when setting up firewall rules, since the original packets might be already modified by the NAT.
- The packets passing through the router are not processed against the rules of neither the input, nor output chains!
- Be careful about changing the default policy action to these chains! You may lose the connection to the router, if you change the policy to drop, and there are no rules in the chain, that allow connection to the router.

Example

The list of currently defined chains can be viewed using the **/ip firewall print** command:

```
[admin@MikroTik] ip firewall> print
  # NAME                                                                POLICY
  0 input                                                               accept
  1 forward                                                             accept
  2 output                                                              accept
[admin@MikroTik] ip firewall>
```

To add a new chain, use the **/ip firewall add** command:

```
[admin@MikroTik] ip firewall> add name=router
[admin@MikroTik] ip firewall> print
  # NAME                                                                POLICY
  0 input                                                               accept
  1 forward                                                             accept
  2 output                                                              accept
  3 router                                                              none
[admin@MikroTik] ip firewall>
```

Firewall Rules

Submenu level : **/ip firewall rule *chain_name***

Description

Management of the firewall rules can be accessed by selecting the desired chain. If you use the WinBox console, select the desired chain and then press the **List** button on the toolbar to open the window with the rules.

Property Description

action (accept | drop | jump | passthrough | reject | return; default: **accept**) − ation to undertake if the packet matches the rule, one of the:

- **accept** − accept the packet. No action, *id est*, the packet is passed through without undertaking any action, except for mangle, and no more rules are processed in the relevant list/chain
- **drop** − Silently drop the packet (without sending the ICMP reject message)
- **jump** − Jump to the chain specified by the value of the **jump−target** argument
- **passthrough** − ignore this rule, except for mangle, go on to the next one Acts the same way as a disabled rule, except for ability to count and mangle packets
- **reject** − reject the packet and send an ICMP reject message
- **return** − return to the previous chain, from where the **jump** took place

disabled (yes | no; default: **no**) − is the rule disabled or not
in−interface (*name*; default: **all**) − interface the packet has entered the router through. If the default value **all** is used, it may include the local loopback interface for packets originated from the router
out−interface (*name*, default: **all**) − interface the packet is leaving the router from. If the default value **all** is used, it may include the local loopback interface for packets with destination to the router
src−port (*port*) − source port number or range (0−65535). 0 means all ports 1−65535
comment (*text*; default: "") − a descriptive comment for the rule
dst−address (*IP adress*; default: **0.0.0.0/0:0−65535**) − destination IP address
jump−target (*name*) − Name of the target chain, if the **action=jump** is used
tcp−options (any | syn−only | non−syn−only; default: **any**) − TCP options
connection (*text*; default: "") − connection mark to match. Only connections (including related) marked in the MANGLE would be matched
dst−netmask (*IP address*) − destination netmask in decimal form x.x.x.x
limit−burst (*integer*; default: **0**) − allowed burst regarding the limit−count/limit−time
protocol (ah | egp | ggp | icmp | ipencap | ospf | rspf | udp | xtp | all | encap | gre | idpr−cmtp | ipip | pup | st | vmtp | ddp | esp | hmp | igmp | iso−tp4 | rdp | tcp | xns−idp; default: **any**) − protocol setting. The value **all** cannot be used, if you want to specify ports
connection−state (any | established | invalid | new | related; default: **any**) − connection state.
dst−port (*integer*) − destination port number or range (0−65535). 0 means all ports 1−65535
limit−count (*integer*; default: **0**) − how many times to use the rule during the **limit−time** period
src−address (*IP adress*; default: **0.0.0.0/0:0−65535**) − source IP address
content (*text*; default: "") − the text packets should contain in order to match the rule
flow − flow mark to match. Only packets marked in the MANGLE would be matched
limit−time (*time*; default **0**) − time interval, used in **limit−count**
src−mac−address (*MAC adress*; default: **00:00:00:00:00:00**) − host's MAC address the packet has been received from.
icmp−options (default: **any:any**) − ICMP options
log (yes | no; default: **no**) − specifies, to log the action or not
src−netmask (*IP address*) − source netmask in decimal form x.x.x.x

Notes

Keep in mind, that **protocol** must be explicity specified, if you want to select **port**.

Example

For instance, we want to reject packets with **dst−port=8080**:

```
[admin@MikroTik] ip firewall rule input> add dst−port=8080 protocol=tcp action=reject
[admin@MikroTik] ip firewall rule input> print
Flags: X − disabled, I − invalid
  0   src−address=0.0.0.0/0:0−65535 in−interface=all
      dst−address=0.0.0.0/0:8080 out−interface=all protocol=tcp
      icmp−options=any:any tcp−options=any connection−state=any flow=""
```

```
      sconnection="" content="" rc-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=reject log=no

[admin@MikroTik] ip firewall rule input>
```

Logging the Firewall Actions

To enable logging of the firewall actions you should set the value of the rule argument **log** to **yes**. Also, the logging facility should be enabled for firewall logs:

```
[admin@MikroTik] system logging facility> set Firewall-Log logging=local
[admin@MikroTik] system logging facility> print
  # FACILITY          LOGGING PREFIX           REMOTE-ADDRESS  REMOTE-PORT ECH
  0 Firewall-Log      local                                                no
  1 System-Info       local                                                no
  2 System-Error      local                                                no
  3 System-Warning    local                                                no
  4 Prism-Info        local                                                no
  5 Web-Proxy-Access  local                                                no
  6 Hotspot-Account   local                                                no
  7 OSPF-Info         local                                                no
  8 Hotspot-Error     local                                                no
  9 IPsec-Event       local                                                no
 10 IKE-Event         local                                                no
 11 IPsec-Warning     local                                                no
 12 System-Echo       local                                                yes

[admin@MikroTik] system logging facility>
```

You can send UDP log messages to a remote syslog host by specifying the remote address and port (usually 514). Local logs can be viewed using the **/log print** command:

```
[admin@MikroTik] > log print without-paging
...
 mar/11/2003 17:44:55 chain added by admin
 mar/11/2003 17:45:51 rule added by admin
 mar/11/2003 18:00:26 web proxy cache size is limited by memory size

[admin@MikroTik] >
```

Network Address Translation

Description

Network Address Translation (NAT) provides ways for hiding local networks as well as to maintain public services on servers from these networks. Besides, through NAT additional applications like transparent proxy service can be made.

Property Description

The **src-nat** and the **dst-nat** have some common properties listed below. In turn, properties specific to each type of NAT will be listed under appropriate headers.

dst-address (*IP adress*; default: **0.0.0.0/0:0-65535**) - destination IP address
src-address (*IP adress*; default: **0.0.0.0/0:0-65535**) - source IP address

flow − flow mark to match. Only packets marked in the MANGLE would be matched
limit−time (*time*; default **0**) − time interval, used in **limit−count**
protocol (ah | egp | ggp | icmp | ipencap | ospf | rspf | udp | xtp | all | encap | gre | idpr−cmtp | ipip | pup | st | vmtp | ddp | esp | hmp | igmp | iso−tp4 | rdp | tcp | xns−idp; default: **any**) − protocol setting. The value **all** cannot be used, if you want to specify ports
icmp−options (default: **any:any**) − ICMP options
content (*text*; default: "") − the text packets should contain in order to match the rule
comment (*text*; default: "") − a descriptive comment for the rule
connection (*text*; default: "") − connection mark to match. Only connections (including related) marked in the MANGLE would be matched
limit−burst (*integer*; default: **0**) − allowed burst regarding the limit−count/limit−time
limit−count(*integer*; default: **0**) − how many times to use the rule during the **limit−time** period
src−netmask (*IP address*) − source netmask in decimal form x.x.x.x
src−port (*port*) − source port number or range (0−65535). 0 means all ports 1−65535
dst−netmask (*IP address*) − destination netmask in decimal form x.x.x.x
dst−port (*integer*) − destination port number or range (0−65535). 0 means all ports 1−65535

Masquerading and Source NAT

Submenu level : **/ip firewall src−nat**

Description

Masquerading is a firewall function that can be used to 'hide' private networks behind one external IP address of the router. For example, masquerading is useful, if you want to access the ISP's network and the Internet appearing as all requests coming from one single IP address given to you by the ISP. The masquerading will change the source IP address and port of the packets originated from the private network to the external address of the router, when the packet is routed through it.

Masquerading helps to ensure security since each outgoing or incoming request must go through a translation process that also offers the opportunity to qualify or authenticate the request or match it to a previous request. Masquerading also conserves the number of global IP addresses required and it lets the whole network use a single IP address in its communication with the world.

Property Description

action (accept | masquerade | nat; default: **accept**) − action to undertake if a packed matched a particular **src−nat** rule, one of the:

- **accept** − Accept the packet. No action, *id est*, the packet is passed through without undertaking any action, except for mangle, and no more rules are processed in the relevant list/chain
- **masquerade** − use masquerading for the packet and substitute the source address:port of the packet with the ones of the router. In this case, the **to−src−address** argument value is not taken into account and it does not need to be specified, since the router's local address is used
- **nat** − perform Network Address Translation. The **to−src−address** should be specified (not required with **action=masquerade**)

out−interface (*name*; default: **all**) − interface the packet is leaving the router from. If the default value **all** is used, it may include the local loopback interface for packets with destination to the router
to−src−address (*IP address*; default: **0.0.0.0**) − source address to replace original source address with
to−src−port (*integer*; default: **0−65535**) − source port to replace original source port with

Example

To use masquerading, a source NAT rule with **action=masquerade** should be added to the **src−nat** rule set:

```
[admin@MikroTik] ip firewall src−nat> add src−address=10.5.91.0/24:0 \
\... out−interface=Public action=masquerade
[admin@MikroTik] ip firewall src−nat> print
Flags: X − disabled, I − invalid, D − dynamic
  0   src−address=10.5.91.0/24:0−65535 dst−address=0.0.0.0/0:0−65535
      out−interface=Public protocol=all icmp−options=any:any flow=""
      connection="" content="" limit−count=0 limit−burst=0 limit−time=0s
      action=masquerade to−src−address=0.0.0.0 to−src−port=0−65535

[admin@MikroTik] ip firewall src−nat>
```

If the packet matches the **masquerade** rule, then the router opens a connection to the destination, and sends out a modified packet with its own address and a port allocated for this connection. The router keeps track about masqueraded connections and performs the "demasquerading" of packets, which arrive for the opened connections. For filtering purposes, you may want to specify the **to−src−ports** argument value, say, to **60000−65535**.

If you want to change the source *address:port* to specific *adress:port*, use the **action=nat** instead of **action=masquerade**:

```
[admin@MikroTik] ip firewall src−nat> add src−address=192.168.0.1/32 action=nat \
\... out−interface=Public to−src−address=10.10.10.5
[admin@MikroTik] ip firewall src−nat> print
Flags: X − disabled, I − invalid, D − dynamic
  4   src−address=192.168.0.1/32:0−65535 dst−address=0.0.0.0/0:0−65535
      out−interface=Public protocol=all icmp−options=any:any flow=""
      connection="" content="" limit−count=0 limit−burst=0 limit−time=0s
      action=nat to−src−address=10.10.10.5 to−src−port=0−65535

[[admin@MikroTik] ip firewall src−nat>
```

Here, the
src−address − can be IP host's address, for example, 192.168.0.1/32, or network address 192.168.0.0/24
to−src−address − can be one address, or a range, say 10.0.0.217−10.0.0.219. The addresses should be added to the router's interface, or should be routed to it from the gateway router.

The source nat can masquerade several private networks, and use individual **to−src−address** for each of them.

Redirection and Destination NAT

Submenu level : **/ip firewall dst−nat**

Description

Redirection and destination NAT should be used when you need to give access to services located on a private network from the outside world.

Property Description

action (accept | nat | redirect; default: **accept**) − action to undertake if a packed matched a particular **dst−nat** rule, one of the:

- **accept** − Accept the packet. No action, *id est*, the packet is passed through without undertaking any action, except for mangle, and no more rules are processed in the relevant list/chain
- **redirect** − redirects to the local address:port of the router. In this case, the **to−dst−address** argument value is not taken into account and it does not need to be specified, since the router's local address is used.
- **nat** − perform Network Address Translation. The **to−dst−address** should be specified (not required with **action=redirect**)

in−interface (*name*; default: **all**) − interface the packet has entered the router through. If the default value **all** is used, it may include the local loopback interface for packets originated from the router
to−dst−port (*integer*; default: **0−65535**) − destination port to replace original with
src−mac−address (*MAC adress*; default: **00:00:00:00:00:00**) − host's MAC address the packet has been received from
to−dst−address (*IP address*; default: **0.0.0.0**) − destination IP address to replace original with

Example

To add a destination NAT rule that gives access to the http server 192.168.0.4 on the local network via external address 10.0.0.217, use the following command:

```
[admin@MikroTik] ip firewall dst−nat> add action=nat protocol=tcp \
\... dst−address=10.0.0.217/32:80 to−dst−address=192.168.0.4
[admin@MikroTik] ip firewall dst−nat> print
Flags: X − disabled, I − invalid, D − dynamic
  0   src−address=0.0.0.0/0:0−65535 in−interface=all
      dst−address=10.0.0.217/32:80 protocol=tcp icmp−options=any:any flow=""
      connection="" content="" src−mac−address=00:00:00:00:00:00
      limit−count=0 limit−burst=0 limit−time=0s action=nat
      to−dst−address=192.168.0.4 to−dst−port=0−65535

[admin@MikroTik] ip firewall dst−nat>
```

Here, if you want to redirect to the router's local address, use **action=redirect** and do not specify the **to−dst−address**.

Understanding REDIRECT and MASQUERADE

REDIRECT is similar to regular destination NAT in the same way as MASQUERADING is similar to source NAT − masquerading is source NAT, except you do not have to specify **to−src−address** − outgoing interface address is used automatically. The same with REDIRECT − it is destination NAT where **to−dst−address** is not used − incoming interface address is used instead. So there is no use of specifying **to−src−address** for src−nat rules with **action=masquerade**, and no use of specifying **to−dst−address** for dst−nat rules with **action=redirect**. **Note** that **to−dst−port** is meaningful for REDIRECT rules − this is port on which service on router that will handle these requests is sitting (e.g. web proxy).

When packet is dst−natted (no matter − **action=nat** or **action=redirect**), dst address is changed. Information about translation of addresses (including original dst address) is kept in router's internal tables. Transparent web proxy working on router (when web requests get redirected to proxy port on router) can access this information from internal tables and get address of web server from them. If you are dst−natting to some different proxy server, it has no way to find web server's address from IP header (because dst

address of IP packet that previously was address of web server has changed to address of proxy server). Starting from HTTP/1.1 there is special header in HTTP request which tells web server address, so proxy server can use it, instead of dst address of IP packet. If there is no such header (older HTTP version on client), proxy server can not determine web server address and therefore can not work.

It means, that it is impossible to correctly transparently redirect HTTP traffic from router to some other transparent−proxy box. Only correct way is to add transparent proxy on the router itself, and configure it so that your "real" proxy is parent−proxy. In this situation your "real" proxy does not have to be transparent any more, as proxy on router will be transparent and will forward proxy−style requests (according to standard; these requests include all necessary information about web server) to "real" proxy.

Marking the Packets (Mangle) and Changing the MSS

Submenu level : **/ip firewall mangle**

Description

Packets entering the router can be marked for further processing them against the rules of firewall chains, source or destination NAT rules, as well as for applying queuing to them.

It is also possible to mark the packets associated (including related) with the same connection as the marked packet (in other words, to mark a connection with all related connections, you need to mark only one packet belonging to that connection).

You may also want to change the TCP Maximum Segment Size (MSS), to a value which is your desired MTU value less 40. The MSS can be set only for TCP SYN packets.

Property Description

action (accept | passthrough; default: **accept**) − ation to undertake if the packet matches the rule, one of the:

- **accept** − accept the packet applying the appropriate attributes (marks, MSS), and no more rules are processed in the list
- **passthrough** − apply the appropriate attributes (marks, MSS), and go on to the next rule

disabled (yes | no; default: **no**) − is the rule disabled or not
in−interface (*name*; default: **all**) − interface the packet has entered the router through. If the default value **all** is used, it may include the local loopback interface for packets originated from the router
src−address (*IP adress*; default: **0.0.0.0/0:0−65535**) − source IP address
src−netmask (*IP address*) − source netmask in decimal form x.x.x.x
src−port (*port*) − source port number or range (0−65535). 0 means all ports 1−65535
comment (*text*; default: "") − a descriptive comment for the rule
dst−address (*IP adress*; default: **0.0.0.0/0:0−65535**) − destination IP address
dst−netmask (*IP address*) − destination netmask in decimal form x.x.x.x
dst−port (*integer*) − destination port number or range (0−65535). 0 means all ports 1−65535
tcp−options (any | syn−only | non−syn−only; default: **any**) − TCP options
icmp−options (default: **any:any**) − ICMP options
protocol (ah | egp | ggp | icmp | ipencap | ospf | rspf | udp | xtp | all | encap | gre | idpr−cmtp | ipip | pup | st | vmtp | ddp | esp | hmp | igmp | iso−tp4 | rdp | tcp | xns−idp; default: **any**) − protocol setting. The value **all** cannot be used, if you want to specify ports
content (*text*; default: "") − the text packets should contain in order to match the rule
flow (*text*; default: "") − flow mark to match. Only packets marked in the MANGLE would be matched

connection (*text*; default: "") − connection mark to match. Only connections (including related) marked in the MANGLE would be matched
limit−burst (*integer*; default: **0**) − allowed burst regarding the limit−count/limit−time
limit−count (*integer*; default: **0**) − how many times to use the rule during the **limit−time** period
limit−time (*time*; default **0**) − time interval, used in **limit−count**
src−mac−address (*MAC adress*; default: **00:00:00:00:00:00**) − host's MAC address the packet has been received from.
log (yes | no; default: **no**) − specifies, to log the action or not
mark−flow (*text*; default: "") − change flow−mark of the packet to this value
mark−connection (*text*; default: "") − change connection−mark of the packet to this value
tcp−mss (*intereg* | dont−change; default: **dont−change** − change MSS of the packet or:

- **dont−change** − leave MSS of the packet as is

Example

Specify the value for the **mark−flow** argument and use **action=passthrough**, for example:

```
[admin@MikroTik] ip firewall mangle> add action=passthrough mark−flow=abc−all
[admin@MikroTik] ip firewall mangle> print
Flags: X − disabled, I − invalid, D − dynamic
  0   src−address=0.0.0.0/0:0−65535 in−interface=all
      dst−address=0.0.0.0/0:0−65535 protocol=all tcp−options=any
      icmp−options=any:any flow="" connection="" content=""
      src−mac−address=00:00:00:00:00:00 limit−count=0 limit−burst=0
      limit−time=0s action=passthrough mark−flow=abc−all tcp−mss=dont−change
      mark−connection=""

[admin@MikroTik] ip firewall mangle>
```

To change the MSS, adjust the **tcp−mss** argument. For example, if your if you have encrypted PPPoE link with MTU = 1492, you can set the mangle rule as follows:

```
[admin@MikroTik] ip firewall mangle> add protocol=tcp tcp−options=syn−only\
\.. action=passthrough tcp−mss=1448
[admin@MikroTik] ip firewall mangle> print
Flags: X − disabled, I − invalid, D − dynamic
  0   src−address=0.0.0.0/0:0−65535 in−interface=all
      dst−address=0.0.0.0/0:0−65535 protocol=tcp tcp−options=syn−only
      icmp−options=any:any flow="" connection="" content=""
      src−mac−address=00:00:00:00:00:00 limit−count=0 limit−burst=0
      limit−time=0s action=passthrough mark−flow="" tcp−mss=1448
      mark−connection=""

[admin@MikroTik] ip firewall mangle>
```

Connection Tracking

Submenu level : **/ip firewall connection**

Description

This feature provides a facility for monitoring connections made through the router and their states.

Property Description

src−address (read−only: *IP address:port*) − the source address and port the connection is established from
dst−address (read−only: *IP address:port*) − the destination address and port the connection is established to
protocol (read−only: *text*) − IP protocol name or number
tcp−state (read−only: *text*) − the state of TCP connection
timeout (read−only: *time*) − the amount of time until the connection will be timed out
reply−src−address (read−only: *IP address:port*) − the source address and port the reply connection is established from
reply−dst−address (read−only: *IP address:port*) − the destination address and port the reply connection is established to
assured (*read−only*: true | false) − shows whether the connection is assured
icmp−id (*read−only*: *integer*) − contains the ICMP ID. Each ICMP packet gets an ID set to it when it is sent, and when the receiver gets the ICMP message, it sets the same ID within the new ICMP message so that the sender will recognize the reply and will be able to connect it with the appropriate ICMP request
icmp−option (*read−only*: *integer:integer*) − the ICMP type and code fields
reply−icmp−id (*read−only*: *integer*) − contains the ICMP ID of received packet
reply−icmp−option (*read−only*: *integer:integer*) − the ICMP type and code fields of received packet
unreplied (*read−only*: true | false) − shows whether the request was unreplied

Connection timeouts

Here comes a list of connection timeouts:

- TCP SYN sent (First stage in establishing a connection) = 2min.
- TCP SYN recvd (Second stage in establishing a connection) = 60sec.
- Established TCP connections (Third stage) = 5 days.
- TCP FIN wait (connection termination) = 2min.
- TCP TIME wait (connection termination) = 2min.
- TCP CLOSE (remote party sends RTS) = 10sec.
- TCP CLOSE wait (sent RTS) = 60sec.
- TCP LAST ACK (received ACK) = 30sec.
- TCP Listen (ftp server waiting for client to establish data connection) = 2min.
- UDP timeout = 30sec.
- UDP with reply timeout (remote party has responded) = 180sec.
- ICMP timeout = 30sec.
- All other = 10min.

Example

```
[admin@MikroTik] ip firewall connection> print
Flags: U − unreplied, A − assured
  #    SRC−ADDRESS           DST−ADDRESS           PR.. TCP−STATE   TIMEOUT
  0  A 10.5.91.205:1361      10.5.0.23:22          tcp  established 4d23h59m55s
  1  A 10.5.91.205:1389      10.5.5.2:22           tcp  established 4d23h59m21s
  2  A 10.5.91.205:1373      10.5.91.254:3986      tcp  established 4d23h59m56s
  3  A 10.5.91.205:1377      159.148.172.3:23      tcp  established 4d23h35m14s
  4  A 80.232.241.3:1514     159.148.172.204:1723  tcp  established 4d23h59m53s
  5    159.148.172.204       80.232.241.3          47               9m21s
[admin@MikroTik] ip firewall connection>
```

Service Ports

Submenu level : **/ip firewall service−port**

Description

This submenu allows to configure Connection Tracking 'helpers' for various protocols. They are used to provide correct NAT traversal for the traffic of these protocols.

Property Description

name (*read−only*: *name*) − protocol name
ports (*integer*) − port range that is used by the protocol

Example

To disable **h323** service port:

```
[admin@MikroTik] ip firewall service−port> set h323 disabled=yes
[admin@MikroTik] ip firewall service−port> print
Flags: X − disabled
  #   NAME                                                                PORTS
  0   ftp                                                                 21
  1   pptp
  2   gre
  3 X h323
  4   mms
  5   irc                                                                 6667
  6   quake3

[admin@MikroTik] ip firewall service−port>
```

Troubleshooting

- *I set the policy for the input chain to **drop**, and I lost connection to the router*
  You should add rules to the chain allowing required communications, and only then change the default policy of the chain!
- *I put up filtering rules, but they seem not to work*
  Use the Firewall logging to see, whether you are matching the packets with your rules or not! The most common mistake is wrong address/netmask, e.g., 10.0.0.217/24 (wrong), 10.0.0.217/32 (right), or 10.0.0.0/24 (right).
- *I am trying to use policy routing based on source addresses and masquerading, but it does not work.*
  Masqueraded packets have source address 0.0.0.0 at the moment when they are processed according to the routing table. Therefore it is not possible to have masquerading with different source address. See the Routes Manual for more information.

General Network Suggestions

- Implement an environment where users are required to log on to use computer resources. This provides a foundation from which suspicious activity can be traced.
- Make use of ***HotSpot*** technology. Doing so provides safe, yet flexible network resources access to end user.

- Provide sufficient training to end−users. Especially be sure that users are aware of the dangers of not logging off their computers. Such dangers include the ability of a third−party to sit at an "open" computer and assume the user’s identity. The unauthorized person has all the rights and privileges of the logged in user. Any suspicious activity will be traced back to the user’s login, not to the unauthorized person.
- Make use of user activities and system activity logs analysis. Doing so enables the organization to detect suspicious activity before a full−blown break−in occurs.
- Some public structures like libraries, univercities, airport and some schools have "public" computers anyone can use. In order to minimize the threat of unauthorized access to network resources, install these computers on a "public" network segment, so that internal network resources can not be reachable without authorization.

IP Firewall Applications

In this section some IP firewalling common applications and examples of them are discussed.

Basic Firewall Building Principles

Assume we have a router that connects a customer's network to the Internet. The basic firewall building principles can be grouped as follows:

- **Protection of the router from unauthorized access**
  Connections to the addresses assigned to the router itself should be monitored. Only access from certain hosts to certain TCP ports of the router should be allowed.

  This can be done by putting rules in the input chain to match packets with the destination address of the router entering the router through all interfaces.
- **Protection of the customer's hosts**
  Connections to the addresses assigned to the customer's network should be monitored. Only access to certain hosts and services should be allowed.

  This can be done by putting rules in the forward chain to match packets passing through the router with the destination addresses of customer's network.
- **Using source NAT (masquerading) to 'Hide' the Private Network behind one External Address**
  All connections form the private addresses are masqueraded, and appear as coming from one external address − that of the router.

  This can be done by enabling the masquerading action for source NAT rules.
- **Enforcing the Internet Usage Policy from the Customer's Network**
  Connections from the customer's network should be monitored.

  This can be done by putting rules in the forward chain, or/and by masquerading (source NAT) only those connections, that are allowed.

Filtering has some impact on the router's performance. To minimize it, the filtering rules that match packets for established connections should be placed on top of the chain. These are TCP packets with options **non−syn−only**.

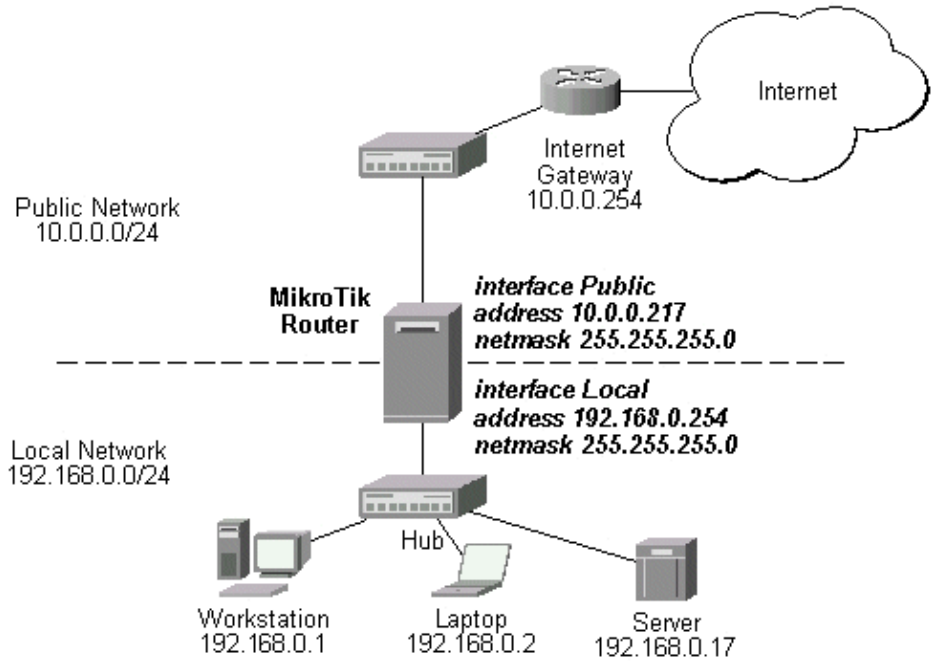
Examples of setting up firewalls are discussed below.

Example of Firewall Filters

Assume we want to create a firewall, that:

- protects the MikroTik router from unauthorized access from anywhere. Only access from the 'trusted' network 10.5.8.0/24 is allowed.
- protects the customer's hosts within the network 192.168.0.0/24 from unauthorized access from anywhere.
- gives access from the Internet to the http and smtp services on 192.168.0.17
- Allows only ICMP ping from all customer's hosts and forces use of the proxy server on 192.168.0.17

The basic network setup is in the following diagram:



The IP addresses and routes of the MikroTik router are as follows:

```
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.217/24      10.0.0.0        10.0.0.255      Public
  1   192.168.0.254/24   192.168.0.0     192.168.0.255   Local
[admin@MikroTik] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.254      1        Public
    1 DC 192.168.0.0/24     r 0.0.0.0         0        Local
    2 DC 10.0.0.0/24        r 0.0.0.0         0        Public
[admin@MikroTik] >
```

Protecting the Router

To protect the router from unauthorized access, we should filter out all packets with the destination addresses of the router, and accept only those are allowed. Since all packets with destination to the router's address are processed against the input chain, we can add the following rules to it:

```
[admin@MikroTik] > ip firewall rule input
[admin@MikroTik] ip firewall rule input> add protocol=tcp tcp-option=non-syn-only \
\... connection-state=established comment="Allow established TCP connections"
[admin@MikroTik] ip firewall rule input> add protocol=udp comment="Allow UDP connections"
[admin@MikroTik] ip firewall rule input> add protocol=icmp comment="Allow ICMP messages"
[admin@MikroTik] ip firewall rule input> add src-addr=10.5.8.0/24 \
\... comment="Allow access from 'trusted' network 10.5.8.0/24"
[admin@MikroTik] ip firewall rule input> add action=reject log=yes \
\... comment="Reject and log everything else"
[admin@MikroTik] ip firewall rule input> print
Flags: X - disabled, I - invalid, D - dynamic
  0   ;;; Allow established TCP connections
      src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=all protocol=tcp
      icmp-options=any:any tcp-options=non-syn-only
      connection-state=established flow="" connection="" content=""
      src-mac-address=00:00:00:00:00:00 limit-count=0 limit-burst=0
      limit-time=0s action=accept log=no

  1   ;;; Allow UDP connections
      src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=all protocol=udp
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=accept log=no

  2   ;;; Allow ICMP messages
      src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=all protocol=icmp
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=accept log=no

  3   ;;; Allow access from 'trusted' network 10.5.8.0/24 of ours
      src-address=10.5.8.0/24:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=all protocol=all
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=accept log=no

  4   ;;; Reject and log everything else
      src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=all protocol=all
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=reject log=yes

[admin@MikroTik] ip firewall rule input>
```

Thus, the input chain will accept only allowed connections and reject and log everything else.

Protecting the Customer's Network

To protect the customer's network, we should match all packets with destination address 192.168.0.0/24 that are passing through the router. This can be done in the forward chain. We can match the packets against the IP addresses in the forward chain, and then jump to another chain, say, **customer**. We create the new chain and add rules to it:

```
[admin@MikroTik] ip firewall> add name=customer
[admin@MikroTik] ip firewall> print
  # NAME                                                                POLICY
  0 input                                                               accept
  1 forward                                                             accept
  2 output                                                              accept
  3 router                                                              none
  4 customer                                                            none
[admin@MikroTik] ip firewall> rule customer
[admin@MikroTik] ip firewall rule customer> add protocol tcp tcp-option non-syn-only \
\... connection-state=established comment="Allow established TCP connections"
[admin@MikroTik] ip firewall rule customer> add protocol udp \
\... comment="Allow UDP connections"
[admin@MikroTik] ip firewall rule customer> add protocol icmp \
\... comment="Allow ICMP messages"
[admin@MikroTik] ip firewall rule customer> add protocol tcp tcp-option syn-only \
\... dst-address 192.168.0.17/32:80 \
\... comment="Allow http connections to the server at 192.168.0.17"
[admin@MikroTik] ip firewall rule customer> add protocol tcp tcp-option syn \
\... dst-address 192.168.0.17/32:25 \
\... comment="Allow smtp connections to the server at 192.168.0.17"
[admin@MikroTik] ip firewall rule customer> add protocol tcp tcp-option syn \
\... src-port 20 dst-port 1024-65535 \
\... comment="Allow ftp data connections from servers on the Internet"
[admin@MikroTik] ip firewall rule customer> add action reject log yes \
\... comment="Reject and log everything else"
[admin@MikroTik] ip firewall rule customer> print
Flags: X - disabled, I - invalid
  0   ;;; Allow established TCP connections
      src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=all protocol=tcp
      icmp-options=any:any tcp-options=non-syn-only
      connection-state=established flow="" connection="" content=""
      src-mac-address=00:00:00:00:00:00 limit-count=0 limit-burst=0
      limit-time=0s action=accept log=no

  1   ;;; Allow UDP connections
      src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=all protocol=udp
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=accept log=no

  2   ;;; Allow ICMP messages
      src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=all protocol=icmp
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=accept log=no

  3   ;;; Allow http connections to the server at 192.168.0.17
      src-address=0.0.0.0/0:0-65535 in-interface=all
```

```
      dst-address=192.168.0.17/32:80 out-interface=all protocol=tcp
      icmp-options=any:any tcp-options=syn-only connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=accept log=no

  4   ;;; Allow smtp connections to the server at 192.168.0.17
      src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=192.168.0.17/32:25 out-interface=all protocol=tcp
      icmp-options=any:any tcp-options=syn-only connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=accept log=no

  5   ;;; Allow ftp data connections from servers on the Internet
      src-address=0.0.0.0/0:20 in-interface=all
      dst-address=0.0.0.0/0:1024-65535 out-interface=all protocol=tcp
      icmp-options=any:any tcp-options=syn-only connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=accept log=no

  6   ;;; Reject and log everything else
      src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=all protocol=all
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=reject log=yes

[admin@MikroTik] ip firewall rule customer>
```

Note about the rule #5: active ftp data connections are made from the server's port 20 to the client's tcp port above 1024.

All we have to do now is to put rules in the forward chain, that match the IP addresses of the customer's hosts on the Local interface and jump to the customer chain:

```
[admin@MikroTik] ip firewall rule forward> add out-interface=Local action=jump \
\... jump-target=customer
[admin@MikroTik] ip firewall rule forward> print
Flags: X - disabled, I - invalid
  0   src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=Local protocol=all
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=jump jump-target=customer log=no

[admin@MikroTik] ip firewall rule forward>
```

Thus, everything that passes the router and leaves the **Local** interface (destination of the customer's network) will be processed against the firewall rules of the **customer** chain.

Enforcing the "Internet Policy"

To force the customer's hosts to access the Internet only through the proxy server at 192.168.0.17, we should put following rules in the forward chain:

```
[admin@MikroTik] ip firewall rule forward> add protocol icmp out-interface Public \
\... comment="Allow ICMP ping packets"
[admin@MikroTik] ip firewall rule forward> add src-address 192.168.0.17/32 out-interface \
\... Public comment="Allow outgoing connections form the server at 192.168.0.17"
```

```
[admin@MikroTik] ip firewall rule forward> add action reject out-interface Public log yes \
\... comment="Reject and log everything else"
[admin@MikroTik] ip firewall rule forward> print
Flags: X - disabled, I - invalid
  0   src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=Local protocol=all
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=jump jump-target=customer log=no

  1   ;;; Allow ICMP ping packets
      src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=Public protocol=icmp
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=accept log=no

  2   ;;; Allow outgoing connections form the server at 192.168.0.17
      src-address=192.168.0.17/32:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=Public protocol=all
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=accept log=no

  3   ;;; Reject and log everything else
      src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 out-interface=Public protocol=all
      icmp-options=any:any tcp-options=any connection-state=any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=reject log=yes

[admin@MikroTik] ip firewall rule forward>
```

Example of Source NAT (Masquerading)

If you want to "hide" the private LAN 192.168.0.0/24 "behind" one address 10.0.0.217 given to you by the ISP (see the network diagram in the Application Example above), you should use the source network address translation (masquerading) feature of the MikroTik router. The masquerading will change the source IP address and port of the packets originated from the network 192.168.0.0/24 to the address 10.0.0.217 of the router when the packet is routed through it.

To use masquerading, a source NAT rule with action 'masquerade' should be added to the firewall configuration:

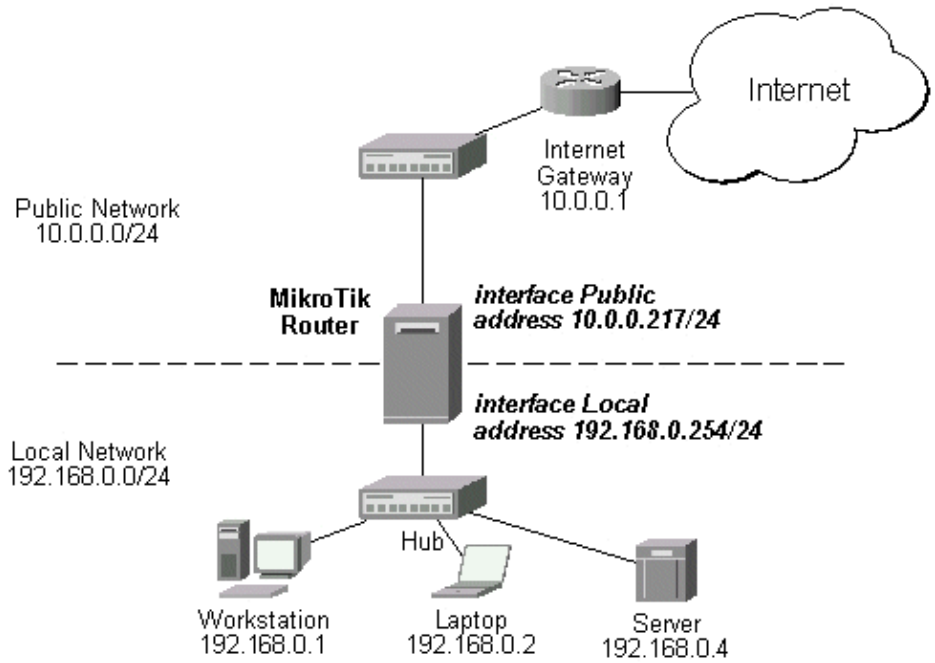
```
[admin@MikroTik] ip firewall src-nat> add action=masquerade out-interface=Public
[admin@MikroTik] ip firewall src-nat> print
Flags: X - disabled, I - invalid
  0   src-address=0.0.0.0/0:0-65535 dst-address=0.0.0.0/0:0-65535
      out-interface=Public protocol=all icmp-options=any:any flow=""
      connection="" content="" limit-count=0 limit-burst=0 limit-time=0s
      action=masquerade to-src-address=0.0.0.0 to-src-port=0-65535

[admin@MikroTik] ip firewall src-nat>
```

All outgoing connections from the network 192.168.0.0/24 will have source address 10.0.0.217 of the router and source port above 1024. No access from the Internet will be possible to the Local addresses. If you want to allow connections to the server on the local network, you should use destination Network

Address Translation (NAT).

Example of Destination NAT

Assume you need to configure the MikroTik router for the following network setup, where the server is located in the private network area:



The server has address 192.168.0.4, and we are running web server on it that listens to the TCP port 80. We want to make it accessible from the Internet at address:port 10.0.0.217:80. This can be done by means of destination Network Address Translation (NAT) at the MikroTik Router. The Public address:port 10.0.0.217:80 will be translated to the Local address:port 192.168.0.4:80. One destination NAT rule is required for translating the destination address and port:

```
[admin@MikroTik] ip firewall dst-nat> add action=nat protocol=tcp \
\... dst-address=10.0.0.217/32:80 to-dst-address=192.168.0.4
[admin@MikroTik] ip firewall dst-nat> print
Flags: X - disabled, I - invalid
  0   src-address=0.0.0.0/0:0-65535 in-interface=all
      dst-address=10.0.0.217/32:80 protocol=tcp icmp-options=any:any flow=""
      connection="" content="" src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=nat to-dst-address=192.168.0.4
      to-dst-port=0-65535

[admin@MikroTik] ip firewall dst-nat>
```

Additional Resources

Read about connection tracking at
http://www.cs.princeton.edu/~jns/security/iptables/iptables_conntrack.html
Read more about NAT in RFC2663

IP Route Management

Document revision 1.4 (01−Jun−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

The following Manual discusses managing the IP routes, equal−cost multi−path (ECMP) routing technique as well as policy−based routing, which give opportunity to select routes in order to restrict the use of network resources to certain classes of customers.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/ip route***, ***/ip policy−routing***
Protocols utilized : *IP (RFC791)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)

Packet Filter (Firewall) and Network Address Translation (NAT)

Description

MikroTik RouterOS has following types of routes:

- **Connected Routes** are created automatically when adding address to an interface. These routes specify networks, which can be accessed directly through the interface.
- **Static Routes** are user−defined routes that specify the router that can forward traffic to the specified network. They are useful for specifying the default gateway.

You do not need to add routes to networks directly connected to the router, since they are added automatically when adding the IP addresses. However, unless you use some routing protocol (RIP or OSPF), you may want to specify static routes to specific networks, or the default route.

More than one gateway for one destination network may be used. This approach ir called 'Equal Cost Multipath Routing' and is used for load balancing (**Note** that this does not provide failover). With equal cost multipath, a router potentially has several available next hops towards any given destination. A new gateway is chosen for each new source/destination IP pair. This means that, for example, one FTP connection will use only one link, but new connection to a different server will use other link. This also means that routes to often−used sites will always be over the same provider. But on big backbones this should distribute traffic fine. Also this has another good feature − single connection packets do not get reordered and therefore do not kill TCP performance.

Equal cost multipath routes can be created by routing protocols (RIP or OSPF), or adding a static route with multiple gateways (in the form **gateway=x.x.x.x,y.y.y.y**) The routing protocols may create routes with equal cost automatically, if the cost of the interfaces is adjusted properly. For more information on using the routing protocols, please read the corresponding section of the Manual.

Static Routes

Submenu level : **/ip route**

Property Description

dst−address (*IP address/mask*) − destination address and network mask, where netmask is number of bits which indicate natwork number
netmask (*IP address*) − network mask
gateway (*IP address*) − gateway host, that can be reached directly through some of the interfaces. You can specify multiple gateways separated by comma "," for equal cost multipath routes. See more information on that below
preferred−source (*IP address*; default: **0.0.0.0**) − source address of packets leaving the router via this route. Must be a valid address of the router, which is assigned to the router's interface, through which the packet leaves

- **0.0.0.0** − determined at the time of sending the packet out through the interface
  distance (*integer*; default: **1**) − administrative distance of the route. When forwarding a packet the router will use the route with the lowest administrative distance and reachable gateway.

  Statistics:

gateway−state (r | u) − shows the status of the next hop. Can be **r** (reachable) or **u** (unreachable)
interface (*name*) − interface through which the gateway can be reached

- **(unknown)** − the gateway cannot be reached directly, or the route has been disabled

Notes

You can specify more than one or two gateways in the route. Moreover, you can repeat some routers in the list several times to do a kind of cost setting for gateways.

Example

To add two static routes to networks 192.168.0.0/16 and 0.0.0.0/0 (the default destination address) on a router with two interfaces and two IP addresses:

```
[admin@MikroTik] ip route> add dst−address=192.168.0.0/16 gateway=10.10.10.2
[admin@MikroTik] ip route> add gateway 10.10.10.1
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, r − rip, o − ospf, b − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 192.168.0.0/16     r 10.10.10.2      1        Local
    1  S 0.0.0.0/0          r 10.10.10.1      1        Public
    2 DC 10.10.10.0/24      r 0.0.0.0         0        Public

[admin@MikroTik] ip route> print detail
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, r − rip, o − ospf, b − bgp
    0  S dst−address=192.168.0.0/16 preferred−source=0.0.0.0
         gateway=10.10.10.2 gateway−state=reachable distance=1
         interface=Local

    1  S dst−address=0.0.0.0/0 preferred−source=0.0.0.0 gateway=10.10.10.1
         gateway−state=reachable distance=1 interface=Public

    2 DC dst−address=10.10.10.0/24 preferred−source=10.10.10.1
         gateway=0.0.0.0 gateway−state=reachable distance=0 interface=Public
```

To set the **192.168.0.0/16** network is reachable via both **10.10.10.2** and **10.10.10.254** gateways:

```
[admin@MikroTik] ip route> set 0 gateway=10.10.10.2,10.10.10.254
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, r − rip, o − ospf, b − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 192.168.0.0/16     r 10.10.10.2      1        Local
                            r 10.10.10.254             Local
    1  S 0.0.0.0/0          r 10.10.10.1      1        Public
    2 DC 10.10.10.0/24      r 0.0.0.0         0        Public

[admin@MikroTik] ip route>
```

Policy Routing

Description

Policy routing allows select routes in order to variate the use of network resources to certain classes of users (in other words, you can set different routes to the same networks depending on some classifiers). This is implemented using multiple routing tables and a list of rules specifying how these tables should be used.

The Policy Routing is implemented in the MikroTik RouterOS based on source and destination addresses of a packet, the interface the packet arrives to the router and the firewall mark that may be associated with some packets.

When finding the route for a packet, the packet is matched against policy routing rules one after another, until some rule matches the packet. Then action specified in that rule is executed. If no rule matches the packet, it is assumed that there is no route to given host and appropriate action is taken (packet dropped and ICMP error sent back to the source).

If a routing table does not have a route for the packet, next rule after the one that directed to the current table is examined, until the route is found, end of rule list is reached or some rule with action drop or unreachable is hit. Thus it is good to have last rule say "from everywhere to everywhere, all interfaces, lookup main route table", because then gateways can be found (connected routes are entered in the main table only).

Note that the only way for packet to be forwarded is to have some rule direct to some routing table that contains route to packet destination.

Routing Tables

Submenu level : **/ip policy−routing**

Description

Routing tables is a way to organize routing rules into groups for a purpose of easy management. These tables can be created/deleted in the **/ip policy−routing** menu.

The routes in the routing tables are managed the same way as the static routes described above, but in the submenu **/ip policy−routing table *name*** submenu, where ***name*** is name of the table

Property Description

name (*name*) − table name

Notes

There is always the table **main** − this one can not be deleted and its name can not be changed. The **main** table can be managed in in the **/ip route** submenu as well:

```
[admin@MikroTik] ip policy−routing> table main
[admin@MikroTik] ip policy−routing table main> print
Flags: X − disabled, I − invalid, D − dynamic, R − rejected
  #    TYPE    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
  0    static  192.168.1.0/24     r 192.168.0.50    1        Local
  1    static  0.0.0.0/0          r 10.0.0.1        1        Public
  2 D  connect 192.168.0.0/24     r 0.0.0.0         0        Local
```

```
  3 D  connect 10.0.0.0/24        r 0.0.0.0         0        Public
[admin@MikroTik] ip policy-routing table main>
[admin@MikroTik] ip policy-routing table main> /ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 192.168.1.0/24     r 192.168.0.50    1        Local
    1  S 0.0.0.0/0          r 10.0.0.1        1        Public
    2 DC 192.168.0.0/24     r 0.0.0.0         0        Local
    3 DC 10.0.0.0/24        r 0.0.0.0         0        Public
[admin@MikroTik] ip policy-routing table main>
```

Example

To add a new table named **mt**:

```
[admin@MikroTik] ip policy-routing> add name=mt
[admin@MikroTik] ip policy-routing> print
Flags: D - dynamic
  #   NAME
  0   mt
  1 D main
[admin@MikroTik] ip policy-routing>
```

To add the route to the **10.5.5.0/24** via **10.0.0.22** gateway to the **mt** table:

```
[admin@MikroTik] ip policy-routing> table mt
[admin@MikroTik] ip policy-routing table mt> add dst-address=10.5.5.0/24 \
\... gateway=10.0.0.22
[admin@MikroTik] ip policy-routing table mt> print
Flags: X - disabled, I - invalid, D - dynamic, R - rejected
  #    TYPE    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
  0    static  10.5.5.0/24        r 10.0.0.22       1        Public
[MikroTik] ip policy-routing table mt>
```

Policy rules

Submenu level : **/ip policy-routing rule**

Property Description

src-address (*IP adress/mask*) - source IP address/mask
dst-address (*IP adress/mask*) - destination IP address/mask
interface (*name* | all; default: **all**) - interface name through which the packet arrives. Should be **all** for the rule that should match locally generated or masqueraded packets, since at the moment of processing the routing table these packets have interface name set to loopback
flow (*name*; default: "") - flow mask of the packet to be matched by this rule
action (drop | unreachable | lookup; default: **unreachable**) - action to be processed on packets to be matched by this rule:

- **drop** - silently drop packet
- **unreachable** - reply that destination host is unreachable
- **lookup** - lookup route in given routing table

Notes

Policy routing will not function 'as desired' for packets originated from the router or masqueraded packets. It is because these packets have source address 0.0.0.0 at the moment when they are processed by the routing table. Therefore it is not possible to match masqueraded packets by source address with policy routing rule. You should use matching by flow together with packet marking instead.

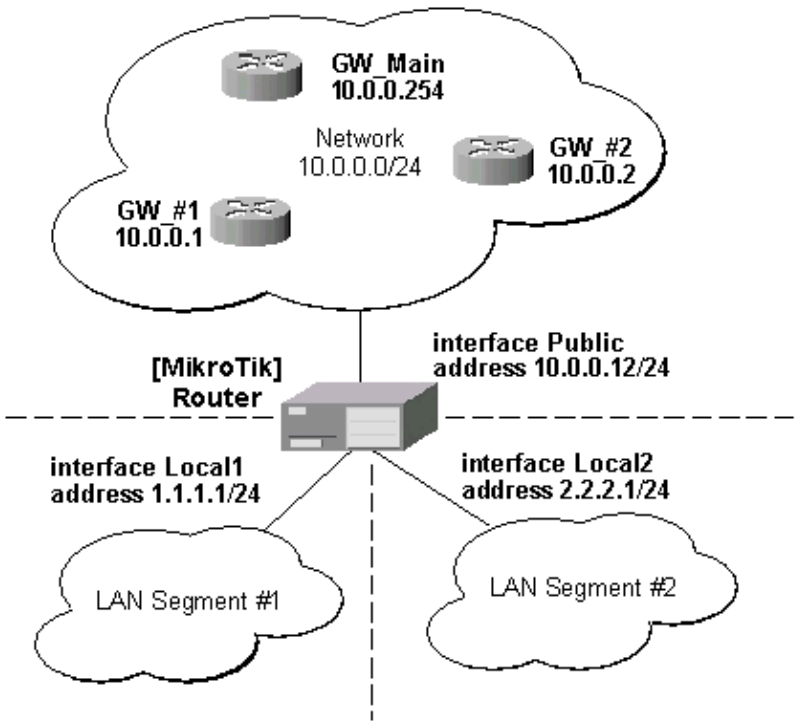
Example

To add the rule specifying that all the packets from the **10.0.0.144** host should lookup the **mt** routing table:

```
[admin@MikroTik] ip policy-routing rule> add src-address=10.0.0.144/32 \
\... table=mt action=lookup
[admin@MikroTik] ip policy-routing rule> print
Flags: X - disabled, I - invalid
  #   SRC-ADDRESS        DST-ADDRESS        INTE... FLOW    ACTION      TABLE
  0   0.0.0.0/0          0.0.0.0/0          all             lookup      main
  1   10.0.0.144/32      0.0.0.0/0          all             lookup      mt
[admin@MikroTik] ip policy-routing rule>
```

Application Example

Suppose we want packets coming from 1.1.1.0/24 to use gateway 10.0.0.1 and packets from 2.2.2.0/24 to use gateway 10.0.0.2. And the rest of packets will use gateway 10.0.0.254 (assuming we already have it):



Command sequence to achieve this:

1. Add 2 new routing tables:

```
[admin@MikroTik] ip policy-routing> add name=from_net1; add name=from_net2
[admin@MikroTik] ip policy-routing> print
Flags: D - dynamic
  #   NAME
  0   from_net1
  1   from_net2
  2 D main

[admin@MikroTik] ip policy-routing>
```

2. Create the default route in each of the tables:

```
[admin@MikroTik] ip policy-routing> table from_net1 add gateway=10.0.0.1
[admin@MikroTik] ip policy-routing> table from_net2 add gateway=10.0.0.2
[admin@MikroTik] ip policy-routing> table from_net1 print
Flags: X - disabled, I - invalid, D - dynamic, R - rejected
  #    TYPE    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
  0    static  0.0.0.0/0          u 10.0.0.1        1        Public

[admin@MikroTik] ip policy-routing> table from_net2 print
Flags: X - disabled, I - invalid, D - dynamic, R - rejected
  #    TYPE    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
  0    static  0.0.0.0/0          u 10.0.0.2        1        Public

[admin@MikroTik] ip policy-routing>
```

3. Create rules that will direct traffic from sources to given tables, and arrange them in the desired order:

```
[admin@MikroTik] ip policy-routing> rule
[admin@MikroTik] ip policy-routing rule> print
Flags: X - disabled, I - invalid
  #   SRC-ADDRESS        DST-ADDRESS        INT... FLOW       ACTION
  0   0.0.0.0/0          0.0.0.0/0          all               lookup

[admin@MikroTik] ip policy-routing rule> add src-address=1.1.1.1/32 \
\... action=lookup table=main
[admin@MikroTik] ip policy-routing rule> add src-address=2.2.2.1/32 \
\... action=lookup table=main
[admin@MikroTik] ip policy-routing rule> add src-address=1.1.1.0/24 \
\... action=lookup table=from_net1
[admin@MikroTik] ip policy-routing rule> add src-address=2.2.2.0/24 \
\... action=lookup table=from_net2
[admin@MikroTik] ip policy-routing rule> print
Flags: X - disabled, I - invalid
  #   SRC-ADDRESS        DST-ADDRESS        INT... FLOW       ACTION
  0   1.1.1.1/32         0.0.0.0/0          all               lookup
  1   2.2.2.1/32         0.0.0.0/0          all               lookup
  2   1.1.1.0/24         0.0.0.0/0          all               lookup
  3   2.2.2.0/24         0.0.0.0/0          all               lookup
  4   0.0.0.0/0          0.0.0.0/0          all               lookup

[admin@MikroTik] ip policy-routing rule>
```

Here the rules #0 and #1 are needed to correctly process connections initiated from the local addresses of the router. Namely, the 'connected' routes from the main table should be used instead of using the default routes from tables **from_net1** or **from_net2**. Rules #2 and #3 handls packets originated from locally connected networks and rule #4 looks after packets originated from all other sources.

Additional Resources

Recommended readings for guidelines on routing issues:

- RFC2328
- RFC2992
- RFC1102

Services, Protocols, and Ports

Document revision 1.2 (10−Oct−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

This document lists protocols and ports used by various MikroTik RouterOS services. It helps you to determine why your MikroTik router listens to certain ports, and what you need to block/allow if you want to prevent or grant access to the certain services. Please see the relevant sections of the Manual for more explanations.

Specifications

Packages required : *Depends on actual service*
License required : *Depends on actual service*
Home menu level : ***/ip service***
Protocols utilized : *Depends on actual service*
Hardware usage: *Depends on actual service*

Related Documents

Firewall Filters and Network Address Translation (NAT)
Certificate Management

Modifying service settings

Submenu level : **/ip service**

Property Description

name (*name*) − service name
port (1...65535) − port the service listens on
address (*IP address/mask*; default: **0.0.0.0/0**) − IP address from which the service is accessible
certificate (*name* | none; default: **none**)− name of the certificate used by this service (absent for the services that do not need certificates)

Example

To set **www** service to use **8081** port accesible from the **10.10.10.0/24** network:

```
[admin@MikroTik] ip service> print
Flags: X - disabled, I - invalid
  #   NAME                                PORT  ADDRESS            CERTIFICATE
  0   telnet                              23    0.0.0.0/0
  1   ftp                                 21    0.0.0.0/0
  2   www                                 80    0.0.0.0/0
  3   hotspot                             8088  0.0.0.0/0
  4   ssh                                 22    0.0.0.0/0
  5   hotspot-ssl                         443   0.0.0.0/0          none

[admin@MikroTik] ip service> set www port=8081 address=10.10.10.0/24
[admin@MikroTik] ip service> print
Flags: X - disabled, I - invalid
  #   NAME                                PORT  ADDRESS            CERTIFICATE
  0   telnet                              23    0.0.0.0/0
  1   ftp                                 21    0.0.0.0/0
  2   www                                 8081  10.10.10.0/24
  3   hotspot                             8088  0.0.0.0/0
  4   ssh                                 22    0.0.0.0/0
  5   hotspot-ssl                         443   0.0.0.0/0          none

[admin@MikroTik] ip service>
```

List of Services

Below is list of protocols and ports used by MikoTik RouterOS services. Some services require additional package to be installed, as well as to be enabled by administrator, e.g., bandwidth server.

```
Port      Description
-----------------------------------------------------------------------
20/tcp    File Transfer [Default Data]
21/tcp    File Transfer [Control] (Change under /ip service)
22/tcp    SSH Remote Login Protocol (Only with ssh package)
23/tcp    Telnet
53/tcp    Domain Name Server (Only with dns-cache package)
53/udp    Domain Name Server (Only with dns-cache package)
67/udp    Bootstrap Protocol Server, DHCP Server (only with dhcp package)
68/udp    Bootstrap Protocol Client, DHCP Client (only with dhcp package)
80/tcp    World Wide Web HTTP (Change under /ip service)
123/tcp   Network Time Protocol (Only with ntp package)
161/tcp   SNMP (Only with snmp package)
500/udp   IKE protocol (Only with ipsec package)
179/tcp   Border Gateway Protocol (Only with bgp package)
1719/udp  h323gatestat (Only with telephony package)
1720/tcp  h323hostcall (Only with telephony package)
1723/tcp  pptp (Only with pptp package)
2000/tcp  bandwidth-test server
3986/tcp  proxy for winbox
3987/tcp  sslproxy for secure winbox (Only with ssh package)
5678/udp  MikroTik Neighbor Discovery
8080/tcp  HTTP Alternate (Only with web-proxy package, can be changed)
/1        ICMP - Internet Control Message
/4        IP - IP in IP (encapsulation)
/47       GRE - General Routing Encapsulation (Only for pptp and eoip)
```

```
/50       ESP - Encap Security Payload for IPv6 (Only with ipsec package)
/51       AH - Authentication Header for IPv6 (Only with ipsec package)
/89       OSPFIGP - OSPF Interior Gateway Protocol
-------------------------------------------------------------------------
```

Additional Resources

Complete list of protocol numbers can be found at http://www.iana.org/assignments/protocol-numbers
Complete list of port numbers can be found at http://www.iana.org/assignments/port-numbers

Universal Client Interface

Document revision 1.2 (01−Jun−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

Universal Client Interface allows to work with clients regardless of their IP addresses translating these addresses to the ones you are able to work with. It gives a possibility to provide network access (for example, Internet access) to mobile clients that are not willing to change their networking settings. The feature is intended to use with HotSpot, but may be useful even without HotSpot.

Specifications

Packages required : ***hotspot***
License required : *Any*
Home menu level : ***/ip hotspot universal***
Protocols utilized : *None*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
IP Pools
DHCP Client and DHCP Server
HotSpot Gateway
Firewall Filters and Network Address Translation (NAT)

Description

Universal client accepts any incoming address from a connected network interface and does one to one translation so that data may be routed through standard IP networks. Clients may use any preconfigured address. If the Universal client feature is set to translate the client to a real IP address, then the client may even run a server or any other connection that requires a real IP address. It is possible to add static entry, so that some clients will get the specified addresses.

Universal client is changing source address of each packet just after it is received by the router (even **mangle** 'sees' the translated address).

Note also that **arp** mode should be **enabled** on the interface you set Universal Client Interface on.

Universal Client Interface Setup

Submenu level : **/ip hotspot universal**

Property Description

interface (*name*) − interface to run universal client on
address−pool (*name*) − IP address pool name
arp (all−arp | no−arp; default: **all−arp**) − ARP hadling mode:

- **all−arp** − respond to all ARP requests
- **no−arp** − respond to ARP requests normally

use−dhcp (yes | no; default: **yes**) − do not translate the addresses assigned by DHCP server
idle−timeout (*time*; default: **5m**) − idle timeout (maximal period of inactivity) for client added dynacally

Notes

Settig **arp** in **all−arp** is generally a good idea because in most cases you cannot know what is the gateway's IP address configured on the clients.

Example

To enable Universal Client Interface on **ether1** interface that will take the addresses to translate to from the **exp** pool:

```
[admin@MikroTik] ip hotspot universal> add address−pool=exp interface=ether1
[admin@MikroTik] ip hotspot universal> print
Flags: X − disabled, I − invalid
  #   INTERFACE ADDRESS−POOL ARP     USE−DHCP IDLE−TIMEOUT
  0 X ether1    exp          all−arp yes      5m

[admin@MikroTik] ip hotspot universal> enable 0
[admin@MikroTik] ip hotspot universal> print
Flags: X − disabled, I − invalid
  #   INTERFACE ADDRESS−POOL ARP     USE−DHCP IDLE−TIMEOUT
  0   ether1    exp          all−arp yes      5m

[admin@MikroTik] ip hotspot universal>
```

Universal Client List

Submenu level : **/ip hotspot universal client**

Description

The list shows the current translation table. There are three ways a client may be added to the table:

- Each time router receives a packet from an unknown client (determined by three properties: **mac−address**, **address** and **interface**), it adds the client to the list
- Client may be added by DHCP server
- Client may be added administratively specifying three properties: **mac−address**, **address** and **interface**

Property Description

mac−address (*MAC address*) − client's MAC address
address (*IP address*) − client's IP address
to−address (*IP address*) − IP address to translate **address** to
interface (*name*) − interface name the client is connected to

Statistics:

idle−time (*time*) − inactivity time
uptime (*time*) − how long the client is active
bytes−in (*integer*) − the amount of bytes received from the client
bytes−out (*integer*) − the amount of bytes sent to the client
packets−in (*integer*) − the amount of packets received from the client
packets−out (*integer*) − the amount of packets sent to the client

Example

To add an entry specifying that IP address **10.20.30.40** should be translated to **10.0.0.20** for packets coming from client with MAC address **01:23:45:67:89:AB** on **ether1** interface:

```
[admin@MikroTik] ip hotspot universal client> add mac−address=01:23:45:67:89:AB
address=10.20.30.40 interface=ether1 to−address=10.0.0.20
[admin@MikroTik] ip hotspot universal client> print
Flags: X − disabled, I − invalid, H − DHCP, D − dynamic
  #   MAC−ADDRESS       ADDRESS         TO−ADDRESS      INTERFACE IDLE−TIME
  0   01:23:45:67:89:AB 10.20.30.40     10.0.0.20       ether1    1s
```

Service Port

Submenu level : **/ip hotspot universal service−port**

Description

Just like for classic NAT, the Universal Client Interface 'breaks' some protocols that are incompatible with address translation. To leave these protocols consistent, helper modules must be used. For the Universal Client Iterface the only such a module is for FTP protocol

Property Description

name (*name*) − protocol name
ports (*list*: *integer*) − list of the ports on which the protocol is working

Example

To set the FTP protocol uses bot 20 and 21 TCP port:

```
[admin@MikroTik] ip hotspot universal service-port> print
Flags: X - disabled
  #   NAME                                                                PORTS
  0   ftp                                                                 21
[admin@MikroTik] ip hotspot universal service-port> set ftp ports=20,21
[admin@MikroTik] ip hotspot universal service-port> print
Flags: X - disabled
  #   NAME                                                                PORTS
  0   ftp                                                                 20
                                                                          21
[admin@MikroTik] ip hotspot universal service-port>
```

Universal Plug and Play

Document revision 1.3 (29−Dec−2003)
This document applies to the MikroTik RouterOS v2.7

Table of Contents

Summary

The MikroTik RouterOS supports Universal Plug and Play architecture for transparent peer−to−peer network connectivity of personal computers and network−enabled intelligent devices or appliances. UPnP builds enables these devices to automatically connect with one another and work together to make networking possible for more people.

Specifications

Packages required : *None*
License required : *Any*
Home menu level : ***/ip upnp***
Standarts and Technologies : *TCP/IP (RFC1180), HTTP (RFC2616), XML (XML)*
Hardware usage : *may require additional memory*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Firewall Filters and Network Address Translation (NAT)

Description

UPnP enables data communication between any two devices under the command of any control device on the network. Universal Plug and Play is completely independent of any particular physical medium. It supports networking with automatic discovery without any initial configuration, whereby a device can dynamically join a network. DHCP and DNS servers are optional and will be used if available on the network. UPnP implements simple yet powerfull NAT traversal solution, that enables the client to get full peer−to−peer network support from behind the NAT.

There are two interface types for UPnP: **internal** (the one local clients are connected to) and **external** (the one the Internet is connected to). A router may only have one external interface with a 'public' IP address on it, and as many internal IP addresses as needed, all with source−NATted 'internal' IP addresses.

The UPnP protocol is used for most of DirectX games as well as for various Windows Messenger features (remote asisstance, application sharing, file transfer, voice, video) from behind a firewall.

Enabling Universal Plug−n−Play

Submenu level : **/ip upnp**

Property Description

enabled (yes | no; default: **no**) − whether UPnP feature is enabled

Example

To enable UPnP feature:

```
[admin@MikroTik] ip upnp> set enable=yes
[admin@MikroTik] ip upnp> print
    enabled: yes
[admin@MikroTik] ip upnp>
```

UPnP Interfaces

Submenu level : **/ip upnp interfaces**

Property Description

interface (*name*) − interface name UPnP will be run on
type (external | internal | unused) − interface type, one of the:

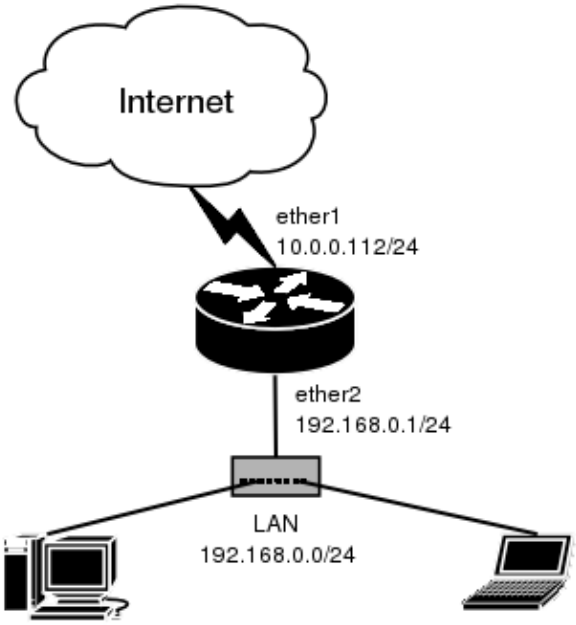
- **external** − the interface global IP address is assigned to
- **internal** − router's local interface
- **unused** − the interface is not used by UPnP

Notes

It is highly recommended to upgrade DirectX runtime libraries to version 9.0a DirectX 9.0 and Windows Messenger to version 5.0 Windows Messenger 5.0 in order to get things to work properly.

Example

Suppose we have a following example:



We have masquerading already enabled on our router:

```
[admin@MikroTik] ip upnp interfaces> /ip firewall src-nat print
Flags: X - disabled, I - invalid, D - dynamic
  0   src-address=0.0.0.0/0:0-65535 dst-address=0.0.0.0/0:0-65535
      out-interface=ether1 protocol=all icmp-options=any:any flow=""
      connection="" content="" limit-count=0 limit-burst=0 limit-time=0s
      action=masquerade to-src-address=0.0.0.0 to-src-port=0-65535

[admin@MikroTik] ip upnp interfaces>
```

Now all we have to do is to add interfaces and enable UPnP:

```
[admin@MikroTik] ip upnp interfaces> add interface=ether1 type=external
[admin@MikroTik] ip upnp interfaces> add interface=ether2 type=internal
[admin@MikroTik] ip upnp interfaces> print
Flags: X - disabled
  #   INTERFACE TYPE
  0 X ether1    external
  1 X ether2    internal

[admin@MikroTik] ip upnp interfaces> enable 0,1
[admin@MikroTik] ip upnp interfaces> .. set enabled=yes
[admin@MikroTik] ip upnp interfaces>
```

Additional Resources

UPnP forum

WEB Proxy

Document revision 1.3 (12−Nov−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

The MikroTik RouterOS has the squid proxy server implementation.

Proxy server features:

- Regular http proxy
- Transparent proxy. Can be transparent and regular at the same time
- Access list by source, destination, URL and requested method
- Cache access list (specifies which objects to cache, and which not)

- Direct Access List (specifies which resources should be accessed directly, and which − through an another proxy server)
- Logging facility

Specifications

Packages required : ***web−proxy***
License required : *Basic*
Home menu level : ***/ip web−proxy***
Protocols utilized : *HTTP/1.0 (RFC1945), HTTP/1.1 (RFC2616), FTP (RFC959)*
Hardware usage: *uses disk space, if available (see description below)*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Firewall Filters and Network Address Translation (NAT)
Log Management

Description

The web proxy can be used as transparent and normal web proxy at the same time. In transparent mode it is possible to use it as standard web proxy, too. However, in this case, proxy users may have trouble to reach web pages which are accessed transparently.

When setting up Web proxy, make sure it serves only your clients, and is not misused as relay. Please read the security notice in the Access List Section!

Note that it may be useful to have Web proxy running even with no cache when you want to use it as something like HTTP and FTP firewall (for example, denying access to mp3 files) or to redirect requests to external proxy transparently

MikroTik Web Proxy Setup

Submenu level : **/ip web−proxy**

```
[admin@MikroTik] ip web−proxy> print
                enabled: no
                address: 0.0.0.0:3128
               hostname: ""
      transparent−proxy: no
           parent−proxy: 0.0.0.0:0
    cache−administrator: "webmaster"
        max−object−size: 4096 kB
         max−cache−size: unlimited
                 status: stopped
     reserved−for−cache: 10240 kB
[admin@MikroTik] ip web−proxy>
```

Proprety Description

enabled (yes | no, default: **no**) − whether web−proxy is enabled or not
address (*IP address:port*, default: **0.0.0.0:3128**) − IP address (**0.0.0.0** for any) and port (mandatory) on which proxy will be listening for requests
hostname (*string*, default: "") − hostname (DNS or IP address) of the web proxy
transparent−proxy (yes | no, default: **no**) − use transparent mode
parent−proxy (*IP address:port*, default: **0.0.0.0:0**) − upper−level proxy

- **0.0.0.0:0** − disables parent proxy server

max−object−size (*integer*, default: **4096**) − objects larger than this size will not be saved on disk. The value is specified in kilobytes. If you wish to get a high bytes hit ratio, you should probably increase this (one 32 MB object hit counts for 3200 10KB hits). If you wish to increase speed more than your want to save bandwidth you should leave this low
max−cache−size (none | unlimited | *integer*) − maximal cache size in MB

- **none** − web proxy will not use cache
- **unlimited** − web proxy will use as much drivespace, as possible

Statistics:

status (*string*) − displays status of the proxy server:

- **stopped** − proxy is disabled and is not running
- **rebuilding−cache** − proxy is enabled and running, existing cache is being verified
- **running** − proxy is enabled and running
- **stopping** − proxy is shutting down (max 10s)
- **clearing−cache** − proxy is stopped, cache files are being removed
- **creating−cache** − proxy is stopped, cache directory structure is being created
- **dns−missing** − proxy is enabled, but not running because of unknown DNS server (you should specify it under **/ip dns**)
- **invalid−address** − proxy is enabled, but not running because of invalid address (you should change address or port)
- **invalid−cache−administrator** − proxy is enabled, but not running because of invalid cache−administrator's e−mail address
- **invalid−hostname** − proxy is enabled, but not running because of invalid hostname (you should set a valid hostname value)
- **error−logged** − proxy is not running because of unknown error. This error is logged as System−Error. Please, send us this error and some description, how it happened

reserved−for−cache (*integer*) − maximal cache size, that is accessible to web−proxy

Notes

By default the proxy cache can use as much disk space as there is allocated for it. When the system allocates the space for the proxy cache, 1/7th of the total partition (disk) size is reserved for the system, but not less than 50MB. The rest is left for the proxy cache. The system RAM size is considered as well when allocating the cache size. The cache size is limited so, that there are at least 11.2MB of RAM per 1GB of cache plus 32MB of RAM is reserved for the system. **max−cache−size** is also taken in account, so the cache will not occupy more than it is specified in this property. The effective limit is calculated as a minimum of all three limits.

Expire time of cache entries can be different for each HTML page (specified in headers). But, if there is no such header, the entry will be considered fresh for max 72 hours.

Example

To enable the proxy on port 8080:

```
[admin@MikroTik] ip web-proxy> set enabled=yes address=0.0.0.0:8080
[admin@MikroTik] ip web-proxy> print
                enabled: yes
                address: 0.0.0.0:8080
               hostname: ""
      transparent-proxy: no
           parent-proxy: 0.0.0.0:0
    cache-administrator: "webmaster"
        max-object-size: 4096 kB
         max-cache-size: unlimited
                 status: rebuilding-cache
     reserved-for-cache: 10240 kB
[admin@MikroTik] ip web-proxy>
```

Monitoring the Web Proxy

Command name : **/ip web-proxy monitor**

Property Description

Statistics:

status (*string*) - the same as for **/ip web-proxy print**
uptime (*time*) - uptime of the proxy server
clients (*integer*) - number of present and past proxy clients with different IP addresses (in current uptime)
requests (*integer*) - total number of requests to the proxy (in current uptime)
hits (*integer*) - number of requests satisfied with proxy's cache (in current uptime)
cache-size (*integer*) - current cache size in kilobytes
received-from-servers (*integer*) - how many kilobytes did proxy receive from remote servers (in current uptime)
sent-to-clients (*integer*) - how many kilobytes did proxy send to the clients to resolve their requests (in current uptime)
hits-sent-to-clients (*integer*) - how many kilobytes of sent traffic were taken from the cache (in current uptime)

Example

To monitor the web proxy:

```
[admin@MikroTik] > ip web-proxy monitor
                   status: running
                   uptime: 4d19h8m14s
                  clients: 9
                 requests: 10242
                     hits: 3839
               cache-size: 328672 kB
    received-from-servers: 58108 kB
          sent-to-clients: 65454 kB
     hits-sent-to-clients: 7552 kB
```

```
[admin@MikroTik] >
```

Access List

Submenu level : **/ip web−proxy access**

Description

Access list is implemented in the same way as MikroTik firewall rules. Rules are processed from the top to the bottom. First matching rule specifies decision of what to do with this connection. Connections can be matched by its source address, destination address, destination port, substring of requested url or request method. If none of these parameters is specified, every connection will match this rule.

If connection is matched by a rule, action property of this rule specifies whether connection will be allowed or not. If connection does not match any rule, it will be allowed.

Property Description

src−address (*IP address/mask*, default: **0.0.0.0/0**) − source addresst
dst−address (*IP address/mask*, default: **0.0.0.0/0**) − destination address
dst−port (*string*, default: "") − destination port list
url (*string*) − the URL of the request (regular expression)
method (any | connect | delete | get | head | options | post | put | trace, default: **any**) − method of the request (see RFC2616 for details)
action (allow | deny, default: **allow**)− action to take

Notes

There is one rule by default, that disallows **connect** method connections other than to 443 (https) and to 563 (snews) ports. **connect** method is a security hole that allow connections (transparent tunneling) to any computer using any protocol. It is used mostly by spammers, as they found it very convinient to use others' mail (SMTP) servers as anonymous mail relay to send spam over the Internet.

It is strongly recommended to deny all IP addresses except those behind the router as the proxy still may be used to access your internal−use−only (intranet) web servers. Also, consult examples in Firewall Manual on how to protect your router.

Details about regular expressions used in **url** field can be found here:
http://www.cs.utah.edu/dept/old/texinfo/regex/regex_toc.html

Examples

The default rule:

```
[admin@MikroTik] ip web−proxy access> print
Flags: X − disabled
  0   ;;; allow CONNECT only to SSL ports 443 [https] and 563 [snews]
      src−address=0.0.0.0/0 dst−address=0.0.0.0/0 dst−port=!443,563 url=""
      method=connect action=deny

[admin@MikroTik] ip web−proxy access> print brief
```

To disallow download of MP3 files and FTP connections other than from the **10.0.0.1** server:

```
[admin@MikroTik] ip web-proxy access> add url=".mp3" action=deny
[admin@MikroTik] ip web-proxy access> add src-address=10.0.0.1/32 action=allow
[admin@MikroTik] ip web-proxy access> add url="ftp://" action=deny
[admin@MikroTik] ip web-proxy access> print
Flags: X - disabled
  0   ;;; allow CONNECT only to SSL ports 443 [https] and 563 [snews]
      src-address=0.0.0.0/0 dst-address=0.0.0.0/0 dst-port=!443,563 url=""
      method=connect action=deny

  1   src-address=0.0.0.0/0 dst-address=0.0.0.0/0 dst-port="" url=".mp3"
      method=any action=deny

  2   src-address=10.0.0.1/32 dst-address=0.0.0.0/0 dst-port="" url=""
      method=any action=allow

  3   src-address=0.0.0.0/0 dst-address=0.0.0.0/0 dst-port="" url="ftp://"
      method=any action=deny


[admin@MikroTik] ip web-proxy access>
```

Direct Access List

Submenu level : **/ip web-proxy direct**

Description

If **parent-proxy** is specified, it is possible to tell proxy server whether to try to pass the request to the parent proxy or to resolve it connecting to the requested server directly. Direct Access List is managed just like Proxy Access List described in the previous chapter except the action argument.

Property Description

src-address (*IP address/mask*, default: **0.0.0.0/0**) - source addresst
dst-address (*IP address/mask*, default: **0.0.0.0/0**) - destination address
dst-port (*string*, default: "") - destination port list
url (*string*) - the URL of the request (regular expression)
method (any | connect | delete | get | head | options | post | put | trace, default: **any**) - method of the request (see RFC2616 for details)
action (allow | deny, default: **allow**)- action to take:

- **allow** - always resolve matching requests directly, not through parent proxy
- **deny** - resolve matching requests through parent proxy if there is one. If there in no parent proxy, action will be the same as with **allow**

Notes

Default action (if no rules specified or request did not match any) is **deny**.

Managing the Cache

Submenu level : **/ip web−proxy cache**

Description

Cache access list specifies, which requests (domains, servers, pages) have to be cached locally by web proxy, and which not.

Access list is implemented exactly the same way as web proxy access list. Default action is to cache object (if no matching rule is found).

Proprety Description

src−address (*IP address/mask*, default: **0.0.0.0/0**) − source addresst
dst−address (*IP address/mask*, default: **0.0.0.0/0**) − destination address
dst−port (*string*, default: "") − destination port list
url (*string*) − the URL of the request (regular expression)
method (any | connect | delete | get | head | options | post | put | trace, default: **any**) − method of the request (see RFC2616 for details)
action (allow | deny, default: **allow**)− action to take:

Notes

By default, one cache access rule is already added:

```
[admin@MikroTik] ip web−proxy cache> print
Flags: X − disabled
  0   src−address=0.0.0.0/0 dst−address=0.0.0.0/0 dst−port=""
      url="cgi−bin \?" method=any action=deny


[admin@MikroTik] ip web−proxy cache>
```

This rule defines, that all runtime generated pages (which are located within cgi−bin directories or contain '?' in url) has not to be cached.

Objects, which are larger than max−object−size, are not cached.

Rebuilding the Cache

Command name : **/ip web−proxy clear−cache**

Description

Web proxy will automatically detect any problems with cache and will try to solve them without loosing any cache data. But in case of a heavy damage to the file system, the web proxy can't rebuild cache data. Cache can be deleted and new cache directories created using the command.

Example

```
[admin@MikroTik] ip web-proxy> clear-cache
Clear all web proxy cache, yes? [y/N]: y
cache will be cleared shortly
[admin@MikroTik] ip web-proxy>
```

Setup Example

For web proxy setup, do the following:

- Specify at least one dns server for the router:

  ```
  /ip dns set primary-dns=159.148.60.2
  ```
- Set IP address and port on which proxy will listen for requests:

  ```
  /ip web-proxy set address=0.0.0.0:8080
  ```
- If this proxy has to use another proxy, specify it:

  ```
  /ip web-proxy set parent-proxy=192.168.1.1:8080
  ```

  otherwise disable it:

  ```
  /ip web-proxy set parent-proxy=0.0.0.0:0
  ```
- Specify cache administrator's e-mail address:

  ```
  /ip web-proxy set cache-administrator=support@mt.lv
  ```
- Specify hostname (DNS or IP address) of the web proxy:

  ```
  /ip web-proxy set hostname=proxy.mt.lv
  ```
- Allow access to web proxy oly from our network (for example, **192.168.0.0/16**):

  ```
  /ip web-proxy access add src-address=!192.168.0.0/16 action=deny
  ```
- Enable the proxy service:

  ```
  /ip web-proxy set enabled=yes
  ```

Now it is possible to use this proxy, by setting it as a proxy for you web browser.

Transparent Mode

To enable the transparent mode, firewall rule in destination nat has to be added, specifying which connections (to which ports) should be transparently redirected to the proxy. For example, we have the following web-proxy settings:

```
[admin@MikroTik] ip web-proxy> print
                enabled: yes
                address: 0.0.0.0:8080
               hostname: "proxy.mt.lv"
      transparent-proxy: yes
           parent-proxy: 10.5.5.1:8080
    cache-administrator: "support@mt.lv"
        max-object-size: 10000 kB
```

```
         max-cache-size: unlimited
                 status: running
     reserved-for-cache: 2633728 kB
[admin@MikroTik] ip web-proxy>
```

If we want all connections coming from interface ether1 and going to port 80 to handle with web proxy transparently, and if our web proxy is listening on port 8080, then we add following destination nat rule:

```
[admin@MikroTik] ip firewall dst-nat> add in-interface=ether1 protocol=tcp \
dst-address=!10.0.0.1/32:80 action=redirect to-dst-port=8080
[admin@MikroTik] ip firewall dst-nat> print
Flags: X - disabled, I - invalid
  0   src-address=0.0.0.0/0:0-65535 in-interface=ether1
      dst-address=!10.0.0.1/32:80 protocol=tcp icmp-options=any:any flow=""
      src-mac-address=00:00:00:00:00:00 limit-count=0 limit-burst=0
      limit-time=0s action=redirect to-dst-address=0.0.0.0 to-dst-port=8080

[admin@MikroTik] ip firewall dst-nat>
```

Here, the router's address and port 80 (10.0.0.1/32:80) have been excluded from redirection to preserve the winbox functionality which uses TCP port 80 on the router. More than one redirect rule can be added to redirect more than one port.

Note: only HTTP traffic is supported by web proxy transparent mode. HTTPS and FTP are not going to work this way!

Troubleshooting

- **When I turned on transparent proxy and redirected TCP port 80 to it, my WinBox stopped working.**
  TCP port 80 is used by WinBox when connecting to the router. You should exclude the router's address:80 from redirection by using rule
  /ip firewall dst-nat add dst-address=address/32:80 protocol=tcp action=accept
  BEFORE the redirect rule. Alternatively, you can use just one rule
  /ip firewall dst-nat add dst-address=!address/32:80 protocol=tcp action=redirect to-dst-port=8080
- **I use firewall to block access to the router from the Internet. My proxy does not work.**
  Make sure you allow established TCP connections with tcp option **non-syn-only** to the router before blocking everything else. The rule is like this:
  /ip firewall rule input add protocol=tcp tcp-options=non-syn-only connection-state=established

Queues and Data Rate Management

Document revision 1.7 (01−Aug−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

Queuing is a mechanism that control data rate allocation, delay variability, timely delivery, and delivery reliability. The MikroTik RouterOS supports the following queuing mechanisms:

PFIFO − Packets First−In First−Out,
BFIFO − Bytes First−In First−Out,

SFQ − Stochastic Fair Queuing
RED − Random Early Detection
HTB − Hierarchical Token Bucket

The queuing can be used for limiting the data rate for certain IP addresses, protocols or ports. The queuing is performed for packets leaving the router through a real interface. It means that the queues should always be configured on the outgoing interface regarding the traffic flow. There are two additional virtual interfaces which are used to limit all the traffic coming to (**global−in**) or leaving (**global−out**) the router regardless of physical interface.

Specifications

Packages required : ***system***
License required : *Basic (DEMO license is limited to 4 queues)*
Home menu level : ***/queue***
Protocols utilized : *None*
Hardware usage: *significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Packet Filter (Firewall) and Network Address Translation (NAT)

Description

Classless Queues

There are four types of simple queues implemented in RouterOS: PFIFO, BFIFO, SFQ and RED. With Bytes First−In First−Out (BFIFO) and Packets First−In First−Out (PFIFO) packets are served in the same order as they are received. The only difference between BFIFO and PFIFO is that PFIFO has a length measured in packets, BFIFO in bytes. Generally, you do not want to use BFIFO or PFIFO as traffic shapers. It's better to use them just for statistics as they are pretty fast. The only exception is when you are running out of resources with RED and/or with complicated queue tree.

Stochastic Fair Queuing (SFQ) cannot limit traffic at all. Its main idea is to equalize sessions (not computer traffic, but session traffic, it is sometimes mentioned as SFQ drawback) when your link is completely full. It works in round−robin fashion, giving each session a chance to send **sfq−allot** bytes. Its algorithm can distinguish only 1024 sessions, and that is why several sessions can be treated as one. Each **sfq−perturb** seconds it drops internal table mixing all the connections and creates a new table. As it is very fast, you may want to use it as a child queue.

The normal behavior of queues is called tail−drop. Tail−drop works by queuing up to a certain amount, then dropping all traffic that 'spills over'. Random Early Detection (RED is also known as Random Early Drop because it actually works that way) statistically drops packets from flows before it reaches its hard limit. This causes a congested backbone link to slow more gracefully. It starts dropping packets when threshold reaches **red−min−threshold** mark randomly with increasing probability as threshold rising. Maximum probability is used when traffic reaches **red−max−threshold** mark. Then packets are simply thrown away. **burst** parameter is the number of packets allowed to burst through the interface when the link is empty (generally value of

(**min**+**min**+**max**)/3 works fine). The minimum value that can be used here is equal to the value of **red−min−threshold**.

Classful Queues

Classful queues are very useful if you have different kinds of traffic which should have different treatment. Generally, we can set only one queue on the interface, but in RouterOS even simple queues (known as classless queues) are attached to the main (attached to the root, which represent real interface) Hierarchical Token Bucket (HTB) and thus have some properties derived from that parent queue. With classful queues it is possible to deploy hierarchical queue trees. For example, we can set a maximum data rate for a workgroup and then distribute that amount of traffic between the members of that group as we can do with simple queues attached to the main HTB, but with upper limit.

Each queue represents a virtual interface with the allowed data rate. It can be borrowed from sibling queues (queues that are children of one queue) when **max−limit** is greater than **limit−at**. If so, the queue would use over the allocated data rate whenever possible. Only when other queues are getting too long and a connection is not to be satisfied, then the borrowing queues would be limited at their allocated data rate.

When a parent is allowed to send some amount of traffic, it asks its inner queues in order of **priority** (priorities are processed one after another, from 1 to 8, where **1** means the highest priority). When the a queue reached its **limit−at** value, its priority is not to be taken in account, such a queue will be less−prioritative than the ones not reached this limit.

Information Rates and Contention Ratios

Quality of Service (QoS) means that router should prioritize and shape network traffic. QoS is not so much about limiting, it is more about providing quality. The main terms used to describe the level of QoS for network applications are:

- CIR (Committed Information Rate) − the guaranteed data rate. It means that traffic not exceeding this rate should always be delivered.
- MIR (Maximal Information Rate) − the maximal data rate router will provide.
- Contention Ratio − the ratio to which the defined data rate is shared between users (i.e., data rate is allocated to a number of subscribers). For example, the contention ratio of 1:4 means that the allocated data rate may be shared between no more than 4 users.
- Priority − the order of importance in what traffic will be processed. You can give priority to some traffic in order it to be handeled befor some other traffic.

  MikroTik RouterOS may be used to provide CIR and MIR with some contention level and priority. Here we will talk in terms of queues (which represent either real or virtual interface) and classes (children of a queue; each class has an another queue attached to it).
- **limit−at** property is used to specify CIR. If the queue will be able to provide that data rate, it will (i.e, the parent queue (and the link the router is connected to) should be able to provide the total data rate equal or greater that the sum of all CIRs the queue should satisfy in order to quarantee these CIRs). CIRs will be satisfied in order of their **priority**.
- **max−limit** property is used to specify MIR. If the queue has satisfied all the CIRs and it is able to provide some additional data rate, it will try to distribute that additional data rate between all its classes regardless of their priorities and not exceeding their MIRs.
- Filters in RouterOS are very powerful and flexible. Providing Contention Ratio is only one application of what they can do. Using firewall mangle you can mark some a number of hosts with a flow−mark, so the data rate allocated for that mark will be shared between these hosts.

Virtual Interfaces

In addition to real interfaces, there are two vitrual interfaces you can attach queues to:

- **global−out** − represents all the output interfaces in general. Queues attached to it applies before the ones attached to a specific interface.
- **global−in** − represents all the input interfaces in general (INGRESS queue). Please **note** that queues attached to **global−in** applies to incomming traffic, not outgoing. **global−in** queueing is taking place just after mangle and before DST−NAT.

Universal Client and Simple Queues

Universal client should catch traffic when it leaves the router − in order to be able to function properly. But interface queues are made to catch the traffic last. Due to this obvious conflict, a not expected behavior (which cannot be considered as a bug, it is just the way interface queues work) is observed: while firewall filters and virtual interface queues (**global−in** and **global−out**) are working with the translated addresses, simple queues attached to the interface Universal Client is functioning on 'see' the original, not translated, IP address. So if it is necessary to match the download (outgoing to the client) traffic by the address Universal Client assigns to the client, either **global−out** queue or queue trees should be used.

Queue Types

Submenu level : **/queue type**

Description

The queue types are used to specify some common argument values for queues. There are four default built−in queue types: **default**, **ethernet−default**, **wireless−default**, and **synchronous−default**. The built−in queue types cannot be removed.

Property Description

name (*name*)− name for the queue type
kind (pfifo | bfifo | red | sfq; default: **pfifo**) − kind of the queuing algorithm used:

- **pfifo** − Packets First−In First−Out
- **bfifo** − Bytes First−In First−Out
- **red** − Random Early Detection
- **sfq** − Stochastic Fair Queuing

bfifo−limit (*integer*; default: **15000**) − BFIFO queue limit. Maximum packet number that queue can hold
pfifo−limit (*integer*; default: **10**) − PFIFO queue limit. Maximum byte number that queue can hold
red−limit (*integer*; default: **60**) − RED queue limit
red−min−threshold (*integer*; default: **10**) − RED minimum threshold
red−max−threshold (*integer*; default: **50**) − RED maximum threshold
red−burst (*integer*; default: **20**) − RED burst
sfq−perturb (*integer*; default: **5**) − how often to change hash function
sfq−allot (*integer*; default: **1514**) − amount of data in bytes that can be sent in one round−robin round

Notes

For small limitations (64kbps, 128kbps) RED is more preferable. For larger speeds PFIFO will be as good as RED. RED consumes much more memory and CPU than PFIFO &BFIFO.

Example

To add **red** queue type with minimum threshold of 0, without any burst and named **CUSTOMER−def**:

```
[admin@MikroTik] queue type> add name=CUSTOMER−def kind=red \
\... red−min−threshold=0 red−burst=0
[admin@MikroTik] queue type> print
  0 name=default kind=none bfifo−limit=15000 pfifo−limit=10 red−limit=60
    red−min−threshold=10 red−max−threshold=50 red−burst=20 sfq−perturb=5
    sfq−allot=1514

  1 name=ethernet−default kind=none bfifo−limit=15000 pfifo−limit=10
    red−limit=60 red−min−threshold=10 red−max−threshold=50 red−burst=20
    sfq−perturb=5 sfq−allot=1514

  2 name=wireless−default kind=sfq bfifo−limit=15000 pfifo−limit=10
    red−limit=60 red−min−threshold=10 red−max−threshold=50 red−burst=20
    sfq−perturb=5 sfq−allot=1514

  3 name=synchronous−default kind=red bfifo−limit=15000 pfifo−limit=10
    red−limit=60 red−min−threshold=10 red−max−threshold=50 red−burst=20
    sfq−perturb=5 sfq−allot=1514

  4 name=CUSTOMER−def kind=red bfifo−limit=15000 pfifo−limit=10 red−limit=60
    red−min−threshold=0 red−max−threshold=50 red−burst=0 sfq−perturb=5
    sfq−allot=1514

[admin@MikroTik] queue type>
```

Interface Default Queues

Submenu level : **/queue interface**

Property Description

interface (*name*) − interface name
queue (*name*; default: **default**) − default queue for the interface

Example

To change the default queue type to **wireless−default** for the **prism1** interface:

```
[admin@MikroTik] queue interface> print
  # INTERFACE                              QUEUE
  0 ether1                                 default
  1 prism1                                 default
[admin@MikroTik] queue interface> set prism1 queue=wireless−default
[admin@MikroTik] queue interface> print
  # INTERFACE                              QUEUE
  0 ether1                                 default
```

```
  1 prism1                                wireless-default
[admin@MikroTik] queue interface>
```

Configuring Simple Queues

Submenu level : **/queue simple**

Description

Simple queues can be used to set up data rate management for the whole traffic leaving an interface, or for certain source and/or destination addresses. For more sophisticated queue setup use the queue trees described further on.

Property Description

name (*name*; default: **queue1**) − name of the queue
src-address (*IP address/mask*) − source IP address
dst-address (*IP address/mask*) − destination IP address
interface (*name*) − outgoing interface of the traffic flow

- **global-in** − match all incomming traffic
- **global-out** − match all outgoing traffic

limit-at (*integer*; default: **0**) − allocated stream data rate (bits/s)

- **0** − no limit

queue (*name*; default: **default**) − queue type. If you specify the queue type other than **default**, then it overrides the default queue type set for the interface under **/queue interface**
priority (1...8; default: **8**) − flow priority, **1** is the highest
max-limit (*integer*; default: **0**) − maximum stream data rate (bits/s)

- **0** − no limit

Notes

max-limit must be equal or greater than **limit-at**.

Simple queues are applied before queue trees.

Queue rules are processed in the order they appear in the list. If some packet matches the queue rule, then the queuing mechanism specified in that rule is applied to it, and no more rules are processed for that packet.

Example

To add a simple queue that will limit traffic destined to **192.168.0.0/24** network on **ether1** interface to **128000** bits per second:

```
[admin@MikroTik] queue simple> add dst-address=192.168.0.0/24 interface=ether1\
\... max-limit=128000
[admin@MikroTik] queue simple> print
Flags: X - disabled, I - invalid, D - dynamic
  0   name="queue1" src-address=0.0.0.0/0 dst-address=192.168.0.0/24
      interface=ether1 limit-at=0 queue=default priority=8 max-limit=128000
```

```
[admin@MikroTik] queue simple> print
```

Configuring Queue Trees

Submenu level : **/queue tree**

Description

The queue trees should be used when you want to use sophisticated data rate allocation based on protocols, ports, groups of IP addresses, etc.

Property Description

name (*name*; default: **queue1**) − descriptive name for the queue
parent (*name*) − name of the parent queue. The top−level parents are the available interfaces (actually, main HTB). Lower level parents can be other queues. Dynamic queues (created with the simple queue tool) cannot be used as parents

- **global−in** − match all incomming traffic
- **global−out** − match all outgoing traffic

flow (*name*; default: "") − flow mark of the packets to be queued. Flow marks can be assigned to the packets under **/ip firewall mangle** when the packets enter the router through the incoming interface
limit−at (*integer*; default: **0**) − maximum stream data rate (bits/s)

- **0** − no limit

queue (*name*; default: **default**) − queue type
priority (1...8; default: **8**) − flow priority, **1** is the highest
max−limit (*integer*; default: **0**) − maximum stream data rate (bits/s)

- **0** − no limit

Notes

max−limit must be equal or greater than **limit−at**.

To apply queues on flows, the mangle feature should be used first to mark incomming packets.

If you have added a simple queue, it is listed as dynamic one in this list:

Simple queues are applied before queue trees.

```
[admin@MikroTik] queue simple> print
Flags: X − disabled, I − invalid, D − dynamic
  0   name="simple queue" src−address=0.0.0.0/0 dst−address=192.168.0.0/24
      interface=ether1 limit−at=0 queue=default priority=8 max−limit=128000


[admin@MikroTik] queue simple> .. tree
[admin@MikroTik] queue tree> print
Flags: X − disabled, I − invalid, D − dynamic
  0  D name="simple queue" parent=ether1 flow="" limit−at=0 queue=default
       priority=8 max−limit=128000
```

```
[admin@MikroTik] queue tree>
```

Example

To mark all the thaffic going from web−servers (**TCP** port **80**) with **abc−http** mark:

```
[admin@MikroTik] ip firewall mangle> add action=passthrough mark−flow=abc−http \
\... protocol=tcp src−port=80
[admin@MikroTik] ip firewall mangle> print
Flags: X − disabled, I − invalid
  0   src−address=0.0.0.0/0:80 in−interface=all dst−address=0.0.0.0/0:0−65535
      protocol=tcp tcp−options=any icmp−options=any:any flow=""
      src−mac−address=00:00:00:00:00:00 limit−count=0 limit−burst=0
      limit−time=0s action=passthrough mark−flow=abc−http tcp−mss=dont−change

[admin@MikroTik] ip firewall mangle>
```

You can add queue using the **/queue tree add** command:

```
[admin@MikroTik] queue tree> add name=HTTP parent=ether1 flow=abc−http \
max−limit=128000
[admin@MikroTik] queue tree> print
Flags: X − disabled, I − invalid, D − dynamic
  0  D name="simple queue" parent=ether1 flow="" limit−at=0 queue=default
       priority=8 max−limit=128000

  1    name="HTTP" parent=ether1 flow="abc−http" limit−at=0 queue=default
       priority=8 max−limit=128000


[admin@MikroTik] queue tree>
```

Troubleshooting

- *The queue is not added for the correct interface.*
  Add the queue to the interface through which the traffic is leaving the router. Queuing works only for packets leaving the router!
- *The source/destination addresses of the packets do not match the values specified in the queue setting*
  Make sure the source and destination addresses, as well as network masks are specified correctly! The most common mistake is wrong address/netmask, e.g., 10.0.0.217/24 (wrong), 10.0.0.217/32 (right), or 10.0.0.0/24 (right).
- *The simple queuing does not work when masquerading is in use.*
  Masquerading changes the source address of packets leaving the router ('outgoing' traffic). Therefore the simple queuing rule should match packets having the router's external address as source. Alternatively, queue trees could be used for marked packets. Use the MANGLE feature to mark the packets.

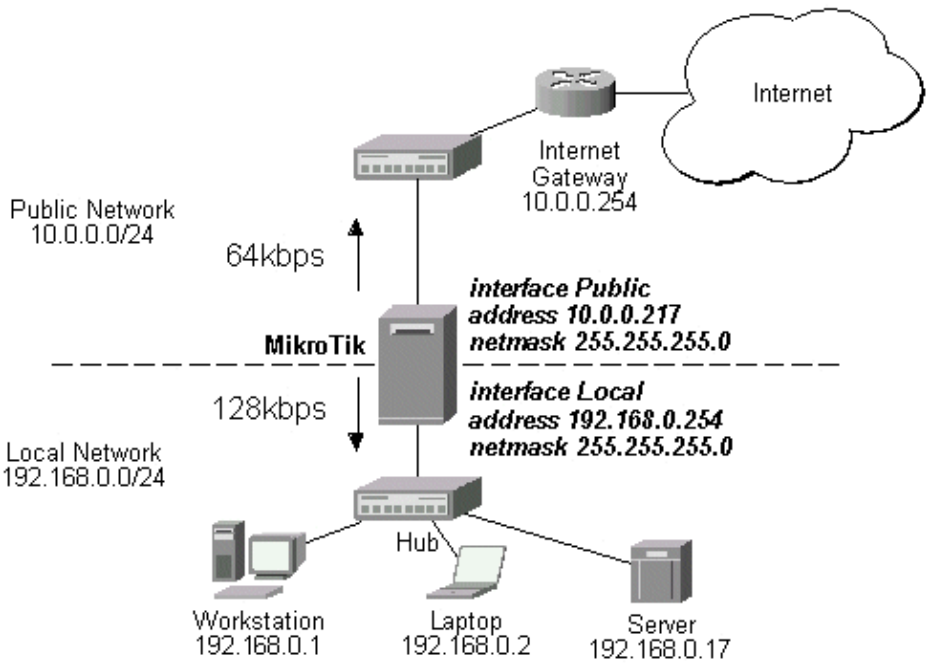
Queue Applications

One of the ways to avoid network traffic ‘jams’ is usage of traffic shaping in large networks. Traffic shaping and data rate allocation is implemented in the MikroTik RouterOS as queuing mechanism. Thus, the network administrator is able to allocate a definite portion of the total data rate and grant it to a

particular network segment or interface. Also the data rate of particular nodes can be limited by using this mechanism.

Example of Emulating a 128k/64k Line

Assume we want to emulate a 128k download and 64k upload line connecting IP network 192.168.0.0/24. The network is served through the Local interface of customer's router. The basic network setup is in the following diagram:



The IP addresses and routes of the MikroTik router are as follows:

```
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.217/24      10.0.0.217      10.0.0.255      Public
  1   192.168.0.254/24   192.168.0.0     192.168.0.255   Local
[admin@MikroTik] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        Public
    1 DC 192.168.0.0/24     r 0.0.0.0         0        Local
    2 DC 10.0.0.0/24        r 0.0.0.0         0        Public
[admin@MikroTik] >
```

Assume you want to limit the data rate to 128kbps on downloads and 64kbps on uploads for all hosts on the LAN. Data rate limitation is done by applying queues for outgoing interfaces regarding the traffic flow. It is enough to add two queues at the MikroTik router:

```
[admin@MikroTik] queue simple> add name=Down interface Local max-limit 128000
[admin@MikroTik] queue simple> add name=UP interface Public max-limit 64000
[admin@MikroTik] queue simple> print
Flags: X - disabled, I - invalid, D - dynamic
```

```
  0   name="Down" src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Local
      limit-at=0 queue=default priority=8 max-limit=128000

  1   name="UP" src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Public
      limit-at=0 queue=default priority=8 max-limit=64000

[admin@MikroTik] queue simple> .. tree print
Flags: X - disabled, I - invalid, D - dynamic
  0  D name="Down" parent=Local flow="" limit-at=0 queue=default priority=8
       max-limit=128000

  1  D name="UP" parent=Public flow="" limit-at=0 queue=default priority=8
       max-limit=64000


[admin@MikroTik] queue simple>
```

Leave all other parameters as set by default. The limit is approximately 128kbps going to the LAN and 64kbps leaving the client's LAN. Please note, that the queues have been added for the outgoing interfaces regarding the traffic flow.

To monitor the traffic flow through the interface while doing file transfer, use the **/interface monitor-traffic** command:

```
[admin@MikroTik] interface> monitor-traffic Public once
    received-packets-per-second: 9
       received-bits-per-second: 4.32kbps
        sent-packets-per-second: 6
           sent-bits-per-second: 65.58kbps

[admin@MikroTik] interface> monitor-traffic Public once
    received-packets-per-second: 7
       received-bits-per-second: 3.36kbps
        sent-packets-per-second: 10
           sent-bits-per-second: 65.15kbps

[admin@MikroTik] interface> monitor-traffic Public once
    received-packets-per-second: 11
       received-bits-per-second: 5.66kbps
        sent-packets-per-second: 7
           sent-bits-per-second: 52.70kbps

[admin@MikroTik] interface>
```

If you want to exclude the server from being limited, add two queues for it with **max-limit=0** (no limit) and move them to the top:

```
[admin@MikroTik] queue simple> add name=Serv_D interface=Local \
\... dst-address=192.168.0.17/32 max-limit=0
[admin@MikroTik] queue simple> add name=Serv_U interface Public \
\... src-address=192.168.0.17/32 max-limit=0
[admin@MikroTik] queue simple> print
Flags: X - disabled, I - invalid, D - dynamic
  0   name="Down" src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Local
      limit-at=0 queue=default priority=8 max-limit=128000

  1   name="UP" src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Public
      limit-at=0 queue=default priority=8  max-limit=64000
```

```
  2   name="Serv_D" src-address=0.0.0.0/0 dst-address=192.168.0.17/32
      interface=Local limit-at=0 queue=default priority=8 max-limit=0

  3   name="Serv_U" src-address=192.168.0.17/32 dst-address=0.0.0.0/0
      interface=Public limit-at=0 queue=default priority=8 max-limit=0

[admin@MikroTik] queue simple> move 2 0
[admin@MikroTik] queue simple> move 3 1
[admin@MikroTik] queue simple> print
Flags: X - disabled, I - invalid, D - dynamic
  0   name="Serv_D" src-address=0.0.0.0/0 dst-address=192.168.0.17/32
      interface=Local limit-at=0 queue=default priority=8 max-limit=0

  1   name="Serv_U" src-address=192.168.0.17/32 dst-address=0.0.0.0/0
      interface=Public limit-at=0 queue=default priority=8 max-limit=0

  2   name="Down" src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Local
      limit-at=0 queue=default priority=8 max-limit=128000

  3   name="UP" src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Public
      limit-at=0 queue=default priority=8 max-limit=64000

[admin@MikroTik] queue simple>
```

Example of Using Masquerading

If masquerading is used for the local address space 192.168.0.0/24 of the client computers in the previous example setup, then the outgoing traffic has masqueraded source address 10.0.0.217, i.e., the outgoing packets have external address of the router as the source.

If you use simple queues, as in the previous example, the queuing rule for incoming traffic should match the customer's local addresses, whereas the rule for outgoing traffic should match the router's external address as the source address. The previous example would work fine, but you cannot exclude the server from being limited.

To apply specific queuing for the server, use **/ip firewall mangle** to mark the packets originated from the server:

```
[admin@MikroTik] ip firewall mangle> add src-address=192.168.0.17/32 \
\... action=passthrough mark-flow=Serv_Up
[admin@MikroTik] ip firewall mangle> add in-interface=Local action=passthrough \
\... mark-flow=Local-all
[admin@MikroTik] ip firewall mangle> print
Flags: X - disabled, I - invalid
  0   src-address=192.168.0.17/32:0-65535 in-interface=all
      dst-address=0.0.0.0/0:0-65535 protocol=all tcp-options=any
      icmp-options=any:any src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=passthrough mark-flow=Serv_Up
      tcp-mss=dont-change

  1   src-address=0.0.0.0/0:0-65535 in-interface=Local
      dst-address=0.0.0.0/0:0-65535 protocol=all tcp-options=any
      icmp-options=any:any src-mac-address=00:00:00:00:00:00 limit-count=0
      limit-burst=0 limit-time=0s action=passthrough mark-flow=Local-all
      tcp-mss=dont-change

[admin@MikroTik] ip firewall mangle>
```

Add a queue to the queue tree, which uses the flow mark:

```
[admin@MikroTik] queue tree> add name=Server parent=Public flow=Serv_Up
[admin@MikroTik] queue tree> add name=Workst parent=Public flow=Local-all \
\... max-limit=64000
[admin@MikroTik] queue tree> print
Flags: X - disabled, I - invalid, D - dynamic
  0    name=Server parent=Public flow=Serv_Up limit-at=0 queue=default
       priority=8 max-limit=0

  1    name=Workst parent=Public flow=Local-all limit-at=0 queue=default
       priority=8 max-limit=128000

[admin@MikroTik] queue tree>
```

Thus, we used queue trees for limiting the upload. Use the same simple queues as in the previous example for limiting the download.

Example of Guaranteed Quality of Service

This example shows how to limit data rate on a channel and guarantee minimum speed to the FTP server allowing other traffic to use the rest of the channel.

Assume we want to emulate a 128k download and 64k upload line connecting IP network 192.168.0.0/24 as in the previous examples. But if these speeds are the best that you can get from your Internet connection, you may want to guarantee certain speeds to the 192.168.0.17 server so that your customers could download from and upload to this server with the speeds not dependent on the other traffic using the same channel (for example, we will guarantee this server the minimum data rate of 32k for each flow direction).

First of all, you should limit the interface speed:

```
[admin@MikroTik] queue tree> add name=Up parent=Public max-limit=64000
[admin@MikroTik] queue tree> print
Flags: X - disabled, I - invalid, D - dynamic
  0    name="Up" parent=Public flow="" limit-at=0 queue=default priority=8
       max-limit=64000

[admin@MikroTik] queue tree>
```

Next, mark the traffic from the FTP server. We will mark only TCP ports 20-21 because these ports are used to send and receive FTP data and control messages.

```
[admin@MikroTik] ip firewall mangle> add src-address=192.168.0.17/32:20-21 \
\... protocol=tcp mark-flow=Server_Up in-interface=Local
[admin@MikroTik] ip firewall mangle> print
Flags: X - disabled, I - invalid, D - dynamic
  0   src-address=192.168.0.17/32:20-21 in-interface=Local
      dst-address=0.0.0.0/0:0-65535 protocol=tcp tcp-options=any
      icmp-options=any:any flow="" src-mac-address=00:00:00:00:00:00
      limit-count=0 limit-burst=0 limit-time=0s action=accept
      mark-flow=Server_Up tcp-mss=dont-change

[admin@MikroTik] ip firewall mangle>
```

The second mangle rule will match the rest of the traffic from the network:

```
[admin@MikroTik] ip firewall mangle> add src-address=0.0.0.0/0 \
\... mark-flow=Local_Up in-interface=Local
[admin@MikroTik] ip firewall mangle> print
Flags: X - disabled, I - invalid, D - dynamic
  0   src-address=192.168.0.17/32:20-21 in-interface=Local
      dst-address=0.0.0.0/0:0-65535 protocol=tcp tcp-options=any
      icmp-options=any:any flow="" src-mac-address=00:00:00:00:00:00
      limit-count=0 limit-burst=0 limit-time=0s action=accept
      mark-flow=Server_Up tcp-mss=dont-change

  1   src-address=0.0.0.0/0:0-65535 in-interface=Local
      dst-address=0.0.0.0/0:0-65535 protocol=tcp tcp-options=any
      icmp-options=any:any flow="" src-mac-address=00:00:00:00:00:00
      limit-count=0 limit-burst=0 limit-time=0s action=accept
      mark-flow=Local_Up tcp-mss=dont-change

[admin@MikroTik] ip firewall mangle>
```

Finally shaping the traffic:

```
[admin@MikroTik] queue tree> add name=Server_Up parent=Up limit-at=32000 \
\... flow=Server_Up max-limit=64000 priority=7
[admin@MikroTik] queue tree> add name=Local_Up parent=Up limit-at=0 \
\... flow=Local_Up
[admin@MikroTik] queue tree> print
Flags: X - disabled, I - invalid, D - dynamic
  0    name="Up" parent=Public flow="" limit-at=0 queue=default priority=8
       max-limit=64000

  1    name="Server_Up" parent=Up flow="Server_Up" limit-at=32000 queue=default
       priority=7 max-limit=64000

  2    name="Local_Up" parent=Up flow="Local_Up" limit-at=0 queue=default
       priority=8 max-limit=0

[admin@MikroTik] queue tree>
```

Thus, we used queue trees for limiting the upload. The download speed can be limited the same way simply changing the interface names and matching the packets destinated to the server (use 'external' server address if you are using DST-NAT):

```
[admin@MikroTik] queue tree> add name=Down parent=Local max-limit=128000
[admin@MikroTik] queue tree> print
Flags: X - disabled, I - invalid, D - dynamic
  0    name="Up" parent=Public flow="" limit-at=0 queue=default priority=8
       max-limit=64000

  1    name="Server_Up" parent=Up flow="Server_Up" limit-at=32000 queue=default
       priority=7 max-limit=64000

  2    name="Local_Up" parent=Up flow="Local_Up" limit-at=0 queue=default
       priority=8 max-limit=0

  3    name="Down" parent=Local flow="" limit-at=0 queue=default priority=8
       max-limit=128000

[admin@MikroTik] queue tree> /ip firewall mangle
[admin@MikroTik] ip firewall mangle> add dst-address=192.168.0.17/32:20-21 \
\... protocol=tcp mark-flow=Server_Down in-interface=Public
[admin@MikroTik] ip firewall mangle> add dst-address=0.0.0.0/0 \
```

```
\... mark-flow=Local_Down in-interface=Public
[admin@MikroTik] ip firewall mangle> print
Flags: X - disabled, I - invalid, D - dynamic
  0   src-address=192.168.0.17/32:20-21 in-interface=Local
      dst-address=0.0.0.0/0:0-65535 protocol=tcp tcp-options=any
      icmp-options=any:any flow="" src-mac-address=00:00:00:00:00:00
      limit-count=0 limit-burst=0 limit-time=0s action=accept
      mark-flow=Server_Up tcp-mss=dont-change

  1   src-address=0.0.0.0/0:0-65535 in-interface=Local
      dst-address=0.0.0.0/0:0-65535 protocol=tcp tcp-options=any
      icmp-options=any:any flow="" src-mac-address=00:00:00:00:00:00
      limit-count=0 limit-burst=0 limit-time=0s action=accept
      mark-flow=Local_Up tcp-mss=dont-change

  2   src-address=0.0.0.0/0:0-65535 in-interface=Public
      dst-address=192.168.0.17/32:20-21 protocol=tcp tcp-options=any
      icmp-options=any:any flow="" src-mac-address=00:00:00:00:00:00
      limit-count=0 limit-burst=0 limit-time=0s action=accept
      mark-flow=Server_Down tcp-mss=dont-change

  3   src-address=0.0.0.0/0:0-65535 in-interface=Public
      dst-address=0.0.0.0/0:0-65535 protocol=tcp tcp-options=any
      icmp-options=any:any flow="" src-mac-address=00:00:00:00:00:00
      limit-count=0 limit-burst=0 limit-time=0s action=accept
      mark-flow=Local_Down tcp-mss=dont-change

[admin@MikroTik] ip firewall mangle> /queue tree
[admin@MikroTik] queue tree> add name=Server_Down parent=Down limit-at=32000 \
\... flow=Server_Down max-limit=128000 priority=7
[admin@MikroTik] queue tree> add name=Local_Down parent=Down limit-at=0 \
\... flow=Local_Down
[admin@MikroTik] queue tree> print
Flags: X - disabled, I - invalid, D - dynamic
  0    name="Up" parent=Public flow="" limit-at=0 queue=default priority=8
       max-limit=64000

  1    name="Server_Up" parent=Up flow="Server_Up" limit-at=32000 queue=default
       priority=7 max-limit=64000

  2    name="Local_Up" parent=Up flow="Local_Up" limit-at=0 queue=default
       priority=8 max-limit=0

  3    name="Down" parent=Local flow="" limit-at=0 queue=default priority=8
       max-limit=128000

  4    name="Server_Down" parent=Down flow="Server_Down" limit-at=32000
       queue=default priority=7 max-limit=128000

  5    name="Local_Down" parent=Down flow="Local_Down" limit-at=0 queue=default
       priority=8 max-limit=0

[admin@MikroTik] queue tree>
```

Additional Resources

- Home of Hierarchical Token Bucket (HTB)
- Paper on Random Early Detection (RED)
- More complete information on Traffic Cotrol

Open Shortest Path First (OSPF) Routing Protocol

Document revision 1.4 (08−Aug−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

MikroTik RouterOS implements OSPF Version 2 (RFC 2328). The OSPF protocol is the link−state protocol that takes care of the routes in the dynamic network structure that can employ different paths to its subnetworks. It always chooses shortest path to the subnetwork first.

Supports event logging.

Specifications

Packages required : ***routing***
License required : *Any*
Home menu level : ***/routing ospf***
Protocols utilized : *OSPF (RFC2328)*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Routes, Equal Cost Multipath Routing, Policy Routing
Log Management

Description

Open Shortest Path First (OSPF) dynamic routing protocol distributes routing information between the routers belonging to a single autonomous system (AS). An AS is a group of routers exchanging routing information via a common routing protocol.

When deploy the OSPF, all routers should be configured in a coordinated manner (**note** that it also means that the routers should have the same MTU for the all the networks advertized by OSPF protocol). Routers belonging to one area should have the same area ID configured. Although Mikrotik RouterOS supports multiple areas, it is not likely that you will deploy structures with many of them.

After you have divided your networks in areas, you have to configure the following settings on each OSPF router:

1. Change general OSPF settings of redistributing connected, static and default routes. The default route should be distributed only from border routers of your area;
2. Configure additional areas, if any;
3. If you're using encryption, you also should configure keys in **/routing ospf interface** command level;
4. Add OSPF network records for all networks you want the OSPF to run on.

The OSPF is started after adding record to the ospf network list.

Note! The OSPF protocol is started only on interfaces configured under the **/routing ospf network**

The routes learned by OSPF protocol are installed in the route list with the distance of **110**.

OSPF Setup

Submenu level : **/routing ospf**

```
[admin@MikroTik] routing ospf> print
                 router-id: 0.0.0.0
         distribute-default: never
    redistribute-connected: no
       redistribute-static: no
          redistribute-rip: no
          redistribute-bgp: no
            metric-default: 1
          metric-connected: 20
             metric-static: 20
                metric-rip: 20
                metric-bgp: 20
[admin@MikroTik] routing ospf>
```

Property Description

router-id (*IP address*; default: **0.0.0.0**) – the Router ID. If not specified, OSPF uses the largest IP address configured on the interfaces as its router ID
distribute-default (default: **never**) :

- **never** - do not send own default route to other routers
- **if-installed-as-type-1** - send the default route (as type 1 metric) only if it has been installed (a static default route, or route added by DHCP, PPP, etc.)
- **if-installed-as-type-2** - send the default route (as type 2 metric) only if it has been installed (a static default route, or route added by DHCP, PPP, etc.)
- **always-as-type-1** - always send the default route (as type 1 metric)
- **always-as-type-2** - always send the default route (as type 2 metric)

redistribute-connected (as-type-1 | as-type-2 | no; default: **no**) - if set, the router will redistribute the information about all connected routes, i.e., routes to networks, that can be directly reached from the router
redistribute-static (as-type-1 | as-type-2 | no; default: **no**) - if set, the router will redistribute the information about all static routes added to its routing database, i.e., routes, that have been created using the **/ip route add** command
redistribute-rip (as-type-1 | as-type-2 | no; default: **no**) - If set, the router will redistribute the information about all routes learned by the RIP protocol
redistribute-bgp (as-type-1 | as-type-2 | no; default: **no**) - If set, the router will redistribute the information about all routes learned by the BGP protocol
metric-default (*integer*; default: **1**) - cost of the default route
metric-connected (*integer*; default: **20**) - cost of connected routes
metric-static (*integer*; default: **20**) - cost of static routes
metric-rip (*integer*; default: **20**) - cost of the routes learned by the RIP protocol
metric-bgp (*integer*; default: **20**) - cost of the routes learned by the BGP protocol

Notes

Within an area, only the router that is connected to an another AS (i.e. border router) should have the propagation of the default route enabled.

OSPF protocol will try to use the shortest path (path with the least total cost) if available.

OSPF protocol supports two types of metrics:
- **type 1** metrics are internal ('cheap') metrics
- **type 2** metrics are external ('expensive') metrics. Any **type 2** metric is considered greater than the cost of any internal path

Example

To enable OSPF protocol redisrtibute routes to the connected networks as type 1 metrics with the cost of **1**:

```
[admin@MikroTik] routing ospf> set redistribute-connected=as-type-1 \
\... metric-connected=1
[admin@MikroTik] routing ospf> print
                 router-id: 0.0.0.0
        distribute-default: never
    redistribute-connected: as-type-1
       redistribute-static: no
          redistribute-rip: no
          redistribute-bgp: no
            metric-default: 1
          metric-connected: 1
             metric-static: 20
                metric-rip: 20
                metric-bgp: 20
[admin@MikroTik] routing ospf>
```

OSPF Areas

Submenu level : **/routing ospf area**

Property Description

name (*name*; default: "") − area name
area-id (*IP address*; default: **0.0.0.0**) − area ID
default-cost (*integer*; default: **1**) − cost for the default summary route used for a stub area. Only for area boundary router
stub (yes | no; default: **no**) − area type
authentication (md5 | none | simple; default: **none**) − authentication method for OSPF:
- **none** − no authentication
- **simple** − plain text authentication
- **md5** − Keyed Message Digest 5 (MD5) authentication

Notes

There is one area that is configured by default − the backbone area (area ID **0.0.0.0**). **name** and **area-id** cannot be changed for this area.

Example

To define an additional OSPF area named **local_10** with ID=**0.0.10.5**:

```
[admin@MikroTik] routing ospf area> add area-id=0.0.10.5 name=local_10
[admin@MikroTik] routing ospf area> print
Flags: X - disabled, I - invalid
  #   NAME                    AREA-ID         STUB DEFAULT-COST AUTHENTICATION
  0   backbone                0.0.0.0                           none
  1   local_10                0.0.10.5        no   1            none

[admin@MikroTik] routing ospf area>
```

OSPF Network

Submenu level : **/routing ospf network**

Description

To start the OSPF protocol, you have to define the networks on which OSPF runs and the area ID for those networks.

Property Description

area (*name*; default: **backbone**) - area to be associated with the address range
network (*IP address/mask*; default: **0.0.0.0/0**) - the network associated with the area The network argument allows defining one or multiple interfaces to be associated with a specific OSPF area. Only directly connected networks of the router may be specified

Notes

For P2P links here you should set exactly the same as the network address is (that is remote point IP address). In this case, the correct netmask bits should be **32**

Example

To enable OSPF protocol on **10.10.1.0/24** network, and include it to the **backbone** area:

```
[admin@MikroTik] routing ospf network> add area=backbone network=10.10.1.0/24
[admin@MikroTik] routing ospf network> print
Flags: X - disabled
  #   NETWORK            AREA
  0   10.10.1.0/24       backbone
[admin@MikroTik] routing ospf>
```

OSPF Interfaces

Submenu level : **/routing ospf interface**

Description

To run OSPF you don't have to configure interfaces. This command level is only for additional configuration of OSPF specific interface parameters.

Prorerty Description

interface (*name*; default: **all**) − interface on which OSPF runs

- **all** sets the defaults, that will be used for all the interfaces not having specific settings

cost (*integer: 1..65535*; default: **1**) − interface cost expressed as the link state metric
priority (*integer: 0..255*; default: **1**) − router priority. It helps to determine the designated router for the network. When two routers attached to a network both attempt to become the designated router, the one with the higher router priority takes precedence
authentication−key (*string*; default: "") − authentication key to be used by neighboring routers that are using OSPF's simple password authentication
retransmit−interval (*time*; default: **5s**) − time between retransmitting lost link state advertisements. When a router sends a link state advertisement (LSA) to its neighbor, it keeps the LSA until it receives back the acknowledgment. If it receives no acknowledgment in seconds, it will retransmit the LSA
transmit−delay (*time*; default: **1s**) − link state transmit delay is the estimated time it takes to transmit a link state update packet on the interface
hello−interval (*time*; default: **10s**) − the interval between hello packets that the router sends on the interface. The smaller the **hello−interval**, the faster topological changes will be detected, but more routing traffic will ensue. This value must be the same for all routers on a specific network
dead−interval (*time*; default: **40s**) − interval after which a neighbor is declared dead. The interval is advertised in the router's hello packets. This value must be the same for all routers and access servers on a specific network

Example

To add an entry that specifies that **ether2** interface should send Hello packets every 5 seconds:

```
[admin@MikroTik] routing ospf> interface add interface=ether2 hello−interval=5s
[admin@MikroTik] routing ospf> interface print
  0 interface=ether2 cost=1 priority=1 authentication−key=""
    retransmit−interval=5s transmit−delay=1s hello−interval=5s
    dead−interval=40s


[admin@MikroTik] routing ospf>
```

OSPF Virtual Links

Submenu level : **/routing ospf network**

Description

Virtual links connect physically separate components of backbone area. The two endpoints of a virtual link are area border routers. The virtual link must be configured in both routers.

Property Description

neighbor−id (*IP address*; default: **0.0.0.0**) − **router−id** of the neighbour
transit−area (*name*; default: **backbone**) − non−backbone area the two routers have in common

Notes

Virtual links cannot be established through stub areas.

Example

To add a virtual link with the **10.0.0.201** router through the **ex** area:

```
[admin@MikroTik] routing ospf virtual−link> add neighbor−id=10.0.0.201 \
\... transit−area=ex
[admin@MikroTik] routing ospf virtual−link> print
Flags: X − disabled, I − invalid
  #   NEIGHBOR−ID     TRANSIT−AREA
  0   10.0.0.201      ex
[admin@MikroTik] routing ospf virtual−link>
```

OSPF Neighbours

Submenu level : **/routing ospf neighbor**

Description

The submenu provides an access to the list of OSPF neighbors for the router, with brief statistics

Proprety Description

router−id (*read−only*: *IP address*) − **router−id** parameter of the OSPF neighbour
address (*read−only*: *IP address*) − appropriate IP address of the OSPF neighbor
priority (*read−only*: *integer*) − priority of neighbor which is used in designated router elections on this network
state (*read−only*: *text*) − state of the connection:

- **Down** − the connection is down
- **Attempt** − sending Hello packet
- **Init** − Hello packet received from the neighbour
- **2−Way** − bidirectional communication established
- **ExStart** − negotiating Exchange state
- **Exchange** − exchanging with hole Link−State DataBase
- **Loading** − receiving information from the neighbour
- **Full** − the neighboring routers are fully adjacent (the link−state databases are completely synchronized)

state−changes (*read−only*: *integer*) − number of state changes of the connection
ls−retransmits (*read−only*: *integer*) − number of Link State retransmits
ls−requests (*read−only*: *integer*) − number of Link State requests
db−summaries (*read−only*: *integer*) − number of records in link−state database advertised by the neighbour
dr−id (*read−only*: *IP address*) − router id of designated router for this neighbor

backup−dr−id (*read−only*: *IP address*) − router id of backup designated router for this neighbor

Notes

The list also shows the router itself in this list.

Example

The following text can be viewed just after adding an OSPF network:

```
[admin@MikroTik] routing ospf> neighbor print
 router−id=10.0.0.204 address=10.0.0.204 priority=1 state="2−Way"
    state−changes=0 ls−retransmits=0 ls−requests=0 db−summaries=0
    dr−id=0.0.0.0 backup−dr−id=0.0.0.0

[admin@MikroTik] routing ospf>
```

An Example of Running OSPF

After configuring OSPF on a number of interconnected routers, dynamic routes should appear in the **ip route print** list:

```
[admin@MikroTik] ip route> print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S ;;; our default gateway
         0.0.0.0/0          r 10.0.0.1        1        ether1
    1 DC 192.168.0.0/24     r 0.0.0.0         0        ether4
    2 DO 10.10.10.0/24      r 10.10.1.1       110      ether2
    3 DC 10.10.1.0/24       r 0.0.0.0         0        ether2
    4 DC 10.0.0.0/24        r 0.0.0.0         0        ether1
[admin@MikroTik] routing ospf>
```

In this case, we have one one route connected through 10.10.1.1 router (item #2). As current router distributes its routes too (including default one), in 10.10.1.1 router we have:

```
[admin@Remote] > ip route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0 DO 0.0.0.0/0          r 10.10.1.2       110      ether1
    1 DO 192.168.0.0/24     r 10.10.1.2       110      ether1
    2 DC 10.10.10.0/24      r 0.0.0.0         0        radiolan1
    3 DC 10.10.1.0/24       r 0.0.0.0         0        ether1
    4 DO 10.5.5.0/24        r 10.10.1.2       110      ether1
    5 DO 10.0.0.0/24        r 10.10.1.2       110      ether1
[admin@Remote] >
```

OSPF Troubleshooting

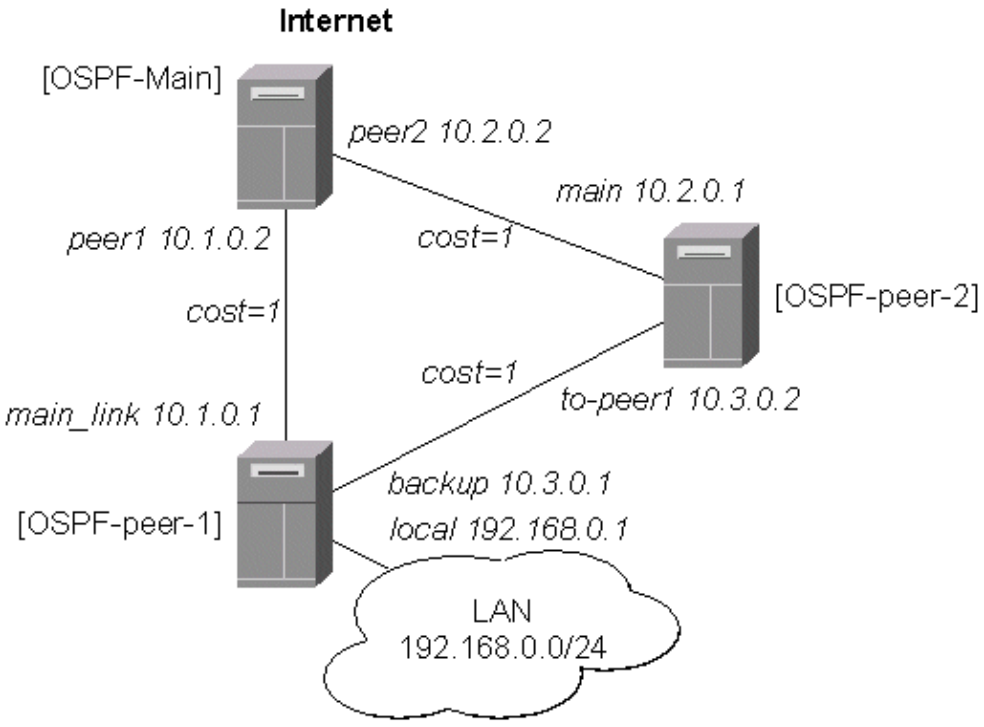
- *OSPF does not work on point−to−point link (PPP, PPPoE, PPTP)*
  Make sure you include the remote address of the point−to−point link into the **/router ospf network** record. For example, if you have

```
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.7.1.3/24        10.7.1.0        10.7.1.255      backbone
  1   192.168.223.55/25  192.168.223.0   192.168.223.127 aironet
  2 D 10.2.0.7/32        10.2.0.8        0.0.0.0         pptp-out1
[admin@MikroTik] ip address>
```

Use **/router ospf network add network=10.2.0.8/32 area=backbone**.

OSPF Backup without using Tunnel

This example shows how to use OSPF for backup purposes, if you are controlling all the involved routers, and you can run OSPF on them.



Let us assume that the link between the routers OSPF−Main and OSPF−peer−1 is the main one. If it goes down, we want the traffic switch over to the link going through the router OSPF−peer−2.

For this:

1. We introduce an OSPF area with area ID=0.0.0.1, which includes all three routers shown on the diagram.
2. Only the OSPF−Main router will have the default route configured. Its interfaces peer1 and peer2 will be configured for the OSPF protocol. The interface main_gw will not be used for distributing the OSPF routing information.
3. The routers OSPF−peer−1 and OSPF−peer−2 will distribute their connected route information, and receive the default route using the OSPF protocol.

OSPF_Main Router Setup

The IP address configuration of the [OSPF_Main] router is as follows:

```
[admin@OSPF−Main] interface> /ip address print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.214/24      10.0.0.0        10.0.0.255      main_gw
  1   10.1.0.2/24        10.1.0.0        10.1.0.255      peer1
  2   10.2.0.2/24        10.2.0.0        10.2.0.255      peer2
[admin@OSPF−Main] interface>
```

OSPF settings:

```
[admin@OSPF−Main] > routing ospf print
                 router−id: 0.0.0.0
        distribute−default: if−installed−as−type−2
    redistribute−connected: as−type−1
       redistribute−static: as−type−2
          redistribute−rip: no
          redistribute−bgp: no
            metric−default: 1
          metric−connected: 0
             metric−static: 0
                metric−rip: 0
                metric−bgp: 0

[admin@OSPF−Main] > routing ospf area print
Flags: X − disabled
  0   name=backbone area−id=0.0.0.0 default−cost=0 stub=no
      authentication=none

  1   name=local_10 area−id=0.0.0.1 default−cost=0 stub=no
      authentication=none

[admin@OSPF−Main] > routing ospf network print
Flags: X − disabled
  #   NETWORK            AREA
  0   10.1.0.0/24        local_10
  1   10.2.0.0/24        local_10
[admin@OSPF−Main] >
```

OSPF−peer−1 Router Setup

The IP address configuration of the [OSPF−peer−1] router is as follows:

```
[admin@OSPF−peer−1] > ip address print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.1.0.1/24        10.1.0.0        10.1.0.255      main_link
  1   10.3.0.1/24        10.3.0.0        10.3.0.255      backup
  2   192.168.0.1/24     192.168.0.0     192.168.0.255   local
[admin@OSPF−peer−1] >
```

OSPF settings:

```
[admin@OSPF−peer−1] > routing ospf print
                 router−id: 0.0.0.0
```

```
        distribute-default: never
    redistribute-connected: as-type-1
       redistribute-static: no
          redistribute-rip: no
          redistribute-bgp: no
            metric-default: 1
          metric-connected: 0
             metric-static: 0
                metric-rip: 0
                metric-bgp: 0
[admin@OSPF-peer-1] > routing ospf area print
Flags: X - disabled
  0   name=backbone area-id=0.0.0.0 default-cost=0 stub=no
      authentication=none

  1   name=local_10 area-id=0.0.0.1 default-cost=0 stub=no
      authentication=none

[admin@OSPF-peer-1] > routing ospf network print
Flags: X - disabled
  #   NETWORK            AREA
  0   10.3.0.0/24        local_10
  1   10.1.0.0/24        local_10
[admin@OSPF-peer-1] >
```

OSPF−peer−2 Router Setup

The IP address configuration of the [OSPF−peer−2] router is as follows:

```
[admin@OSPF-peer-2] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.2.0.1/24        10.2.0.0        10.2.0.255      main
  1   10.3.0.2/24        10.3.0.0        10.3.0.255      to-peer1
[admin@OSPF-peer-2] >
```

OSPF settings:

```
[admin@OSPF-peer-2] > routing ospf print
                 router-id: 0.0.0.0
        distribute-default: never
    redistribute-connected: as-type-1
       redistribute-static: no
          redistribute-rip: no
          redistribute-bgp: no
            metric-default: 1
          metric-connected: 0
             metric-static: 0
                metric-rip: 0
                metric-bgp: 0
[admin@OSPF-peer-2] > routing ospf area print
Flags: X - disabled
  0   name=backbone area-id=0.0.0.0 default-cost=0 stub=no
      authentication=none

  1   name=local_10 area-id=0.0.0.1 default-cost=0 stub=no
      authentication=none

[admin@OSPF-peer-2] > routing ospf network print
```

```
Flags: X - disabled
  #   NETWORK            AREA
  0   10.2.0.0/24        local_10
  1   10.3.0.0/24        local_10
[admin@OSPF-peer-2] >
```

Routing Tables

After the three routers have been set up as described above, and the links between them are operational, the routing tables of the three routers should look as follows:

On the main OSPF router:

```
[admin@OSPF-Main] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        main_gw
    1 DO 192.168.0.0/24     r 10.1.0.1        110      peer1
    2 DC 10.2.0.0/24        r 0.0.0.0         0        peer2
    3 DO 10.3.0.0/24        r 10.2.0.1        110      peer2
                            r 10.1.0.1                 peer1
    4 DC 10.1.0.0/24        r 0.0.0.0         0        peer1
    5 DC 10.0.0.0/24        r 0.0.0.0         0        main_gw

[admin@OSPF-Main] >
```

On the Peer 1:

```
[admin@OSPF-peer-1] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0 DO 0.0.0.0/0          r 10.1.0.2        110      main_link
    1 DC 192.168.0.0/24     r 0.0.0.0         0        local
    2 DO 10.2.0.0/24        r 10.1.0.2        110      main_link
                            r 10.3.0.2                 backup
    3 DC 10.3.0.0/24        r 0.0.0.0         0        backup
    4 DC 10.1.0.0/24        r 0.0.0.0         0        main_link
    5 DO 10.0.0.0/24        r 10.1.0.2        110      main_link
[admin@OSPF-peer-1] >
```

On the Peer 2:

```
[admin@OSPF-peer-2] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0 DO 0.0.0.0/0          r 10.2.0.2        110      main
    1 DO 192.168.0.0/24     r 10.3.0.1        110      to-peer1
    2 DC 10.2.0.0/24        r 0.0.0.0         0        main
    3 DC 10.3.0.0/24        r 0.0.0.0         0        to-peer1
    4 DO 10.1.0.0/24        r 10.3.0.1        110      to-peer1
                            r 10.2.0.2                 main
    5 DO 10.0.0.0/24        r 10.2.0.2        110      main
[admin@OSPF-peer-2] >
```

Please note the three equal cost multipath routes (multiple gateways for one destination) in this setup. They have been created by the OSPF, because there is equal cost to go, for example, from the router OSPF−peer−2 to the network 10.1.0.0/24.

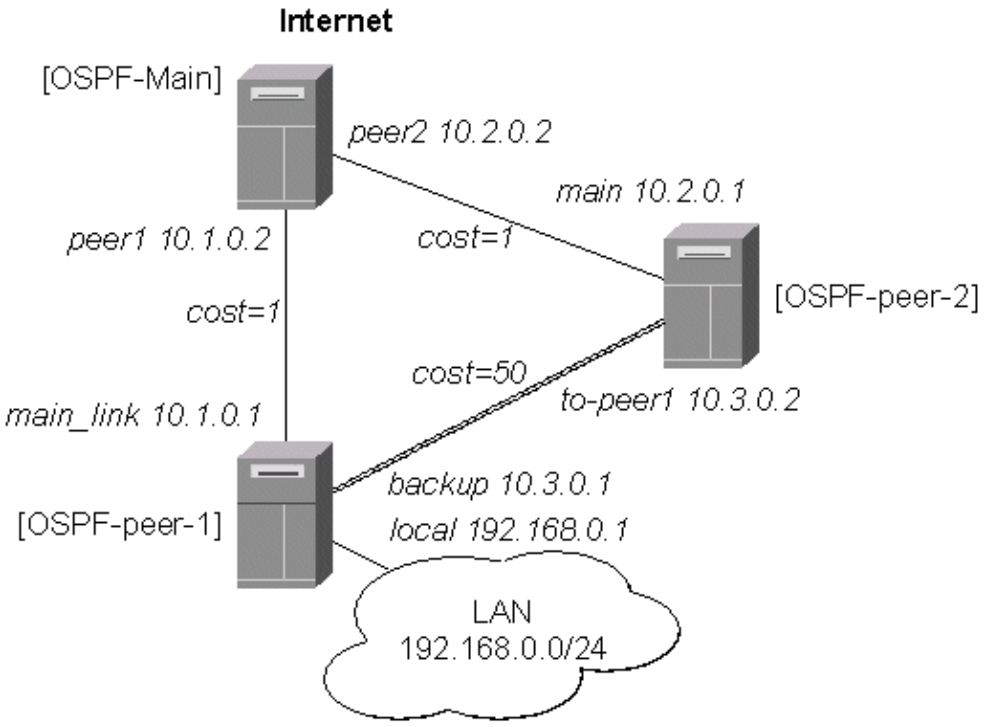
> The cost is calculated as the sum of costs over each hop to the destination. Unless this is not specially desired, we may want to avoid such situations, i.e., and adjust the cost settings for the interfaces (links) accordingly.

Routing Tables with Revised Link Cost

Let us assume, that the link between the routers OSPF−peer−1 and OSPF−peer−2 has a higher cost (might be slower, we have to pay more for the traffic through it, etc.). Since we have left all ospf interface cost settings as default (cost=1), we need to change the following settings:

```
[admin@OSPF−peer−1] > routing ospf interface add interface=backup cost=50
[admin@OSPF−peer−2] > routing ospf interface add interface=to−peer2 cost=50
```

The revised network diagram:



After changing the cost settings, we have only one equal cost multipath route left − to the network 10.3.0.0/24 from the OSPF−Main router:

On the main OSPF router:

```
[admin@OSPF−Main] > ip route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        main_gw
    1 DO 192.168.0.0/24     r 10.1.0.1        110      peer1
```

```
    2 DC 10.2.0.0/24        r 0.0.0.0         0        peer2
    3 DO 10.3.0.0/24        r 10.2.0.1        110      peer2
                            r 10.1.0.1                 peer1
    4 DC 10.1.0.0/24        r 0.0.0.0         0        peer1
    5 DC 10.0.0.0/24        r 0.0.0.0         0        main_gw

[admin@OSPF-Main] >
```

On the Peer 1:

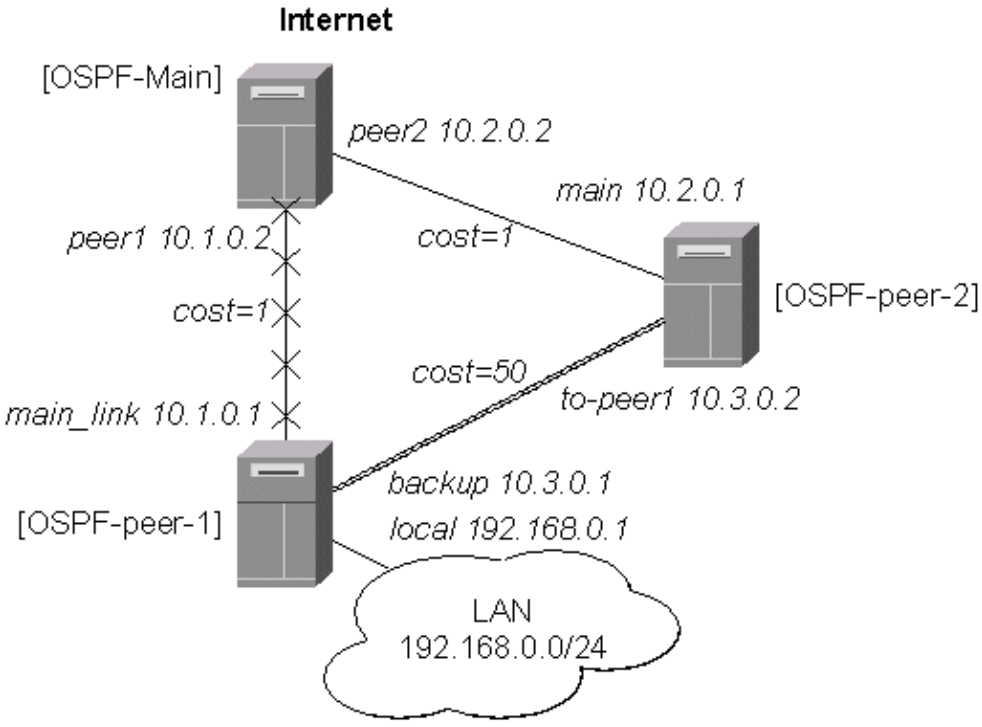
```
[admin@OSPF-peer-1] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0 DO 0.0.0.0/0          r 10.1.0.2        110      main_link
    1 DC 192.168.0.0/24     r 0.0.0.0         0        local
    2 DO 10.2.0.0/24        r 10.1.0.2        110      main_link
    3 DC 10.3.0.0/24        r 0.0.0.0         0        backup
    4 DC 10.1.0.0/24        r 0.0.0.0         0        main_link
    5 DO 10.0.0.0/24        r 10.1.0.2        110      main_link
[admin@OSPF-peer-1] >
```

On the Peer 2:

```
[admin@OSPF-peer-2] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0 DO 0.0.0.0/0          r 10.2.0.2        110      main
    1 DO 192.168.0.0/24     r 10.3.0.1        110      to-peer1
    2 DC 10.2.0.0/24        r 0.0.0.0         0        main
    3 DC 10.3.0.0/24        r 0.0.0.0         0        to-peer1
    4 DO 10.1.0.0/24        r 10.2.0.2        110      main
    5 DO 10.0.0.0/24        r 10.2.0.2        110      main
[admin@OSPF-peer-2] >
```

Functioning of the Backup

If the link between routers OSPF−Main and OSPF−peer−1 goes down, we have the following situation:



The OSPF routing changes as follows:

On the main OSPF router:

```
[admin@OSPF−Main] > ip route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        main_gw
    1 DO 192.168.0.0/24     r 10.2.0.1        110      peer2
    2 DC 10.2.0.0/24        r 0.0.0.0         0        peer2
    3 DO 10.3.0.0/24        r 10.2.0.1        110      peer2
    4 DC 10.1.0.0/24        r 0.0.0.0         0        peer1
    5 DC 10.0.0.0/24        r 0.0.0.0         0        main_gw

[admin@OSPF−Main] >
```

On the Peer 1:

```
[admin@OSPF−peer−1] > ip route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0 DO 0.0.0.0/0          r 10.3.0.2        110      backup
    1 DC 192.168.0.0/24     r 0.0.0.0         0        local
    2 DO 10.2.0.0/24        r 10.3.0.2        110      backup
    3 DC 10.3.0.0/24        r 0.0.0.0         0        backup
    4 DC 10.1.0.0/24        r 0.0.0.0         0        main_link
    5 DO 10.0.0.0/24        r 10.3.0.2        110      backup
[admin@OSPF−peer−1] >
```

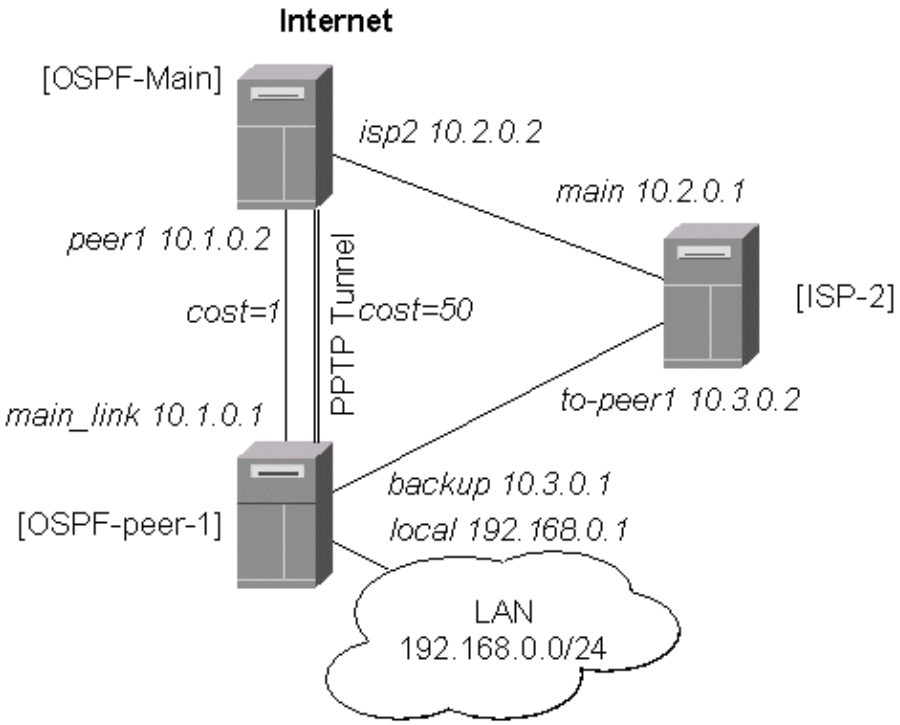
On the Peer 2:

```
[admin@OSPF-peer-2] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0 DO 0.0.0.0/0          r 10.2.0.2        110      main
    1 DO 192.168.0.0/24     r 10.3.0.1        110      to-peer1
    2 DC 10.2.0.0/24        r 0.0.0.0         0        main
    3 DC 10.3.0.0/24        r 0.0.0.0         0        to-peer1
    4 DO 10.1.0.0/24        r 10.2.0.2        110      main
    5 DO 10.0.0.0/24        r 10.2.0.2        110      main
[admin@OSPF-peer-2] >
```

The change of the routing takes approximately 40 seconds (the hello-interval setting). If required, this setting can be adjusted, but it should be done on all routers within the OSPF area!

OSPF Backup using Encrypted Tunnel through a Third Party

This example shows how to use OSPF for backup purposes, if you have to use third party link for backup, and you are not controlling the routers on the backup link.



Let us assume that the link between the routers OSPF-Main and OSPF-peer-1 is the main one. When the main link goes down, the backup link should go through the ISP-2 router. Since we cannot control the ISP-2 router, we cannot run OSPF on the backup router like in the previous example with OSPF-peer-2. Therefore we have to create a tunnel between the routers OSPF-Main and OSPF-peer-1 that goes through the ISP-2 router. Thus, we will have two links between the routers, and the traffic should switch over to the backup when the main link goes down.

For this:

1. We create a PPTP tunnel between our two routers, which goes over the ISP-2 router. Please

consult the PPTP Interface Manual on how to create PPTP tunnels.

2. Only the OSPF−Main router will have the default route configured. Its interfaces peer1 and pptp−in1 will be configured for the OSPF protocol. The interface main_gw will not be used for distributing the OSPF routing information.
3. The router OSPF−peer−1 will distribute its connected and static route information, and receive the default route from OSPF−main using the OSPF protocol.

OSPF_Main Router Setup

The PPTP static server configuration is as follows:

```
[admin@OSPF−Main] > ip route add dst−address=10.3.0.1/32 gateway=10.2.0.1
[admin@OSPF−Main] > ppp secret add name=ospf service=pptp password=asdf4 \
\... local−address=10.4.0.2 remote−address=10.4.0.1
[admin@OSPF−Main] > interface pptp−server add name=pptp−in1 user=ospf
[admin@OSPF−Main] > interface pptp−server server set enabled=yes
[admin@OSPF−Main] > interface pptp−server print
Flags: X − disabled, D − dynamic, R − running
  #     NAME                USER         MTU   CLIENT−ADDRESS  UPTIME   ENC...
  0     pptp−in1            ospf
[admin@OSPF−Main] >
```

The IP address configuration of the [OSPF_Main] router is as follows:

```
[admin@OSPF−Main] > ip address print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.214/24      10.0.0.0        10.0.0.255      main_gw
  1   10.2.0.2/24        10.2.0.0        10.2.0.255      isp2
  2   10.1.0.2/24        10.1.0.0        10.1.0.255      peer1
  3 D 10.4.0.2/32        10.4.0.1        0.0.0.0         pptp−in1
[admin@OSPF−Main] >
```

OSPF settings:

```
[admin@OSPF−Main] routing ospf> print
                 router−id: 0.0.0.0
        distribute−default: if−installed−as−type−1
    redistribute−connected: as−type−1
       redistribute−static: no
          redistribute−rip: no
          redistribute−bgp: no
            metric−default: 1
          metric−connected: 20
             metric−static: 20
                metric−rip: 20
                metric−bgp: 20
[admin@OSPF−Main] routing ospf> interface add interface=pptp−in1 cost=50
[admin@OSPF−Main] routing ospf> interface print
  0 interface=pmi cost=150 priority=1 authentication−key=""
    retransmit−interval=5s transmit−delay=1s hello−interval=10s
    dead−interval=40s

[admin@OSPF−Main] routing ospf> area print
Flags: X − disabled, I − invalid
  #   NAME                    AREA−ID         STUB DEFAULT−COST AUTHENTICATION
  0   backbone                0.0.0.0                           none
```

```
[admin@OSPF−Main] routing ospf> network print
Flags: X − disabled, I − invalid
  #   NETWORK            AREA
  0   10.1.0.0/24        backbone
  1   10.4.0.1/32        backbone
[admin@OSPF−Main] routing ospf>
```

Note, that the OSPF is configured only for the peer1 and pptp−in1 interfaces. Since the pptp−in1 is a point−to−point interface, the network address has 32 bits.

OSPF−peer−1 Router Setup

The PPTP client configuration is as follows:

```
[admin@OSPF−peer−1] > ip route add dst−address=10.2.0.2/32 gateway=10.3.0.2
[admin@OSPF−peer−1] > interface pptp−client add name=pptp−out1 user=ospf \
\... connect−to=10.2.0.2 password=asdf4 mtu=1500 mru=1500
[admin@OSPF−peer−1] > interface pptp−client enable pptp−out1
[admin@OSPF−peer−1] > interface pptp−client print
Flags: X − disabled, R − running
  0  R name="pptp−out1" mtu=1500 mru=1500 connect−to=10.2.0.2 user="ospf"
       password="asdf4" profile=default add−default−route=no

[admin@OSPF−peer−1] > interface pptp−client monitor pptp−out1
      status: "connected"
      uptime: 39m46s
    encoding: "none"

[admin@OSPF−peer−1] >
```

The IP address configuration of the [OSPF−peer−1] router is as follows:

```
[admin@OSPF−peer−1] > ip address print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.1.0.1/24        10.1.0.0        10.1.0.255      main_link
  1   10.3.0.1/24        10.3.0.0        10.3.0.255      backup
  2   192.168.0.1/24     192.168.0.0     192.168.0.255   local
  3 D 10.4.0.1/32        10.4.0.2        0.0.0.0         pptp−out1
[admin@OSPF−peer−1] >
```

OSPF settings:

```
[admin@OSPF−peer−1] routing ospf> print
                 router−id: 0.0.0.0
        distribute−default: never
    redistribute−connected: as−type−1
       redistribute−static: no
          redistribute−rip: no
          redistribute−bgp: no
            metric−default: 1
          metric−connected: 20
             metric−static: 20
                metric−rip: 20
                metric−bgp: 20
[admin@OSPF−peer−1] routing ospf> interface add interface=pptp−out1 cost=50
[admin@OSPF−peer−1] routing ospf> interface print
```

```
  0 interface=pptp-out1 cost=50 priority=1 authentication-key=""
    retransmit-interval=5s transmit-delay=1s hello-interval=10s
    dead-interval=40s

[admin@OSPF-peer-1] routing ospf> area print
Flags: X - disabled, I - invalid
  #   NAME                    AREA-ID         STUB DEFAULT-COST AUTHENTICATION
  0   backbone                0.0.0.0                           none

[admin@OSPF-peer-1] routing ospf> network print
Flags: X - disabled, I - invalid
  #   NETWORK            AREA
  0   10.1.0.0/24        backbone
  1   10.4.0.2/32        backbone
[admin@OSPF-peer-1] routing ospf>
```

Routing Tables

After the PPTP tunnel and OSPF protocol between two routers has been set up as described above, and the links between them are operational, the routing tables of the two routers should look as follows:

```
[admin@OSPF-Main] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        main_gw
    1  S 10.3.0.1/32        r 10.2.0.1        1        isp2
    2 DO 192.168.3.0/24     r 10.1.0.1        110      peer1
    3 DO 192.168.0.0/24     r 10.1.0.1        110      peer1
    4 DO 10.4.0.2/32        r 10.1.0.1        110      peer1
    5 DC 10.4.0.1/32        r 0.0.0.0         0        pptp-in1
    6 DO 10.3.0.0/24        r 10.1.0.1        110      peer1
    7 DC 10.2.0.0/24        r 0.0.0.0         0        isp2
    8 DO 10.2.0.2/32        r 10.1.0.1        110      peer1
    9 DC 10.1.0.0/24        r 0.0.0.0         0        peer1
   10 DC 10.0.0.0/24        r 0.0.0.0         0        main_gw
[admin@OSPF-Main] >

================================================================================

[admin@OSPF-peer-1] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 10.2.0.0/24        r 10.3.0.2        1        backup
    1  S 192.168.3.0/24     r 192.168.0.20    1        local
    2  S 10.2.0.2/32        r 10.3.0.2        1        backup
    3 DO 0.0.0.0/0          r 10.1.0.2        110      main_link
    4 DC 192.168.0.0/24     r 0.0.0.0         0        local
    5 DC 10.4.0.2/32        r 0.0.0.0         0        pptp-out1
    6 DO 10.4.0.1/32        r 10.1.0.2        110      main_link
    7 DC 10.3.0.0/24        r 0.0.0.0         0        backup
    8 DC 10.1.0.0/24        r 0.0.0.0         0        main_link
    9 DO 10.0.0.0/24        r 10.1.0.2        110      main_link
[admin@OSPF-peer-1] >
```

Functioning of the Backup

If the link between routers OSPF−Main and OSPF−peer−1 goes down, the OSPF routing changes as follows:

```
[admin@OSPF−Main] > ip route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        main_gw
    1  S 10.3.0.1/32        r 10.2.0.1        1        isp2
    2 DO 192.168.3.0/24     r 10.4.0.1        110      pptp−in1
    3 DO 192.168.0.0/24     r 10.4.0.1        110      pptp−in1
    4 DO 10.4.0.2/32        r 10.4.0.1        110      pptp−in1
    5 DC 10.4.0.1/32        r 0.0.0.0         0        pptp−in1
    6 DO 10.3.0.0/24        r 10.4.0.1        110      pptp−in1
    7 DC 10.2.0.0/24        r 0.0.0.0         0        isp2
    8 DO 10.2.0.2/32        r 10.4.0.1        110      pptp−in1
    9 DC 10.1.0.0/24        r 0.0.0.0         0        peer1
   10 DC 10.0.0.0/24        r 0.0.0.0         0        main_gw
[admin@OSPF−Main] >

=======================================================

[admin@OSPF−peer−1] > ip route print
Flags: X − disabled, I − invalid, D − dynamic, J − rejected,
C − connect, S − static, R − rip, O − ospf, B − bgp
    #    DST−ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 10.2.0.0/24        r 10.3.0.2        1        backup
    1  S 192.168.3.0/24     r 192.168.0.20    1        local
    2  S 10.2.0.2/32        r 10.3.0.2        1        backup
    3 DO 0.0.0.0/0          r 10.4.0.2        110      pptp−out1
    4 DC 192.168.0.0/24     r 0.0.0.0         0        local
    5 DC 10.4.0.2/32        r 0.0.0.0         0        pptp−out1
    6 DO 10.4.0.1/32        r 10.4.0.2        110      pptp−out1
    7 DC 10.3.0.0/24        r 0.0.0.0         0        backup
    8 DC 10.1.0.0/24        r 0.0.0.0         0        main_link
    9 DO 10.0.0.0/24        r 10.4.0.2        110      pptp−out1
[admin@OSPF−peer−1] >
```

As we see, all routing goes through the PPTP tunnel now.

Additional Resources

Recommended readings for guidelines on building OSPF networks:

- http://www.ietf.org/rfc/rfc2328.txt
- OSPF Design Guide, Cisco Systems
- Designing Large−Scale IP Internetworks, Cisco Systems

Routing Prefix Lists

Document revision 1.0 (21−Jan−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

Prefix lists are used to filter routes received from or sent to other routers.

Specifications

Packages required : ***routing***
License required : *Any*
Home menu level : ***/routing prefix−list***
Protocols utilized : *None*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Routes, Equal Cost Multipath Routing, Policy Routing
RIP, Router Information Protocol
BGP, Border Gateway Protocol

Description

Filtering by prefix list involves matching the prefixes of routes with those listed in the prefix list. When there is a match, the route is used. The prefix lists are used when specifying the BGP peers under **/routing bgp peer** or RIP interfaces under **/routing rip interface**.

To match a prefix−list entry, route should have its prefix (i.e. destination address) matching **prefix** property of the entry, and it should have the length of its prefix (i.e. mask of destination address) matching **prefix−length** property of the entry.

Prefix List Setup

Submenu level : **/routing prefix−list**

Property Description

name (*name*, default: "") − name for the prefix list
default−action (accept | reject, default: **accept**) − default action for all members of this list

Notes

An empty prefix list permits or rejects all prefixes.

Exapmle

To add a **cybernet** list that rejects the routes by default:

```
[admin@MikroTik] routing prefix−list> add name=cybernet default−action=reject
[admin@MikroTik] routing prefix−list> print
  # NAME                                                    DEFAULT−ACTION
  0 cybernet                                                reject
[admin@MikroTik] routing prefix−list>
```

Prefix List Rules

Submenu level : **/routing prefix−list list *listname***

Property Description

prefix (*IP address/mask*, default: **0.0.0.0/0**) − network prefix
prefix−length (*integer−integer*, default: 0−32) − length (range) of the network prefix in bits
action (accept | reject, default: **accept**) − action for the list member

Notes

There are two different values to match − prefix (i.e. destination address of the route applying the network mask) and prefix length. Prefix length match network mask of the received route.

For example, if **prefix**=172.16.0.0/16 and **prefix=length**=16−24, then received route for 172.16.24.0/24 will match, but route for 172.16.24.0/25 will not.

Example

To accept the routes to the **172.16.0.0/16** network and any of it's subnetworks that has their network mask between 16 and 24.

```
[admin@MikroTik] routing prefix-list> list cybernet
[admin@MikroTik] routing prefix-list list cybernet> add prefix=172.16.0.0/16 \
\... prefix-length=16-24
[admin@MikroTik] routing prefix-list list cybernet> print
  # PREFIX             PREFIX-LENGTH ACTION
  0 172.16.0.0/16      16-24         accept
[admin@MikroTik] routing prefix-list list cybernet>
```

Routing Information Protocol (RIP)

Document revision 1.1 (02−May−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

MikroTik RouterOS implements RIP Version 1 (RFC1058) and Version 2 (RFC 2453). RIP lets routers in the same autonomous system exchange routing information. It always uses the best path (the path with te fewest number of hops (i.e. routers)) available.

Specifications

Packages required : ***routing***
License required : *Any*

Home menu level : ***/routing rip***
Protocols utilized : *RIPv1 (RFC1058), RIPv2 (RFC2453)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Routes, Equal Cost Multipath Routing, Policy Routing
Prefix Lists

Description

Routing Information Protocol (RIP) is one protocol in a series of routing protocols based on Bellman−Ford (or distance vector) algorithm. This Interior Gateway Protocol (IGP) lets routers in the same autonomous system exchange routing information in the way of periodic RIP updates. Routers transmit their own RIP updates to neighboring networks and listen to the RIP updates from the routers on those neighboring networks to ensure their routing table reflects current state of the network and all the best paths are available. Best path is a path with the fewest hops (routers gateways).

The routes learned by RIP protocol are installed in the route list with the distance of **120**.

RIP Routing Setup

Submenu level : **/routing rip**

```
[admin@MikroTik] routing rip> print
       redistribute-static: no
    redistribute-connected: no
         redistribute-ospf: no
          redistribute-bgp: no
             metric-static: 1
          metric-connected: 1
               metric-ospf: 1
                metric-bgp: 1
              update-timer: 30s
             timeout-timer: 3m
             garbage-timer: 2m
[admin@MikroTik] routing rip>
```

Proprety Description

redistribute−static (yes | no, default: **no**) − redistribution of static routes to neighbor routers
redistribute−connected (yes | no, default: **no**) − redistribution of connected routes to neighbor routers
redistribute−ospf (yes | no, default: **no**) − redistribution of routes learned by OSPF to neighbor routers
redistribute−bgp (yes | no, default: **no**) − redistribution of routes learned by BGP to neighbor routers
metric−static (*integer*, default: **1**) − metric, the distance to the destination for static routes
metric−connected (*integer*, default: **1**) − metric, the distance to the destination for connected routes
metric−ospf (*integer*, default: **1**) − metric, the distance to the destination for OSPF routes
metric−bgp (*integer*, default: **1**) − metric, the distance to the destination for BGP routes
update−timer (*time*, default: **30s**) − time period for RIP update to start

timeout−timer (*time*, default: **3m**)− time period after route is not valid more
garbage−timer (*time*, default: **2m**)− time period after dropped out route is dropped from neighbor router table

Notes

The maximum metric of RIP route can be **15**. Metric higher than **15** is considered 'infinity' and routes with such metric are considered unreachable. Thus RIP cannot be used on networks with more than 15 hops between any two routers, and using redistribute metrics larger that **1** further reduces this maximum hop count.

Example

To enable RIP protocol to redistribute the routes to the connected networks:

```
[admin@MikroTik] routing rip> set redistribute-connected=yes
[admin@MikroTik] routing rip> print
       redistribute-static: no
    redistribute-connected: yes
         redistribute-ospf: no
          redistribute-bgp: no
             metric-static: 1
          metric-connected: 1
               metric-ospf: 1
                metric-bgp: 1
              update-timer: 30s
             timeout-timer: 3m
             garbage-timer: 2m
[admin@MikroTik] routing rip>
```

RIP Interfaces

Submenu level : **/routing rip interface**

Description

To run RIP you don't have to configure interfaces. This command level is only for additional configuration of RIP specific interface parameters.

Prorerty Description

interface (*name*, default: **all**) − interface on which rus RIP
- **all** sets the defaults, that will be used for all the interfaces not having specific settings
  send (v1 | v1−2 | v2, default: **v2**) − distributed RIP protocol versions
  receive (v1 | v1−2 | v2, default: **v2**) − RIP protocol versions the router can receive
  authentication (none | simple | md5, default: **none**) − authentication method for RIP messages:
- **none** − no authentication
- **simple** − clear text authentication
- **md5** − Keyed Message Digest 5 (MD5) authentication
  authentication−key (*string*, default: "") − authentication key for RIP messages
  prefix−list−in (*name*, default: "") − name of the filtering prefix list for receiving routes
  prefix−list−out (*name*, default: "") − name of the filtering prefix list for advertising routes

Notes

Security issue: it is recommended not to use RIP version 1 when it is possible.

Example

To add an entry that specifies that sending routes to the **ether1** interface, prefix list **plout** should be applied:

```
[admin@MikroTik] routing rip> interface add interface=ether1 \
\... prefix-list-out=plout
[admin@MikroTik] routing rip> interface print
Flags: I - inactive
  0   interface=ether1 receive=v2 send=v2 authentication=none
      authentication-key="" prefix-list-in=plout prefix-list-out=none


[admin@MikroTik] routing rip>
```

RIP Networks

Submenu level : **/routing rip network**

Description

To start the RIP protocol, you have to define the networks on which RIP runs.

Proprety Description

address (*IP address/mask*, dafault: **0.0.0.0/0**) - the network on which RIP is running. It allows defining one or multiple interfaces RIP to be run on. Only directly connected networks of the router may be specified **network** (*IP address*, default: **0.0.0.0**) - specifies the network mask of the **address** (if it is not specified in the **address** argument)

Notes

For P2P links here you should set exactly the same as the network address is (that is remote point IP address). In this case, the correct netmask bits should be **32**

Example

To enable RIP protocol on **10.10.1.0/24** network:

```
[admin@MikroTik] routing rip network> add address=10.10.1.0/24
[admin@MikroTik] routing rip network> print
  # ADDRESS
  0 10.10.1.0/24
[admin@MikroTik] routing rip>
```

RIP Neighbors

Description

The submenu is used to define a neighboring router with which to exchange routing information. Normally there is no need to add the neighbors, if the multicasting is working properly within the network. If there are problems with exchanging the routing information, the neighbors can be added to the list. It will force to exchange the routing information with the neighbor.

Proprety Description

address (*IP address*, dafault: **0.0.0.0**) − the neighbour's address

Example

To force RIP protocol to exchange routing information with the **10.0.0.1** router:

```
[admin@MikroTik] routing rip> neighbor add address=10.0.0.1
[admin@MikroTik] routing rip> neighbor print
Flags: I − inactive
  #   ADDRESS
  0   10.0.0.1
[admin@MikroTik] routing rip>
```

RIP Routes

Submenu level : **/routing rip route**

Property Description

Statistics:

dst−address (*IP address/mask*) − destination network address and netmask
gateway (*IP address*) − last gateway to destination address
metric (*integer*) − distance vector length to the network
from (*IP address*) − from which router this route was received

Notes

This list shows the routes learned by all dynamic routing protocols (RIP, OSPF, BGP)

Example

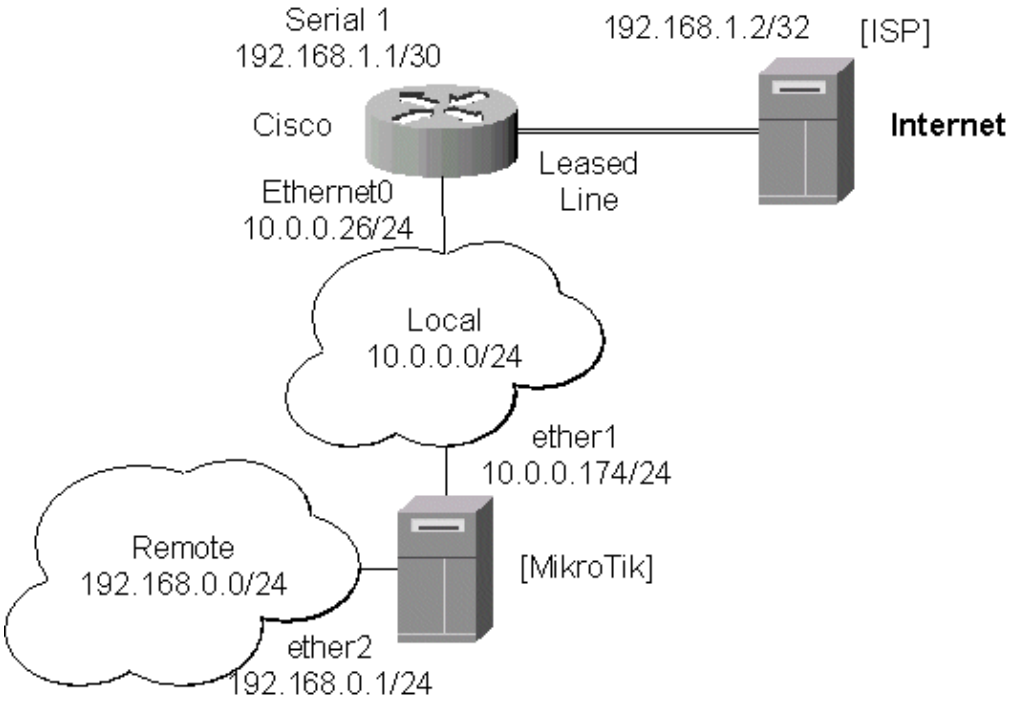
To view the routes:

```
[admin@MikroTik] routing rip route> print
Flags: S − static, R − rip, O − ospf, C − connect, B − bgp
  0 O dst−address=0.0.0.0/32 gateway=10.7.1.254 metric=1 from=0.0.0.0

...
```

```
 33 R dst-address=159.148.10.104/29 gateway=10.6.1.1 metric=2 from=10.6.1.1

 34 R dst-address=159.148.10.112/28 gateway=10.6.1.1 metric=2 from=10.6.1.1

[admin@MikroTik] routing rip route>
```

RIP Examples

Let us consider an example of routing information exchange between MikroTik router, a Cisco router, and the ISP (also mikrotik) routers:



The Configuration of the MikroTik Router

The configuration of the MikroTik router is as follows:

```
[admin@MikroTik] > interface print
Flags: X - disabled, D - dynamic, R - running
  #    NAME                TYPE             MTU
  0  R ether1               ether            1500
  1  R ether2               ether            1500
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.174/24      10.0.0.174      10.0.0.255      ether1
  1   192.168.0.1/24     192.168.0.0     192.168.0.255   ether2
[admin@MikroTik] > ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0 DC 192.168.0.0/24     r 0.0.0.0         0        ether2
    1 DC 10.0.0.0/24        r 0.0.0.0         0        ether1
[admin@MikroTik] >
```

Note, that no default route has been configured. The route will be obtained using the RIP. The necessary configuration of the RIP general settings is as follows:

```
[admin@MikroTik] routing rip> set redistribute-connected=yes
[admin@MikroTik] routing rip> print
       redistribute-static: no
    redistribute-connected: yes
         redistribute-ospf: no
          redistribute-bgp: no
             metric-static: 1
          metric-connected: 1
               metric-ospf: 1
                metric-bgp: 1
              update-timer: 30s
             timeout-timer: 3m
             garbage-timer: 2m

[admin@MikroTik] routing rip>
```

The minimum required configuration of RIP interface is just enabling the network associated with the **ether1** interface:

```
[admin@MikroTik] routing rip network> add address=10.0.0.0/2
[admin@MikroTik] routing rip network> print
  # ADDRESS
  0 10.0.0.0/24

[admin@MikroTik] routing rip network>
```

Note, that there is no need to run RIP on the **ether2**, as no propagation of RIP information is required into the Remote network in this example. The routes obtained by RIP can be viewed in the **/routing rip route** menu:

```
[admin@MikroTik] routing rip> route print
Flags: S - static, R - rip, O - ospf, C - connect, B - bgp
  0 R dst-address=0.0.0.0/0 gateway=10.0.0.26 metric=2 from=10.0.0.26

  1 C dst-address=10.0.0.0/24 gateway=0.0.0.0 metric=1 from=0.0.0.0

  2 C dst-address=192.168.0.0/24 gateway=0.0.0.0 metric=1 from=0.0.0.0

  3 R dst-address=192.168.1.0/24 gateway=10.0.0.26 metric=1 from=10.0.0.26

  4 R dst-address=192.168.3.0/24 gateway=10.0.0.26 metric=1 from=10.0.0.26

[admin@MikroTik] routing rip>
```

The regular routing table is:

```
[MikroTik] routing rip> /ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  R 0.0.0.0/0          r 10.0.0.26       120      ether1
    1  R 192.168.3.0/24     r 10.0.0.26       120      ether1
    2  R 192.168.1.0/24     r 10.0.0.26       120      ether1
    3 DC 192.168.0.0/24     r 0.0.0.0         0        ether2
    4 DC 10.0.0.0/24        r 0.0.0.0         0        ether1
```

```
[admin@MikroTik] routing rip>
```

As we can see, the MikroTik router has learned RIP routes from the Cisco router.

The Configuration of the Cisco Router

```
Cisco#show running-config
...
interface Ethernet0
 ip address 10.0.0.26 255.255.255.0
 no ip directed-broadcast
!
interface Serial1
 ip address 192.168.1.1 255.255.255.252
 ip directed-broadcast
!
router rip
 version 2
 redistribute connected
 redistribute static
 network 10.0.0.0
 network 192.168.1.0
!
ip classless
!
...
```

The routing table of the Cisco router is:

```
Cisco#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 192.168.1.2 to network 0.0.0.0

     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, Ethernet0
R    192.168.0.0/24 [120/1] via 10.0.0.174, 00:00:19, Ethernet0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial1
R    192.168.3.0/24 [120/1] via 192.168.1.2, 00:00:05, Serial1
R*   0.0.0.0/0 [120/1] via 192.168.1.2, 00:00:05, Serial1
Cisco#
```

As we can see, the Cisco router has learned RIP routes both from the MikroTik router (192.168.0.0/24), and from the ISP router (0.0.0.0/0 and 192.168.3.0/24).

Additional Resources

Links for RIP documentation:

- http://www.ietf.org/rfc/rfc1058.txt - RIPv1 Protocol
- http://www.ietf.org/rfc/rfc2453.txt - RIPv2 Protocol

- Cisco Systems RIP protocol overview

Border Gateway Protocol (BGP) Routing Protocol

Document revision 1.1 (23−Nov−2002)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

The Border Gateway Protocol (BGP) allows setting up an interdomain dynamic routing system that automatically generates the routing table for routing between autonomous systems (AS).

MikroTik RouterOS supports BGP Version 4, as defined in RFC1771.

The MikroTik RouterOS implementation of the BGP has filtering (using prefix lists) feature.

Specifications

Packages required : ***routing***
License required : *Any*
Home menu level : ***/routing bgp***
Protocols utilized : *BGP (RFC1771)*
Hardware usage: *requires additional RAM for storing routing information (128MB recommended)*

Related Documents

Software Package Installation and Upgrading

IP Addresses and Address Resolution Protocol (ARP)
Routes, Equal Cost Multipath Routing, Policy Routing
Prefix Lists

Description

The Border Gateway Protocol (BGP) is an Exterior Gateway Protocol (EGP). It allows setting up an interdomain routing system that automatically guarantees the loop−free exchange of routing information between autonomous systems (AS). It is widely used in companies assigned with a definite IP address ranges and connected to a nubmer of ISPs simultaneously so that if one of the links is down, the IP address ranges are still reachable via an another ISP.

The MikroTik RouterOS implementation of the BGP supports filtering with prefix lists, that is used for filtering received and sent routing information

The routes learned by BGP protocol are installed in the route list with the distance of **200** for iBGP (Internal BGP) routes and of **20** for eBGP (External BGP) routes.

BGP Setup

Submenu level : **/routing bgp**

```
[admin@MikroTik] routing bgp> print
                    enabled: no
                         as: 1
                  router-id: 0.0.0.0
         redistribute-static: no
     redistribute-connected: no
           redistribute-rip: no
          redistribute-ospf: no
                      state: disabled
[admin@MikroTik] routing bgp>
```

Property Description

enabled (yes | no, default: **no**) − enable or disable the BGP
as (*integer*, default: **1**) − autonomous system number
router−id (*IP address*, default: **0.0.0.0**) – the Router ID
redistribute−connected (yes | no, default: **no**) – if enabled, the router will redistribute the information about all connected routes, i.e., routes to the networks, that can be directly reached from the router
redistribute−static (yes | no, default: **no**) – if enabled, the router will redistribute the information about all static routes added to its routing database, i.e., routes, that have been created using the **/ip route add** command on the router
redistribute−rip (yes | no, default: **no**) – if enabled, the router will redistribute the information about all routes learned by the RIP protocol
redistribute−ospf (yes | no, default: **no**) – if enabled, the router will redistribute the information about all routes learned by the OSPF protocol

Statistics:

state (disabled | running | terminating) − status of the BGP:

- **disabled** − not working, has been disabled
- **running** − working
- **terminating** − shutting dows, flushing all route information

Notes

Usually you want to redistribute connected and static routes, if any. Therefore change the settings for these arguments and proceed to the BGP networks.

Example

To enable BGP protocol specifying that router **192.168.0.206**, that belongs to the **65002** AS, should redistribute the connected routes:

```
[admin@MikroTik] routing bgp> set enabled=yes router-id=192.168.0.206 as=65002 \
\... redistribute-connected=yes
[admin@MikroTik] routing bgp> print
                   enabled: yes
                        as: 65002
                 router-id: 159.148.147.206
        redistribute-static: no
    redistribute-connected: yes
          redistribute-rip: no
         redistribute-ospf: no
                     state: running
[admin@MikroTik] routing bgp>
```

BGP Network

Submenu level : **/routing bgp network**

Description

BGP Networks is a list of the networks to be advertized.

Property Description

network (*IP address/mask*, default: **0.0.0.0/0**) − network to advertize.

Notes

You can add to the list as many networks as required.

The router is not checking whether the network is in the routing table, it always advertises all the routes that are specified here.

Note the difference with OSPF, that use network list for different purpose − to determine where to send updates.

Example

To specify the router should advertize the **159.148.150.192/27** network:

```
[admin@MikroTik] routing bgp network> add network=159.148.150.192/27
[admin@MikroTik] routing bgp network> print
  # NETWORK
  0 192.168.150.192/27
[admin@MikroTik] routing bgp network>
```

BGP Peers

Submenu level : **/routing bgp peer**

Description

You need to specify the BGP peer with whom you want to exchange the routing information. The BGP exchanges routing information only if it can establish a TCP connection to its peer. You can add as many peers as required

Property Description

remote−address (*IP address*, default: **0.0.0.0**) − address of the remote peer
remote−as (*integer*, default: **0**) − AS number of the remote peer
multihop (yes | no, default: **no**) − if enabled, allows BGP sessions, even when the neighbor is not on a directly connected segment. The multihop session is not established if the only route to the multi−hop peer's address is the default route (0.0.0.0/0)
route−reflect (yes | no, default: **no**) − defines whether to redistribute further the routes learned from the router of the same AS or not. If enabled, can significantly reduce traffic between routers in the same AS
prefix−list−in (*name*, default: "") − name of the filtering prefix list for receiving routes
prefix−list−out (*name*, default: "") − name of the filtering prefix list for advertising routes

Statistics:

state (connected | not−connected) − the status of the BGP connection to the peer
routes−received − the number of received routes from this peer

Example

To enable routing information exchange with the neigbour (non−multihop) **192.168.0.254** that belongs to **65002** AS:

```
[admin@MikroTik] routing bgp peer> add remote−address=192.168.0.254 remote−as=217
[admin@MikroTik] routing bgp peer> print
  # REMOTE−ADDRESS  REMOTE−AS MULTIHOP ROUTE−REFLECT PREFIX−LIS... PREFIX−LI...
  0 192.168.0.254   65002     no       no            none          none
[admin@MikroTik] routing bgp> peer print status
  # REMOTE−ADDRESS  REMOTE−AS STATE          ROUTES−RECEIVED
  0 192.168.0.254   65002     connected      1
[admin@MikroTik] routing bgp>
```

Troubleshooting

- *The BGP does not learn routes from its peer.*
  Try to see if the peer is directly attached, or you should use the **multihop** flag when defining the peer and static routing to get the connection between the peers.
- *I can ping from one peer to the other one, but no routing exchange takes place.*
  Check the status of the peer using **/routing bgp peer print detail** command. See if you do not have firewall that blocks TCP port 179.

Additional Resources

Recommended readings for guidelines on building BGP networks:

- BGP − 4, http://www.ietf.org/rfc/rfc1771.txt
- http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/icsbgp4.htm
- Designing Large−Scale IP Internetworks, Cisco Systems

Authentication, Authorization and Accounting

Document revision 1.14 (06−Oct−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

Authentication, Authorization and Accounting feature provides a possibility of local and/or remote (on RADIUS server) Point−to−Point and HotSpot user management and traffic accounting (all IP traffic passing the router is accounted)

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/user***, ***/ppp***, ***/ip accounting***, ***/radius***
Protocols utilized : *RADIUS (RFC2865)*
Hardware usage: *local traffic accounting requires some memory*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
HotSpot Gateway

Description

The MikroTik RouterOS provides scalable Authentication Athorization and Accounting (AAA) functionality.

Local authentication is done consulting User Database and Profile Database. The configuration is collected from the respective item in User Database (determined by the username), from the item in Profile Database,

that is associated with this item and from the item in Profile Database, that is set as default for the service the user is authenticating to. Settings received from the default profile for the service is overriden by the respective settings from the user's profile, and the resulting settings are overriden by the respective settings taken from the User Database (the only exception is that concrete IP addresses take precedence over IP pools in the **local−address** and **remote−address** settings, as described later on).

RADIUS authentication gives the ISP or network administrator the ability to manage P2P user access and accounting from one server throughout a large network. The MikroTik RouterOS has a RADIUS client which can authenticate for PPP, PPPoE, PPTP, L2TP and ISDN connections. The attributes received from RADIUS server override the ones set in the default profile, but if some parameters are not received they are taken from the respective default profile.

Traffic is accounted locally with Cisco **IP pairs** and snapshot image can be gathered using Syslog utilities. If RADIUS accounting is enabled, accounting information is also sent to the RADIUS server default for that service.

Router User AAA

Description

The router user can manage the router connecting from the local console, via serial terminal, telnet, SSH and Winbox. Router user permissions are determined by the group the user belongs to.

Router User Groups

Submenu level : **/user group**

Property Description

name (*name*) − group name
policy (*multiple choice*: local | telnet | ssh | ftp | reboot | read | write | policy | test | web) − group rights:

- **local** − User can log on locally via console
- **telnet** − User can log on remotely via telnet
- **ssh** − User can log on remotely via secure shell
- **ftp** − User can log on remotely via ftp and send and retrieve files from the router
- **reboot** − User can reboot the router
- **read** − User can retrieve the configuration
- **write** − User can retrieve and change the configuration
- **policy** − Manage user policies, add and remove user
- **test** − User can run ping, traceroute, bandwidth test
- **web** − user can log on remotely via winbox

Notes

There are three system groups which cannot be deleted:

```
[admin@MikroTik] user group> print
  0 ;;; users with read only permission
    name="read"
    policy=local,telnet,ssh,!ftp,reboot,read,!write,!policy,test,web
```

```
  1 ;;; users with write permission
    name="write"
    policy=local,telnet,ssh,!ftp,reboot,read,write,!policy,test,web

  2 ;;; users with complete access
    name="full" policy=local,telnet,ssh,ftp,reboot,read,write,policy,test,web

[admin@MikroTik] user group>
```

Exclamation sign (**!**) just before policy name means **NOT**.

Example

To add **reboot** group that is allowed to reboot the router locally or using telnet, as well as read the router's configuration:

```
[admin@MikroTik] user group> add name=reboot policy=telnet,reboot,read
[admin@MikroTik] user group> print
  0 ;;; users with read only permission
    name="read"
    policy=local,telnet,ssh,!ftp,reboot,read,!write,!policy,test,web

  1 ;;; users with write permission
    name="write"
    policy=local,telnet,ssh,!ftp,reboot,read,write,!policy,test,web

  2 ;;; users with complete access
    name="full" policy=local,telnet,ssh,ftp,reboot,read,write,policy,test,web

  3 name="reboot"
    policy=!local,telnet,!ssh,!ftp,reboot,read,!write,!policy,!test,!web

[admin@MikroTik] user group>
```

Router Users

Submenu level : **/user**

Property Description

name (*name*) − user name. Must start with an alphanumeric character and may contain alphanumeric characters, "*", "_", ".", "@"
group (*name*) − name of the group the user belongs to
password (*string*; default: "")− user password. If not specified, it is left blank (hit 'Enter' when logging in). It conforms to standard Unix characteristics of passwords. Can contain letters, digits, "*" and "_"
address (*IP address/mask*; default: **0.0.0.0/0**)− IP address form which the user is allowed to log in
netmask (*IP address*) − network mask of addresses assigned to the user

Notes

There is one predefined user that cannot be deleted:

```
[admin@MikroTik] user> print
Flags: X − disabled
  #   NAME                                            GROUP ADDRESS
```

```
  0   ;;; system default user
      admin                                           full  0.0.0.0/0

[admin@MikroTik] user>
```

When the user has logged in he can change his password using the **/password** command. The user is required to enter his/her current password before entering the new password. When the user logs out and logs in for the next time, the new password must be entered.

Example

To add user **joe** with password **j1o2e3** belonging to **write** group:

```
[admin@MikroTik] user> add name=joe password=j1o2e3 group=write
[admin@MikroTik] user> print
Flags: X - disabled
  0   ;;; system default user
      name="admin" group=full address=0.0.0.0/0

  1   name="joe" group=write address=0.0.0.0/0


[admin@MikroTik] user>
```

Monitoring Active Router Users

Command name : **/user active print**

Property Description

Statistics:

when (*date*) − log−in time
name (*name*) − user name
address (*IP address*) − IP address from which the user is accessing the router

- **0.0.0.0** − if the user is logged in locally

via (console | telnet | ssh | web) − access method

Example

```
[admin@MikroTik] user> active print
Flags: R - radius
  #   WHEN                 NAME                            ADDRESS         VIA
  0   feb/21/2003 17:48:21 admin                           0.0.0.0         console
  1   feb/24/2003 22:14:48 admin                           10.0.0.144      ssh
  2   mar/02/2003 23:36:34 admin                           10.0.0.144      web

[admin@MikroTik] user>
```

Router User Remote AAA

Submenu level : **/user aaa**

```
[admin@MikroTik] user aaa> print
```

```
        use-radius: no
        accounting: yes
    interim-update: 0s
     default-group: read
[admin@MikroTik] user aaa>
```

Property Description

use-radius (yes | no, default: **no**) - whether user database in a RADIUS server should be consulted
accounting (yes | no, default: **yes**) - whether RADIUS accounting is used
interim-update (*time*, default: **0s**) - Interim-Update time interval
default-group (*name*; default: **read**) - group used by default for users authenticated via RADIUS server

Notes

RADIUS user database is consulted only if the required username is not found in local user database

Example

To enable RADIUS AAA:

```
[admin@MikroTik] user aaa> set use-radius=yes
[admin@MikroTik] user aaa> print
        use-radius: yes
        accounting: yes
    interim-update: 0s
     default-group: read
[admin@MikroTik] user aaa>
```

Local Point-to-Point AAA

Local P2P User Profiles

Submenu level : **/ppp profile**

Description

P2P profiles are used to define default values to users managed in **/ppp secret** submenu. Settings in **/ppp secret** override corresponding **/ppp profile** settings except in the case when **local-address** or **remote-address** are configured in both **/ppp secret** and **/ppp profile**, but in one of them ip pool is referred, concrete IP addresses always take precedence.

Property Descripion

name (*name*) - profile name
local-address (*IP address* | *name*; default: **0.0.0.0**) - either address or pool of the P2P server
remote-address (*IP address* | *name*; default: **0.0.0.0**) - either address or pool of the P2P client
session-timeout (*time*; default: **0s**) - the maximum time the connection can stay up

- **0s** - no timeout

idle-timeout (*time*; default: **0s**) - the link will be terminated if there is no activity within the time set

- **0s** - no timeout

use-compression (yes | no, default: **no**) - defines whether compress traffic or not

use−vj−compression (yes | no, default: **no**) − use Van Jacobson header compression
use−encryption (yes | no, default: **no**) − defines whether encrypt traffic or not
require−encryption (yes | no, default: **no**) − defines whether require encryption from the client or simply prefer it
only−one (yes | no, default: **no**) − allow only one connection at a time
tx−bit−rate (*integer*, default: **0**) − Transmit bitrate in bits/s
rx−bit−rate (*integer*, default: **0**) − Receive bitrate in bits/s
incoming−filter (*name*; default: "") − firewall chain name for incoming packets. If not empty for each packet coming from client, this firewall chain will get control
outgoing−filter (*name*; default: "") − firewall chain name for outgoing packets. If not empty for each packet coming to client, this firewall chain will get control
wins−server (*string*; default: "") − the Windows DHCP client will use this as the default WINS server. Two comma−separated WINS servers can be specified to be used by P2P user as primary and secondary WINS servers

Notes

One default profile is created:

```
[admin@MikroTik] ppp profile> print
Flags: * − default
  0 * name="default" local−address=0.0.0.0 remote−address=0.0.0.0
      session−timeout=0s idle−timeout=0s use−compression=no
      use−vj−compression=no use−encryption=yes require−encryption=no
      only−one=no tx−bit−rate=0 rx−bit−rate=0 incoming−filter=""
      outgoing−filter="" wins−server=""

[admin@MikroTik] ppp profile>
```

Use VJ compression only if You have to because it may slow down the communications on bad or congested channels.

tx−bit−rate and **rx−bit−rate** are used for PPPoE connections only.

Example

To add the profile **ex** that will assign the router itself the **10.0.0.1** address, and the addresses from the **ex** pool to the clients:

```
[admin@MikroTik] ppp profile> add name=ex local−address=10.0.0.1 remote−address=ex
[admin@MikroTik] ppp profile> print
Flags: * − default
  0 * name="default" local−address=0.0.0.0 remote−address=0.0.0.0
      session−timeout=0s idle−timeout=0s use−compression=no
      use−vj−compression=no use−encryption=yes require−encryption=no
      only−one=no tx−bit−rate=0 rx−bit−rate=0 incoming−filter=""
      outgoing−filter="" wins−server=""

  1   name="ex" local−address=10.0.0.1 remote−address=ex session−timeout=0s
      idle−timeout=0s use−compression=no use−vj−compression=no
      use−encryption=no require−encryption=no only−one=no tx−bit−rate=0
      rx−bit−rate=0 incoming−filter="" outgoing−filter="" wins−server=""
```

```
[admin@MikroTik] ppp profile>
```

Local P2P User Database

Submenu level : **/ppp secret**

Description

P2P User Database stores P2P users and defines owner and profile for each of them.

Property Description

name (*name*) − user name
service (any | async | isdn | l2tp | pppoe | pptp; default: **any**) − specifies service that will use this user
caller−id (*string*; default: "") :

- PPTP and L2TP − the IP address which a client must connect from
- PPPoE − the MAC address (written in CAPITAL letters) which the client must connect from
- ISDN − the caller's number (that may or may not be provided by the operator) that the client may dial−in from
- if not set − there are no restrictions on from where clients may connect

password (*string*; default: "") − user password
profile (*name*; default: **default**) − profile name for the user
local−address (*IP address* | *name*; default: **0.0.0.0**) − either address or pool of the P2P server
remote−address (*IP address* | *name*; default: **0.0.0.0**) − either address or pool of the P2P client
routes − routes that appear on the server when the client is connected. The route format is: "dst−address gateway metric" (for example, "10.1.0.0/ 24 10.0.0.1 1"). Several routes may be specified separated with commas

Example

To add the user **ex** with **lkjrht** password for PPTP service only and with **ex** profile:

```
[admin@MikroTik] ppp secret> add name=ex password=lkjrht service=pptp profile=ex
[admin@MikroTik] ppp secret> print
Flags: X − disabled
  #   NAME              SERVICE CALLER−ID       PASSWORD        PROFILE
  0   ex                pptp                    lkjrht          ex
[admin@MikroTik] ppp secret> print detail
Flags: X − disabled
  0   name="ex" service=pptp caller−id="" password="lkjrht" profile=ex
      local−address=0.0.0.0 remote−address=0.0.0.0 routes=""


[admin@MikroTik] ppp secret>
```

Monitoring Active P2P Users

Command name : **/ppp active print**

Property Description

Statistics:

name (*name*) − user name
service (async | isdn | l2tp | pppoe | pptp) − what service the user is using
caller−id (*string*) − unique client identifier
address (*IP address*) − the IP address the client got from the server
uptime (*time*) − uptime
encoding (*string*) − encryption and encoding (if asymmetric, separated with '/') being used in this connection

Example

```
[admin@MikroTik] ppp profile> .. active print
Flags: R − radius
  #   NAME         SERVICE CALLER−ID         ADDRESS         UPTIME   ENCODING
  0   ex           pptp    10.0.0.148        10.1.0.148      1d15h... MPPE12...

[admin@MikroTik] ppp profile> .. active print detail
Flags: R − radius
  0   name="ex" service=pptp caller−id="10.0.0.148" address=10.1.0.148
      uptime=1d15h4m41s encoding="MPPE128 stateless"

[admin@MikroTik] ppp profile>
```

P2P User Remote AAA

Submenu level : **/ppp aaa**

```
[admin@MikroTik] ppp aaa> print
        use−radius: no
        accounting: yes
    interim−update: 0s
[admin@MikroTik] ppp aaa>
```

Property Description

use−radius (yes | no, default: **no**) − whether user database in a RADIUS server should be consulted
accounting (yes | no, default: **yes**) − whether RADIUS accounting is used
interim−update (*time*, default: **0s**) − Interim−Update time interval

Notes

RADIUS user database is consulted only if the required username is not found in local user database

Example

To enable RADIUS AAA:

```
[admin@MikroTik] ppp aaa> set use−radius=yes
[admin@MikroTik] ppp aaa> print
        use−radius: yes
        accounting: yes
```

```
    interim-update: 0s
[admin@MikroTik] ppp aaa>
```

Local IP Traffic Accounting

Local IP Traffic Accounting Setup

Submenu level : **/ip accounting**

```
[admin@MikroTik] ip accounting> print
      enabled: no
    threshold: 256
[admin@MikroTik] ip accounting>
```

Description

As each packet passes through the router, the packet source and destination address is matched to an IP pair in the accounting table and the traffic for that pair is increased. The source and destination users for PPP, PPTP, PPPoE, ISDN and HotSpot client traffic is accounted too. Both the number of packets and number of bytes are accounted.

If no matching IP or user pair exists, a new entry to the table will be created.

Note that for bidirectional connections two entries will be created.

Only packets that enter and leave the router are accounted. Packets that are dropped in the router are not counted. Packets that are sent from the router itself are not counted – such as packets used for administration connections (i.e. web and telnet connections to the router). Packets that are NATted on the router will be accounted for with the actual IP addresses on each side. Packets that are going through bridged interfaces (i.e. inside the bridge interface) are also accounted correctly.

Property Description

enabled (yes | no; default: **no**) − whether local IP traffic accounting is enabled
threshold (*integer*; default: **256**) − maximum number of IP pairs in the accounting table (maximal value is **8192**)

Notes

Each IP pair uses approximately 100 bytes

When the threshold limit is reached, no new IP pairs will be added to the accounting table. Each packet that is not accounted in the accounting table will then be added to the **uncounted** counter. To see if the limit on pairs has been reached, check the **uncounted** counter:

```
[admin@MikroTik] ip accounting uncounted> print
    packets: 0
      bytes: 0
```

Example

To enable traffic accounting:

```
[admin@MikroTik] ip accounting> set enabled=yes
[admin@MikroTik] ip accounting> print
      enabled: yes
    threshold: 256
[admin@MikroTik] ip accounting>
```

Local IP Traffic Acounting Table

Submenu level : **/ip accounting snapshot**

Description

When a snapshot is made for data collection, the accounting table is cleared and new IP pairs and traffic data are added. The more frequently traffic data is collected, the less likelihood that the IP pairs threshold limit will be reached.

Property Description

Statistics:

src−address (*IP address*) − source address
dst−address (*IP address*) − destination address
packets (*integer*) − total number of packets matched by this entry
bytes (*integer*) − total number of bytes matched by this entry
src−user (*string*) − sender's name (if aplicable)
dst−user (*string*) − recipient's name (if aplicable)

Notes

Usernames are shown only if the users are connected to the router via a P2P tunnel or ar authenticated by HotSpot.

Before the first snapshot is taken, the table is empty.

Example

To take a new snapshot:

```
[admin@MikroTik] ip accounting> snapshot take
```

To view the current snapshot:

```
[admin@MikroTik] ip accounting> snapshot print
  # SRC−ADDRESS     DST−ADDRESS     PACKETS    BYTES      SRC−USER   DST−USER
  0 10.5.8.8        10.0.0.4        194        15132
  1 10.0.0.4        10.5.8.8        194        15132
  2 10.0.0.144      10.5.8.23       4960       4097835
  3 10.5.8.23       10.0.0.144      4807       3843113
[admin@MikroTik] ip accounting> snapshot print
```

Web Access to the Local IP Traffic Accounting Table

Submenu level : **/ip accounting web−access**

```
[admin@MikroTik] ip accounting web−access> print
    accessible−via−web: no
               address: 0.0.0.0/0
[admin@MikroTik] ip accounting web−access>
```

Description

The web page report makes it possible to use the standard Unix/Linux tool wget to collect the traffic data and save it to a file or to use MikroTik shareware Traffic Counter to display the table. If the web report is enabled and the web page is viewed, the **snapshot** will be made when connection is initiated to the web page. The **snapshot** will then be displayed on the web page. TCP protocol used by http connections with the wget tool guarantees that none of the traffic data will be lost. The **snapshot** image will be made when the connection from wget is initiated. Web browsers or wget should connect to URL **http://*routerIP*/accounting/ip.cgi**

Property Description

accessible−via−web (yes | no; default: **no**) − whether the snapshot is available via web
address (*IP address/mask*; default: **0.0.0.0/0**) − IP address range that is allowed to access the sapshot

Example

To enable web access from 10.0.0.1 server only:

```
[admin@MikroTik] ip accounting web−access> set accessible−via−web=yes \
\... address=10.0.0.1/32
[admin@MikroTik] ip accounting web−access> print
    accessible−via−web: yes
               address: 10.0.0.1/32
[admin@MikroTik] ip accounting web−access>
```

RADIUS Client Setup

Submenu level : **/radius**

Description

This table sets the RADIUS servers the router is using to authenticate users.

Property Description

service (*multiple choice*:hotspot | login | ppp | telephony | wireless; default: "") − services that use this RADIUS server:

- **hotspot** − HotSpot authentication
- **login** − local user authentication
- **ppp** − P2P client authentication
- **telephony** − accounting for IP telephony

- **wireless** − wireless client authentication (client's MAC address is sent as **User−Name**)

called−id (*string*; default: "") − depending on P2P protocol:

- ISDN − phone number dialed (MSN)
- PPPoE − service name
- PPTP and L2TP − server IP address

domain (*string*; default: "") − Windows client's domain
address (*IP address*; default: **0.0.0.0**) − IP address of the RADIUS server
secret − shared secret to access the server
authentication−port (*integer*; default: **1812**) − server's port for authentication
accounting−port (*integer*; default: **1813**) − server's port for accounting
timeout (*time*; default: **100ms**) − timeout, after which the request should be resent

Notes

The order of the items is important.

Windows clients send their usernames in form: **domain\username**

Example

To set the RADIUS server HotSpot and PPP services will be using has **10.0.0.3** IP address and **ex** shared secret is:

```
[admin@MikroTik] radius> add service=hotspot,ppp address=10.0.0.3 secret=ex
[admin@MikroTik] radius> print
Flags: X − disabled
  #   SERVICE         CALLED−ID     DOMAIN        ADDRESS         SECRET
  0   ppp,hotspot                                 10.0.0.3        ex

[admin@MikroTik] radius>
```

AAA for the respective services should be enabled too:

```
[admin@MikroTik] radius> /ppp aaa set use−radius=yes
[admin@MikroTik] radius> /ip hotspot aaa set use−radius=yes
```

To view some statistics for a client:

```
[admin@MikroTik] radius> monitor 0
             pending: 0
            requests: 10
             accepts: 4
             rejects: 1
             resends: 15
            timeouts: 5
         bad−replies: 0
    last−request−rtt: 0s

[admin@MikroTik] radius>
```

RADIUS Servers Suggested

MikroTik RouterOS RADIUS CLIENT should work well with all RFC compliant servers. It has been

tested with:

FreeRADIUS : http://www.freeradius.org/
XTRadius : http://xtradius.sourceforge.net/ (do not support MS−CHAP)
Steel−Belted Radius : http://www.funk.com/

RADUIS Attributes Utilized

Here you can download MikroTik reference dictionary, which incorporates all the needed RADIUS attributes. This dictionary is the minimal dictionary, which is enough to support all features of MikroTik RouterOS. It is designed for FreeRADIUS, but may also be used with many other UNIX RADIUS servers (eg. XTRadius).

Note that it may conflict with the default configuration files of RADIUS server, which have references to the Attributes, absent in this dictionary. Please correct the configuration files, not the dictionary, as no other Attributes are supported by MikroTik RouterOS.

There is also dictionary.mikrotik that can be included in an existing dictionary to support MikroTik vendor−specific Attributes.

Authentication data sent to server (Access−Request)

```
Service−Type            always is Framed−User (only for P2P)

Framed−Protocol         always is PPP (only for P2P)

NAS−Identifier          router identity

NAS−IP−Address          router IP address

NAS−Port−Type           Async (for async PPP)
                        Virtual (for PPTP and L2TP)
                        Ethernet (for PPPoE and HotSpot)
                        ISDN Sync (for ISDN)

Calling−Station−Id      client MSN (for ISDN)
                        client public IP address (for PPTP and L2TP)
                        client MAC address (with CAPITAL letters) (for PPPoE)
                        client MAC address (with CAPITAL letters) (for HotSpot)

Called−Station−Id       service name (for PPPoE)
                        server IP address (for PPTP and L2TP)
                        interface MSN (for ISDN)
                        HotSpot server MAC address (for HotSpot)

NAS−Port                interface ID that may be used by SNMP client to retrieve
                        statistics information (only for P2P)
                        a unique session ID (for HotSpot)

NAS−Port−Id             serial port name (for async PPP)
                        ethernet interface name server is running on (for PPPoE
                        and HotSpot)

User−Name               client login name

MS−CHAP−Domain          authentication domain if username is in "domain\username"
```

	form (if Windows client set the "include domain name" parameter (only for P2P)

Depending on authentication methods (always CHAP for HotSpot):

User−Password	encrypted password (used with PAP auth.)
CHAP−Password, CHAP−Challenge	encrypted password and challenge (used with CHAP auth.)
MS−CHAP−Response, MS−CHAP−Challenge	encrypted password and challenge (used with MS−CHAPv1 auth.)
MS−CHAP2−Response, MS−CHAP2−Challenge	encrypted password and challenge (used with MS−CHAPv2 auth.)

Data received from server (Access−Accept)

Framed−IP−Address	IP address given to the client NOTE for P2P: If address belongs to networks 127.0.0.0/8, 224.0.0.0/4, 240.0.0.0/4, IP pool is used from the default profile to allocate client IP address NOTE for HotSpot: If address is 255.255.255.254, IP pool is used from hotspot settings. If Framed−IP−Address is specified, Framed−Pool is ignored
Framed−IP−Netmask	client netmask For P2P: If specified, the route will be created to the network Framed−IP−Address belongs to via the Framed−IP−Address gateway. For HotSpot: Framed−IP−Address netmask for DHCP−pool login method.
Framed−Pool	IP pool name (on the router) from which to get IP address for the client. If specified, overrides Framed−IP−Address
Idle−Timeout	idle−timeout parameter
Session−Timeout	session−timeout parameter
Class	cookie, will be included in Accounting−Request unchanged
Framed−Route	routes to add on the server. Format is specified in RFC2865 (Ch. 5.22), can be specified as many times as needed
Filter−Id	firewall filter chain name. It is used to make dynamic firewall rule that will jump to specified chain, if a packet is came to or from the client. Firewall chain name can have suffix .in or .out, that will install rule only for incoming or outgoing traffic. Multiple filter−id can be provided, but only last ones for incoming and outgoing is used
Acct−Interim−Interval	interim−update for RADIUS client, if 0 uses the one specified in RADIUS client

MS−MPPE−Encryption−Policy	require−encryption parameter (only for P2P)
MS−MPPE−Encryption−Type	use−encryption parameter. Non 0 value means use encryption (only for P2P)
Ascend−Data−Rate	tx/rx data rate limitation (for PPPoE and HotSpot). If multiple attributes are provided, first limits tx data rate, second − rx data rate. 0 if unlimited
Ascend−Xmit−Rate	tx data rate limitation (for PPPoE and HotSpot only). It may be used to specify tx limit only instead of sending two sequental Ascend−Data−Rate attributes. 0 if unlimited
Ascend−Client−Gateway	Client gateway for DHCP−pool HotSpot login method (only for HotSpot)
Mikrotik−Recv−Limit	total receive limit in bytes for the client (only for HotSpot)
Mikrotik−Xmit−Limit	total transmit limit in bytes for the client (only for HotSpot)
MS−CHAP2−Success	auth. response if MS−CHAPv2 was used (only for P2P)
MS−MPPE−Send−Key and MS−MPPE−Recv−Key	encryption keys for encrypted PPP, PPTP, L2TP and PPPoE, provided by RADIUS server only if MS−CHAP (both v1 and v2) was used for authentication (for PPP, PPTP, L2TP, PPPoE only)

Note that the received attributes override the default ones (set in the default profile), but if an attribute is not received from RADIUS server, the default one is to be used.

Accounting information sent to server (Accounting−Request)

Acct−Status−Type	Start, Stop, or Interim−Update
Acct−Session−Id	accounting session ID
Service−Type	same as in request (only for P2P)
Framed−Protocol	same as in request (only for P2P)
NAS−Identifier	same as in request
NAS−IP−Address	same as in request
User−Name	same as in request
MS−CHAP−Domain	same as in request (only for P2P)
NAS−Port−Type	same as in request
NAS−Port	same as in request (only for P2P)
NAS−Port−Id	same as in request
Calling−Station−Id	same as in request
Called−Station−Id	same as in request
Acct−Authentic	either authenticated by the RADIUS or Local authority (only for P2P)
Framed−IP−Address	IP address given to the user
Framed−IP−Netmask	same as in request (only for P2P)
Class	RADIUS server cookie
Acct−Delay−Time	how long does the router try to send this Accounting−Request packet

RADIUS attributes additionally included in Stop and Interim−Update Accounting−Request packets:

Acct−Session−Time	connection uptime in seconds

```
Acct-Input-Octects      bytes received from the client
Acct-Input-Packets      packets received from the client
Acct-Output-Octets      bytes sent to the client
Acct-Output-Packets     packets sent to the client
```

Stop Accounting-Request packets can additionally have:

```
Acct-Terminate-Cause    session termination cause (described in RFC2866 Ch. 5.10)
```

RADIUS Attribute Numeric Values

Acct-Authentic		45	RFC2866
Acct-Delay-Time		41	RFC2866
Acct-Input-Octects		42	RFC2866
Acct-Input-Packets		47	RFC2866
Acct-Interim-Interval		85	RFC2869
Acct-Output-Octets		43	RFC2866
Acct-Output-Packets		48	RFC2866
Acct-Session-Id		44	RFC2866
Acct-Session-Time		46	RFC2866
Acct-Status-Type		40	RFC2866
Acct-Terminate-Cause		49	RFC2866
Ascend-Client-Gatway	529	132	
Ascend-Data-Rate	529	197	
Ascend-Xmit-Rate	529	255	
Called-Station-Id		30	RFC2865
Calling-Station-Id		31	RFC2865
CHAP-Challenge		60	RFC2866
CHAP-Password		3	RFC2865
Class		25	RFC2865
Filter-Id		11	RFC2865
Framed-IP-Address		8	RFC2865
Framed-IP-Netmask		9	RFC2865
Framed-Pool		88	RFC2869
Framed-Protocol		7	RFC2865
Framed-Route		22	RFC2865
Idle-Timeout		28	RFC2865
MS-CHAP-Challenge	311	11	RFC2548
MS-CHAP-Domain	311	10	RFC2548
MS-CHAP-Response	311	1	RFC2548
MS-CHAP2-Response	311	25	RFC2548
MS-CHAP2-Success	311	26	RFC2548
MS-MPPE-Encryption-Policy	311	7	RFC2548
MS-MPPE-Encryption-Type	311	8	RFC2548

MS−MPPE−Recv−Key	311	17	RFC2548
MS−MPPE−Send−Key	311	16	RFC2548
Mikrotik−Recv−Limit	14988	1	
Mikrotik−Xmit−Limit	14988	2	
NAS−Identifier		32	RFC2865
NAS−IP−Address		4	RFC2865
NAS−Port		5	RFC2865
NAS−Port−Id		87	RFC2869
NAS−Port−Type		61	RFC2865
Service−Type		6	RFC2865
Session−Timeout		27	RFC2865
User−Name		1	RFC2865
User−Password		2	RFC2865

Certificate Management

Document revision 2.1 (09−Oct−2003)
This document applies to MikroTik RouterOS V2.7

Table of Contents

General Information

Summary

In MikroTik RouterOS certificates are used for SSL security protocol

Specifications

Packages required: ***security***
License required: *Any*
Submenu level: ***/certificate***
Hardware usage: *high CPU usage*

Certificates

Submenu level: ***/certificate***

Property Description

name (*name*) − reference name

subject (*read−only: text*) − subject of the certificate

issuer (*read−only: text*) − issuer of the certificate

serial−number (*read−only: text*) − serial number of the certificate

invalid−before (*read−only: date*) − date the certificate is valid from

invalid−after (*read−only: date*) − date the certificate is valid until

ca (yes | no; default: **yes**) − whether the certificate is used for building or verifying certificate chains (as Certificate Authority)

Command Description

import − install new certificates
- Input parameters:
  - **file−name** − import only this file (all files are searched for certificates by default)
  - **passphrase** − passphrase for the found encrypted private key
- Output parameters:
  - **certificates−imported** − how many new certificates were successfully imported
  - **private−keys−imported** − how many private keys for existing certificates were successfully imported
  - **files−imported** − how many files contained at least one item that was successfully imported
  - **decryption−failures** − how many files could not be decrypted
  - **keys−with−no−certificate** − how many public keys were successfully decrypted, but did not have matching certificate already installed

reset−certificate−cache − delete all cached decrypted public keys and rebuild the certificate cache

decrypt − decrypt and cache public keys
- Input parameters:
  - **passphrase** − passphrase for the found encrypted private key
- Output parameters:
  - **keys−decrypted** − how many keys were successfully decrypted and cached

Notes

Server certificates may have **ca** property set to **no**, but Certificate Authority certificates must have it set to **yes**

Certificates and encrypted private keys are imported from and exported to the router's FTP server. Public keys are not stored on a router in unencrypted form. Cached decrypted private keys are stored in encrypted form, using key that is derived from the router ID. Passphrases are not stored on router.

Configuration backup does not include cached decrypted private keys. After restoring backup all certificates with private keys must be decrypted again, using **decrypt** command with the correct passphrase.

Examples

To import a certificate and the respective private key already uploaded on the router:

```
[admin@MikroTik] certificate> import
passphrase: xxxx
       certificates-imported: 1
       private-keys-imported: 1
              files-imported: 2
         decryption-failures: 0
    keys-with-no-certificate: 1
[admin@MikroTik] certificate> print
Flags: K - decrypted-private-key, Q - private-key, R - rsa, D - dsa
  0 QR name="cert1" subject=C=LV,ST=.,O=.,CN=cert.test.mt.lv
       issuer=C=LV,ST=.,O=.,CN=third serial-number="01"
       invalid-before=sep/17/2003 11:56:19 invalid-after=sep/16/2004 11:56:19
       ca=yes

[admin@MikroTik] certificate> decrypt
passphrase: xxxx
    keys-decrypted: 1
[admin@MikroTik] certificate> print
Flags: K - decrypted-private-key, Q - private-key, R - rsa, D - dsa
  0 KR name="cert1" subject=C=LV,ST=.,O=.,CN=cert.test.mt.lv
       issuer=C=LV,ST=.,O=.,CN=third serial-number="01"
       invalid-before=sep/17/2003 11:56:19 invalid-after=sep/16/2004 11:56:19
       ca=yes

[admin@MikroTik] certificate>
```

Now the certificate may be used by HotSpot servlet:

```
[admin@MikroTik] ip service> print
Flags: X - disabled, I - invalid
  #   NAME                                PORT  ADDRESS            CERTIFICATE
  0   telnet                              23    0.0.0.0/0
  1   ftp                                 21    0.0.0.0/0
  2   www                                 8081  0.0.0.0/0
  3   hotspot                             80    0.0.0.0/0
  4   ssh                                 22    0.0.0.0/0
  5   hotspot-ssl                         443   0.0.0.0/0          none

[admin@MikroTik] ip service> set hotspot-ssl certificate=
cert1  none
[admin@MikroTik] ip service> set hotspot-ssl certificate=cert1
[admin@MikroTik] ip service> print
Flags: X - disabled, I - invalid
  #   NAME                                PORT  ADDRESS            CERTIFICATE
  0   telnet                              23    0.0.0.0/0
  1   ftp                                 21    0.0.0.0/0
  2   www                                 8081  0.0.0.0/0
  3   hotspot                             80    0.0.0.0/0
  4   ssh                                 22    0.0.0.0/0
  5   hotspot-ssl                         443   0.0.0.0/0          cert1

[admin@MikroTik] ip service>
```

Export and Import

Document revision 1.1 (31−Jan−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

Configuration export feature is used to dump the part or whole RouterOS configuation. Then it can be edited and imported to the same or to an another router.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : */*
Protocols utilized : *None*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Configuration Backup and Restore

Description

The configuration export can be used for dumping out MikroTik RouterOS configuration to the console screen or to a text (script) file, which can be downloaded from the router using ftp. The configuration import can be used to import the router configuration script from a text file.

The **export** command prints a script that can be used to restore configuration. The command can be invoked at any menu level, and it acts for that menu level and all menu levels below it. If the argument **from** is used, then it is possible to export only specified items. In this case **export** does not descend recursively through the command hierarchy. **export** also has the argument **file**, which allows you to save the script in a file on the router to retrieve it later via ftp.

The root level command **/import file_name** restores the exported information from the specified file. This is used to restore configuration or part of it after a **/system reset** event or anything that causes configuration data loss.

Note that it is impossible to import the whole router configuration using this feature. It can only be used to import a part of configuration (for example, firewall rules) in order to spare you some typing.

For backing up configuration to a binary file and restoring it without alterations, please refer to the configuration backup and restore section of the MikroTik RouterOS Manual.

The Export Command

Command name : **export**

Example

```
[admin@MikroTik] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.5.5.244/24      10.5.5.244      10.5.5.255      ether1
  1   10.5.5.245/32      10.5.5.245      10.5.5.245      ether1
  2   10.5.5.246/32      10.5.5.246      10.5.5.246      ether1
[admin@MikroTik] ip address>
```

To make an export file:

```
[admin@MikroTik] ip address> export file=address
[admin@MikroTik] ip address>
```

To make an export file from only one item:

```
[admin@MikroTik] ip address> export file=address1 from=1
[admin@MikroTik] ip address>
```

To see the files stored on the router:

```
[admin@MikroTik] > file print
  # NAME                          TYPE         SIZE       CREATION−TIME
  0 address1.rsc                  script       128        mar/26/2002 16:00:13
  1 address.rsc                   script       354        mar/26/2002 15:48:57
[admin@MikroTik] file>
```

To export the setting on the display use the same command but without the **file** argument:

```
[admin@MikroTik] ip address> export from=0,2
/ ip address
add address=10.5.5.244/24 network=10.5.5.244 broadcast=10.5.5.255 interface=ether1
comment="" disabled=no
add address=10.5.5.246/32 network=10.5.5.246 broadcast=10.5.5.246 interface=ether1
comment="" disabled=no
[admin@MikroTik] ip address>
```

The Import Command

Command name : **/import**

Example

To load the saved export file use the following command:

```
[admin@MikroTik] > import address1.rsc
[admin@MikroTik] >
```

Backup and Restore

Document revision 1.0 (31−Jan−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

The configuration backup can be used for backing up MikroTik RouterOS configuration to a binary file, which can be stored on the router or downloaded from it using ftp. The configuration restore can be used for restoring the router's configuration from a backup file. For exporting configuration or part of it to a text (script) file and importing it, please refer to the configuration export and import section of the MikroTik RouterOS Manual.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/system backup***
Protocols utilized : *None*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
Configuration Export and Import

Description

The **save** command is used to store the entire router configuration in a backup file. The file is shown in the **/file** submenu. You can download this it via ftp to keep it as a backup for your configuration.

To restore the system configuration, for example, after a **/system reset**, you can upload that file via ftp and then load that backup file, using **load** command in **/system backup** submenu.

Configuration Save Command

Command name : **/system backup save**

Example

To save the router configuration to file **test**:

```
[admin@MikroTik] system backup> save name=test
Configuration backup saved
[admin@MikroTik] system backup>
```

To see the files stored on the router:

```
[admin@MikroTik] > file print
  # NAME                          TYPE         SIZE       CREATION-TIME
  0 test.backup                   backup       12567      aug/12/2002 21:07:50
[admin@MikroTik] >
```

Configuration Load Command

Command name : **/system backup load**

Example

To load the saved backup file **test**:

```
[admin@MikroTik] system backup> load name=test
Restore and reboot? [y/N]:
```

FTP server

Document revision 1.2 (05−May−2003)
This document applies to the MikroTik RouterOS V2.7

Table Of Contents

Summary

MicroTik RouterOS implements File Transfer Protocol (FTP) server feature. It is intended to use for software packages uploading as well as configuration script exporting and importing procedures.

Specifications

Packages required : *None*
License required : *Any*
Home menu level : ***/file***
Standards and Technologies : *FTP (RFC 959)*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Configuration Export and Import
Configuration Backup and Restore HotSpot Gateway

File Transfer Protocol Server

Submenu level : **/file**

Description

MikroTik RouterOS has an industry standard FTP server feature. It uses ports 20 and 21 for communication with other hosts on the network. Do not disable these ports on your router!

Uploaded files as well as exported configuration or backup files can be accessed under **/file** menu. There you can also delete unwanted files from your router.

Authorization via ftp uses router's system user account names and passwords.

Property Description

name (*read−only*: *text*) − item name
type (*read−only*: file | directory | unknown) − item type
package size (*read−only*: *integer*) − package size
creation−time (*read−only*: *time*) − item creation date and time
package−name (*read−only*: *name*) − package name
package−version (*read−only*: *text*) − package version
package−build−time (*read−only*: *time*) − build time of the package

Example

Suppose we need to upload a software package **system−2.7rc4.npk** to a router with IP address **192.168.0.2** using generic text mode FTP client from a Linux workstation (the procedure and commands are the same also for Windows workstations):

```
user@wildcat:~/work$ ftp 192.168.0.2
Connected to 192.168.0.2.
220 MikroTik FTP server (MikroTik v2.7rc3) ready
Name (192.168.0.2:root): admin
331 Password required for admin
Password:
230 User admin logged in
Remote system type is UNIX.
ftp> binary
200 Type set to I
ftp> send
(local−file) /home/psi/system−2.7rc4.npk
(remote−file) system−2.7rc4.npk
local: /home/psi/system−2.7rc4.npk remote: system−2.7rc4.npk
200 PORT command successful
150 Opening BINARY mode data connection for '/system−2.7rc4.npk'
226 Transfer complete
8391343 bytes sent in 12.61 secs (649.7 kB/s)
ftp> close
221 Closing
ftp> quit
user@wildcat:~/work$
```

Now you can see this package on the router:

```
[admin@MikroTik] file> print
  # NAME                          TYPE         SIZE       CREATION−TIME
  0 system−2.7rc4.npk             package      8391343    apr/30/2003 17:09:55
  1 hotspot                       directory                mar/08/2003 16:14:19

[admin@MikroTik] file>
```

GPS

Document revision 1.1 (25−Jul−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

Global Positioning System (GPS) receiver can be used by MikroTik RouterOS to get the precise location and time (which may be used as NTP time source)

Specifications

Packages required : ***gps***
License required : *Any*
Home menu level : ***/system gps***
Standards and Technologies : *GPS, NMEA 0183, Simple Text Output Protocol*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
Network Time Protocol (NTP)

Description

Global Positioning System (GPS) is used for determining precise location of a GPS receiver. There are two types of GPS service:

- Precise Positioning Service (PPS) that is used only by U. S. and Allied military, certain U. S. Government agencies, and selected civil users specifically approved by the U. S. Government. Its accuracy is 22m horizontally, 27.7m vertically and 200ns of time

Standard Positioning Service (SPS) can be used by civil users worldwide without charge or restrictions except that SPS accuracy is intentionally degradated to 100m horizontally, 156m vertically and 340ns of time

GPS system is based on 24 satellites rotating on 6 different orbital planes with 12h orbital period. It makes that at least 5, but usually 6 or more satellites are visible at any time anywhere on the Earth. GPS receiver calculates more or less precise position (latitude, longitude and altitude) and time based on signals received from 4 satellites (three are used to determine position and fourth is used to correct time), which are broadcasting their current positions and UTC time.

MikroTik RouterOS can communicate with many GPS receivers which are able to send the positioning and time via asynchronous serial line using NMEA 0183, NMEA/RTCM or Simple Text Output Protocol.

Precise time is mainly intended to be used by built−in NTP server, which can use it as a time source without any additional configuration if GPS is configured to set system time.

Synchronizing with a GPS Receiver

Submenu level : **/system gps**

```
[admin@MikroTik] system gps> print
            enabled: no
               port: (unknown)
    set−system−time: yes
[admin@MikroTik] system gps>
```

Property Description

enabled (yes | no) − whether router will communicate with a GPS receiver
port (*name*) − the port will be used to communicate with a GPS receiver
set−system−time − whether to set the system time to what received from a GPS receiver

Notes

If you are synchronizing system time with a GPS device, you should correctly choose time zone if it is different from GMT as satellites are broadcasting GMT (a.k.a. UTC) time.

Example

To enable GPS communication through **serial0** port:

```
[admin@MikroTik] system gps> print
            enabled: no
               port: (unknown)
    set−system−time: yes
[admin@MikroTik] system gps> set enabled=yes port=serial0
[admin@MikroTik] system gps> print
            enabled: yes
               port: serial0
    set−system−time: yes
[admin@MikroTik] system gps>
```

• Monitoring GPS

Command name: **/system gps monitor**

Description

This command is used for monitoring the data received from a GPS receiver

Property Description

date−and−time (*read−only*: *text*) − data and time received from a GPS receiver
longitude (*read−only*: *text*) − longitude of the current location
latitude (*read−only*: *text*) − latitude of the current location
altitude (*read−only*: *text*) − altitude of the current location
speed (*read−only*: *text*) − mean velocity
valid (*read−only*: yes | no) − whether the showings are actually valid (e.g. you can set a GPS receiver to the demo mode to test the connection, in which case you will receive information, but it will not be valid)

Examples

```
[admin@MikroTik] system gps> monitor
    date−and−time: jul/23/2003 12:25:00
        longitude: "E 24 8' 17''"
         latitude: "N 56 59' 22''"
         altitude: "−127.406400m"
            speed: "0.001600 km/h"
            valid: yes

[admin@MikroTik] system gps>
```

Additional Resources

For additional information on how GPS works see:

Global Positioning System − How it Works

Liquid Crystal Display (LCD) Manual

Document revision 1.4 (11−Nov−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

LCDs are used to display system inormation.

The MikroTik RouterOS supports the following LCD hardware:

- Crystalfontz (www.crystalfontz.com) Intelligent Serial LCD Module 632 (16x2 characters) and 634 (20x4 characters)
- Powertip (www.powertip.com.tw) PC2404 (24x4 characters)

Specifications

Packages required : ***lcd***
License required : *Any*
Home menu level : ***/system lcd***
Protocols utilized : *None*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading

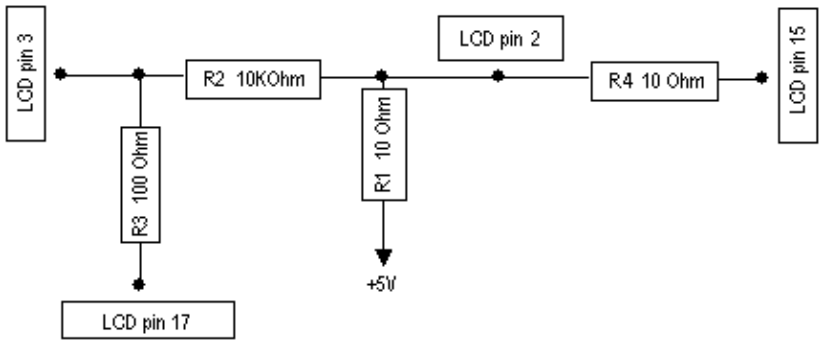
Description

How to Connect PowerTip LCD to a Parallel Port

Data signals are connected that way:

DB25m	Signal	LCD Panel
1	Enable (Strobe)	6
2	Data 0	7
3	Data 1	8
4	Data 2	9
5	Data 3	10
6	Data 4	11
7	Data 5	12
8	Data 6	13
9	Data 7	14
14	Register Select	4
18−25, GND	Ground	1,5,16

Powering:



As there are only 16 pins for the PC1602 modules, you need not connect power to the 17th pin

GND and +5V can be taken from computer's internal power supply (use Black wire for GND and Red wire for +5V).

WARNING! Be very careful connecting power supply. We do not recommend using external power supplies. In no event shall MikroTik be liable for any hardware damages.

Note that there are some PowerTip PC2404A modules that have different pin−out. Compare:

From www.powertip.com.tw (probably newer one)
From www.actron.de (probably older one)

Some LCDs may be connected without resistors:

DB25m	Signal	LCD Panel
18−25, GND	Ground	1,3,5,16
+5V	Power	2,15

Crystalfontz LCD installation notes

Before connecting the LCD, please check the availability of ports, their configuration, and free the desired port resource, if required:

```
[admin@MikroTik] port> print
  # NAME                            USED-BY                         BAUD-RATE
  0 serial0                         Serial Console                  9600
  1 serial1                                                         9600
[admin@MikroTik] port>
```

Configuring the LCD's Settings

Submenu level : **system lcd**
[admin@MikroTik] system lcd> print enabled: no type: powertip [admin@MikroTik] system lcd>

Property Description

enabled (yes | no, default: **no**) − turns the LCD on or off
type (powertip | crystalfontz; default: **powertip**) − sets the type of the LCD
serial−port (*name*) − name of the port where the LCD is connected (not shown when **type**=**powertip**)

Example

To enable Powertip parallel port LCD:

```
[admin@MikroTik] system lcd> print
        enabled: no
           type: powertip
[admin@MikroTik] system lcd> set enabled=yes
[admin@MikroTik] system lcd> print
        enabled: yes
           type: powertip
[admin@MikroTik] system lcd>
```

To enable Crystalfontz serial LCD on **serial1**:

```
[admin@MikroTik] system lcd> set type=crystalfontz
ERROR: can't acquire requested port − already used
[admin@MikroTik] system lcd> set type=crystalfontz serial-port=serial1
[admin@MikroTik] system lcd> /port print
  # NAME                            USED-BY                         BAUD-RATE
  0 serial0                         Serial Console                  9600
  1 serial1                         LCD Panel                       9600
[admin@MikroTik] system lcd> print
        enabled: yes
           type: crystalfontz
    serial-port: serial1
```

```
[admin@MikroTik] system lcd>
```

As You see, the first try to set LCD **type** failed because it wanted to use **serial0** (that is commonly used for **Serial Console**) by default.

LCD Information Display Configuration

Submenu level : **system lcd page**

Description

The submenu is used for configuring LCD information display: what pages and how long will be shown.

Property Description

display−time (*time*; default: **5s**) − how long to display the page
description (*string*) − page description

Notes

You cannot neither add your own pages (they are created dynamically depeding on the configuration) nor change pages' description.

Example

To enable displaying all the pages:

```
[admin@MikroTik] system lcd page> print
Flags: X − disabled
  #   DISPLAY−TIME         DESCRIPTION
  0 X 5s                   System date and time
  1 X 5s                   System resources − cpu and memory load
  2 X 5s                   System uptime
  3 X 5s                   Aggregate traffic in packets/sec
  4 X 5s                   Aggregate traffic in bits/sec
  5 X 5s                   Software version and build info
  6 X 5s                   ether1
  7 X 5s                   prism1
[admin@MikroTik] system lcd page> enable [find]
[admin@MikroTik] system lcd page> print
Flags: X − disabled
  #   DISPLAY−TIME         DESCRIPTION
  0   5s                   System date and time
  1   5s                   System resources − cpu and memory load
  2   5s                   System uptime
  3   5s                   Aggregate traffic in packets/sec
  4   5s                   Aggregate traffic in bits/sec
  5   5s                   Software version and build info
  6   5s                   ether1
  7   5s                   prism1
[admin@MikroTik] system lcd page>
```

To set "System date and time" page to be displayed for 10 seconds:

```
[admin@MikroTik] system lcd page> set 0 display-time=10s
[admin@MikroTik] system lcd page> print
Flags: X - disabled
  #   DISPLAY-TIME         DESCRIPTION
  0   10s                  System date and time
  1   5s                   System resources - cpu and memory load
  2   5s                   System uptime
  3   5s                   Aggregate traffic in packets/sec
  4   5s                   Aggregate traffic in bits/sec
  5   5s                   Software version and build info
  6   5s                   ether1
  7   5s                   prism1
[admin@MikroTik] system lcd page>
```

LCD Troubleshooting

1. LCD does not work, cannot be enabled by the **/system lcd set enabled yes** command.

   Probably the selected serial port is used by PPP client or server, or by the serial console.
   Check the availability and use of the ports by examining the output of the **/port print** command.
   Alternatively, select another port for connecting the LCD, or free up the desired port by disabling the related resource.
2. LCD does not work, does not show any information.

   Probably none of the information display items have been enabled.
   Use the **/system lcd page set** command to enable the display.

License Management

Document revision 1.3 (28−Apr−2003)
This document applies to the MikroTik RouterOS v2.7

Table of Contents

Summary

MikroTik RouterOS software has a licensing system with Software License (Software Key) issued for each individual installation of the RouterOS.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/system license***
Protocols utilized : *none*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Basic Setup Guide

Description

The Software License can be obtained through the Account Server at www.mikrotik.com after the MikroTik RouterOS has been installed. The Software ID of the installation is required when obtaining the Software License. Please read the MikroTik RouterOS Basic Setup Guide for detailed explanation of the installation and licensing process.

License Administration

Submenu level : **/system license**

Property Description

key (*string*) − software features unlock key
software−id (*read−only*) − unique identifier of this particular installation
upgradeable−until (*read−only*) − the date until which the software can be upgraded to a newer version

Example

The following example shows, how to change Software Key:

```
[admin@MikroTik] system license> set key=PSJ5−FG3−BCD
[admin@MikroTik] system license> /system reboot
Reboot, yes? [y/N]: y
```

After reboot you will see the new licensing information, for example:

```
[admin@MikroTik] system license> print
          software−id: M61X−UPT
                  key: PSJ5−FG3−BCD
    upgradeable−until: nov/11/2003
[admin@MikroTik] system license>
```

Features List

Submenu level : **/system license feature**

Property Description

AP − enables wireless Access Point feature
synchronous − enables synchronous interface and HotSpot feature as well
radiolan − enables 5.8GHz RadioLAN interface
wireless−2.4gHz − enables wireless client interface and HotSpot feature as well
licensed − basic license for RouterOS

Example

The following example shows how to see the software features that are enabled with the current license:

```
[admin@MikroTik] system license> feature print
Flags: X − disabled
  #   FEATURE
  0 X AP
  1   synchronous
  2 X radiolan
  3   wireless−2.4gHz
  4   licensed
[admin@MikroTik] system license>
```

Here we see, that the software has full license (not the demo version), and the 2.4GHz Wireless and Synchronous features are enabled.

Notes

To enable additional MikroTik RouterOS software features, or to enable upgrading (if it has expired), a new Software Key should be obtained from the Account Server at www.mikrotik.com. The new Software Key must be supplied to the router and the system must be rebooted:

Log Management

Document revision 1.1 (3−Feb−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

Various system events and status information can be logged. Logs can be saved in a file on the router or sent to a remote server running a syslog daemon. MikroTik provides a shareware Windows Syslog daemon, which can be downloaded from www.mikrotik.com.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/system logging***, ***/log***
Protocols utilized : *Syslog (Syslog)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading

Description

The logging feature sends all of your actions on the router to a log file or to a logging daemon. Router has several global configuration settings that are applied to logging. Logs have different facilities. Logs from each facility can be configured to be discarded, logged locally or remotely.

General Settings

Submenu level : **/syestem logging**

```
[admin@MikroTik] system logging> print
    default−remote−address: 0.0.0.0
       default−remote−port: 0
              buffer−lines: 100
[admin@MikroTik] system logging>
```

Property Description

default−remote−address (*IP address*; default: **0.0.0.0**) − remote log server IP address. Used when remote logging is enabled but no IP address of the remote server is specified
default−remote−port (*integer*; default: **0** − remote log server UDP port. Used when remote logging is enabled but no UDP port of the remote server is specified
buffer−lines (*integer*; default: **100**) − number of lines kept in local buffer. Contents of the local logs can be viewed using the **/log print** command. When number of lines in local log buffer is exceeded, lines from the beginning of buffer are deleted

Example

To use the **10.5.13.11** host, listening on **514** port, as the default remote system−log server:

```
[admin@MikroTik] system logging> print
    default−remote−address: 10.5.13.11
       default−remote−port: 514
              buffer−lines: 100
[admin@MikroTik] system logging>
```

Log Classification

Submenu level : **/system logging facility**

Property Description

facility (*name*) − name of the log group, message type
logging (none | local | remote; default: **local**) − type of logging:

- **local** − logs are stored in local log buffer. Local logs can be viewed using **/log print** command
- **none** − logs from this source are discarded
- **remote** − logs are sent to remote log server
  prefix (*string*; default: "") − local log prefix
  remote−address (*IP address*; default: "") − remote log server IP address. Used when logging type is remote. If not set, default log server IP address is used
  remote−port (*integer*; default: "") − Remote log server UDP port. Used when logging type is remote. If not set, default log server UDP port is used
  echo (yes | no; default: **no**) − whether to echo the message of this type to the active (logged−in) consoles

Notes

You cannot add, delete or rename the facilities: they are added and removed with the packages they are associated with.

System−Echo facility has its default **echo** property set to **yes**

Example

To force the router to send **Firewall−Log** to the **10.5.13.11** server, and **System−Info**, **System−Error** and **System−Warning** to the **10.5.13.12** server:

```
[admin@MikroTik] system logging facility> set Firewall−Log logging=remote remot
−address=10.5.13.11 remote−port=514
[admin@MikroTik] system logging facility> set System−Info,System−Error,System−W
rning logging=remote remote−address=10.5.13.12 remote−port=514
[admin@MikroTik] system logging facility> print
  # FACILITY          LOGGING PREFIX           REMOTE−ADDRESS  REMOTE−PORT ECHO
  0 Firewall−Log      remote                   10.5.13.11      514         no
  1 PPP−Account       local                                                no
  2 PPP−Info          local                                                no
  3 PPP−Error         local                                                no
  4 System−Info       remote                   10.5.13.12      514         no
  5 System−Error      remote                   10.5.13.12      514         no
  6 System−Warning    remote                   10.5.13.12      514         no
  7 Telephony−Info    local                                                no
  8 Telephony−Error   local                                                no
  9 Prism−Info        local                                                no
 10 Web−Proxy−Access  local                                                no
 11 ISDN−Info         local                                                no
 12 Hotspot−Account   local                                                no
 13 OSPF−Info         local                                                no
 14 Hotspot−Error     local                                                no
 15 IPsec−Event       local                                                no
 16 IKE−Event         local                                                no
 17 IPsec−Warning     local                                                no
 18 System−Echo       local                                                yes

[admin@MikroTik] system logging facility>
```

Log Messages

Submenu level : **/log**

Property Description

time (*string*) − data and time of the event
message (*string*) − message text

Notes

print command has new argument − **follow**, that cam be uset to monitor the log−message list, and see the new messages just as they arrive. Use [Ctrl]+[C] to exit from this mode

Example

To view the local logs:

```
[admin@MikroTik] log> print
 TIME                 MESSAGE
 jan/28/2003 21:28:34 user admin logged out via console
 jan/28/2003 21:28:34 user admin logged out via console
 jan/28/2003 21:54:35 user admin logged in via console
 jan/28/2003 22:09:24 ipsec peer removed by admin
 jan/28/2003 22:10:19 ipsec policy changed by admin
 jan/28/2003 23:18:02 pool our-dhcp-clients added by admin
 jan/29/2003 05:55:44 log configuration changed by admin
 jan/29/2003 06:13:33 log configuration changed by admin
 jan/29/2003 06:13:33 log configuration changed by admin
-- more
```

To monitor the system log:

```
[admin@MikroTik] log> print follow
 jan/28/2003 21:28:34 user admin logged out via console
 jan/28/2003 21:28:34 user admin logged out via console
 jan/28/2003 21:54:35 user admin logged in via console
 jan/28/2003 22:09:24 ipsec peer removed by admin
 jan/28/2003 22:10:19 ipsec policy changed by admin
 jan/28/2003 23:18:02 pool our-dhcp-clients added by admin
 jan/29/2003 05:55:44 log configuration changed by admin
 jan/29/2003 06:13:33 log configuration changed by admin
 jan/29/2003 06:13:33 log configuration changed by admin
-- Ctrl-C to quit. New entries will appear at bottom.
```

MAC Telnet Server and Client

Document revision 1.2 (07−May−2003)
This document applies to the MikroTik RouterOS V2.7

Contents of the Manual

Summary

MAC telnet is used to provide access to a router that have no IP address set. It works just like IP telnet. MAC telnet is possible between two MikroTik RouterOS routers only.

Specifications

Packages required : *None*
License required : *Any*
Home menu level : ***/tool, /tool mac−server***
Protocols utilized : *MAC Telnet*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Ping
MNDP (MikroTik Neighbor Discovery Protocol)

MAC Telnet Server

Submenu level : /**tool mac−server**

Property Description

interface (*name*|all) − interface name
- **all** − all interfaces

Notes

There is an interface list in configured in the submenu level. If you add some interfaces to this list, you allow MAC telnet to that interface. Disabled (**disabled=yes**) item means that interface in not in the list rather than that MAC telnet is disabled on that interface.

MAC Telnet server already has an entry allowing MAC telnets from **all** interfaces after MikroTik router installation:

```
[admin@MikroTik] tool mac-server> print
Flags: X - disabled
  #   INTERFACE
  0   all

[admin@MikroTik] tool mac-server>
```

Example

To enable MAC telnet server on **ether1** interface only:

```
[admin@MikroTik] tool mac-server> print
Flags: X - disabled
  #   INTERFACE
  0   all

[admin@MikroTik] tool mac-server> remove 0
[admin@MikroTik] tool mac-server> add interface=ether1 disabled=no
[admin@MikroTik] tool mac-server> print
Flags: X - disabled
  #   INTERFACE
  0   ether1

[admin@MikroTik] tool mac-server>
```

Monitoring Active Session List

Submenu level : **/tool mac-server sessions**

Property Description

interface (*read-only*: *name*) - interface the client is connected to
src-address (*read-only*: *MAC address*) - MAC address the client is connected from
uptime (*read-only*: *time*) - how long the client is connected to the server

MAC Telnet Client

Command name:**/system mac-telnet**

Example

```
[admin@MikroTik] tool> mac-telnet "00:40:63:C1:23:C4"
Login: admin
Password:
Trying 00:40:63:C1:23:C4...
Connected to 00:40:63:C1:23:C4

  MMM      MMM       KKK                          TTTTTTTTTTT      KKK
  MMMM    MMMM       KKK                          TTTTTTTTTTT      KKK
  MMM MMMM MMM  III  KKK  KKK  RRRRRR     OOOOOO      TTT     III  KKK  KKK
  MMM  MM  MMM  III  KKKKK     RRR  RRR  OOO  OOO     TTT     III  KKKKK
  MMM      MMM  III  KKK KKK   RRRRRR    OOO  OOO     TTT     III  KKK KKK
  MMM      MMM  III  KKK  KKK  RRR  RRR   OOOOOO      TTT     III  KKK  KKK

  MikroTik RouterOS v2.7 (c) 1999-2003       http://www.mikrotik.com/

Terminal linux detected, using multiline input mode
[admin@10.5.7.1] >
```

Network Time Protocol (NTP)

Document revision 1.3 (04−Sep−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

NTP protocol allows synchronizing time among computers in network. The best is if there is internet connection available and local NTP server is synchronized to correct time source. List of public NTP servers is available: http://www.eecis.udel.edu/~mills/ntp/servers.html

Specifications

Packages required : ***ntp***
License required : *Any*
Home menu level : ***/system ntp***
Protocols utilized : *NTP (RFC958)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Firewall Filters and Network Address Translation (NAT)

Description

Network Time Protocol (NTP) is used to synchronize time with some NTP servers in a network. MikroTik RouterOS privides both NTP client and NTP server.

NTP client synchronizes local clock with some other time source (NTP server). There are 4 modes in which NTP client can operate:

- In **unicast** (Client/Server) mode NTP client connects to specified NTP server. IP address of NTP server must be set in ntp−server and/or second−ntp−server parameters. At first client synchronizes to NTP server. Afterwards client periodically (64..1024s) sends time requests to NTP server. Unicast mode is the only one which uses ntp−server and second−ntp−server parameters.
- In **broadcast** mode NTP client listens for broadcast messages sent by NTP server. After receiving first broadcast message, client synchronizes local clock using unicast mode, and afterwards does not send any packets to that NTP server. It uses received broadcast messages to adjust local clock.
- **multicast** mode acts the same as broadcast mode, only instead of broadcast messages (IP address 255.255.255.255) multicast messages are received (IP address 224.0.1.1).
- **manycast** mode actually is unicast mode only with unknown IP address of NTP server. To discover NTP server, client sends multicast message (IP 239.192.1.1). If NTP server is configured to listen for these multicast messages (manycast mode is enabled), it replies. After client receives reply, it enters unicast mode and synchronizes to that NTP server. But in parallel client continues to look for more NTP servers by sending multicast messages periodically.

NTP Client

Submenu level : **/system ntp client**

```
[admin@MikroTik] > system ntp client print
          enabled: no
             mode: unicast
      primary-ntp: 0.0.0.0
    secondary-ntp: 0.0.0.0
           status: stopped
[admin@MikroTik] >
```

Property Description

enabled (yes | no, default: **no**) − whether the NTP client is enabled
mode (unicast | broadcast | multicast | manycast, default: **unicast**) − NTP client mode
primary−ntp (*IP address*, default: **0.0.0.0**) − specifies IP address of the primary NTP server
secondary−ntp (*IP address*, default: **0.0.0.0**) − specifies IP address of the secondary NTP server

Statistics:

status (*string*) − status of NTP client:
- **stopped** − NTP is not running (NTP is disabled)
- **error** − there was some internal error starting NTP service. (please, try to restart (disable and enable) NTP service)
- **started** − NTP client service is started, but NTP server is not found, yet
- **failed** − NTP server sent invalid response to our NTP client. (NTP server is not synchronous to some other time source)
- **reached** − NTP server contacted. Comparing local clock to NTP server's clock. (duration of this phase − approx 30 sec)
- **timeset** − local time changed to NTP server's time. (duration of this phase − approx 30 sec)
- **synchronized** − local clock is synchronized to NTP server's clock. NTP server is activated.
- **using−local−clock** − using local clock as time source (server enabled while client disabled)

Example

To enable the NTP client to synchronize with the **159.148.60.2** server:

```
           status: reached
[admin@MikroTik] system ntp client> print
          enabled: yes
             mode: unicast
      primary-ntp: 159.148.60.2
    secondary-ntp: 0.0.0.0
           status: synchronized
[admin@MikroTik] system ntp client>
```

NTP Server

Submenu level : **/system ntp server**

```
[admin@MikroTik] > system ntp server print
      enabled: no
    broadcast: no
    multicast: no
     manycast: yes
[admin@MikroTik] >
```

Property Description

enabled (yes | no, default: **no**) − whether the NTP client is enabled
broadcast (yes | no, default: **no**) − whether NTP broadcast message is sent to 255.255.255.255 every 64s
multicast (yes | no, default: **no**) − whether NTP multicast message is sent to 224.0.1.1 every 64s
manycast (yes | no, default: **yes**) − whether NTP server listens for multicast messages sent to 239.192.1.1 and responds to them

Notes

NTP server activates only when local NTP client is in **synchronized** or **using−local−clock** mode.

If NTP server is disabled, all NTP requests are ignored.

If NTP server is enabled, all individual time requests are answered.

CAUTION! Using **broadcast**, **multicast** and **manycast** modes is dangerous! Intruder (or simple user) can set up his own NTP server. If this new server will be chosen as time source for Your server, it will be possible for this user to change time on Your server at his will.

Example

To enable NTP server to answer unicast requests only:

```
[admin@MikroTik] system ntp client> .. server print
      enabled: yes
    broadcast: no
    multicast: no
     manycast: no
```

```
[admin@MikroTik] system ntp client>
```

Time Zone

Submenu level : **/system clock**

```
[admin@MikroTik] > system clock print
         time: aug/12/2002 18:31:20
    time-zone: +00:00
[admin@MikroTik] >
```

Notes NTP changes local clock to UTC (GMT) time by default

Example

Time zone is specified as a difference between local time and GMT time. For example, if GMT time is 18:00:00, but correct local time is 19:00:00, then time-zone has to be set to +1 hour:

```
[admin@MikroTik] > system clock set time-zone=1
[admin@MikroTik] > system clock print
         time: aug/12/2002 19:31:57
    time-zone: +01:00
[admin@MikroTik] >
```

If local time is before GMT time, time-zone value will be negative. For example, if GMT is 18:00:00, but correct local time is 15:00:00, time-zone has to be set to -3 hours:

```
[admin@MikroTik] > system clock set time-zone=-3
[admin@MikroTik] > system clock print
         time: aug/12/2002 15:32:20
    time-zone: -03:00
[admin@MikroTik] >
```

Scripting Manual

Document revision 1.7 (15−May−2003)
This document applies to the MikroTik RouterOS V2.7

Table Of Contents

Summary

Scripting gives a way to automate some router maintenance tasks by writing scripts to be executed if some event occurs. To write a script, the administrator must learn console commands described in the relevant documentation. Scripts may be written for the System Scheduler (see relevant manual), the Traffic Monitoring Tool (see relevant manual), and for the Netwatch Tool.

Specifications

Packages required : *None*
License required : *Any*
Home menu level : ***/system script***
Protocols utilized : *None*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading

Description

Although 2.7 console syntax has many changes from previous versions, most users will not notice any differences. However, if you are using scripting capabilities of RouterOS, it is recommended to read this section, even if you have some experience with previous versions.

This is more an introductory text, less a reference. It freely uses commands and concepts before explaining them, to make it as short, simple and comprehensive as possible. It might be necessary to read it several times. Many examples are given, because it is the best way to explain most things.

Command Syntax

Description

Console commands in RouterOS 2.7 are made from the following parts:

```
PREFIX PATH PATH_ARGUMENT COMMAND NAMELESS_ARGUMENTS ARGUMENTS
```

Property Description

PREFIX (: | /) − optional
PATH (multiple choice: *text*, ..) − a sequence of command level names. It is also optional, but the processing of commands without given path may change in future versions, so PATH is highly recommended. ".." means parent level path
PATH_ARGUMENT is required by some command levels (like **/ip firewall rule**), and is not allowed anywhere else
COMMAND (*text*) − command name from the command level specified by path
NAMELESS_ARGUMENTS (*text*) − arguments, specific to each command. Values of these arguments are written in fixed order after the name of a command, and only after all nameless argument values any named arguments can be given
ARGUMENTS (*text*) − sequence of argument names (like **/user print brief without−paging**). For arguments that need values, argument name is followed by a **=**, followed by a value of an argument

Notes

Variable substitution, command substitution and expressions are allowed only for PATH_ARGUMENT and command argument values. PREFIX, PATH, command name and argument names can only be given directly, as a word. So

```
:put (1 + 2)
```

is valid and

```
(":pu" . "t") 3
```

is not.

Example

```
The console commands' parts can be seen in the following examples:
```

/ping 10.0.0.13 count=5 PREFIX − "/" COMMAND − "ping" NAMELESS_ARGUMENTS − "10.0.0.13" ARGUMENTS − "count=5"

```
... ip firewall rule input
PATH − ".. ip firewall rule"
PATH_ARGUMENT − "input"

:for i from=1 to=10 do={:put $i}

PREFIX − ":"
COMMAND − "for"
NAMELESS_ARGUMENTS − "i"
ARGUMENTS − "from=1 to=10 do={:put $i}"

/interface monitor−traffic ether1,ether2,ipip1

PREFIX − "/"
PATH − "interface"
COMMAND − "monitor−traffic"
NAMELESS_ARGUMENTS − "ether1,ether2,ipip1"
```

Grouping

Description

It is possible to execute several commands from the same command level, by grouping them with braces '{}'.

Notes

You should not change current command level in scripts by typing just it's path, without any command, like you when working with console interactively. Such changes have no effect in scripts. Consider:

```
[admin@MikroTik] ip address> /user {
{... /ip route
{... print
{... }
Flags: X − disabled
  0   ;;; system default user
      name="admin" group=full address=0.0.0.0/0

  1   name="x" group=write address=0.0.0.0/0

  2   name="y" group=read address=0.0.0.0/0


[admin@MikroTik] ip route>
```

Although the current command level is changed to **/ip route**, it has effect only on next command entered from prompt, **print** command is still considered to be **/user print**.

Example

We will add two users to the **user** menu in the example below:

```
[admin@MikroTik] ip address> /user {
{... add name=x password=y group=write
{... add name=y password=z group=read
{... print
{... }
Flags: X − disabled
  0   ;;; system default user
      name="admin" group=full address=0.0.0.0/0

  1   name="x" group=write address=0.0.0.0/0

  2   name="y" group=read address=0.0.0.0/0


[admin@MikroTik] ip address>
```

Variables

Description

Console allows you to create and use global (system wide) and local (only usable within one script) variables. Variables can be accessed by writing '$' followed by name of variable. Variable names can contain letters, digits and '−' character. A variable must be declared prior to using it in scripts. There are three types of declaration available:

- **global**

Global variables can be accessed by all scripts and console logins on the same router. There is no way currently to remove global variable, except rebooting router. Variables are not kept across reboots.

- **local**

Local variables are not shared with any other script, other instance of the same script or other console logins. Its value is lost when script finishes or when variable name is freed by **:unset**.

- loop index variables

These are used only in **do=** block of commands and are removed after command completes.

- **monitor** commands, that have **do=** argument

See details below.

You can assign new value to variable using **:set** command. It has two unnamed arguments: the name of the variable and the new value of the variable.

Notes

Loop variables "shadows" already introduced local variables with the same name.

Introducing variable has no effect on other scripts that may be running. It just tells the current script what variable names can be used, and where to get their values. After variable is no longer needed, it's name can be freed by **:unset** command. If you free local variable, it's value is lost. If you free global variable, it's value is still kept in router, it just becomes inaccessible from current script.

Example

```
[admin@MikroTik] ip route> :put $a
ERROR: unknown variable a
[admin@MikroTik] ip route>
```

You must first declare a variable.
Like this:

```
[admin@MikroTik] ip route> /
[admin@MikroTik] > :global g1
[admin@MikroTik] > :set g1 "this is global variable"
[admin@MikroTik] > :put $g1
this is global variable
[admin@MikroTik] >
```

or like this:

```
[admin@MikroTik] > :local l1
[admin@MikroTik] > :set l1 "this is local variable"
[admin@MikroTik] > :put $l1
this is local variable
[admin@MikroTik] >
```

or, finally, like this:

```
[admin@MikroTik] > :for l1 from=1 to=3 do={:put $l1}
1
2
3
[admin@MikroTik] > :put $l1
this is local variable
[admin@MikroTik] >
```

The following example will create a local variable with start value 0 and then will increase it by 1:

```
[admin@MikroTik] > :local counter
[admin@MikroTik] > :set counter 0
[admin@MikroTik] > :put $counter
0
[admin@MikroTik] > :set counter ($counter + 1)
[admin@MikroTik] > :put $counter
1
[admin@MikroTik] >
```

Because increasing or decreasing variable's value by one is such a common case, there are two commands that do just that. **:incr** increases value of variable by 1, and **:decr** decreases it by 1.

```
[admin@MikroTik] > :incr counter
[admin@MikroTik] > :put $counter
2
[admin@MikroTik] >
```

Command substitution, return values

Description

Some console commands are most useful if their output can be used as an argument value in other commands. In console, this is done by "returning" value from commands. Return value is not displayed on the screen. When you type such a command between square brackets '[' ']', this command is executed and it's return value is used as the value of these brackets. This is called command substitution.

Example

Consider **find** command.

```
[admin@MikroTik] > /interface
[admin@MikroTik] interface> find type=ether
[admin@MikroTik] interface>
```

It displays nothing on screen, and returns internal numbers of items with matching property values. This is how return value looks:

```
[admin@MikroTik] interface> :put [find type=ether]
*A,*B
[admin@MikroTik] interface>
```

and this is how it can be used in other commands

```
[admin@MikroTik] interface> enable [find type=ether]
[admin@MikroTik] interface>
```

Besides **find**, some other commands also return useful values. **/ping** returns number of successful pings:

```
[admin@MikroTik] interface> :put [/ping 10.0.0.1 count=3]
10.0.0.1 64 byte pong: ttl=64 time<1 ms
10.0.0.1 64 byte pong: ttl=64 time<1 ms
10.0.0.1 64 byte pong: ttl=64 time<1 ms
3 packets transmitted, 3 packets received, 0 packet loss
round-trip min/avg/max = 0/0.0/0 ms
3
[admin@MikroTik] interface>
```

:set returns value of it's second argument. **:time** returns the measured time value. **:incr** and **:decr** return new value of variable. Another important case is **add** command, which return internal number of newly created item.

```
[admin@MikroTik] interface> /user
[admin@MikroTik] user> :put [add name=z password=x group=full]
```

```
*7
[admin@MikroTik] user>
```

This way you can store it in variable for later use.

Operators

Description

Console can do simple calculations with numbers, time values, ip addresses, strings and lists. It is achieved by writing expressions and putting them in parentheses '(' and ')'.

Supported operations are:
- **!** − logical NOT

Unary operator. Argument is a boolean value. Result is an opposite boolean value.
- **−** − unary minus

Unary operator. Argument and result is a number.
- **~** − bit inversion

Unary operator. Inverts bits in IP address.
- **+** − sum

Binary operator. Adds two numbers, two time values, or add number to an IP address.
- **−** − subtraction

Binary operator. Subtracts two numbers one from another, two time values, two IP addresses or an IP address and a number.
- ***** − multiplication

Binary operator. Multiplies two numbers, or multiply a time value by a number.
- **/** − division

Binary operator. Divides one number by another (gives an integer), or a time value by a number (gives time value).
- comparison operators

< − less
> − more
<= − less or equal
>= − more or equal

Binary operators. Compare two numbers, two time values, or two IP addresses. Give boolean value.
!= − not equal
= − equal
Binary operators. Compare two values of the same type. Arrays are equal if their respective elements are equal.

- **− logical AND**

- || − logical OR

Binary operators. Logical operation on two boolean values. Result of **is true, if both operands are true. Result of || is true if either operand is true.**

- **bitwise operators**

− bitwise and
| − bitwise or
^ − bitwise xor
Binary operators. Bitwise operations on two IP addresses. Result is also an IP address.
<< − shift left
>> − shift right
Binary operators. Shift IP value left or right by given amount of bits. First argument is an IP address and second argument is an integer. Result is an IP address.

- **. − concatenation**

Binary operator. Concatenates two strings, or appends one list to another, or appends an element to a list.

Example

```
[admin@MikroTik] user> :put (1 + 2)
3
[admin@MikroTik] user> /interface
[admin@MikroTik] interface> :put ([find type=ipip ] . [find type=ether ])
*6,*A,*B
[admin@MikroTik] interface>
```

logical NOT

```
[admin@MikroTik] interface> :put (!true)
false
[admin@MikroTik] interface> :put (!(2>3))
true
[admin@MikroTik] interface>
```

unary minus

```
[admin@MikroTik] interface> :put (−1 :put (−−1)
1
```

bit inversion

```
[admin@MikroTik] interface> :put (~255.255.0.0)
0.0.255.255
[admin@MikroTik] interface>
```

sum

```
[admin@MikroTik] interface> :put (3s + 5s)
8s
[admin@MikroTik] interface> :put (10.0.0.15 + 0.0.10.0)
ERROR: cannot add ip address to ip address
[admin@MikroTik] interface> :put (10.0.0.15 + 10)
10.0.0.25
[admin@MikroTik] interface>
```

subtraction

```
[admin@MikroTik] interface> :put (15 − 10)
5
[admin@MikroTik] interface> :put (10.0.0.15 − 10.0.0.3)
12
[admin@MikroTik] interface> :put (10.0.0.15 − 12)
10.0.0.3
[admin@MikroTik] interface> :put (15h − 2s)
14h59m58s
[admin@MikroTik] interface>
```

multiplication

```
[admin@MikroTik] interface> :put (12s * 4)
48s
[admin@MikroTik] interface> :put (−5 * −2)
10
[admin@MikroTik] interface>
```

division

```
[admin@MikroTik] interface> :put (10s / 3)
3s333.333ms
[admin@MikroTik] interface> :put (5 / 2)
2
[admin@MikroTik] interface>
```

comparison

```
[admin@MikroTik] interface> :put (10.0.2.3<=2.0.3.10)
false
[admin@MikroTik] interface> :put (100000s>27h)
true
[admin@MikroTik] interface> :put (60s,1d!=1m,3600s)
false
[admin@MikroTik] interface> :put (bridge=routing)
false
[admin@MikroTik] interface> :put (yes=false)
false
[admin@MikroTik] interface> :put (true=aye)
ERROR: cannot compare if truth value is equal to string
[admin@MikroTik] interface>
```

logical AND, logical OR

```
[admin@MikroTik] interface> :put ((yes &yes) || (yes &no))
true
[admin@MikroTik] interface> :put ((no || no) &(no || yes))
false
[admin@MikroTik] interface>
```

bitwise AND, bitwise OR, bitwise XOR

```
[admin@MikroTik] interface> :put (10.16.0.134 &~255.255.255.0)
0.0.0.134
[admin@MikroTik] interface>
```

shift operators

```
[admin@MikroTik] interface> :put (~((0.0.0.1 <<7) - 1))
255.255.255.128
[admin@MikroTik] interface>
```

concatenation

```
[admin@MikroTik] interface> :put (1 . 3)
13
[admin@MikroTik] interface> :put (1,2 . 3)
1,2,3
[admin@MikroTik] interface> :put (1 . 3,4)
13,4
[admin@MikroTik] interface> :put (1,2 . 3,4)
1,2,3,4
[admin@MikroTik] interface> :put ((1 . 3) + 1)
ERROR: cannot add string to integer number
[admin@MikroTik] interface>
```

Value types

Description

Console can work with several types of values. Currently it distinguishes between strings, boolean values, numbers, time intervals, IP addresses, internal numbers and lists. Currently console tries to convert any value to the most specific type first, backing up if it fails. This is the order in which console attempts to convert a value:

- list
- internal number
- number
- IP address
- time value
- boolean value
- string value

There is no way to explicitly control this type conversion, but it most likely will be changed in future versions. Meanwhile, this can help to explain why console sometimes "corrupts" values, that are meant to be strings, but look like one of the above types:

```
[admin@MikroTik] interface> :put 1s1d90039
2d1h40s
[admin@MikroTik] interface>
```

In console integers are internally represented as 64 bit signed numbers, so the range of variable values can be from −9223372036854775808 to 9223372036854775807. It is possible to input them as hexadecimal numbers, by prefixing with "0x":

```
[admin@MikroTik] interface> :put 0x123ABCDEF4567890
1313569907099990160
[admin@MikroTik] interface> /
[admin@MikroTik] >
```

Lists are written as comma separated sequence of values. Putting whitespaces around commas is not recommended, because it might confuse console about word boundaries.

```
[admin@MikroTik] > :foreach i in 1,2,3 do {:put $i}
1
2
3
[admin@MikroTik] > :foreach i in 1, 2, 3 do {:put $i}
ERROR: no such argument (2,)
[admin@MikroTik] >
```

Boolean values are written as either **true** or **false**. Console also accepts **yes** for **true**, and **no** for **false**.

Internal numbers begin with '*'.

Time intervals are written as sequence of numbers, that can be followed by letters specifying the units of time measure. The default is a second. Numbers may have decimal point. It is also possible to use the HH:MM:SS notation. Here are some examples:

```
[admin@MikroTik] > :put "1000s"
16m40s
[admin@MikroTik] > :put "1day 1day 1day"
3d
[admin@MikroTik] > :put "1day day 1day"
1day day 1day
[admin@MikroTik] > :put "1.5hours"
1h30m
[admin@MikroTik] > :put "1:15"
1h15m
[admin@MikroTik] > :put "0:3:2.05"
3m2s50ms
[admin@MikroTik] >
```

Accepted time units:

d, **day**, **days** − unit is 24 hours
h, **hour**, **hours** − unit is 1 hour
m − unit is 1 minute
s − unit is 1 second
ms − unit is 1 millisecond (0.001 second)

Common Commands

Description

Console has many built−in commands that start with ':' prefix. Although they don't change configuration directly, they are useful for writing scripts. You can see entire list of such commands by pressing '?' after typing the ':' prefix:

```
[admin@MikroTik] > :

        local  introduces local variable
       global  introduces global variable
        unset  forgets variable
          set  creates or changes variable value
          put  prints argument on the screen
        while  executes command while condition is true
           if  executes command if condition is true
           do  executes command
         time  times command
         incr  increments variable
         decr  decrements variable
          for  executes command for a range of integer values
      foreach  executes command for every element in a list
        delay  does nothing for a while (default 1 second)
  environment  information about variables
          log  add entry in the system logs
[admin@MikroTik] > :
```

:local, **:global**, **:unset**, **:set**, **:incr** and **:decr** commands are explained in the section about variables. All other commands will be explained in this section.

- **:put** − takes only one, unnamed argument. It is displayed on the screen. Cannot be used in scripts, because scripts don't have a place to display values on.
- **:if** − this is a conditional, or branching command. It has one unnamed argument which must be a condition, *id est* an expression that must return boolean value. If computing condition returns **true**, commands that are given as value for **do** argument are executed, otherwise **else** commands are. **else** argument is optional.

```
[admin@MikroTik] > :if (yes) do={:put yes} else={:put no}
true
[admin@MikroTik] > :if ([/ping 10.0.0.1 count=1] = 0) do {:put "gw unreachable"}
10.0.0.1 pong timeout
1 packets transmitted, 0 packets received, 100% packet loss
gw unreachable
[admin@MikroTik] >
```

There are four loop control commands in console. They all have **do** statement, which holds console commands that have to be executed repeatedly.

- **:while** − this command has one unnamed argument, a condition. It is evaluated every time before executing **do** statement. If result is not a boolean value, error is reported. If the result of condition is **true**, commands are executed once, and the condition is evaluated again, and this repeats until **false**.
- **:do** − It has one unnamed argument, which holds the console commands that must be executed. It is

similar to the **do** statement of other commands. If no other arguments are given, **:do** just executes these commands once. There is not much use in that. If you specify a condition as a value for **while** argument, it is evaluated after executing commands, and if it returns **true**, **do** statement is executed again, and this is repeated until the condition returns **false**. If you specify a condition for **if** argument, it is computed only once, before doing anything else, and if it is **false**, nothing is done. If it is **true**, everything is executed as usual. Note that **:do A while=B** is different from **:while B do=A**, because **:do** evaluates condition after executing command, not before, like **:while** does. However, **:do A if=B** and **:if B do=A** do exactly the same thing.

- **:for** − It has one unnamed argument, the name of the loop variable. **from** argument is the starting value for the loop counter, **to** value is the final value. This command counts loop variable up or down starting at **from** and ending with **to**, inclusive, and for each value it executes the **do** statement. It is possible to change the increment from the default 1 (or −1), by specifying the **step** argument.

```
[admin@MikroTik] > :for i from=1 to=100 step=37  do={:put ($i . " − " . 1000/$i)}
1 − 1000
38 − 26
75 − 13
[admin@MikroTik] >
```

- **:foreach** − The unnamed argument is the name of the loop variable. **in** argument is treated as a list. Each value in sequence is assigned to the loop variable, and **do** statement is executed for this value. If **in** value is not a list, **do** statement is executed only once. If **in** value is empty, **do** statement isn't executed at all. This is made to work good with **find** command, which return lists of internal numbers, and may return empty list or just one internal number. This example prints all ethernet interfaces, each followed by all addresses that are assigned to it:

```
[admin@MikroTik] > :foreach i in=[/interface find type=ether ] do={
{... :put [/interface get $i name]
{... :foreach j in=[/ip address find interface=$i] do={
{{... :put [/ip address get $j address]
{{... }
{... }
ether1
ether2
10.0.0.65/24
[admin@MikroTik] >
```

- **:delay** − This command does nothing for a given amount of time. The unnamed argument should be a time interval value. It is optional, and if **:delay** is executed without any arguments, it does nothing for one second.
- **:time** − This command takes one unnamed argument containing console commands. Commands are executed, and the time it took to execute them is printed, and returned.

```
[admin@MikroTik] > :time {:delay 1756ms}
1.755333s
[admin@MikroTik] > :put [:time {:delay}]
1.007464s
1s7.464ms
[admin@MikroTik] >
```

- **:log** − This command adds an entry to the system logs. **message** argument is the text of log entry. **facility** argument tells at which logging facility (see **/system logging facility**) this message should be logged, the default is **System−Info**.

```
[admin@MikroTik] > :log facility=System−Warning message="Very Bad Thing happened"
[admin@MikroTik] >
```

- **:environment print** − This command prints information about variables. All global variables in the system are listed under heading **Global Variables**. All variable names that are introduced in this script (local variables introduced by **:local** or created by **:for** or **:foreach**, global variables introduced by **:global**, in short, all variables that can be used from the current script) are listed under heading **Local Variables**.

```
[admin@MikroTik] > :environment print
Global Variables
g1=this is global variable
Local Variables
g1=this is global variable
l1=this is local variable
counter=2
[admin@MikroTik] >
```

This can be useful in debugging scripts, or just for figuring out how variables work in console. Suppose we don't want to use variable "g1" anymore:

```
[admin@MikroTik] > :unset g1
[admin@MikroTik] > :environment print
Global Variables
g1=this is global variable
Local Variables
l1=this is local variable
counter=2
[admin@MikroTik] > :put $g1
ERROR: unknown variable g1
[admin@MikroTik] >
```

Here, although such global variable still exists (and we can get it back with **:global g1** command), it is **unknown** because we have told current script to forget about it.

```
[admin@MikroTik] > :global g1
[admin@MikroTik] > :put $g1
this is global variable
[admin@MikroTik] >
```

Special Commands

Monitor

It is possible to access values that are shown by most monitor commands from scripts. If monitor command has **do** argument, it can be supplied either script name (see **/system scripts**), or console commands.

Get

It is also possible to access from scripts values that are shown by most print commands. Most command levels that have **print** command, also have **get** command. It has one or two unnamed arguments. If this command level deals with a list of items, first argument is a name or internal number of an item. Second argument is a name of item's property which should be returned.

Notes

Monitor command with **do** argument can also be called directly from scripts. It will not print anything then, just execute the given script.

Names of properties that can be accessed by **get** are the same as shown by **print** command, plus names of item flags (like the **disabled** in the example below). You can use tab key completions to see what properties any particular **get** command can return.

Monitor Example

In the example below monitor command will execute given script each time it prints stats on the screen, and it will assign all printed values to local variables with the same name:

```
[admin@MikroTik] > /interface
[admin@MikroTik] interface> monitor-traffic ether2 once do={:environment print}
    received-packets-per-second: 2
       received-bits-per-second: 960.00bps
        sent-packets-per-second: 0
           sent-bits-per-second: 0.00bps

Global Variables
Local Variables
sent-bits-per-second=0
received-packets-per-second=2
received-bits-per-second=960
sent-packets-per-second=0
[admin@kzd] interface>
```

Get Example

The example below will get ether1 status from the interfaces list. If ether1 is disabled, it will return the value **true**:

```
[admin@MikroTik] interface> :put [/interface get ether1  disabled ]
true
[admin@MikroTik] interface>
```

If command level has general settings, **get** command only takes the name of property:

```
[admin@MikroTik] interface> :put [/system clock get time ]
feb/28/2003 12:44:39
[admin@MikroTik] interface>
```

Additional Features

It is possible to include comments in console scripts. If script line starts with '#', all characters until newline are ignored.

It is possible to put multiple commands on a single line, separating them by ';'. Console treats ';' as end of line when separating script text into commands.

If you want to use any of {}[]"'\$ characters in a string, you have to prefix them with '\' character. Console takes any character following '\' literally, without assigning any special meaning to it, except for such cases:

```
\a      bell (alarm), character code 7
\b      backspace, character code 8
\f      form feed, character code 12
\n      newline, character code 10
\r      carriage return, character code 13
\t      tabulation, character code 9
\v      vertical tabulation, character code 11
\_      space, character code 32
```

Also, '\' followed by any amount of whitespace characters (spaces, newlines, carriage returns, tabulations), followed by newline is treated as a single whitespace, except inside quotes, where it is treated as nothing. This is used by console to break up long lines in scripts generated by export commands.

Scripts

Submenu level : **/system script**

Description

In RouterOS v2.7, a script may be started in three ways:

- according to a specific time or an interval of time
- on an event − for example, if the netwatch tool sees that an address does not respond to pings
- by another script

Property Description

name (*name*; default: **script*N***) − name of the script to be referenced when invoking it
source (*text*; default: "") − the script itself
owner (*name*; default: **admin**) − the name of the user who created the script
run−count (*integer*; default: **0**) − usage counter. This counter is incremented each time the script is executed, it can be reset to zero by setting 'run−counter=0'. The counters will reset after reboot.
last−started (*time*) − date and time when the script has been last invoked. The argument is shown only if the 'run−count=0'.
policy (multiple choice: ftp, local, policy, read, reboot, ssh, telnet, test, web, write; default: **reboot,read,write,policy,test**) − the name(s) of the specific policy. Can be choosen of the:

- **ftp** − user can log on remotely via ftp and send and retrieve files from the router
- **local** − user can log on locally via console
- **policy** − manage user policies, add and remove user
- **read** − user can retrieve the configuration
- **reboot** − user can reboot the router
- **ssh** − user can log on remotely via secure shell
- **telnet** − user can log on remotely via telnet
- **test** − user can run ping, traceroute, bandwidth test
- **web** − user can log on remotely via http
- **write** − user can retrieve and change the configuration

Notes

You can't do more in the scripts than you are allowed to do by your current user name, that is, you can't use disabled policies. For example, if there is a policy group in **/user group** which allows you **ssh,local,telnet,read,write,policy,test,web** and if this group is assigned to yout user name then you can't make a script that reboots the router.

You can execute a script by using the **run** command.

Example

The following example is a script for writing message "hello" to the system log:

```
[admin@MikroTik] system script> add name=log-test source={:log message=hello}
[admin@MikroTik] system script> print
  0 name="log-test" source=":log message=hello" owner="admin"
    policy=reboot,read,write,policy,test run-count=0

[admin@MikroTik] system script>
```

Task Management

Submenu level : **/system script job**

Description

This facility is used to manage the active or scheduled tasks. You can see the status of all currently active tasks using the **print** command.

Property Description

name (*name*) - name of the script to be referenced when invoking it. **source** (*text*) - the script itself **owner** (*text*; default: **admin**) - the name of the user who created the script

Example

For example, we have a script that delays some process for 10 minutes:

```
[admin@MikroTik] system script> add name=DelayeD source={:delay 10m}
[admin@MikroTik] system script> print
  0 name="log-test" source=":log message=hello" owner=admin
    last-started=feb/27/2003 11:05:19 run-count=1

  1 name="DelayD" source=":delay 10m" owner="admin"
    policy=reboot,read,write,policy,test run-count=0
[admin@MikroTik] system script> run DelayeD
[admin@MikroTik] system script> job print
  # SCRIPT  OWNER                    STARTED
  0 DelayeD admin                    feb/27/2003 11:17:33

[admin@MikroTik] system script>
```

You can cancel execution of a script by removing it from the jobs list:

```
[admin@MikroTik] system script> job remove 0
[admin@MikroTik] system script> job print
[admin@MikroTik] system script> print
  0 name="log-test" source=":log message=hello" owner="admin"
    policy=reboot,read,write,policy,test last-started=feb/27/2003 11:05:13
    run-count=1

  1 name="DelayD" source=":delay 10m" owner="admin"
    policy=reboot,read,write,policy,test last-started=feb/27/2003 11:17:33
    run-count=1

[admin@MikroTik] system script>
```

Script Editor

Submenu level : **/system script edit**

Description

system script edit is simple full-screen editor for scripts. It's used for multiline script writing. To run the script editor just type **system script edit *script-name* source**, where ***script-name*** is the name of the script you want to edit.

Special Keys

Delete - delete character a cursor position
Ctrl-h, backspace - delete character before cursor. Unindent line
Tab - indent line
Ctrl-b, LeftArrow - move cursor left
Ctrl-f, RightArrow - move cursor right
Ctrl-p, UpArrow - move cursor up
Ctrl-n, DownArrow - move cursor down
Ctrl-a, Home - move cursor to the beginning of line or script
Ctrl-e, end - move cursor to the end of line or script
Ctrl-y - insert contents of cut buffer at cursor position
Ctrl-k - delete characters from cursor position to the end of line
Ctrl-u - undo editing action
Ctrl-o - exit editor and accept changes
Ctrl-x - exit editor and discard changes

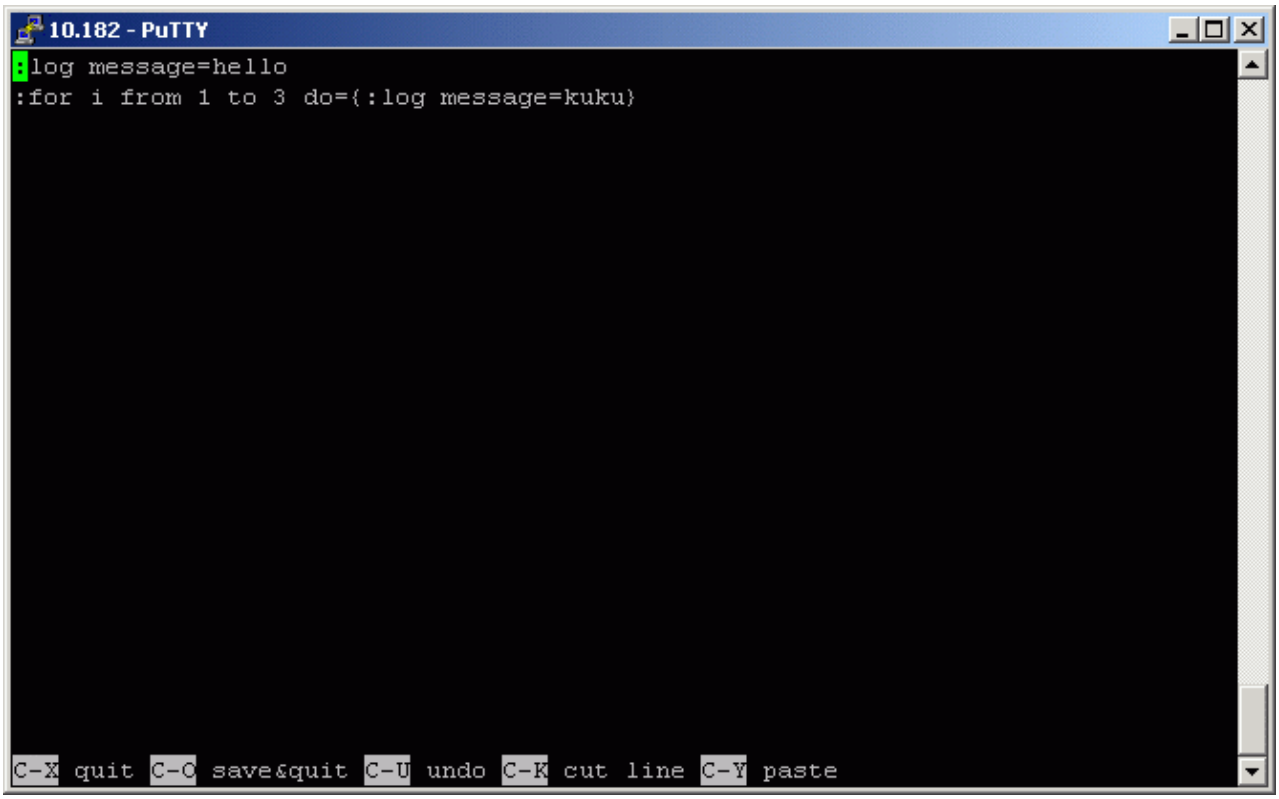
Notes

All characters that are deleted by backspace, delete, Ctrl-k keys are accumulated in cut buffer. Pressing any other key finishes adding to this buffer (Ctrl-y can paste it's contents), and next delete operation will replace it's contents. Undo doesn't change contents of cut buffer.

Editor works only on VT102 compatible terminals (terminal names "vt102", "linux", "xterm", "rxvt" are recognized as VT102 at the moment). Delete, backspace and cursor keys might not work with all terminal programs, use 'Ctrl' alternatives in such cases.

Example

The following example shows the process of sript editing using **edit** command:



This script is used for writing message "hello" and 3 messages "kuku" to the system log.

Network Watching Tool

Specifications

Packages required : ***advanced−tools***
License required : *Any*
Home menu level : ***/tool netwatch***
Protocols utilized : *None*
Hardware usage: *not significant*

Description

Netwatch monitors state of hosts on the network. It does so by sending ICMP pings to list of specified IP addresses. For each entry in netwatch table you can specify IP address, ping interval and console scripts. The main advantage of netwatch is it's ability to issue arbitrary console commands on host state changes.

Property Description

host (*IP address*; default: **0.0.0.0**) − IP address of host that should be monitored
interval (*time*; default: **1s**) − time between pings. Lowering this will make state changes more responsive,

but can create unnecessary traffic and consume system resources
timeout (*time*; default: **1s**) − timeout for each ping. If no reply from a host is received in this time, the host is considered unreachable (**down**)
up−script (*name*) − console script that is executed once when state of a host changes from **unknown** or **down** to **up**
down−script (*name*) − console script that is executed once when state of a host changes from **unknown** or **up** to **down**
since (read−only: *time*) − time when state of host changed last time
status (read−only: up | down | unknown) − tells the current status of the host

- **up** − the host is up
- **down** − the host is down
- **unknown** − when any properties of this list entry are changed, or it is enabled or disabled

Example

This example will run the scripts gw_1 or gw_2 which change the default gateway depending on the status of one of the gateways:

```
[admin@MikroTik] system script> add name=gw_1 source={/ip route set
{... [/ip route find dst 0.0.0.0] gateway 10.0.0.1}
[admin@MikroTik] system script> add name=gw_2 source={/ip route set
{.. [/ip route find dst 0.0.0.0] gateway 10.0.0.217}
[admin@MikroTik] system script> /tool netwatch
[admin@MikroTik] tool netwatch> add host=10.0.0.217 interval=10s timeout=998ms \
\... up−script=gw_2 down−script=gw_1
[admin@MikroTik] tool netwatch> print
Flags: X − disabled
  #   HOST            TIMEOUT              INTERVAL             STATUS
  0   10.0.0.217      997ms                10s                  up
[admin@MikroTik] tool netwatch> print detail
Flags: X − disabled
  0   host=10.0.0.217 timeout=997ms interval=10s since=feb/27/2003 14:01:03
      status=up up−script=gw_2 down−script=gw_1

[admin@MikroTik] tool netwatch>
```

Without scripts, netwatch can be used just as an information tool to see which links are up, or which specific hosts are running at the moment.

Let's look at the example above − it changes default route if gateway becomes unreachable. How it's done? There are two scripts. The script "gw_2" is executed once when status of host changes to **up**. In our case, it's equivalent to entering this console command:

```
[MikroTik] > /ip route set [/ip route find dst 0.0.0.0] gateway 10.0.0.217
```

The **/ip route find dst 0.0.0.0** command returns list of all routes whose **dst−address** value is zero. Usually that's the default route. It is substituted as first argument to **/ip route set** command, which changes gateway of this route to 10.0.0.217

The script "gw_1" is executed once when status of host becomes **down**. It does the following:

```
[MikroTik] > /ip route set [/ip route find dst 0.0.0.0] gateway 10.0.0.1
```

It changes the default gateway if 10.0.0.217 address has become unreachable.

Here's another example, that sends email notification whenever the 10.0.0.215 host goes down:

```
[admin@MikroTik] system script> add name=e-down source={/tool e-mail send
{... from="rieks@mt.lv" server="159.148.147.198" body="Router down"
{... subject="Router at second floor is down" to="rieks@latnet.lv"}
[admin@MikroTik] system script> add name=e-up source={/tool e-mail send
{... from="rieks@mt.lv" server="159.148.147.198" body="Router up"
{.. subject="Router at second floor is up" to="rieks@latnet.lv"}
[admin@MikroTik] system script>
[admin@MikroTik] system script> /tool netwatch
[admin@MikroTik] system netwatch> add host=10.0.0.215 timeout=999ms \
\... interval=20s up-script=e-up down-script=e-down
[admin@MikroTik] tool netwatch> print detail
Flags: X - disabled
  0   host=10.0.0.215 timeout=998ms interval=20s since=feb/27/2003 14:15:36
      status=up up-script=e-up down-script=e-down

[admin@MikroTik] tool netwatch>
```

System Scheduler

Specifications

Packages required : *None*
License required : *Any*
Home menu level : ***/system scheduler***
Protocols utilized : *none*
Hardware usage: *not significant*

Description

System sheduler provides a vay to execute scripts at designated time.

Property Description

name (*name*) − name of the task
interval (*time interval*; default: **0s**) − interval between two script executions, if time **interval** is set to zero, the script is only executed at it's start time, otherwise it is executed repeatedly at the time interval specified
run-count (read-only: *integer*) − to monitor script usage, this counter is incremented each time the script is executed
script (*name*) − name of the script. The script must be present at **/system script**
start-date (*date*) − date of first execution
start-time (*time*) − time of first execution

Notes

Rebooting the router will reset **run-count** counter.

If more than one script has to be executed at one time, they are executed in the order they appear in the scheduler configuration. This can be important if, for example, one scheduled script is used to disable another. The order of scripts can be changed with the **move** command.

If a more complex execution pattern is needed, it can usually be done by scheduling several scripts, and making them enable and disable each other.

Example

We will add a task that executes the script **log−test** every hour:

```
[admin@MikroTik] system script> add name=log−test source=:log
[admin@MikroTik] system script> print
  0 name="log−test" source=":log" owner=admin run−count=0

[admin@MikroTik] system script> .. scheduler
[admin@MikroTik] system scheduler> add name=run−1h interval=1h script=log−test
[admin@MikroTik] system scheduler> print
Flags: X − disabled
  #   NAME      SCRIPT   START−DATE  START−TIME INTERVAL             RUN−COUNT
  0   run−1h    log−test oct/30/2008 15:08:22   1h                   1
[admin@MikroTik] system scheduler>
```

In another example there will be two scripts added that will change the bandwidth setting of a queue rule "Cust0". Everyday at 9AM the queue will be set to 64Kb/s and at 5PM the queue will be set to 128Kb/s. The queue rule, the scripts, and the scheduler tasks are below:

```
[admin@MikroTik] queue simple> add name=Cust0 interface=ether1 \
\... dst−address=192.168.0.0/24 limit−at=64000
[admin@MikroTik] queue simple> print
Flags: X − disabled, I − invalid
  0   name="Cust0" src−address=0.0.0.0/0 dst−address=192.168.0.0/24
      interface=ether1 limit−at=64000 queue=default priority=8 bounded=yes

[admin@MikroTik] queue simple> /system script
[admin@MikroTik] system script> add name=start_limit source={/queue simple set \
\... Cust0 limit−at=64000}
[admin@MikroTik] system script> add name=stop_limit source={/queue simple set \
\... Cust0 limit−at=128000}
[admin@MikroTik] system script> print
  0 name="start_limit" source="/queue simple set Cust0 limit−at=64000"
    owner=admin run−count=0

  1 name="stop_limit" source="/queue simple set Cust0 limit−at=128000"
    owner=admin run−count=0

[admin@MikroTik] system script> .. scheduler
[admin@MikroTik] system scheduler> add interval=24h name="set−64k" \
\... start−time=9:00:00 script=start_limit
[admin@MikroTik] system scheduler> add interval=24h name="set−128k" \
\... start−time=17:00:00 script=stop_limit
[admin@MikroTik] system scheduler> print
Flags: X − disabled
  #   NAME      SCRIPT   START−DATE  START−TIME INTERVAL             RUN−COUNT
  0   set−64k   start... oct/30/2008 09:00:00   1d                   0
  1   set−128k  stop_... oct/30/2008 17:00:00   1d                   0
[admin@MikroTik] system scheduler>
```

The following example schedules script that sends each week backup of router configuration by e−mail.

```
[admin@MikroTik] system script> add name=e−backup source={/system backup
{... save name=email; /tool e−mail send to="root@host.com" subject=[/system
```

```
{... identity get name]" Backup" file=email.backup}
[admin@MikroTik] system script> print
  0 name="e-backup" source="/system backup save name=ema... owner=admin
    run-count=0

[admin@MikroTik] system script> .. scheduler
[admin@MikroTik] system scheduler> add interval=7d name="email-backup" \
\... script=e-backup
[admin@MikroTik] system scheduler> print
Flags: X - disabled
  #   NAME      SCRIPT   START-DATE  START-TIME INTERVAL            RUN-COUNT
  0   email-... e-backup oct/30/2008 15:19:28   7d                  1
[admin@MikroTik] system scheduler>
```

Do not forget to set the e-mail settings, i.e., the SMTP server and From: address under **/tool e-mail**. For example:

```
[admin@MikroTik] tool e-mail> set server=159.148.147.198 from=SysAdmin@host.com
[admin@MikroTik] tool e-mail> print
    server: 159.148.147.198
      from: SysAdmin@host.com
[admin@MikroTik] tool e-mail>
```

Example below will put 'x' in logs each hour from midnight till noon:

```
[admin@MikroTik] system script> add name=enable-x source={/system scheduler
{... enable x}
[admin@MikroTik] system script> add name=disable-x source={/system scheduler
{... disable x}
[admin@MikroTik] system script> add name=log-x source={:log message=x}
[admin@MikroTik] system script> .. scheduler
[admin@MikroTik] system scheduler> add name=x-up start-time=00:00:00 \
\... interval=24h script=enable-x
[admin@MikroTik] system scheduler> add name=x-down start-time=12:00:00
\... interval=24h script=disable-x
[admin@MikroTik] system scheduler> add name=x start-time=00:00:00 interval=1h \
\... script=log-x
[admin@MikroTik] system scheduler> print
Flags: X - disabled
  #   NAME      SCRIPT   START-DATE  START-TIME INTERVAL            RUN-COUNT
  0   x-up      enable-x oct/30/2008 00:00:00   1d                  0
  1   x-down    disab... oct/30/2008 12:00:00   1d                  0
  2   x         log-x    oct/30/2008 00:00:00   1h                  0
[admin@MikroTik] system scheduler>
```

Traffic Monitor

Specifications

Packages required : *None*
License required : *Any*
Home menu level : ***/tool traffic-monitor***
Protocols utilized : *None*
Hardware usage: *not significant*

Description

The traffic monitor tool is used to execute console scripts on when interface traffic crosses some given thresholds. Each item in traffic monitor list consists of its name (which is useful if you want to disable or change properties of this item from another script), some parameters specifying traffic condition and the pointer to a script or scheduled event to execute when this condition is met.

Property Description

name (*name*) − name of the traffic monitor item
interface (*name*) − interface to monitor
threshold (*integer*; default: **0**) − traffic threshold
trigger (above | always | below; default: **above**) − condition on which to execute script

- **above** − the script will be run each time traffic exceeds the threshold
- **always** − triggers scripts on both **above** and **below** conditions
- **below** − triggers script in the opposite condition, when traffic drops under the threshold

traffic (transmitted | received; default: **transmitted**) − type of traffic to monitor

- **transmitted** − transmitted packets
- **received** − received packets

on−event (*name*) − Script source. Must be present under **/system script**

Example

The example monitor enables the interface ether2, if the received traffic exceeds 15kbps on ether1, and disables the interface ether2, if the received traffic falls below 12kbps on ether1.

```
[admin@MikroTik] system script> add name=eth−up source={/interface enable ether2}
[admin@MikroTik] system script> add name=eth−down source={/interface disable
{... ether2}
[admin@MikroTik] system script> /tool traffic−monitor
[admin@MikroTik] tool traffic−monitor> add name=turn_on interface=ether1 \
\... on−event=eth−up threshold=15000 trigger=above traffic=received
[admin@MikroTik] tool traffic−monitor> add name=turn_off interface=ether1 \
\... on−event=eth−down threshold=12000 trigger=below traffic=received
[admin@MikroTik] tool traffic−monitor> print
Flags: X − disabled, I − invalid
  #   NAME           INTERFACE     TRAFFIC     TRIGGER THRESHOLD  ON−EVENT
  0   turn_on        ether1        received    above   15000      eth−up
  1   turn_off       ether1        received    below   12000      eth−down
[admin@MikroTik] tool traffic−monitor>
```

Sigwatch

Specifications

Packages required : ***advanced−tools***
License required : *Any*
Home menu level : ***/tool sigwatch***
Protocols utilized : *None*
Hardware usage: *not significant*

Description

Sigwatch can be used to monitor state of serial port pins.

Property Description

name − name of the sigwatch item
log (yes | no; default: **no**) − add or not message in form "name−of−sigwatch−item: signal changed [to high | to low]" to System−Info facility whenever this sigwatch item is triggered
script (*name*) − script that is executed whenever this item is triggered
on−condition (on | off; default: **on**) − on what condition to trigger actions of this item

- **on** − trigger when state of pin changes to high
- **off** − trigger when state of pin changes to low
- **change** − trigger whenever state of pin changes. If state of pin changes rapidly, there might be triggered only one action for several state changes

port (*name*) − serial port to monitor
signal (dtr | rts | cts | dcd | ri | dsr; default: **rts**) − name of signal or number of pin (for standard 9−pin connector) to monitor

- **dtr** − Data Terminal Ready − pin #4
- **rts** − Request To Send − pin #7
- **cts** − Clear To Send − pin #8
- **dcd** − Data Carrier Detect − pin #1
- **ri** − Ring Indicator − pin #9
- **dsr** − Data Set Ready − pin #6

count (read−only: *integer*) − how many time event for this item was triggered. Count is reset on reboot and on most item configuration changes
state (read−only: *text*) − last remembered state of monitored signal

Notes

You can type actual script source instead of the script name from **/system script** list.

Example

In the following example we'll add new sigwatch item that monitors whether serial1 port has **cts** signal.

```
[admin@10.179] tool sigwatch> pr
Flags: X − disabled
  #   NAME                                  PORT    SIGNAL     ON−CONDITION LOG
  0   test                                  serial1 cts        change       no

[admin@MikroTik] tool sigwatch>
```

By typing a command **print detail interval=1s** we can prove whether a cable is connected or disconnected. See the **state** argument − if the cable is connected to the serial port, it shows **on**, when disconnected − **off**:

```
[admin@MikroTik] tool sigwatch> print detail
Flags: X − disabled
  0   name="test" port=serial1 signal=cts on−condition=change log=no script=""
      count=1 state=on

[admin@MikroTik] tool sigwatch> print detail
```

```
Flags: X - disabled
  0   name="test" port=serial1 signal=cts on-condition=change log=no script=""
      count=1 state=on

[admin@MikroTik] tool sigwatch> print detail
Flags: X - disabled
  0   name="test" port=serial1 signal=cts on-condition=change log=no script=""
      count=2 state=off

[admin@MikroTik] tool sigwatch> print detail
Flags: X - disabled
  0   name="test" port=serial1 signal=cts on-condition=change log=no script=""
      count=2 state=off

[admin@MikroTik] tool sigwatch>
```

In the **port** menu it's seen what **signal** is used by serial cable. For example, without any cables it looks like this:

```
[admin@MikroTik] port> print stats
  0 name="serial0" line-state=dtr,rts

  1 name="serial1" line-state=dtr,rts
[admin@MikroTik] port>
```

But after adding a serial cable to the serial port:

```
[admin@MikroTik] port> print stats
  0 name="serial0" line-state=dtr,rts

  1 name="serial1" line-state=dtr,rts,cts
[admin@MikroTik] port>
```

It means that the line-state beside the **dtr** and **rts** signals has also **cts** when a serial cable is connected.

The example below will execute a script whenever **on-condition** changes to **off**:

```
[admin@10.MikroTik] tool sigwatch> pr detail
Flags: X - disabled
  0   name="cts_rest" port=serial1 signal=cts on-condition=off log=no
      script=/system shutdown count=0 state=on
```

It means that if a serial cable is connected to the serial port, all works fine, but as soon as it's disconnected - the router shuts down. It will continue all the time until the serial cable will not be connected again.

Serial Console and Terminal

Document revision 1.1 (02−May−2003)
This document applies to the MikroTik RouterOS v2.7

Table of Contents

Summary

The Serial Console and Terminal are tools, used to communicate with devices and other systems that are interconnected via serial port. The serial terminal may be used to monitor and configure many devices – including modems, network devices (including MikroTik routers), and any device that can be connected to a serial (asynchronous) port.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/system***
Protocols utilized : *RS−232*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading

Description

The Serial Console (managed side) feature allows configuring one serial port of the MikroTik router for access to the router's Terminal Console over the serial port. A special null−modem cable is required to connect the router's serial port with the workstation's or laptop's serial (COM) port. A terminal emulation

program, e.g., HyperTerminal, should be run on the workstation. You can also use MikroTik RouterOS to connect to an another Serial Console (for example, on a Cisco router)

Several customers have described situations where the Serial Terminal (managing side) feature would be useful:

- in a mountaintop where a MikroTik wireless installation sits next to equipment (including switches and Cisco routers) that can not be managed in−band (by telnet through an IP network)
- monitoring weather−reporting equipment through a serial−console.
- connection to a high−speed microwave modem that needed to be monitored and managed by a serial−console connection.

With the serial−terminal feature of the MikroTik, up to 132 (and, may be, even more) devices can be monitored and controlled.

Serial Console Configuration

A special null−modem cable should be used for connecting to the serial console. The Serial Console cabling diagram for DB9 connectors is as follows:

Router Side (DB9f)	Signal	Direction	Side (DB9f)
1,6	CD, DSR	IN	4
2	RxD	IN	3
3	TxD	OUT	2
4	DTR	OUT	1,6
5	GND	−	5
7	RTS	OUT	8
8	CTS	IN	7

Setting Serial Console

Submenu level : **/system serial−console**

```
[admin@MikroTik] system serial-console> print
    enabled: no
       port: serial0
[admin@MikroTik] system serial-console>
```

Property Description

enabled (yes | no; default: **no**) − whether serial console is enabled
port (*name*; default: **serial0**) − which port should the serial terminal listen on

Example

To enable Serial Console:

```
[admin@MikroTik] system serial-console> set enabled=yes
[admin@MikroTik] system serial-console> print
    enabled: yes
       port: serial0
[admin@MikroTik] system serial-console>
```

To check if the port is available or used:

```
[admin@MikroTik] system serial-console> /port print detail
  0 name=serial0 used-by=Serial Console baud-rate=9600 data-bits=8 parity=none
    stop-bits=1 flow-control=none

  1 name=serial1 used-by="" baud-rate=9600 data-bits=8 parity=none stop-bits=1
    flow-control=none

[admin@MikroTik] system serial-console>
```

Usind Serial Terminal

Command name : **/system serial-terminal**

Description

The command is used to communicate with devices and other systems that are connected to router via serial port.

All keyboard input is forwarded to the serial port and all data from the port is output to the connected device. After exiting with [Ctrl]+[Q], the control signals of the port are lowered. The speed and other parameters of serial port may be configured in the **/port** directory of router console. No terminal translation on printed data is performed. It is possible to get the terminal in an unusable state by outputting sequences of inappropriate control characters or random data. Do not connect to devices at an incorrect speed and avoid dumping binary data.

Property Description

port (*name*) - which port to use

Notes

[Ctrl]+[Q] and [Ctrl]+[X] have special meaning and are used to provide a possibility of exitting from nested serial-terminal sessions:

To send [Ctrl]+[X] to to serial port, press [Ctrl]+[X] [Ctrl]+[X]
To send [Ctrl]+[Q] to to serial port, press [Ctrl]+[X] [Ctrl]+[Q].

Example

To conect to a device connected to the **serail1** port:

```
[admin@MikroTik] system> serial-terminal serial1

[Type Ctrl-Q to return to console]
```

```
[Ctrl-X is the prefix key]
```

Troubleshooting

- *An error appears when trying to enable the Serial Console.*
  This situation can occur when the Serial console is set on the port which is already been used by another device such as a ppp-server, ppp-client, LCD etc, e.g.:

```
[admin@MikroTik] system serial-console> print
    enabled: no
       port: serial0
[admin@MikroTik] system serial-console> set enabled=yes
ERROR: can't acquire requested port
```

  Check the available ports using the **/port print detail** command:

```
[admin@MikroTik] system serial-console> /port print
  0 name=serial0 used-by=LCP Panel baud-rate=9600 data-bits=8 parity=none
    stop-bits=1 flow-control=none

  1 name=serial1 used-by="" baud-rate=9600 data-bits=8 parity=none stop-bits=1
    flow-control=none
```

  The Serial Console port must be set to serial1, since the serial0 port is already used by another device:

```
[admin@MikroTik] system serial-console> set port=serial1 enable=yes
[admin@MikroTik] system serial-console> print
    enabled: yes
       port: serial1
[admin@MikroTik] system serial-console>
```

- *The port parameter settings for baud rate, stop bits, etc., do not match the settings of your terminal.*
  Adjust the port settings of your Terminal program to the settings of MikroTik router (see **/port print detail**).

Additional Resources

http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html
http://www.ctsystems.org/rs.htm

SSH (Secure Shell) Server and Client

Document revision v 1.1 (25−Apr−2003)
This document applies to the MikroTik RouterOS V2.7

Contents of the Manual

Summary

SSH Client authenticates server and encrypts traffic between the client and server. You can use SSH just the same way as telnet − you run the client, tell it where you want to connect to, give your username and password, and everything is the same after that. After that you won't be able to tell that you're using SSH. The SSH feature can be used with various SSH Telnet clients to securely connect to and administrate the router.

The MikroTik RouterOS supports:

- SSH 1.3, 1.5, and 2.0 protocol standards
- server functions for secure administration of the router
- telnet session termination with 40 bit RSA SSH encryption is supported
- secure ftp is not supported
- Winbox connection encryption (TSL)

The MikroTik RouterOS has been tested with the following SSH telnet terminals:

- PuTTY
- Secure CRT
- Most SSH compatible telnet clients

Specifications

Packages required : **security**
License required : Any
Home menu level : **/system ssh**
Protocols utilized : SSH
Hardware usage : not significant

Related Documents

Software Package Installation and Upgrading

SSH Server

Submenu level : /**ip service**

Description

SSH Server is already up and running after Mikro Tik router installation. The default port of the service is 22. You can set a different port number.

Property Description

name (*name*) − service name
port (*1...65535*) − port the service listens on
address (*IP address/mask;* default: **0.0.0.0/0**) − IP address from which the service is accessible

Example

```
[admin@MikroTik] ip service>set ssh port=51
[admin@MikroTik] ip service> print
Flags: X − disabled, I − invalid
  #   NAME                                                                  PORT  ADDRESS
  0   telnet                                                                23    0.0.0.0/0
  1   ftp                                                                   21    0.0.0.0/0
  2   www                                                                   80    0.0.0.0/0
  3   ssh                                                                   51    0.0.0.0/0
[admin@MikroTik] ip service>
```

SSH Client

Command name:**/system ssh**

Example

```
[admin@MikroTik] /system ssh 10.0.0.211 user=admin port=22
admin@10.0.0.211's password:


  MMM      MMM       KKK                          TTTTTTTTTTT      KKK
  MMMM    MMMM       KKK                          TTTTTTTTTTT      KKK
  MMM MMMM MMM  III  KKK  KKK  RRRRRR     OOOOOO      TTT     III  KKK  KKK
  MMM  MM  MMM  III  KKKKK     RRR  RRR  OOO  OOO     TTT     III  KKKKK
  MMM      MMM  III  KKK KKK   RRRRRR    OOO  OOO     TTT     III  KKK KKK
  MMM      MMM  III  KKK  KKK  RRR  RRR   OOOOOO      TTT     III  KKK  KKK

  MikroTik RouterOS v2.7 (c) 1999−2002       http://www.mikrotik.com/
```

```
Terminal vt100 detected, using single line input mode
[admin@10.0.0.211] >
```

Additional Resources

Links for Windows Client:

http://www.zip.com.au/~roca/ttssh.html
http://www.chiark.greenend.org.uk/~sgtatham/putty.html
http://pgpdist.mit.edu/FiSSH/index.html
http://telneat.lipetsk.ru/
http://support.jgaa.com/?cmd=ShowArticle
http://akson.sgh.waw.pl/~chopin/ssh/index_en.html
http://cs.mscd.edu/MSSH/index.html
http://www.networksimplicity.com/openssh/

Other links:

http://www.openssh.com/
http://www.freessh.org/

Support Output File

Document revision 1.2 (10−Mar−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

The support file is used for debugging MikroTik RouterOS and to solve the support questions faster. All MikroTik Router information is saved in a binary file, which is stored on the router and can be downloaded from the router using ftp.

Specifications

Packages required : **system**
License required : Any
Home menu level : **/system**
Hardware usage: There is no significant resource usage

Generating Support Output File

Command name : **/system sup−output**

Example

To make a Support Output File:

```
[admin@MikroTik] > system sup−output
creating supout.rif file, might take a while
..................
Accomplished!
[admin@MikroTik] >
```

To see the files stored on the router:

```
[admin@MikroTik] > file print
  # NAME                          TYPE         SIZE       CREATION−TIME
  0 supout.rif                    unknown      38662      feb/28/2003 16:12:04
[admin@MikroTik] >
```

Connect to the router using FTP and download the supout.rif file using BINARY file transfer mode. Send the supout.rif file to MikroTik Support support@mikrotik.com with detailed description of the problem.

System Resource Management

Document revision 20−Jan−2003
This document applies to the MikroTik RouterOS v2.7

Table of Contents

Summary

MikroTik RouterOS offers several features for monitoring and managing the system resources.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/system***
Protocols utilized : *None*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
Network Time Protocol (NTP)

System Resource Monitor

Submenu level : **/system resource**

Example

To view the basic system resource status:

```
[admin@MikroTik] system resource> print
             uptime: 1d23h32m6s
        free-memory: 1112 kB
       total-memory: 29528 kB
                cpu: "WinChip"
           cpu-load: 0
     free-hdd-space: 6400 kB
    total-hdd-space: 46478 kB
[admin@MikroTik] system resource>
```

To view the current system CPU usage and free memory:

```
[admin@MikroTik] system resource> monitor
       cpu-used: 3
    free-memory: 1112

[admin@MikroTik] system resource>
```

Notes

The propery names are self-explanatory.

In **monitor** command printout the values for cpu usage and free memory are in percentage and kilobytes, respectively.

IRQ Usage Monitor

Command name : **/system resource irq print**

Description

IRQ usage shows which IRQ (Interrupt requests) are currently used by hardware.

Example

```
[admin@MikroTik] system resource> irq print
Flags: U - unused
   IRQ OWNER
   1   keyboard
   2   APIC
 U 3
   4   sync1
   5   pc1
 U 6
 U 7
 U 8
 U 9
   10  ether2
   11  ether1
 U 12
   13  FPU
   14  IDE 1
[admin@MikroTik] system resource>
```

IO Port Usage Monitor

Command name : **/system resource io print**

Description

IO usage shows which IO (Input/Output) ports are currently used by hardware.

Example

```
[admin@MikroTik] system resource> io print
 PORT-RANGE        OWNER
 20-3F             APIC
 40-5F             timer
 60-6F             keyboard
 80-8F             DMA
 A0-BF             APIC
 C0-DF             DMA
 F0-FF             FPU
 1F0-1F7           IDE 1
 300-33F           pc1
 3C0-3DF           VGA
 3F6-3F6           IDE 1
 CF8-CFF           [PCI conf1]
 1000-100F         [Silicon Integrated Systems [SiS] 5513 [IDE]]
 1000-1007         IDE 1
 1008-100F         IDE 2
 6000-60FF         [Realtek Semiconductor Co., Ltd. RTL-8139]
 6000-60FF         [8139too]
 6100-61FF         [Realtek Semiconductor Co., Ltd. RTL-8139 (#2)]
 6100-61FF         [8139too]
```

```
[admin@MikroTik] system resource>
```

Reboot

Command name : **/system reboot**

Description

The system reboot is required when upgrading or installing new software packages. The packages are installed during the system shutdown.

The reboot process sends termination signal to all running processes, unmounts the file systems, and reboots the router.

Notes

Only users, which are members of groups with reboot privileges are permitted to reboot the router

Reboot can be called from scripts, in which case it does not prompt for confirmation

Example

```
[admin@MikroTik] system> reboot
Reboot, yes? [y/N]: y
system will reboot shortly
```

Shutdown

Command name : **/system shutdown**

Description

Before turning the power off for the router, the system should be brought to halt. The shutdown process sends termination signal to all running processes, unmounts the file systems, and halts the router.

For most systems, it is necessary to wait approximately 30 seconds for a safe power down.

Notes

Only users, which are members of groups with reboot privileges are permitted to shutdown the router

Shutdown can be called from scripts, in which case it does not prompt for confirmation

Example

```
[admin@MikroTik] system> shutdown
Shutdown, yes? [y/N]: y
system will shutdown promptly
```

Configuration Reset

Command name : **/system reset**

Description

The command clears all configuration of the router and sets it to the default including the login name and password ('admin' and no password).

The router is rebooted after the reset command.

Example

```
[admin@MikroTik] system> reset
Dangerous! Reset anyway? [y/N]:
```

Router Identity

Submenu level : **/system identity**

Description

The router identity is displayed before the command prompt. It is also used for DHCP client as 'host name' parameter when reporting it to the DHCP server.

Example

To view the router identity:

```
[admin@MikroTik] system identity> print
    name: "MikroTik"
[admin@MikroTik] system identity>
```

To set the router identity:

```
[admin@MikroTik] system identity> set name=Our_GW
[admin@Our_GW] system identity>
```

Date and Time

Submenu level : **/system clock**

Property Description

time (*string*) − date and time in format: "mm/DD/YYYY HH:MM:SS"
time−zone (*string*) − UTC timezome

Notes

It is recommended that you reboot the router after time change to obviate the possible errors in time measurments and logging

Date and time settings become permanent and effect BIOS settings.

Example

To view the current date and time settings

```
[admin@MikroTik] system clock> print
         time: aug/09/2002 21:27:29
    time-zone: +03:00
[admin@MikroTik] system resource>
```

To set the system date and time:

```
[adminMikroTik] system clock> set date=mar/26/2002 time=14:41:00 time-zone=+02:00
[admin@MikroTik] system clock> print
         time: mar/26/2002 16:41:12
    time-zone: +02:00
[admin@MikroTik] system clock>
```

Configuration Change History

Submenu level : **/system history** Command name : **/undo** Command name : **/redo**

Description

The history of system configuration changes is held until the next router shutdown. The invoked commands can be 'undone' (in reverse order they have been invoked). The 'undone' commands may be 'redone' (in reverse order they have been 'undone').

Command Description

/undo − undoes previous configuration changing command (except another **/undo** command)
/redo − undoes previous **/undo** command
/system history print − print a list of last configuration changes, specifying whether the action can be undone or redone

Notes

Floating−undo actions are created within the current SAFE mode session. They are automatically converted to undoable and redoable when SAFE mode terminated successfully, and are all undone irreverively when SAFE mode terminated insuccessfully.

Undo command cannot undo commands past start of the SAFE mode.

Example

To show the list of configuration changes:

```
[admin@MikroTik] system history> print
Flags: U − undoable, R − redoable, F − floating−undo
   ACTION                                  BY                 POLICY
 U new traffic monitor script added
 U DNS server configuration changed
 U device changed
 U marking rule moved                      admin
 U route changed
 U route added
 U routing table added
 U ipsec manual sa ex1 added
[admin@MikroTik] system history>
```

The **undo** command's effect:

```
[MikroTik] system history> /undo
[admin@MikroTik] system history> print
Flags: U − undoable, R − redoable
   ACTION                                  BY                 POLICY
 R new traffic monitor script added
 U DNS server configuration changed
 U device changed
 U marking rule moved                      admin
 U route changed
 U route added
 U routing table added
 U ipsec manual sa ex1 added
[admin@MikroTik] system history>
```

Telnet Server and Client

Document revision 1.2 (05−May−2003)
This document applies to the MikroTik RouterOS v2.7

Table of Contents

Summary

MikroTik RouterOS has a build−in Telnet server and client features. These two are used to communicate with other systems over a network.

Specifications

Packages required : *None*
License required : *Any*
Home menu level : ***/system***, ***/ip service***
Standards and Technologies : *Telnet (RFC 854)*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
Services, Protocols and Ports
Configuration Backup and Restore

Telnet Server

Submenu level : **/ip service**

Description

Telnet protocol is intended to provide a fairly general, bi−directional, eight−bit byte oriented communications facility. The main goal is to allow a standard method of interfacing terminal devices to each other.

MikroTik RouterOS implements industry standard Telnet server. It uses port 23, which must not be disabled on the router in order to use the feature.

You can enable/disable this service or allow the use of the service to certain IP addresses.

Example

```
[admin@MikroTik] ip service> print detail
Flags: X - disabled, I - invalid
  0   name="telnet" port=23 address=0.0.0.0/0

  1   name="ftp" port=21 address=0.0.0.0/0

  2   name="www" port=80 address=0.0.0.0/0

[admin@MikroTik] ip service> set 0 address 10.10.10.0/24
[admin@MikroTik] ip service>
```

Telnet Client

Command name : **/system telnet**

Description

MikroTik RouterOS telnet client is used to connect to other hosts in the network via Telnet protocol.

You can type something that cannot be treated as an IP address into the telnet prompt in order to use advanced telnet mode. This is for advanced users only.

Example

A simple example of Telnet connection:

```
[admin@MikroTik] > /system telnet 192.168.0.2
Trying 192.168.0.2...
Connected to 192.168.0.2.
Escape character is '^]'.

MikroTik v2.7rc4
Login: admin
Password:

  MMM      MMM       KKK                          TTTTTTTTTTT      KKK
  MMMM    MMMM       KKK                          TTTTTTTTTTT      KKK
  MMM MMMM MMM  III  KKK  KKK  RRRRRR     OOOOOO      TTT     III  KKK  KKK
  MMM  MM  MMM  III  KKKKK     RRR  RRR  OOO  OOO     TTT     III  KKKKK
  MMM      MMM  III  KKK KKK   RRRRRR    OOO  OOO     TTT     III  KKK KKK
  MMM      MMM  III  KKK  KKK  RRR  RRR   OOOOOO      TTT     III  KKK  KKK

  MikroTik RouterOS v2.7 (c) 1999-2003       http://www.mikrotik.com/


Terminal unknown detected, using single line input mode
[admin@MikroTik] >
```

Advanced Telnet command mode:

```
[admin@MikroTik] > system telnet
```

```
:
telnet> ?
Commands may be abbreviated.  Commands are:

close           close current connection
logout          forcibly logout remote user and close the connection
display         display operating parameters
mode            try to enter line or character mode ('mode ?' for more)
open            connect to a site
quit            exit telnet
send            transmit special characters ('send ?' for more)
set             set operating parameters ('set ?' for more)
unset           unset operating parameters ('unset ?' for more)
status          print status information
toggle          toggle operating parameters ('toggle ?' for more)
slc             set treatment of special characters

z               suspend telnet
environ         change environment variables ('environ ?' for more)
telnet>
```

UPS Monitor

Document revision 1.1 (21−Jan−2003)
This document applies to the MikroTik RouterOS v2.7

Table of Contents

Summary

The UPS monitor feature works with APC UPS units that support “smart” signaling. This feature enables the network administrator to monitor the UPS and set the router to ‘gracefully’ handle any power outage with no corruption or damage to the router. The basic purpose of this feature is to ensure that the router will come back online after an extended power failure. To do this, the router will monitor the UPS and set itself to hibernate mode when the ‘utility’ power is down and the UPS battery is has less than 10% of its battery power left. The router will then continue to monitor the UPS (while in hibernate mode) and then restart itself after when the ‘utility’ power returns. If the UPS battery is drained and the router loses all power, the router will power back to full operation when the ‘utility’ power returns.

The UPS monitor feature on the MikroTik RouterOS supports:

- hibernate and safe reboot on power and battery failure
- UPS battery test and run time calibration test
- monitoring of all “smart” mode status information supported by UPS
- logging of power changes

Specifications

Packages required : ***ups***
License required : *Any*
Home menu level : ***/system ups***
Protocols utilized : *APC's smart protocol*

Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading

Cabling

The APC UPS (BackUPS Pro or SmartUPS) requires a special serial cable. If no cable came with the UPS, a cable may be ordered from APC or one can be made "in−house". Use the following diagram:

Router Side (DB9f)	Signal	Direction	UPS Side (DB9m)
2	Receive	IN	2
3	Send	OUT	1
5	Ground		4
7	CTS	IN	6

UPS Monitor Setup

Submenu level : **/system ups**

```
[admin@MikroTik] system> ups
[admin@MikroTik] system ups> print
                     enabled: no
                        port: (unknown)
               off−line−time: 5m
                min−run−time: 5m
               alarm−setting: immediate
            rtc−alarm−setting: none
[admin@MikroTik] system ups>
```

Property Description

enabled (yes | no, default: **no**) − status of the monitoring is disabled by default
port (*name*)− s communication port of the router
off−line−time (*time*, default: **5m**) − how long to work on batteries. The router waits that amount of time and then goes into hibernate mode until the UPS reports that the ‘utility’ power is back

- **0** − the router will go into hibernate mode according the **min−run−time** setting and 10% of battery power event. In this case, the router will wait until the UPS reports that the battery power is below 10%

min−run−time (*time*, default: **5m**) − minimal run time remaining
After a ‘utility’ failure, the router will monitor the run−time−left value. When the value reaches the min−run−time value, the router will go to hibernate mode

- **0** − the router will go to hibernate mode when the “battery low” signal is sent indicating that the battery power is below 10%

alarm−setting (delayed | immediate | low−battery | none, default: **immediate**) − UPS sound alarm setting:

- **delayed** − alarm is delayed to the on−battery event
- **immediate** − alarm immediately after the on−battery event

- **low−battery** − alarm only when the battery is low
- **none** − do not alarm

rtc−alarm−setting (delayed | immediate | low−battery | none, default: **none**) − UPS sound alarm setting during run time calibration:

- **delayed** − alarm is delayed to the on−battery event
- **immediate** − alarm immediately after the on−battery event
- **low−battery** − alarm only when the battery is low
- **none** − do not alarm

Statistics:

model (*string*) − less than 32 ASCII character string consisting of the UPS model name (the words on the front of the UPS itself).

version (*string*) − UPS version, consists of three fields: SKU number, firmware revision, country code. The county code may be one of the following:

- **I** − 220/230/240 Vac
- **D** − 115/120 Vac
- **A** − 100 Vac
- **M** − 208 Vac
- **J** − 200 Vac

serial (*string*) − a string of at least 8 characters directly representing the UPS's serial number as set at the factory. Newer SmartUPS models have 12−character serial numbers

manufacture−date (*string*) − the UPS's date of manufacture in the format "mm/dd/yy" (month, day, year)

nominal−battery−voltage (*integer*) − the UPS's nominal battery voltage rating (this is not the UPS's actual battery voltage)

Notes

In order to enable UPS monitor, the serial port should be available:

```
[admin@MikroTik] port> print
  # NAME                              USED-BY                           BAUD-RATE
  0 serial0                           Serial Console                    9600
  1 serial1                                                             9600
[admin@MikroTik] port>
```

Port **serial1** if free in this example.

Example

To enable the UPS monitor for port **serial1**:

```
[admin@MikroTik] system ups> set port=serial1 enabled=yes
[admin@MikroTik] system ups> print
                    enabled: yes
                       port: serial1
              off-line-time: 5m
               min-run-time: 5m
              alarm-setting: immediate
          rtc-alarm-setting: immediate
                      model: "Back-UPS Pro 420"
                    version: "11.4.I"
              serial-number: "NB9941252992"
```

```
           manufacture-date: "10/08/99"
    nominal-battery-voltage: 12
[admin@MikroTik] system ups>
```

Runtime Calibration

Command name : **/system ups run−time−calibration**

Description

The **run−time−calibration** command causes the UPS to start a run time calibration until less than 25% of full battery capacity is reached. This command calibrates the returned run time value.

Notes

The test begins only if battery capacity is 100%.

Example

```
[MikroTik] system ups> run-time-calibration
```

UPS Monitoring

Command name : **/system ups monitor**

Property Description

Statistics:

on−line (yes | no) − whether power is being provided by the external utility (power company)
on−battery (yes | no) − whether UPS battery is supplying power
transfer cause (*string*) − the reason for the most recent transfer to on−battery operation (only shown when the unit is on−battery):

- unacceptable utility voltage rate of change
- detection of high utility voltage
- detection of low utility voltage
- detection of a line voltage notch or spike
- transfer in response to battery−test or run−time−calibration

low−battery − Only shown when the UPS report this status
replace−battery − Only shown when the UPS report this status
overloaded−output − Only shown when the UPS report this status
smart−boost−mode − Only shown when the UPS report this status
smart−ssdd−mode − Only shown when the UPS report this status
run−time−calibration−running − Only shown when the UPS report this status
run−time−left − the UPS's estimated remaining run time in minutes. You can query the UPS when it is operating in the on−line, bypass, or on−battery modes of operation. The UPS's remaining run time reply is based on available battery capacity and output load
battery−charge − the UPS's remaining battery capacity as a percent of the fully charged condition
battery−voltage − the UPS's present battery voltage. The typical accuracy of this measurement is ±5% of the maximum value (depending on the UPS's nominal battery voltage)

line−voltage − the the in−line utility power voltage
output−voltage − the UPS's output voltage
load − the UPS's output load as a percentage of full rated load in Watts. The typical accuracy of this measurement is ±3% of the maximum of 105%
frequency − When operating on−line, the UPS's internal operating frequency is synchronized to the line within variations within 3 Hz of the nominal 50 or 60 Hz. The typical accuracy of this measurement is ±1% of the full scale value of 63 Hz

Example

When running on utility power:

```
[admin@MikroTik] system ups> monitor
                        on−line: yes
                    on−battery: no
                  run−time−left: 11m
                 battery−charge: 100
                battery−voltage: 13
                   line−voltage: 221
                 output−voltage: 221
                           load: 57
                       fequency: 50

[admin@MikroTik] system ups>
```

When running on battery:

```
[admin@MikroTik] system ups> monitor
                        on−line: no
                    on−battery: yes
                transfer−cause: "utility voltage notch or spike detected"
                  run−time−left: 9m
                 battery−charge: 95
                battery−voltage: 11
                   line−voltage: 0
                 output−voltage: 233
                           load: 66
                       fequency: 50

[admin@MikroTik] system ups>
```

Additional Resources

http://www.linuxdoc.org/HOWTO/UPS−HOWTO.html
http://www.sibbald.com/apcupsd/manual/upsbible.html

Bandwidth Test

Document revision 1.4 (06−Aug−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

The Bandwidth Tester can be used to monitor the throughput only to a remote MikroTik router (either wired or wireless) and thereby help to discover network ‘bottlenecks’.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/tool***
Protocols utilized : *TCP (RFC793, UDP (RFC768)*
Hardware usage: *high*

Related Documents

Software Package Installation and Upgrading

Description

Protocol Description

The TCP test uses the standard TCP protocol with acknowledgments and follows the TCP algorithm on how many packets to send according to latency, dropped packets, and other features in the TCP algorithm. Please review the TCP protocol for details on its internal speed settings and how to analyze its behavior. Statistics for throughput are calculated using the entire size of the TCP packet. As acknowledgments are an internal working of TCP, their size and usage of the link are not included in the throughput statistics. Therefore this

statistic is not as reliable as the UDP statistic when estimating throughput.

The UDP tester sends 110% or more packets than currently reported as received on the other side of the link. To see the maximum throughput of a link, the packet size should be set for the maximum MTU allowed by the links – usually this is 1500 bytes. There is no acknowledgment required by UDP; this implementation means that the closest approximation of the throughput can be seen.

Usage Notes

!Caution! Bandwidth Test uses all available bandwidth (by default) and may impact network usability.

Bandwidth Test uses much resources. If you want to test real throughput of a router, you should run bandwidth test through it not from or to it. To do this you need at least 3 routers connected in chain: the Bandwidth Server, the given router and the Bandwidth Client:

Server Configuration

Submenu level : /**tool**

```
[admin@MikroTik] tool bandwidth-server> print
                    enabled: no
                authenticate: yes
    allocate-udp-ports-from: 2000
                max-sessions: 10
[admin@MikroTik] tool>
```

Property Description:

enable (yes | no; default: **no**)− enable client connections for bandwidth test
authenticate (yes | no; default: **yes**)− communicate only with authenticated (by valid username and password) clients
allocate−udp−ports−from (1000...64000; default: **2000**)− allocate UDP ports from
max−sessions (1...1000; default: **10**) − maximal number of bandwidth−test clients

Notes

The list of current connections can be get in **session** submenu:

```
[admin@MikroTik] tool> bandwidth-server session print
  # FROM            PROTOCOL DIRECTION USER
  0 10.0.0.168      udp      send

[admin@MikroTik] tool>
```

Example

To enable bandwidth−test server without client authentication:

```
[admin@MikroTik] tool bandwidth−server> set enabled=yes authenticate=no
[admin@MikroTik] tool bandwidth−server> print
                   enabled: yes
              authenticate: no
    allocate−udp−ports−from: 2000
              max−sessions: 10
[admin@MikroTik] tool>
```

Client Configuration

Command name : /**tool bandwidth−test**

Property Description

address (*IP address*) − IP address of destination host
assume−lost−time (*time*; default: **0s**) − assume that connection is lost if Bandwidth Server is not responding for that time
direction (*receive/transmit/both;* default**: transmit**) − the direction of the test
do (*name* | *string*; default: "") − script source
duration (*time*; default: **0s**) − duration of the test

- **0s** − test duration is not limited

interval (*20ms...5s;*; default: **1s**)− delay between reports (in seconds)
local−tx−speed (*integer*; default: **0**)− transfer test maximum speed (bits per second)

- **0** − no speed limitations

password (*string*; default: "") − password for remote user
protocol (udp | tcp; default: **tcp**)− protocol to use
remote−tx−speed (*integer*; default: **0**)− receive test maximum speed (bits per second)

- **0** − no speed limitations

size (*50..1500;* default: **512**) − packet size in bytes (only for UDP protocol)
user (*name*; default: "") − remote user

Example

To run 15−second long bandwidth−test to the **10.0.0.211** host sending and receiving **1000**−byte UDP packets and using username **admin** to connect

```
[admin@MikroTik] tool> bandwidth−test 10.0.0.211 duration=15s direction=both \
\... size=1000 protocol=udp user=admin
                 status: done testing
                 duration: 15s
              tx−current: 3.62Mbps
    tx−10−second−average: 3.87Mbps
        tx−total−average: 3.53Mbps
              rx−current: 3.33Mbps
    rx−10−second−average: 3.68Mbps
        rx−total−average: 3.49Mbps

[admin@MikroTik] tool>
```

Dynamic DNS (DDNS) Update Tool

Document revision 1.3 (30−Dec−2003)
This document applies to the MikroTik RouterOS V2.7

Contents of the Manual

Summary

Dynamic DNS Update Tool gives a way to keep domain name pointing to dynamic IP address. It works by sending domain name system update request to name server, which has a zone to be updated. Secure DNS updates are also supported. TSIG (transport signature) with HMAC−MD5 algorithm is used in this case to authenticate updates.

Also **note** that the clock of both parties (the router and the DDNS server) must not be different more than by 5 minutes. The update will be ignored otherwise.

Specifications

Packages required : ***advanced−tools***
License required : *Any*
Home menu level : ***/tool***
Protocols utilized : *Dynamic Updates in the DNS (RFC 2136), Secure DNS Dynamic Update (RFC 3007)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading

Description

Dynamic DNS Update is a tool that should be manually run to update dynamic DNS server

Note that you have to have a DNS server that supports DNS updates properly configured

Dynamic DNS Update

Command level : **/tool dns−update**

Property Description

address (*IP address*) − defines IP address associated with the domain name
dns−server (*IP address*) − DNS server to send update to
key (*string*; default: "") − authorization key (password of a kind) to access the server
key−name (*string*; default: "") − authorization key name (username of a kind) to access the server
name (*string*) − name to attach with the IP address
ttl (*integer*; default: **0**) − time to live for the item (in seconds)
zone (*string*) − DNS zone where to update the domain name in

Notes

To delete a DDNS entry that has not yet been timed out, you should add the same entry with **ttl** of **0**>.

Example

To tell **23.34.45.56** DNS server to (re)associate **mydomain** name in the **myzone.com** zone with **68.42.14.4** IP address specifying that the name of the key is **dns−update−key** and the actual key is **update**:

```
[admin@MikroTik] tool> dns−update dns−server=23.34.45.56 name=mydomain \
\... zone=myzone.com address=68.42.14.4 key−name=dns−update−key key=update
```

Additional Resources

DNS related RFCs

ICMP Bandwidth Test

Document revision 1.0 (28−Apr−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

The ICMP Bandwidth Tester (Ping Speed) can be used to approximately evaluate the throughput to **any** remote computer and thereby help to discover network 'bottlenecks'.

Specifications

Packages required : ***advanced−tools***
License required : *Any*
Home menu level : ***/tool***
Standards and Technologies : *ICMP* (RFC792)
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Log Management

ICMP Bandwith Test

Command name : **/tool ping−speed**

Description

The ICMP test uses two standard echo−requests per second. The time between these pings can be changed. Ping packet size variation makes it possible to approximately evaluate connection parameters and speed with different packet sizes. Statistics for throughput is calculated using the size of the ICMP packet, the interval between ICMP echo−request and echo−reply and the differences between parameters of the first and the second packet.

Property Description

do (*name*) − assigned name of the script to start
first−ping−size (*integer: 32..64000*; default: **32**) − first ICMP packet size
second−ping−size (*integer: 32..64000*; default: **1500**) − second ICMP packet size
time between pings (*integer*) − the time between first and second ICMP echo−requests in seconds. A new ICMP−packet pair will never be sent before the previous pair is completely sent and the algorithm itself will never send more than two requests in one second
once (option) − specifies that the ping will be performed only once
interval (*time: 20ms..5s*) − time interval between two ping repetitions

Example

In the following example we will test the bandwith to a host with an IP address **159.148.60.2**. The interval between repetitions will be **1** second.

```
[admin@MikroTik] tool> ping-speed 159.148.60.2 interval=1s
    current: 2.23Mbps
    average: 2.61Mbps

[admin@MikroTik] tool>
```

Packet Sniffer

Document revision 1.6 (02−May−2003)
This document applies to the MikroTik RouterOS v2.7

Table Of Contents

Summary

Packet sniffer is a feature that catches all the data travelling over the network, that it is able to get (when using switched network, a computer may catch only the data addressed to it or is forwarded through it)

Specifications

Packages required : *None*
License required : *Any*
Home menu level : ***/tool sniffer***
Protocols utilized : *none*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading

Description

It allows you to "sniff" packets going through the router (and any other traffic that gets to the router, when there is no switching in the network) and view them using specific software.

Packet Sniffer Configuration

Submenu level : **/tool sniffer**

```
[admin@MikroTik] tool sniffer> print
           interface: all
        only-headers: no
        memory-limit: 10
           file-name: ""
          file-limit: 10
    streaming-enabled: no
     streaming-server: 0.0.0.0
        filter-stream: yes
      filter-protocol: ip-only
      filter-address1: 0.0.0.0/0:0-65535
      filter-address2: 0.0.0.0/0:0-65535
              running: no
[admin@MikroTik] tool sniffer>
```

Property Description

interface (*name* | all; default: **all**) − the name of the interface receives the packets
only-headers (yes | no; default: **no**) − whether to save in the memory packets' headers only (not the whole packet)
memory-limit (*integer*; default: **10**) − maximal amount of memory to use. Sniffer will stop after this limit is reached
file-name (*string*; default: "") − the name of the file that the sniffed packets are to be saved to
file-limit (*integer*; default: **10**) − the limit of the file in KB. Sniffer will stop after this limit is reached
streaming-enabled (yes | no; default: **no**) − whether to send sniffed packets to a remote server
streaming-server (*IP address*; default: **0.0.0.0**) − Tazmen Sniffer Protocol (TZSP) stream receiver
filter-stream (yes | no; default: **yes**) − whether to ignore sniffed packets that are destined to the stream server
filter-protocol (all-frames | ip-only | mac-only-no-ip; default: **ip-only**) − specific protocol group to filter:

- **mac-only-no-ip** − sniff non-IP packets only
- **all-frames** − sniff all packets
- **ip-only** − sniff IP packets only

filter-address1 (*IP address/mask:ports*; default: **0.0.0.0/0:0-65535**) − criterion of choosing the packets to process
filter-address2 (*IP address/mask:ports*; default: **0.0.0.0/0:0-65535**) − criterion of choosing the packets to process
running (yes | no; default: **no**) − if the sniffer is started then the value is **yes** otherwise **no**

Notes

filter−address1 and **filter−address2** are used to specify the two participients in communication (i.e. they will match only in the case one of them matches the source address, and the other one matches the destination address of a packet). These properties are taken in account only if **filter−protocol** is **ip−only**.

Not only **Ethereal** (http://www.ethereal.com) and **Packetyzer** (http://www.packetyzer.com) can receive the sniffer's stream but also MikroTik's program **trafr** (http://www.mikrotik.com/download.html) that runs on any IA32 Linux computer and saves received packets in **libpcap** file format.

Example

In the following example **streaming−server** will be added, streaming will be enabled, **file−name** will be set to *test* and packet sniffer will be started and stopped after some time:

```
[admin@MikroTik] tool sniffer>set streaming−server=10.0.0.241 \
\... streaming−enabled=yes file−name=test
[admin@MikroTik] tool sniffer> prin
            interface: all
         only−headers: no
         memory−limit: 10
            file−name: "test"
           file−limit: 10
    streaming−enabled: yes
     streaming−server: 10.0.0.241
        filter−stream: yes
      filter−protocol: ip−only
      filter−address1: 0.0.0.0/0:0−65535
      filter−address2: 0.0.0.0/0:0−65535
              running: no
[admin@MikroTik] tool sniffer>start
[admin@MikroTik] tool sniffer>stop
```

Running Packet Sniffer

Command name : **/tool sniffer start**, **/tool sniffer stop**, **/tool sniffer save**

Description

The commands are used to control runtime operation of the packet sniffer. The **start** command is used to start/reset sniffering, **stop** − stops sniffering. To save currently sniffed packets in a specific file **save** command is used.

Example

In the following example the packet sniffer will be started and after some time − stopped:

```
[admin@MikroTik] tool sniffer> start
[admin@MikroTik] tool sniffer> stop
```

Below the sniffed packets will be saved in the file named *test*:

```
[admin@MikroTik] tool sniffer> save file−name=test
```

```
[admin@MikroTik] tool sniffer> /file print
  # NAME                          TYPE         SIZE       CREATION-TIME
  0 test                          unknown      1350       apr/07/2003 16:01:52

[admin@MikroTik] tool sniffer>
```

Sniffed Packets

Submenu level : **/tool sniffer packet**

Description

The submenu allows to see the list of sniffed packets.

Property Description

data (*read−only*; *string*) − specified data inclusion in packets
dst−address (*read−only*; *IP address*) − IP destination address
fragment−offset (*read−only*; *integer*) − IP fragment offset
identification (*read−only*; *integer*) − IP identification
ip−header−size (*read−only*; *integer*) − the size of IP header
ip−packet−size (*read−only*; *integer*) − the size of IP packet
ip−protocol (ip | icmp | igmp | ggp | ipencap | st | tcp | egp | pup | udp | hmp | xns−idp | rdp | iso−tp4 | xtp | ddp | idrp−cmtp | gre | esp | ah | rspf | vmtp | ospf | ipip | encap) − the name/number of IP protocol

- **ip** − internet protocol
- **icmp** − internet control message protocol
- **igmp** − internet group management protocol
- **ggp** − gateway−gateway protocol
- **ipencap** − ip encapsulated in ip
- **st** − st datagram mode
- **tcp** − transmission control protocol
- **egp** − exterior gateway protocol
- **pup** − parc universal packet protocol
- **udp** − user datagram protocol
- **hmp** − host monitoring protocol
- **xns−idp** − xerox ns idp
- **rdp** − reliable datagram protocol
- **iso−tp4** − iso transport protocol class 4
- **xtp** − xpress transfer protocol
- **ddp** − datagram delivery protocol
- **idpr−cmtp** − idpr control message transport
- **gre** − general routing encapsulation
- **esp** − IPsec ESP protocol
- **ah** − IPsec AH protocol
- **rspf** − radio shortest path first
- **vmtp** − versatile message transport
- **ospf** − open shortest path first
- **ipip** − ip encapsulation
- **encap** − ip encapsulation

protocol (*read−only*; ip | arp | rarp | ipx | ipv6) − the name/number of ethernet protocol

- **ip** − internet ptotocol

- **arp** − address resolution protocol
- **rarp** − reverse address resolution protocol
- **ipx** − internet packet excange protocol
- **ipv6** − internet protocol next generation

size (*read−only*; *integer*) − size of packet
src−address (*IP address*) Source address
time (*read−only*; *time*) − time when packet arrived
tos (*read−only*; *integer*) − IP Type Of Service
ttl (*read−only*; *integer*) − IP Time To Live

Example

In the example below it's seen, how to get the list of sniffed packets:

```
[admin@MikroTik] tool sniffer packet> pr
  # TIME    INTERFACE SRC-ADDRESS             DST-ADDRESS            IP-.. SIZE
  0 0.12    ether1    10.0.0.241:1839         10.0.0.181:23 (telnet) tcp   46
  1 0.12    ether1    10.0.0.241:1839         10.0.0.181:23 (telnet) tcp   40
  2 0.12    ether1    10.0.0.181:23 (telnet)  10.0.0.241:1839        tcp   78
  3 0.292   ether1    10.0.0.181              10.0.0.4               gre   88
  4 0.32    ether1    10.0.0.241:1839         10.0.0.181:23 (telnet) tcp   40
  5 0.744   ether1    10.0.0.144:2265         10.0.0.181:22 (ssh)    tcp   76
  6 0.744   ether1    10.0.0.144:2265         10.0.0.181:22 (ssh)    tcp   76
  7 0.744   ether1    10.0.0.181:22 (ssh)     10.0.0.144:2265        tcp   40
  8 0.744   ether1    10.0.0.181:22 (ssh)     10.0.0.144:2265        tcp   76
-- more
```

Packet Sniffer Protocols

Submenu level : **/tool sniffer protocol**

Description

In this submenu you can see all kind of protocols that has been sniffed.

Property Description

bytes (*integer*)− total number of data bytes
protocol (ip | arp | rarp | ipx | ipv6) − the name/number of ethernet protocol

- **ip** − internet ptotocol
- **arp** − address resolution protocol
- **rarp** − reverse address resolution protocol
- **ipx** − internet packet excange protocol
- **ipv6** − internet protocol next generation

ip−protocol (ip | icmp | igmp | ggp | ipencap | st | tcp | egp | pup | udp | hmp | xns−idp | rdp | iso−tp4 | xtp | ddp | idrp−cmtp | gre | esp | ah | rspf | vmtp | ospf | ipip | encap) − the name/number of IP protocol

- **ip** − internet protocol
- **icmp** − internet control message protocol
- **igmp** − internet group management protocol
- **ggp** − gateway−gateway protocol
- **ipencap** − ip encapsulated in ip
- **st** − st datagram mode

- **tcp** − transmission control protocol
- **egp** − exterior gateway protocol
- **pup** − parc universal packet protocol
- **udp** − user datagram protocol
- **hmp** − host monitoring protocol
- **xns−idp** − xerox ns idp
- **rdp** − reliable datagram protocol
- **iso−tp4** − iso transport protocol class 4
- **xtp** − xpress transfer protocol
- **ddp** − datagram delivery protocol
- **idpr−cmtp** − idpr control message transport
- **gre** − general routing encapsulation
- **esp** − IPsec ESP protocol
- **ah** − IPsec AH protocol
- **rspf** − radio shortest path first
- **vmtp** − versatile message transport
- **ospf** − open shortest path first
- **ipip** − ip encapsulation
- **encap** − ip encapsulation

packets (*integer*) − the number of packets
port (*name*) − the port of TCP/UDP protocol
share (*integer*) − specific type of traffic compared to all traffic in bytes

Example

```
[admin@MikroTik] tool sniffer protocol> print
  # PROTOCOL IP-PR... PORT          PACKETS   BYTES   SHARE
  0 ip                              77        4592    100 %
  1 ip       tcp                    74        4328    94.25 %
  2 ip       gre                    3         264     5.74 %
  3 ip       tcp      22 (ssh)      49        3220    70.12 %
  4 ip       tcp      23 (telnet)   25        1108    24.12 %

[admin@MikroTik] tool sniffer protocol>
```

Packet Sniffer Hosts

Submenu level : **/tool sniffer host**

Description

The submenu shows the list of hosts that were participating in data excange you've sniffed.

Property Description

address (*read−only*; *IP address*) − the address of the host **peek−rate** (*read−only*; *integer/integer*) − the maximum data−rate received/transmitted
rate (*read−only*; *integer/integer*) − current data−rate received/transmitted
total (*read−only*; *integer/integer*) − total packets received/transmitted

Example

In the following example we'll see the list of hosts:

```
[admin@MikroTik] tool sniffer host> print
  # ADDRESS       RATE         PEEK-RATE           TOTAL
  0 10.0.0.4      0bps/0bps    704bps/0bps         264/0
  1 10.0.0.144    0bps/0bps    6.24kbps/12.2kbps   1092/2128
  2 10.0.0.181    0bps/0bps    12.2kbps/6.24kbps   2994/1598
  3 10.0.0.241    0bps/0bps    1.31kbps/4.85kbps   242/866

[admin@MikroTik] tool sniffer host>
```

Packet Sniffer Connections

Submenu level : **/tool sniffer connection**

Description

Here you can get a list of the connections have been watched during the sniffing time.

Property Description

active (*read-only*; yes | no) - if **yes** the find active connections
bytes (*read-only*; *integer*) - bytes in the current connection
dst-address (*read-only*; *IP address*) - destination address
mss (*read-only*; *integer*) - Maximum Segment Size
resends (*read-only*; *integer*) - the number of packets resends in the current connection
src-address (*read-only*; *IP address*) - source address

Example

The example shows how to get the list of connections:

```
[admin@MikroTik] tool sniffer connection> print
Flags: A - active
  #   SRC-ADDRESS       DST-ADDRESS             BYTES     RESENDS   MSS
  0 A 10.0.0.241:1839   10.0.0.181:23 (telnet)  6/42      60/0      0/0
  1 A 10.0.0.144:2265   10.0.0.181:22 (ssh)     504/252   504/0     0/0

[admin@MikroTik] tool sniffer connection>
```

Ping

Document revision 1.9 (30−Apr−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

Ping uses Internet Control Message Protocol (ICMP) Echo messages to determine if a remote host is active or inactive and to determine the round−trip delay when communicating with it.

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : */, /tool mac−server ping*
Protocols utilized : *ICMP (RFC792)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
ICMP Bandwidth Test
Traceroute MAC Telnet Server and Client

Description

Ping sends ICMP echo (ICMP type 8) message to the host and waits for the ICMP echo−reply (ICMP type 0) from that host. The interval between these events is called *round trip*. If the response (that is called *pong*) has not come until the end of the *interval*, we assume it has timed out. The second significant parameter reported is *ttl* (Time to Live). Is is decremented at each machine in which the packet is processed. The packet will reach its destination only when the *ttl* is greater than the number of routers between the source and the destination

The Ping Command

Command name: **/ping**

Property Description

(*IP address* | *MAC address*) − IP or MAC address for destination host
size (28...65535, default: **64**) − Size of the IP packet (in bytes, including the IP and ICMP headers)
do−not−fragment − If added, packets will not be fragmented
interval (10ms...5s, default: **1s**) − Delay between messages
count (*integer*, default: **0**) − How many times ICMP packets will be sent

- 0 − Ping continues till [Ctrl]+[C] is pressed

ttl (1...255, default: **255**) − Time To Live (TTL) value of the ICMP packet

Notes

If DNS service is configured, it is possible to ping by DNS address. To do it from **Winbox**, you should resolve DNS address first, pressing right mouse button over it address and choosing **Lookup Address**.

Packet size may not be greater than the interface's mtu. If 'pinging' by MAC address, minimal packet size iz **50**.

Only neighbour MikroTik RouterOS routers with MAC−ping feature enabled can be 'pinged' by MAC address.

Examples

```
[admin@MikroTik] > ping 159.148.60.2 count=5 interval=40ms size=64
159.148.60.2 64 byte pong: ttl=247 time=32 ms
159.148.60.2 64 byte pong: ttl=247 time=30 ms
159.148.60.2 64 byte pong: ttl=247 time=40 ms
159.148.60.2 pong timeout
159.148.60.2 64 byte pong: ttl=247 time=28 ms
5 packets transmitted, 4 packets received, 20% packet loss
round−trip min/avg/max = 28/32.5/40 ms
[admin@MikroTik] >
```

MAC Ping Server

Submenu level : **/tool mac−server ping**

```
[admin@MikroTik] tool mac−server ping> print
    enabled: yes
[admin@MikroTik] tool mac−server ping>
```

Property Description

enabled (yes | no) − whether MAC pings to this router are allowed

Example

To disable MAC pings:

```
[admin@MikroTik] tool mac-server ping> set enabled=no
[admin@MikroTik] tool mac-server ping> print
    enabled: no
[admin@MikroTik] tool mac-server ping>
```

Realtime Traffic Monitor (torch)

Document revision 1.2 (17−Apr−2003)
This document applies to the MikroTik RouterOS v2.7

Table Of Contents

Summary

Realtime traffic monitor may be used to monitor the traffic flow through an interface

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/tool***
Protocols utilized : *none*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading

Description

Realtime Traffic Monitor called also torch is used for monitoring traffic going through an interface. You can monitor traffic classified by protocol name, source address, destination address, port. **Torch** shows the protocols you have chosen and mean transmitted and received data rate for each of them.

The Torch Command

Command name : **/tool torch**

Property Description

interface (*name*) − the name of the interface to monitor
protocol (any | any−ip | icmp | igmp | ipip | ospf | pup | tcp | udp| *integer*) − the name or number of the proocol

- **any** − any ethernet or IP protocol
- **any−ip** − any IP protocol
  port (*name* | *integer*) − the name or number of the port
- **any** − any port
  source−address (*IP address/mask*) − source address and network mask to filter the traffic only with such an address, any source address: 0.0.0.0/0
  destination−address (*IP address/mask*) − destination address and network mask to filter the traffic only with such an address, any destination address: 0.0.0.0/0

Notes

If there will be specific port given, then only **tcp**, and **udp** protocols will be filtered i.e. the name of the **protocol** can be **any**, **any−ip**, **tcp**, **udp**.

Except TX and RX, there will be only the field you've specified in command line in the command's output (e.g. you will get **PROTOCOL** column only in case **protocol** property is explicitly specified).

Example

The following example monitors the traffic that goes through the **ether1** interface generated by **telnet** protocol:

```
[admin@MikroTik] tool> torch ether1 port=telnet
 SRC-PORT                    DST-PORT                    TX         RX
 1439                        23 (telnet)                 1.7kbps    368bps

[admin@MikroTik] tool>
```

To see what IP protocols are going through the **ether1** interface:

```
[admin@MikroTik] tool> torch ether1 protocol=any-ip
 PRO.. TX         RX
 tcp   1.06kbps   608bps
 udp   896bps     3.7kbps
 icmp  480bps     480bps
 ospf  0bps       192bps

[admin@MikroTik] tool>
```

To see what IP protocols are interacting with **10.0.0.144/32** host connected to the **ether1** interface:

```
[admin@MikroTik] tool> torch ether1 src-address=10.0.0.144/32 protocol=any
 PRO.. SRC-ADDRESS     TX         RX
 tcp   10.0.0.144      1.01kbps   608bps
 icmp  10.0.0.144      480bps     480bps

[admin@MikroTik] tool>
```

To see what tcp/udp protocols are going through the **ether1** interface:

```
[admin@MikroTik] tool> torch ether1 protocol=any-ip port=any
 PRO.. SRC-PORT                  DST-PORT                  TX         RX
 tcp   3430                      22 (ssh)                  1.06kbps   608bps
 udp   2812                      1813 (radius-acct)        512bps     2.11kbps
```

```
 tcp   1059                      139 (netbios-ssn)         248bps     360bps
[admin@MikroTik] tool>
```

Traceroute

Document revision 1.1 (31−Jan−2003)
This document applies to MikroTik RouterOS v2.7

Table of Contents

Summary

Traceroute determines how packets are being routed to a particular host

Specifications

Packages required : ***system***
License required : *Any*
Home menu level : ***/tool***
Standards and Technologies : *ICMP (RFC792), UDP(RFC768, Traceroute (RFC2925)*
Hardware usage : *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)
Firewall Filters and Network Address Translation (NAT)
ICMP Bandwidth Test
Ping

Description

Traceroute is a TCP/IP protocol−based utility, which allows user to determine how packets are being routed to a particular host. Traceroute works by increasing the time−to−live value of packets and seeing how far they get until they reach the given destination; thus, a lengthening trail of hosts passed through is built up.

Traceroute shows the number of hops to the given host address of every passed gateway. Traceroute utility sends packets three times to each passed gateway so it shows three timeout values for each gateway in ms.

The Traceroute Command

Command name : **/tool traceroute**

Property Description

(*IP address*) − IP address of the host you are tracing route to
port (*integer: 0..65535*) − UDP port number
protocol (UDP | ICMP) − type of protocol to use. If one fails (for example, it is blocked by a firewall) try the other
size (*integer: 28..1500*, default: **64**) − packet size in bytes
timeout (*time: 1s..8s*, default: **1s**) − response waiting timeout, i.e. delay between messages
tos (*integer: 0..255*, default: **0**) − Type Of Service − parameter of IP packet
use−dns (yes | no, default: **no**) − specifies whether to use DNS server, which can be set in **/ip dns** menu

Notes

Traceroute session may be stopped by pressing [Ctrl]+[C].

Example

To trace the route to 216.239.39.101 host using ICMP protocol with packet size of 64 bytes, setting ToS field to 8 and extending the timeout to 4 seconds:

```
[admin@MikroTik] tool> traceroute 216.239.39.101 protocol=icmp size=64 tos=8 timeout=4s
     ADDRESS                                    STATUS
   1 159.148.60.227       3ms      3ms      3ms
   2 195.13.173.221      80ms    169ms     14ms
   3 195.13.173.28        6ms      4ms      4ms
   4 195.158.240.21     111ms    110ms    110ms
   5 213.174.71.49      124ms    120ms    129ms
   6 213.174.71.134     139ms    146ms    135ms
   7 213.174.70.245     132ms    131ms    136ms
   8 213.174.70.58      211ms    215ms    215ms
   9 195.158.229.130    225ms    239ms       0s
  10 216.32.223.114     283ms    269ms    281ms
  11 216.32.132.14      267ms    260ms    266ms
  12 209.185.9.102      296ms    296ms    290ms
  13 216.109.66.1       288ms    297ms    294ms
  14 216.109.66.90      297ms    317ms    319ms
  15 216.239.47.66      137ms    136ms    134ms
  16 216.239.47.46      135ms    134ms    134ms
  17 216.239.39.101     134ms    134ms    135ms
[admin@MikroTik] tool>
```

SNMP Service

Document revision 1.4 (22−Oct−2003)
This document applies to the MikroTik RouterOS V2.7

Table of Contents

Summary

SNMP is a network protocol that allows managing many network devices from one location.

The MikroTik RouterOS supports:

- SNMPv1 only;
- Read−only access is provided to the NMS (network management system);
- User defined communities are supported;
- No Trap support.

Specifications

Packages required : ***system***, *ppp* (optional)
License required : *Any*
Home menu level : ***/snmp***
Protocols utilized : *SNMP (RFC1157)*
Hardware usage: *not significant*

Related Documents

Software Package Installation and Upgrading
IP Addresses and Address Resolution Protocol (ARP)

Description

Mikrotik implemetation of Simple Network Management Protocol (SNMP) provides a possibility to access the configuration and statistics from the remote location. Installation of the SNMP package makes the router an SNMP agent.

SNMP Setup

Submenu level : **/snmp**

```
[admin@MikroTik] snmp> print
     enabled: no
     contact: ""
    location: ""
[admin@MikroTik] snmp>
```

Property Description

enabled (yes | no, default: **no**) − whether the SNMP service is enabled
contact (*string*, default: "") − contact information for the NMS
location (*string*, default: "") − location information for the NMS
Example To enable the service, specifying some inforation:

```
[admin@MikroTik] snmp> set contact=Sysadmin−555−1212 location=MikroTik enabled=yes
[admin@MikroTik] snmp> print
     enabled: yes
     contact: Sysadmin−555−1212
    location: MikroTik
[admin@MikroTik] snmp>
```

SNMP Communities

Submenu level : **/snmp community**

Description

The community is like a 'username' for connecting to the SNMP agent. The default community for SNMP is **public**:

Property Description

name (*name*) − community name
address (*IP address/mask*, default: **0.0.0.0/0**) − allow only requests from these addresses
read−access (yes | no, default: **yes**) − whether the read access is enabled for the community

Example

To view the existing communities:

```
[admin@MikroTik] snmp community> print
  # NAME                                            ADDRESS            READ−ACCESS
```

```
  0 public                                          0.0.0.0/0          yes
[admin@MikroTik] snmp community>
```

To disable read access to the **public** community:

```
[admin@MikroTik] snmp community> print
  # NAME                                            ADDRESS            READ-ACCESS
  0 public                                          0.0.0.0/0          yes
[admin@MikroTik] snmp community> set public read-access=no
[admin@MikroTik] snmp community> print
  # NAME                                            ADDRESS            READ-ACCESS
  0 public                                          0.0.0.0/0          no
```

To add the community called **communa**, that is only accessible from the **159.148.116.0/24** network:

```
[admin@MikroTik] snmp community> add name=communa address=159.148.116.0/24
[admin@MikroTik] snmp community> print
  # NAME                                            ADDRESS            READ-ACCESS
  0 public                                          0.0.0.0/0          no
  1 communa                                         159.148.116.0/24   no
[admin@MikroTik] snmp community>
```

Available MIBs

Mikrotik RouterOS OID: enterprises.14988.1

MIB objects supported

RFC1493

dot1dBridge.dot1dBase.dot1dBaseBridgeAddress
dot1dBridge.dot1dStp.dot1dStpProtocolSpecification
dot1dBridge.dot1dStp.dot1dStpPriority
dot1dBridge.dot1dTp.dot1dTpFdbTable.dot1dTpFdbEntry.dot1dTpFdbAddress
dot1dBridge.dot1dTp.dot1dTpFdbTable.dot1dTpFdbEntry.dot1dTpFdbPort
dot1dBridge.dot1dTp.dot1dTpFdbTable.dot1dTpFdbEntry.dot1dTpFdbStatus

RFC2863

ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInOctets
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInUcastPkts
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutOctets
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutUcastPkts

RFC1213

interfaces.ifNumber
interfaces.ifTable.ifEntry.ifIndex
interfaces.ifTable.ifEntry.ifDescr
interfaces.ifTable.ifEntry.ifType
interfaces.ifTable.ifEntry.ifMtu
interfaces.ifTable.ifEntry.ifSpeed

interfaces.ifTable.ifEntry.ifPhysAddress
interfaces.ifTable.ifEntry.ifAdminStatus
interfaces.ifTable.ifEntry.ifOperStatus
interfaces.ifTable.ifEntry.ifLastChange
interfaces.ifTable.ifEntry.ifInOctets
interfaces.ifTable.ifEntry.ifInUcastPkts
interfaces.ifTable.ifEntry.ifInNUcastPkts
interfaces.ifTable.ifEntry.ifInDiscards
interfaces.ifTable.ifEntry.ifInErrors
interfaces.ifTable.ifEntry.ifInUnknownProtos
interfaces.ifTable.ifEntry.ifOutOctets
interfaces.ifTable.ifEntry.ifOutUcastPkts
interfaces.ifTable.ifEntry.ifOutNUcastPkts
interfaces.ifTable.ifEntry.ifOutDiscards
interfaces.ifTable.ifEntry.ifOutErrors
interfaces.ifTable.ifEntry.ifOutQLen

RFC2011

ip.ipForwarding
ip.ipDefaultTTL
ip.ipAddrTable.ipAddrEntry.ipAdEntAddr
ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex
ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask
ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr
ip.ipAddrTable.ipAddrEntry.ipAdEntReasmMaxSize
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaIfIndex
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaPhysAddress
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaNetAddress
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType

RFC2096

ip.ipForward.ipCidrRouteNumber
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteDest
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteMask
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteTos
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteNextHop
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteIfIndex
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteType
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteProto
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteAge
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteInfo
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteNextHopAS
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteMetric1
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteMetric2
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteMetric3
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteMetric4
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteMetric5
ip.ipForward.ipCidrRouteTable.ipCidrRouteEntry.ipCidrRouteStatus

Note thet obsolete ip.ipRouteTable is also supported.

RFC1213

system.sysDescr
system.sysObjectID
system.sysUpTime
system.sysContact
system.sysName
system.sysLocation
system.sysServices

RFC2790

host.hrSystem.hrSystemUptime
host.hrSystem.hrSystemDate
host.hrStorage.hrMemorySize
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageIndex
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageType
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageDescr
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageAllocationUnits
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageSize
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageUsed

CISCO−AAA−SESSION−MIB

Note that this MIB is supported only when **ppp** package is installed. It reports both ppp and hotspot active users.
enterprises.cisco.ciscoMgmt.ciscoAAASessionMIB.casnMIBObjects.casnActive.casnActiveTableEntries
enterprises.cisco.ciscoMgmt.ciscoAAASessionMIB.casnMIBObjects.casnActive.casnActiveTable.casnActiveEntry.ca
enterprises.cisco.ciscoMgmt.ciscoAAASessionMIB.casnMIBObjects.casnActive.casnActiveTable.casnActiveEntry.ca
enterprises.cisco.ciscoMgmt.ciscoAAASessionMIB.casnMIBObjects.casnActive.casnActiveTable.casnActiveEntry.ca

MIB objects reported as '0'

RFC2863

ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifInMulticastPkts
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifInBroadcastPkts
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifOutMulticastPkts
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifOutBroadcastPkts
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInMulticastPkts
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInBroadcastPkts
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutMulticastPkts
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCOutBroadcastPkts
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHighSpeed

RFC2790

host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageAllocationFailures

Tools for SNMP Data Collection and Analysis

MRTG (Multi Router Traffic Grapher) is the most commonly used SNMP monitor.

http://ee−staff.ethz.ch/~oetiker/webtools/mrtg/

Example of using MRTG with Mikrotik SNMP

Here is a example configuration file for MRTG to monitor network card traffic on Mikrotik RouterOS This file was created with MRTG v2.9.17 cfgmaker on a linux computer. This is a only an example file.

MRTG Sample Configuration

For more information read the MRTG documentation: Configuration Reference

Additional Resources

http://www.ietf.org/rfc/rfc1157.txt
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm