MikroTik RouterOS™ V2.6 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

MikroTik RouterOS™ V2.6 Basic Setup Guide

PDF version

Introduction

Document revision 29−Nov−2002
This document applies to the MikroTik RouterOS™ V2.6

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).

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

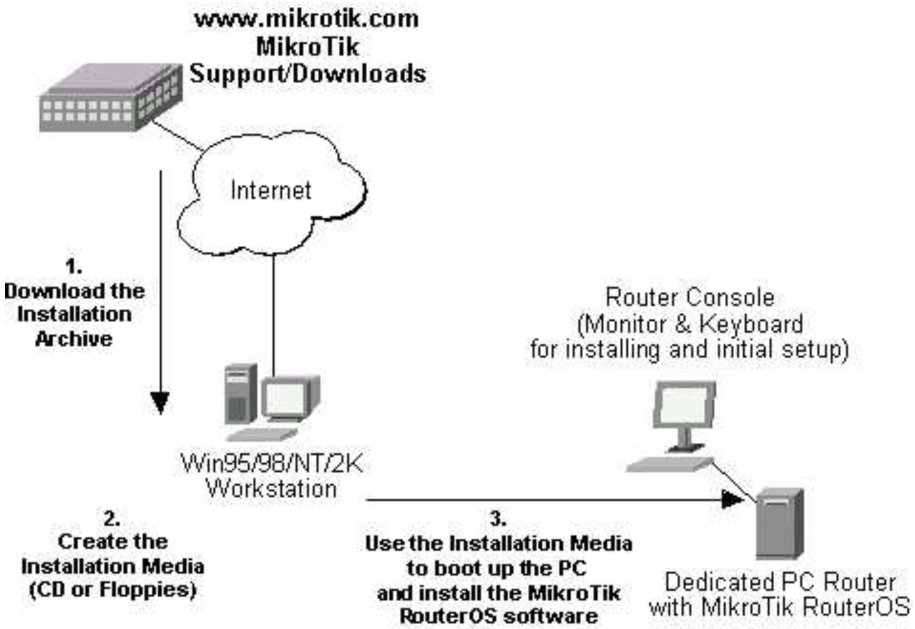
- Remote control with easy real−time Windows application (WinBox)
- Telnet/console/serial console control
- Advanced bandwidth control
- Network firewall with packet−filtering, masquerading, network address translation, logging and connection monitoring
- DHCP support
- HotSpot technology
- Ethernet 10/100/1000Mb/s
- Wireless client and AP 2.4GHz 11 Mb/s
- V.35 synchronous 5Mb/s with frame−relay
- Asynch PPP/RADIUS (up to 32 ports) for modem pools
- Cyclades and LMC DS3 with 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™. The following sections are included in this Guide:

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_v2−6−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_v2−6−x_dd−mmm−yyyy.exe file, which should be run on your Win95/98/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.
- **MikroTik Disk Maker in a set of smaller files**, if you have problems downloading one large file.
- **Netinstall**, if you want to install RouterOS™ over a LAN with one floppy boot disk

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

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** (dual processors are not supported);
- from 32MB to 1GB **RAM** (from 48MB suggested);
- 30MB or more PRIMARY MASTER **IDE HDD or IDE flashdrive**. Note: The hard disk will be entirely reformatted during the installation and all data on it will be lost!
- A **network adapter** (NE2000 compatible PCI or ISA Ethernet card, or any other supported NIC, see specifications of supported interfaces on our web page);

Note that 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

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.

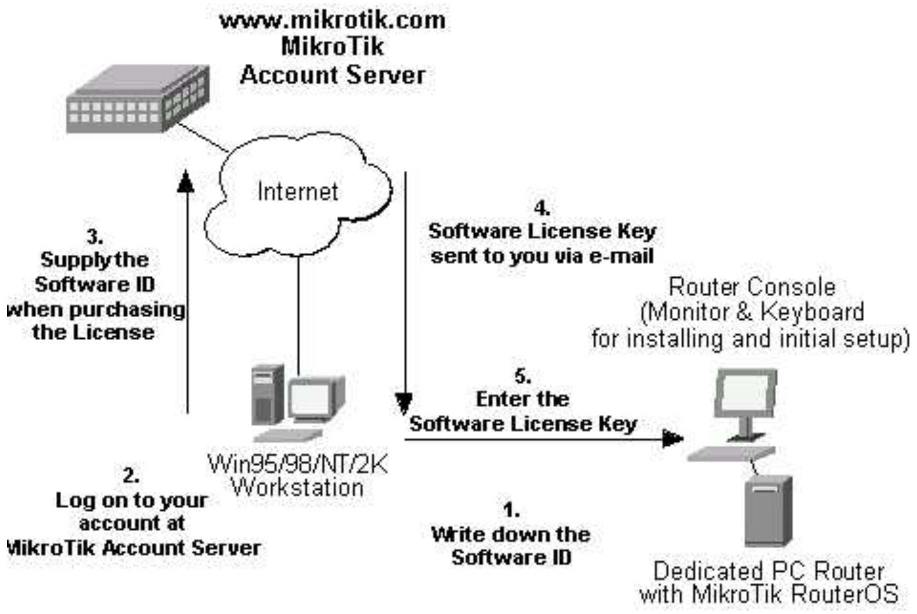
Boot up your dedicated PC router from the Installation Media you created and follow the instructions on the console screen while the HDD is reformatted and MikroTik RouterOS™ installed on it.

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.

Obtaining the Software License

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

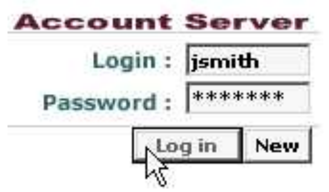


After installing the router and starting it up for the first time you will be given a Software ID.

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.

   > ***Note!*** The CD or Netinstall installation cannot be 'unlocked' with the Free Demo Key. Use the Floppy installation, or, purchase the License Key.
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.

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 v2.6
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 wont 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 to a 'paid' version of your MikroTik RouterOS™ installation, 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: "SB5T-R8T"
                 key: "3YIY-ZV8-DH2"
    upgradable-unitl: may/01/2003
[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=D45G-IJ6-QM3
[admin@MikroTik] system license> /system reboot
Reboot, yes? [y/N]: y
system will reboot shortly
```

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

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 v2.6 (c) 1999-2002       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:

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

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
           redo  Redo previosly undone action
          setup  Do basic setup of system
           undo  Undo previous action
           user  User management
            ppp
           snmp  snmp settings
  isdn-channels  ISDN channel status info
             ip
          queue  Bandwidth management
         system  System information and utilities
           tool
        routing
         export

[admin@MikroTik] > ip ?

      accounting  Traffic accounting
         address  Address management
             arp  ARP entries management
             dns  DNS settings
        firewall  Firewall management
       neighbour  neighbours
         packing  Packet packing settings
            pool  IP address pools
           route  Route management
```

```
          service
  policy-routing
     dhcp-client  DHCP client settings
     dhcp-server  DHCP server settings
       dns-cache
           ipsec
       web-proxy  HTTP proxy
       telephony  IP Telephony interface
          export
[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 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 arguments, that are required, may have no name. Below is a 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 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
```

However, if the netmask argument is not specified, 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

The MikroTik router can be accessed remotely using

- the **telnet** protocol, for example, using the telnet client of your Windows or Unix workstation. Working with the telnet console is the same as working with the monitor and keyboard attached to the router locally.
- the **ftp** for uploading the software upgrade packages or retrieving the exported configuration files.
- the **http** and **WinBox Console**, for example, using the web browser of your workstation.

Overview

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

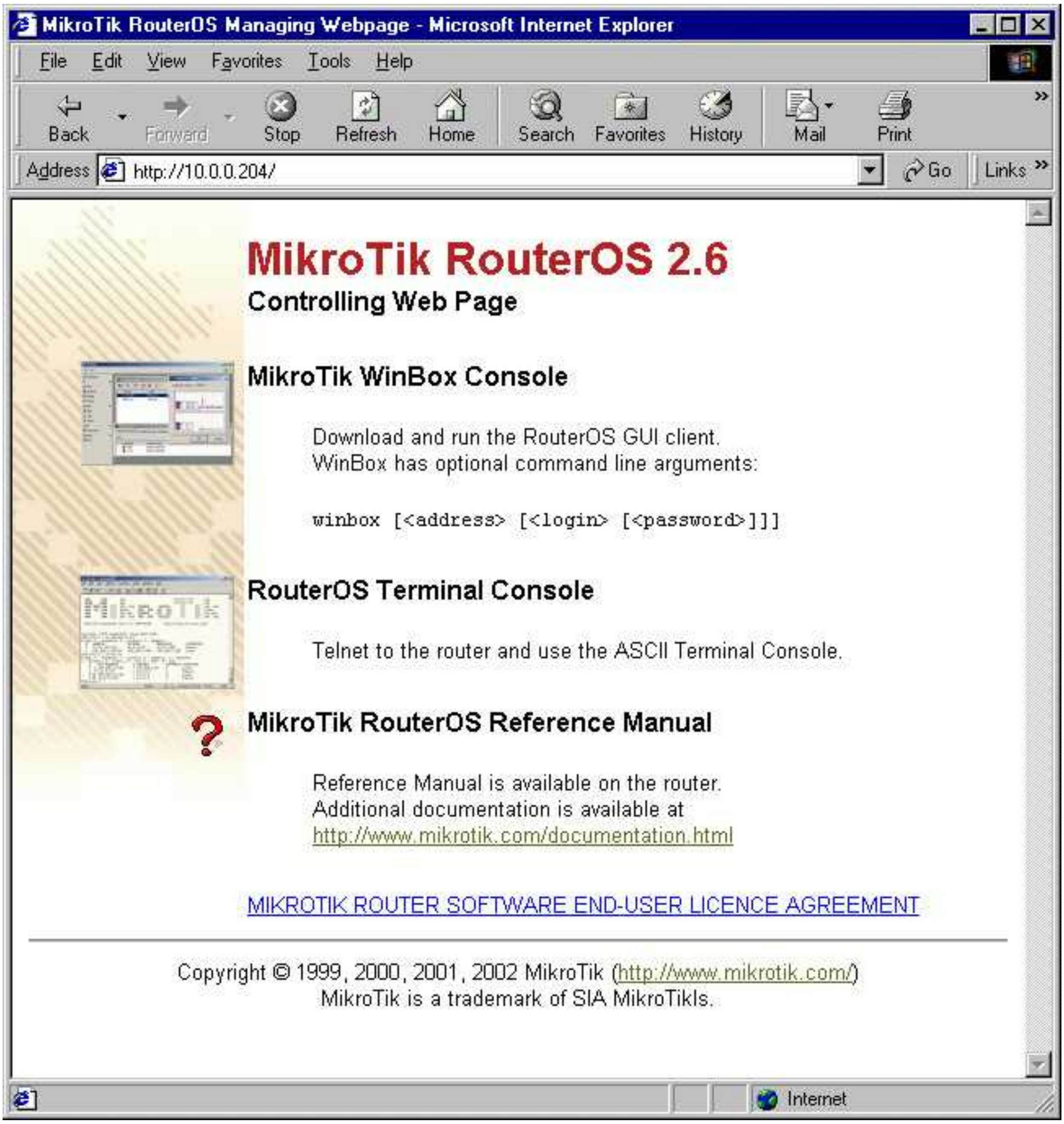
All Winbox interface functions are as close as possible to Console functions: all Winbox functions are exactly in the same place in Terminal Console and vice versa (except functions that are not implemented in Winbox). That is why there are no Winbox sections in the manual.

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.

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), the router's Welcome Page is displayed in the web browser, for example:

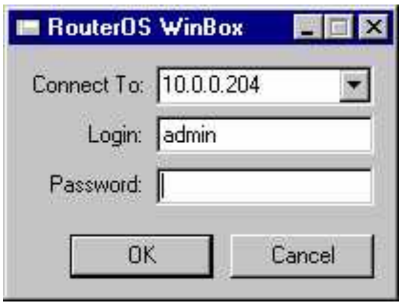


By clicking on the Winbox Console link you can start the winbox.exe download. Choose the option "Run this program from its current location" and click "OK":

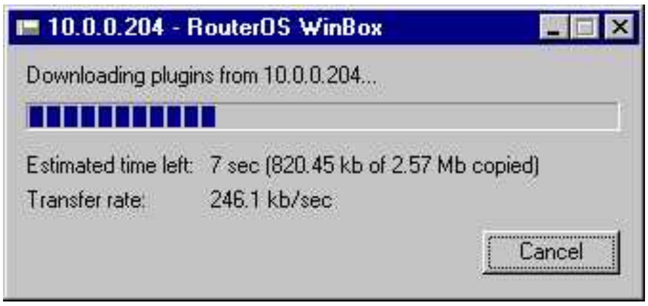
Accept the security warning, if any:

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

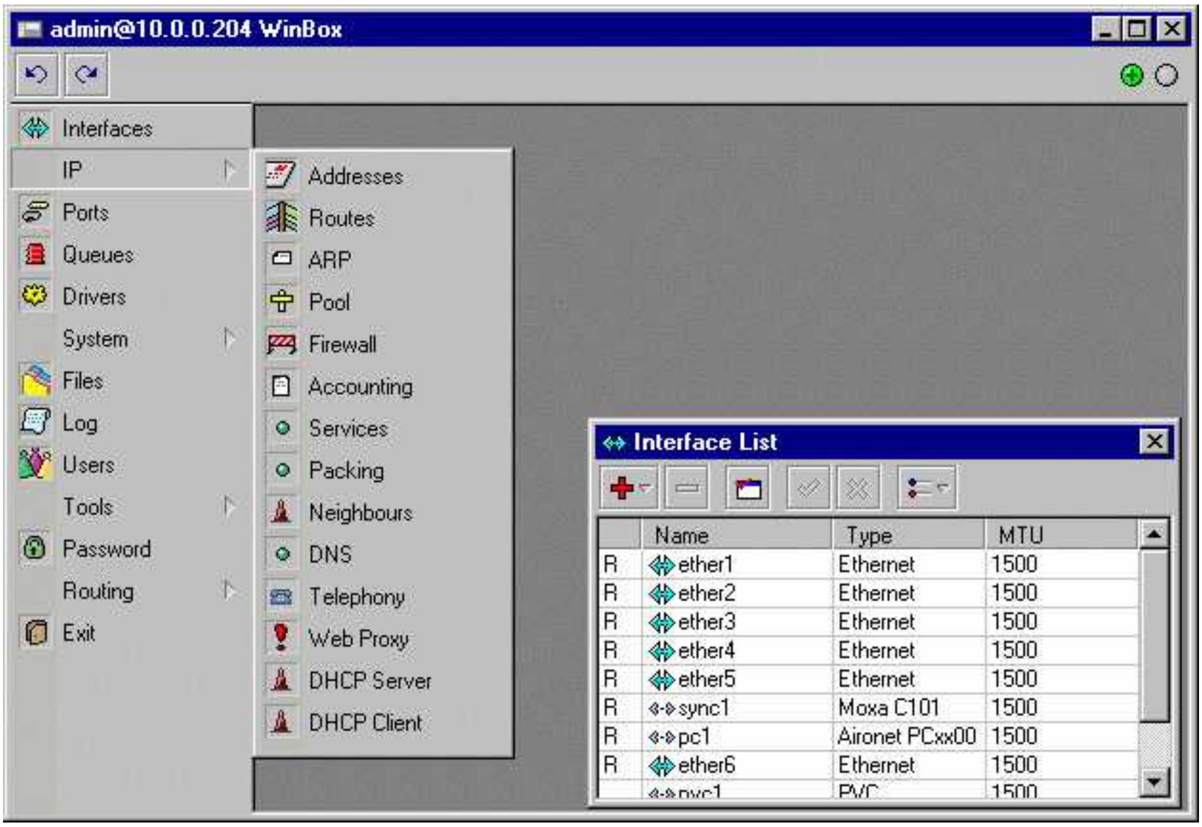
The winbox.exe program opens the Winbox login window. Login to the router by specifying the IP address, user name, and password, for example:



Watch the download process of Winbox plugins:



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



The Winbox Console uses TCP port 3987. 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.

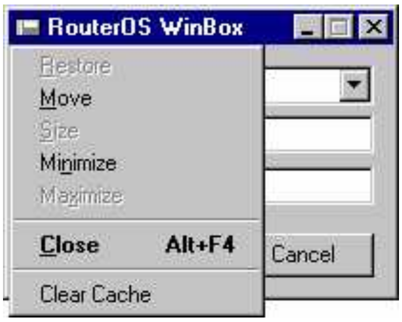
There are some hints for using the Winbox Console:

- 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 the 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

- *Cannot get the MikroTik RouterOS™ Winbox to start. The "Missing RouterOS Winbox plugins" message is displayed.*
  You can try to clear the winbox cache or wipe out the cache folder, and then reload the plugins:
    - ♦ To clear the winbox plugin cache on your computer, choose the Clear Cache option in the Winbox system menu of the login window:



    - ♦ To wipe out the winbox plugin cache on your computer, find the cache file location using the registry Key="HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ShellFolders\AppData"
      For example, for the user 'Administrator' on W2K, the Winbox folder is under C:\Documents and Settings\Administrator\Application Data\Mikrotik
      On W95/98 the Winbox folder is under C:\Windows\Application Data\Mikrotik
- *I still cannot open the Winbox Console*
  The Winbox Console uses TCP port 3987. 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 PCI Ethernet 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 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>
```

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, 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>
```

You can use the number or the name of the interface in the **enable** command.

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

```
[admin@MikroTik] interface> set 0 name=Public
```

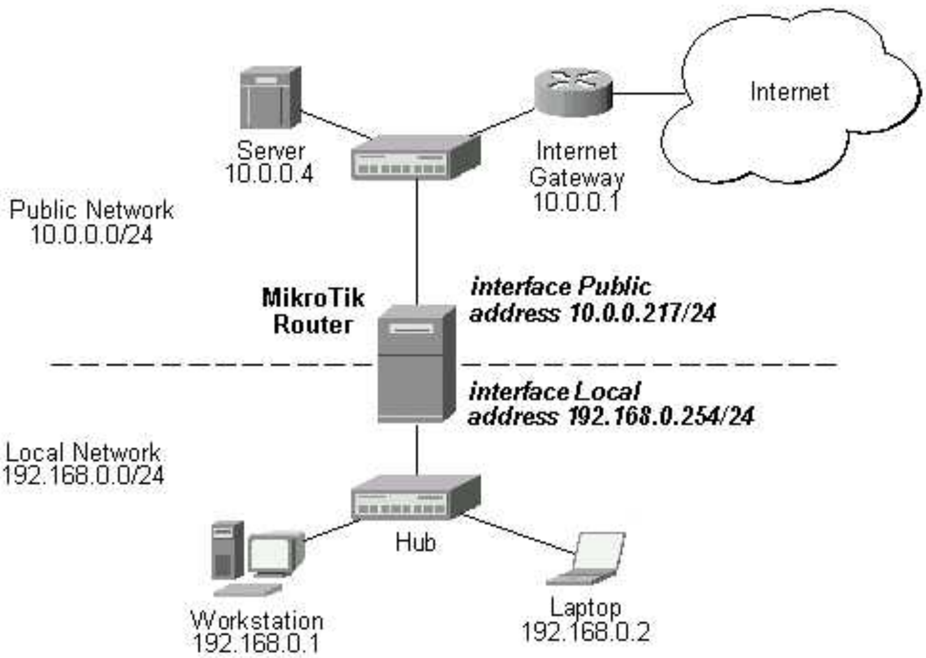
```
[admin@MikroTik] interface> set 1 name=Local
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
  #   NAME                 MTU   TYPE
  0  R Public               ether            1500
  0  R Local                ether            1500
[admin@MikroTik] interface>
```

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.

Adding Addresses

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



Please note that the addresses assigned to different interfaces of the router should belong to different networks. 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 192.168.0.254/24 interface Local
[admin@MikroTik] ip address> add address 10.0.0.217/24 interface Public
[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.

Configuring the Default Route

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

```
[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. This is done by adding the **default route** (destination 0.0.0.0, netmask 0.0.0.0). 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. Note, that 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. **Do not remove the dynamic (D) routes!** They are added automatically and should not be deleted 'by hand'. If you happen to, then reboot the router, the route will show up again.

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:

```
[admin@MikroTik] ip route> /ping 10.0.0.4
10.0.0.4 64 byte pong: ttl=255 time=7 ms
10.0.0.4 64 byte pong: ttl=255 time=5 ms
10.0.0.4 64 byte pong: 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 pong: ttl=255 time<1 ms
192.168.0.1 64 byte pong: ttl=255 time<1 ms
192.168.0.1 64 byte pong: ttl=255 time<1 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0/0.0/0 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:\>
```

You cannot access anything beyond the router (network 10.0.0.0/24 and the Internet), unless you do 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.

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 Examples

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
  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=""
      limit-count=0 limit-burst=0 limit-time=0s action=masquerade
      to-src-address=0.0.0.0 to-src-port=0-65535 bytes=0 packets=0

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

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

Application Example with Bandwidth Management

Mikrotik RouterOS™ V2.6 offers extensive queue management. For information on queue management, please refer to the relevant manual.

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:

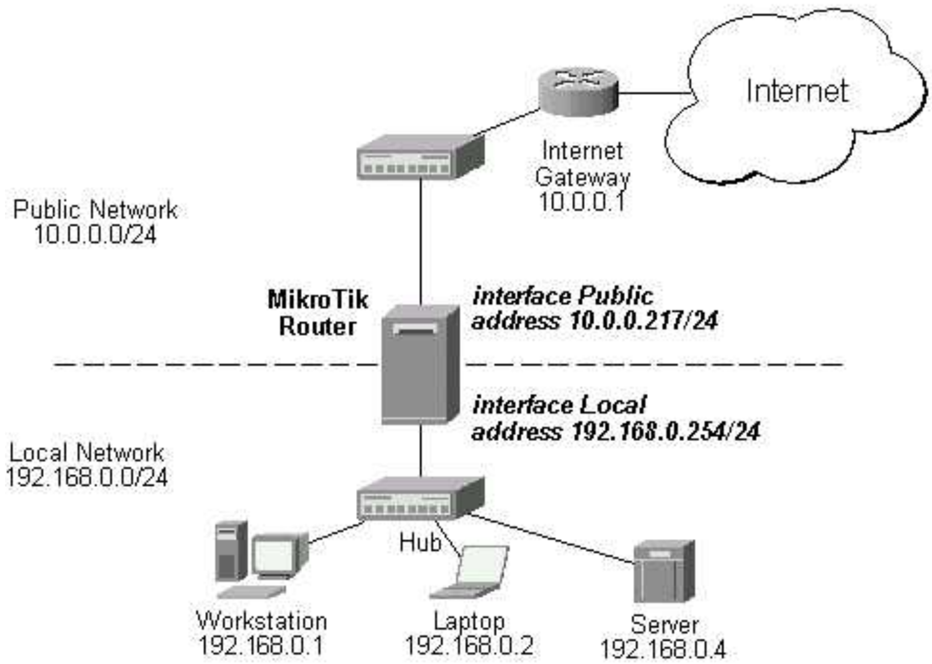
```
[admin@MikroTik] queue simple> add interface Local limit-at 128000
[admin@MikroTik] queue simple> add interface Public limit-at 64000
[admin@MikroTik] queue simple> print
Flags: X - disabled, I - invalid
  0   name="" src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Local
      limit-at=128000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

  1   name="" src-address=0.0.0.0/0 dst-address=0.0.0.0/24 interface=Public
      limit-at=64000 queue=default priority=8 bounded=yes

[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.

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
  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=""
      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>
```

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

MikroTik RouterOS™ V2.6 Reference Manual

PDF version (for printing)

Document revision 21−Jan−2003
This document applies to the MikroTik RouterOS™ V2.6

Terminal Console Manual

Document revision 29−Nov−2002
This document applies to the MikroTik RouterOS v2.6

Overview

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.

Contents of the Manual

The following topics are covered in this 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 the **..** twice:

```
[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' parameters. 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 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 is also migtht 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**.

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.

remove

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

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>
```

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.

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] >
```

Scripting Manual

Document revision 29−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

Scripting gives the administrator a way to execute console commands by writing a script for the router which is executed on the basis of time or events that can be monitored on the router. Some examples of uses of scripting could be: setting bandwidth settings according to time. In RouterOS v2.6, 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

To write a script, the writer must learn all of the 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.

Contents of the Manual

Scripts

The scripts are stored under **/system script**. Use the **add** command to add a new script. The following example is a script for writing message "kuku" to the system log:

```
[admin@MikroTik] system script> add name=log−test source={:log message=kuku}
[admin@MikroTik] system script> print
  0 name="log−test" source=":log message=kuku" owner=admin run−count=0

[admin@MikroTik] system script>
```

Argument description:

name − name of the script to be referenced when invoking it. If not specified, the name is

generated automatically as "scriptX", X=1,2,...
source − the script itself
owner − user's name who created the script
run−count − usage counter. This counter is incremented each time the script is executed, it can be reset to zero by setting 'run−counter=0'
last−started − date and time when the script has been last invoked. The argument is shown only if the 'run−count=0'.

Note that the counters will reset after reboot.

You can execute a script by using the **run** command.

To manage the active or scheduled tasks, use the **/system script job** menu. You can see the status of all currently active tasks using the **print** command. 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=kuku" owner=admin
    last−started=may/09/2001 03:22:19 run−count=1

  1 name="DelayeD" source=":delay 10m" owner=admin run−count=0

[admin@MikroTik] system script> run DelayeD
[admin@MikroTik] system script> job print
  # SCRIPT                                                  STARTED
  0 DelayeD                                                 may/09/2001 03:32:18
[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=kuku" owner=admin
    last−started=may/09/2001 03:36:44 run−count=3

  1 name="DelayeD" source=":delay 10m" owner=admin
    last−started=may/09/2001 03:32:18 run−count=1

[admin@MikroTik] system script>
```

Network Watching Tool

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 ability to issue arbitrary console commands on host state changes. Here's an example configuration of netwatch. It will run the scripts gw_1 or gw_2 which change the default gateway depending on the status of one of the gateways:

```
[MikroTik] system script>
add name=gw_1 source={/ip route set [/ip route find dst 0.0.0.0] gateway 10.0.0.1}
add name=gw_2 source={/ip route set [/ip route find dst 0.0.0.0] gateway 10.0.0.217}
[MikroTik] system script> /tool netwatch
add host=10.0.0.217 interval=10s timeout=998ms up−script=gw_2 down−script=gw_1
[MikroTik] tool netwatch> print
Flags: X − disabled
```

```
  #   HOST            TIMEOUT              INTERVAL             STATUS
  0   10.0.0.217      997ms                10s                  up
[MikroTik] tool netwatch> print detail
Flags: X - disabled
  0   host=10.0.0.217 timeout=997ms interval=10s since=mar/22/2002 11:21:03
      status=up up-script=gw_2 down-script=gw_1

[MikroTik] tool netwatch>
```

Argument description:

host − IP address of host that should be monitored
interval − Time between pings. Lowering this will make state changes more responsive, but can create unnecessary traffic and consume system resources.
timeout − Timeout for each ping. If no reply from host is received in this time, host is considered unreachable (**down**).
up−script − Console script that is executed once when state of host changes from **unknown** or **down** to **up**.
down−script − Console script that is executed once when state of host changes from **unknown** or **up** to **down**.
since − Time when state of host changed last time.
status − tells the current status of the host (up / down / unknown). State of host changes to **unknown** when any properties of this list entry are changed, or it is enabled or disabled. Also, any entry that is added has state **unknown** initially.

Hint: Scripts are not printed by default, to see them, type **print detail**.

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:

```
[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"}
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"}
[MikroTik] system script>
[MikroTik] system script> /tool netwatch
[MikroTik] system script>
```

```
add host=10.0.0.215 timeout=999ms interval=20s \
up-script=e-up down-script=e-up
[MikroTik] tool netwatch> print detail
Flags: X - disabled
  0   host=10.0.0.215 timeout=998ms interval=20s since=mar/22/2002 14:07:36
      status=up up-script=e-up down-script=e-up

[MikroTik] tool netwatch>
```

Writing Scripts

Console scripting introduction

Although 2.6 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 console 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

Console commands in 2.6 are made from the following parts:

```
PREFIX PATH PATH_ARGUMENT COMMAND NAMELESS_ARGUMENTS ARGUMENTS
```

first, few 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"
```

Here are explanations for each part of command:

PREFIX is either '/' or ':'. It is optional
PATH is a sequence of command level names and '..'. It is also optional, but the processing of

commands without given path may change in future versions; so, in your scripts, use path that starts with prefix ('/' or ':') whenever possible

PATH_ARGUMENT is required by some command levels (like **/ip firewall rule**), and is not allowed anywhere else

COMMAND is command name from the command level specified by path

NAMELESS_ARGUMENTS are specific to each command. Values of these arguments are written in fixed order after name of command, and only after all nameless argument values any named arguments can be given

ARGUMENTS are sequence of argument names (like **/user print brief without−paging**). For arguments that take values, argument name is followed by '=', followed by value of argument

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.

Grouping level commands

It is possible to execute several commands from the same command level, by grouping them with '{}'. For example:

```
[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>
```

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**.

Variables

Console allows 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.

```
[admin@MikroTik] ip route> :put $a
ERROR: unknown variable a
[admin@MikroTik] ip route>
```

Before using variable in script, it's name must be introduced. There are several ways to do that:

- With **:global**. It introduces name of global variable, which is created if it doesn't exist already.

  ```
  [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] >
  ```

  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.
- With **:local**. It introduces new local variable, which is not shared with any other script, other instance of the same script, other console logins. It's value is lost when script finishes or when variable name is freed by **:unset**.

  ```
  [admin@MikroTik] > :local l1
  [admin@MikroTik] > :set l1 "this is local variable"
  [admin@MikroTik] > :put $l1
  this is local variable
  [admin@MikroTik] >
  ```
- With **:for** and **:foreach** commands, which introduce loop index variable. It's valid only in the **do=** block of commands and is removed after command completes.

  ```
  [admin@MikroTik] > :for l1 from=1 to=3 do={:put $l1}
  1
  2
  3
  [admin@MikroTik] > :put $l1
  this is local variable
  [admin@MikroTik] >
  ```

  See how loop variable "shadows" already introduced local variable l1. It's value is not overwritten by **:for** loop.
- **monitor** commands, that have **do=** argument. See details below.

  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.

Changing variable values

You can assign new value to variable using **:set** command. It has two unnamed arguments. First is name of variable. Second is the new value of variable.

```
[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] >
```

Variable must contain integer number value, otherwise these commands will fail.

Command substitution, return values

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 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. 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** commands, 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.

Expressions

Console can do a simple math with numbers, time values, ip addresses, and strings and lists. It is done by writing expressions, putting them in parentheses '(' and ')'.

```
[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>
```

Supported operations are

- **!** − logical negation

  Unary operation. Argument is a truth value. Result is an opposite truth value.

  ```
  [admin@MikroTik] interface> :put (!true)
  false
  [admin@MikroTik] interface> :put (!(2>3))
  true
  [admin@MikroTik] interface>
  ```
- **−** − unary minus

  Unary operation. Argument and result is a number.

  ```
  [admin@MikroTik] interface> :put (−1<0)
  true
  [admin@MikroTik] > :put (−−1)
  1
  ```
- **~** − bit inversion

  Unary operations. Inverts bits in IP address.

  ```
  [admin@MikroTik] interface> :put (~255.255.0.0)
  0.0.255.255
  [admin@MikroTik] interface>
  ```
- **+** − sum

  Add together two numbers, two time values, or add number to an IP address.

  ```
  [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>
```

- **−** − difference

  Subtract one number from another, one time value from another. Subtracting a number from IP address gives IP address. Subtracting one IP address from another gives number.

```
[admin@MikroTik] interface> :put (10.0.0.15 + 10)
10.0.0.25
[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

  Multiply two numbers, or multiply a time value by a number.

```
[admin@MikroTik] interface> :put (12s * 4)
48s
[admin@MikroTik] interface> :put (−5 * −2)
10
[admin@MikroTik] interface>
```

- **/** − division

  Divide one number by another (gives an integer), or a time value by a number (gives time value).

```
[admin@MikroTik] interface> :put (10s / 3)
3s333.333ms
[admin@MikroTik] interface> :put (5 / 2)
2
[admin@MikroTik] interface>
```

- **<** − less
  > − more
  <= − less or equal
  >= − more or equal

  Compare two numbers, two time values, or two IP addresses. Gives truth value.

```
[admin@MikroTik] interface> :put (10.0.2.3<=2.0.3.10)
false
[admin@MikroTik] interface> :put (100000s>27h)
true
[admin@MikroTik] interface>
```

- **!=** − not equal
  = − equal

  Compare two values of the same type. Arrays are equal if their respective elements are equal.

```
[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

  Logical operation on two truth values. Result of **&&** is true, if both operands are true. Result of **||** is true if either operand is true.

  ```
  [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

  Bitwise operations on two IP addresses. Result is also an IP address.

  ```
  [admin@MikroTik] interface> :put (10.16.0.134 & ~255.255.255.0)
  0.0.0.134
  [admin@MikroTik] interface>
  ```
- << − shift left
  >> − shift right

  Shift IP value left or right by given amount of bits. First argument is IP address, second argument is integer. Result is IP address.

  ```
  [admin@MikroTik] interface> :put (~((0.0.0.1 << 7) − 1))
  255.255.255.128
  [admin@MikroTik] interface>
  ```
- **..** − concatenation

  Paste together two strings, or append one list to another, or append an element to a list.

  ```
  [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

Console can work with several types of values. Currently it distinguishes between strings, truth values (also known as booleans), 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 value:

- list
- internal number
- number
- ip address
- time value
- truth value
- string value

There is no way to explicitly control this type conversion, but it will most likely change 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 sd90039
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 are 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] >
```

Truth 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 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 "day day day"
3d
[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)

Colon commands

Console has many built−in commands that start with ':' prefix. They don't change configuration directly, but are most useful for writing scripts. You can see list of all such commands by pressing '?' after typing just 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
          log
[admin@MikroTik] > :
```

:local, **:global**, **:unset**, **:set**, **:incr** and **:decr** commands are explained in the section about variables. Here all the other commands will be explained.

- **:put** − takes only one, unnamed argument. It is displayed on screen. Cannot be used in scripts, because scripts don't have anywhere to display values on to.
- **:if** −This is a conditional, or branching command. It has one unnamed argument which must be a condition, that is, an expression that must return truth 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 "gateway unreachable"}
10.0.0.1 pong timeout
1 packets transmitted, 0 packets received, 100% packet loss
gateway unreachable
[admin@MikroTik] >
```

There are four loop control commands in console. They all have **do** argument, which is the 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** commands. If result is not a truth value, error is reported. If the result of condition is **true**, commands are executed once, and the condition is evaluated again, and this is repeated until condition returns **false**
- **:do** − It has one unnamed argument, which is the console commands that must be executed. It is similar to the **do** argument of other commands. If no other arguments are given, **:do** just executes this command 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**, commands are 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 it.

However, **:do A if=B** and **:if B do=A** do exactly the same thing.

* **:for** − It has one unnamed argument, the name of loop variable. **from** argument is the starting value for loop counter, **to** value is the final value. This commands counts loop variable up or down starting at **from** and ending with **to** (inclusive), and for each value it executes the **do** commands. It is possible to change the increment from the default 1 (or −1), by specifying **step** argument.

```
[admin@MikroTik] > :for i from=100 to=1 step=−37 do={:put ($i . " − " . 1000 / $i)}
100 − 10
63 − 15
26 − 38
[admin@MikroTik] >
```

* **:foreach** − The unnamed argument is the name of the loop variable. **in** argument is treated as a list. Each value of this list in sequence is assigned to loop variable, and **do** commands are executed for this value. If **in** value is not a list, **do** commands are executed just once, with this value in the loop variable. If **in** value is empty, **do** commands are not executed at all. This is made to work good with **find** commands, 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 in 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] >
```

Monitor commands

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. If **do** argument is present, monitor command will execute given script after each time it prints stats on the screen, and it will assign all printed values to local variables with the same name:

```
[admin2@kzd] > /interface
[admin2@kzd] 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
[admin2@kzd] interface>
```

Monitor command with **do** argument can also be called directly from scripts. It will not print anything then, but just execute the given script.

Get commands

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 list of items, first argument is name or internal number of item. Second argument is a name of item's property which should be returned.

```
[admin2@kzd] interface> :put [/interface get ether1  disabled ]
true
[admin2@kzd] interface>
```

If command level has general settings, **get** command only takes the name of property:

```
[admin2@kzd] interface> :put [/system clock get time ]
oct/23/2002 01:44:39
[admin2@kzd] interface>
```

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 above). You can use tab key completions to see what properties any particular **get** command can return.

More on syntax

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 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 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.

SSH Installation and Usage

Document revision 29−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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

Contents of the Manual

The following topics are covered in this manual:

Installation

The 'ssh−2.6.x.npk' (less than 1MB) package for installation of SSH is required. The package can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload it to the router with ftp and reboot. No additional settings are required. You may check to see if the SSH package is installed with the command **system package print**

Hardware Resource Usage

The uncompressed package will use approximately 1MB of additional Flash/HD IDE memory. A minimum amount of additional RAM is used. No hardware upgrades are required.

Suggested Windows Client Setup

PuTTY is a free Windows (all Windows) SSH client which needs no installation. It is one .exe file which can be downloaded and run.

Download this program from http://www.chiark.greenend.org.uk/~sgtatham/putty.html

Simple instructions:

1. After downloading, run the program,
2. Set the connection type to SSH,
3. On the first connection to the router a Security Alert will notify that the server's host is not in the registry. Answer 'YES' to trust this server.
4. The normal router login will not be display. Instead, 'login as:' and 'name@xxx.xxx.xxx.xxx's password:' will appear.

Suggested UNIX/Linux Client Setup

SSH client exists (and generally is istalled by default) for all standard Linux distributions. The command: **ssh –l [username] [router address]** will initiate a connection.

Winbox connections are encrypted (TSL) if ssh package is installed.

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&ID=11
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/

Software Package Installation and Upgrading

Document revision 29−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

The MikroTik RouterOS is residing on a formatted HDD specific to your installation and containing software packages. The main package is the **system** software package, which provides the basic functionality of the router. Additional software packages provide support for additional features (e.g., PPPoE, PPTP, PPP, wireless, etc).

Features

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

- Ability to add RouterOS functions by installing additional software packages
- Optimal usage of the storage space by the modular/compressed system
- The 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 installation of the package

Contents of the Manual

The following sections are included in this Manual:

Software Upgrade Instructions

Upgrade of the MikroTik RouterOS can be done by uploading the newer version software packages to the router and rebooting it. **Note!** 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 version of the MikroTik RouterOS system software (and the build number) are shown before the console login prompt, for example:

```
MikroTik v2.6beta4
Login:
```

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 package print
Flags: I - invalid
  #   NAME                  VERSION              BUILD-TIME           UNINSTALL
  0   system                2.6beta4             aug/09/2002 20:22:14 no
  1   rip                   2.6beta4             aug/09/2002 20:33:41 no
  2   ppp                   2.6beta4             aug/09/2002 20:28:01 no
  3   plist                 2.6beta4             aug/09/2002 20:32:58 no
  4   pppoe                 2.6beta4             aug/09/2002 20:29:18 no
  5   pptp                  2.6beta4             aug/09/2002 20:28:43 no
  6   ssh                   2.6beta4             aug/09/2002 20:25:31 no
  7   advanced-tools        2.6beta4             aug/09/2002 20:53:37 no
  8   bgp                   2.6beta4             aug/09/2002 20:34:22 no
  9   ospf                  2.6beta4             aug/09/2002 20:34:08 no
[admin@MikroTik] >
```

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:

```
[admin@MikroTik] > system package set 6 uninstall=yes
[admin@MikroTik] > system package print
Flags: I - invalid
  #   NAME                  VERSION              BUILD-TIME           UNINSTALL
  0   system                2.6beta4             aug/09/2002 20:22:14 no
  1   rip                   2.6beta4             aug/09/2002 20:33:41 no
  2   ppp                   2.6beta4             aug/09/2002 20:28:01 no
  3   plist                 2.6beta4             aug/09/2002 20:32:58 no
  4   pppoe                 2.6beta4             aug/09/2002 20:29:18 no
  5   pptp                  2.6beta4             aug/09/2002 20:28:43 no
  6   ssh                   2.6beta4             aug/09/2002 20:25:31 yes
  7   advanced-tools        2.6beta4             aug/09/2002 20:53:37 no
  8   bgp                   2.6beta4             aug/09/2002 20:34:22 no
  9   ospf                  2.6beta4             aug/09/2002 20:34:08 no
[admin@MikroTik] >
```

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

Software Package Installation Instructions

The software package files are compressed binary files, which can be downloaded from 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−2.6beta4.npk**, **ppp−2.6beta4.npk**, **pppoe−2.6beta4.npk**, etc. To install (upgrade) newer version of the MikroTik RouterOS system software please follow the upgrade instructions below:

- Check the availability of free HDD space on the router using the **/system resource print** command:

```
[admin@MikroTik] > system resource print
             uptime: 2d8h31m33s
        free-memory: 3328 kB
       total-memory: 29504 kB
                cpu: "WinChip"
           cpu-load: 0
     free-hdd-space: 5679 kB
    total-hdd-space: 46478 kB
[admin@MikroTik] >
```

Note! If there is not enough free disk space for storing the upgrade packages, disk space can be freed up by uninstalling some software packages, which provide functionality not required for your needs.

- If the free disk space is sufficient 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.
- 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.

Example output of the **/file print** command:

```
[admin@MikroTik] > file print
  # NAME                                TYPE    SIZE        CREATION-TIME
  0 ssh_host_key.pub                    unknown 332         jan/23/2002 18:45:02
  1 ssh_host_dsa_key.pub                unknown 603         jan/23/2002 18:45:08
  2 cyclades-2.6beta4.npk               package 114321      jan/31/2002 17:45:27
  3 framerelay-2.6beta4.npk             package 94632       jan/31/2002 17:45:29
[admin@MikroTik] >
```

The installation/upgrade process is shown on the console screen (monitor) attached to the router. After successful installation the software packages installed can be viewed using **/system package print** command.

Note!The versions of packages should match the version number of the system software package.

Contents of the Software Packages

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, 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 network monitor and support for other advanced tools	–	–
aironet	Provides support for CISCO Aironet IEEE 802.11b wireless PC/PCI/ISA cards	–	2.4GHz wireless
arlan	Provides support for DSSS 2.4GHz 2mbps Aironet ISA cards	–	2.4GHz wireless
atheros	Provides support for Atheros chipset based IEEE 802.11a wireless cards as clients or as access points	–	2.4GHz wireless (station mode);
			2.4GHz wireless and AP (AP mode)
bgp	Provides BGP support	–	–
cyclades	Provides support for PC300 synchronous interfaces	–	synchronous
ddns	Provides dynamic DNS support	–	–
dhcp	Provides DHCP server and client support	–	–
dns−cache	DNS cache	–	–
farsync	Provides support FarSync interfaces	–	synchronous
framerelay	Provides support for frame relay (used with Moxa C101, Cyclades PC300, or FarSync interfaces	–	–
hotspot	HotSpot gateway	–	any additional license
ipsec	Provides Ipsec support	–	–

isdn	Provides support for ISDN	ppp	–
lcd	Provides LCD monitor support	–	–
moxa−c101	Provides support for Moxa C101 synchronous card	–	synchronous
moxa−c502	Provides support for Moxa C502 synchronous card	–	synchronous
ntp	Provides network time protocol support	–	–
ospf	Provides OSPF support	–	–
plist	Provides Prefix List support for BGP and RIP	–	–
ppp	Provides asynchronous PPP support	–	–
pppoe	Provides PPPoE support	ppp	–
pptp	Provides PPTP support	ppp	–
prism	Provides support for Prism II chipset based IEEE 802.11b wireless cards as clients or as access points	–	2.4GHz wireless (station mode);
			2.4GHz wireless and AP (AP mode)
radiolan	Provides support for 5.8GHz RadioLAN ISA cards	–	radiolan
rip	Provides RIP support	–	–
snmp	Provides read only SNMP support	–	–
ssh	Provides remote access via SSH	–	–
telephony	Provides IP telephony support (H.323) for Quicknet cards	–	–
ups	Provides APC Smart Mode UPS support	–	–
vlan	Provides support fot IEEE 802.1Q Virtual LAN	–	–
wavelan	Provides support for Lucent WaveLAN IEEE 802.11 wireless cards	–	2.4GHz wireless
web−proxy	provides squid based web proxy support	–	–
xpeed	Provides support for Xpeed 300 SDSL cards	–	–

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] system license> print
      software−id: TPNG−SXN
              key: 2C6A−YUE−3H2
```

```
    upgradable-to: dec/01/2002
[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=D45G-IJ6-QM3
[admin@MikroTik] system license> /system reboot
Reboot, yes? [y/N]: y
system will reboot shortly
```

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.

Software Package Resource Usage

The following table shows the required resources of HDD storage and RAM for the various software packages. The total required storage space can be calculated by adding the together the required storage of all installed packages including the system software package.

Note that there are only minimal requirements needed to run the software. Additional resource usage is expected from many packages when they are configured and running (especially from **web−proxy**, **system** and **dns−cache**)

Name	Memory (RAM) usage, MB	Storage (HDD) usage, MB
advanced−tools	0.6	0.4
aironet	1.6	0.2
arlan	1.1	0.2
atheros	2.7	0.7
bgp	1.0	0.9
cyclades	1.5	0.2
ddns	0.4	0.2
dhcp	1.0	0.4
dns−cache	1.5	0.3
farsync	2.2	0.3
framerelay	0.8	0.2
hotspot	1.0	0.4
ipsec	3.0	0.8
isdn	2.7	0.9
lcd	1.9	0.3
moxa−c101	1.9	0.2
moxa−c502	2.0	0.1
ntp	1.3	0.4
ospf	2.1	0.8

plist	0.5	0.2
ppp	2.2	0.9
pppoe	0.2	0.3
pptp	1.0	0.3
prism	2.7	0.7
radiolan	2.0	0.3
rip	1.7	0.5
snmp	1.0	0.3
ssh	2.0	1.8
system	16.5	20.0
telephony	6.0	5.2
ups	0.9	0.3
vlan	1.9	0.2
wavelan	1.9	0.2
web−proxy	1.3	1.0
xpeed	1.8	0.2

Troubleshooting

- *Is it possible to upgrade from 2.5 to 2.6 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 <u>before</u> 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 V2.6*
  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.x has been upgraded to the version 2.6.y, 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™ V2.6 Specifications Sheet

Document revision 21−Nov−2002
This document applies to the MikroTik RouterOS™ V2.6

Hardware

CPU and motherboard − 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 (dual 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 − floppy drive, keyboard, monitor

Basic Network Platform

TCP/IP protocol suite

- **Firewall and NAT**
  packet filtering, source and destination NAT, source MAC, addresses, ports, protocols, protocol options, interfaces.
- **Routing**
  RIP 1 / 2, 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
- **Bandwidth Management**
  per IP / protocol / subnet / port, CBQ, RED, SFQ, byte limited queue, packet limited queue
- **Point−to−Point links**
  ISDN dial−out, ISDN dial−in, RADIUS authentication/accounting, onboard serial ports, PPTP encrypted tunnel (VPN), PPTP Access Concentrator, PPPoE client, PPPoE Access Concentrator (server), modem pool
- **Tunnels**
  IPIP tunnels, EoIP (Ethernet over IP)
- **IPsec**
  IP encryption (IPsec)
- **VLAN**
  Virtual LAN support
- **DHCP**
  DHCP server per interface, DHCP client
- **HotSpot**
  HotSpot Gateway
- **NTP**
  Network Time Protocol server and client
- **Monitoring/Accounting**
  IP traffic accounting, firewall actions logging
- **Tools**
  ping, traceroute, bandwidth test, ping flood, telnet
- **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
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.

Wireless Interfaces

(additional license purchase required)

- **2.4 GHz Wireless 802.11b clients**
  Aironet 4800 ISA/PCI/PC
  Cisco 340/352 ISA/PCI/PC

WaveLAN Bronze/Gold/Silver ISA/PC
Prism II chipset based cards

- **2.4 GHz Wireless 802.11b Access Point**
  Prism II chipset based cards
- **5.2 GHz Wireless 802.11a Access Points and clients**
  Atheros chipset based cards
- **5.8 GHz Wireless**
  10Mbps RadioLAN

Synchronous

(additional license purchase required)

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

Asynchronous Interfaces

- Standard Communication Ports Com1 and Com2
- Moxa Smartio C104H/C168H PCI 4/8 port up to 4 cards (32 ports)
- Cyclades Cyclom−Y and Cyclades−Z Series up to 32 ports per card, up to 4 cards
- 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

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.

Device Driver Management

Document revision 30−Sep−2002
This document applies to the MikroTik RouterOS V2.6

Overview

Device drivers represent the software interface part of installed network devices. For example, the MikroTik RouterOS includes device drivers for NE2000 compatible Ethernet cards and other network devices. Device drivers are included in the system software package and in the additional feature packages.

The device drivers for PCI and PC cards are loaded automatically. Other network interface cards (most ISA and ISDN PCI cards) require the device drivers loaded manually by 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 device driver for a device, which is not supported by the MikroTik RouterOS, please suggest it at our suggestion page on our web site.

Contents of the Manual

The following topics are covered in this manual:

Loading Device Drivers

The drivers for PCI and PCMCIA cards (except the ISDN cards) are loaded automatically at the system startup. Use the **/driver print** command to see the list of loaded drivers:

```
[admin@MikroTik] driver> print
Flags: I − invalid, D − dynamic
  #   DRIVER                       IRQ IO        MEMORY     ISDN−PROTOCOL
  0 D RealTek RTL8129/8139
[admin@MikroTik] driver>
```

As we see, the driver for the Realtek PCI card has been loaded automatically.

If the driver required to be loaded, use the **/driver add** command. The syntax of the command is:

```
[admin@MikroTik] > driver add
Load driver name [irq IRQ] [io IO range start] [mem shared memory].

      copy−from  item number
             io  IO port base address
            irq  IRQ number
  isdn−protocol  ISDN line protocol
         memory  Shared Memory base address
           name  Driver name
```

```
[admin@MikroTik] >
```

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:

```
[admin@MikroTik] driver> add name=?
Name of driver to load.

     3c509  3com 3c509 ISA
  ne2k-isa  ISA NE2000
[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 RTL8129/8139
  1   ISA NE2000                            280
[admin@MikroTik] driver>
```

To see the 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>
```

Note, that the resource list shows only the interfaces, if they are enabled!

Removing Device Drivers

Use the **/driver remove** command to remove device drivers. Unloading of device driver is useful when changing network devices − this can be useful to save system resources in avoiding loading drivers for devices, which have been removed from the system. Device driver needs to be removed and loaded again, if some parameter (memory range, i/o base address) has been changed for the adapter card. The device drivers can be removed only if the appropriate interface has been disabled.

Notes on PCMCIA Adapters

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

- Vadem VG−469 PCMCIA−ISA adapter,
- RICOH PCMCIA−PCI Bridge with R5C475 II or RC476 II chip (one or two PCMCIA ports)
- CISCO/Aironet PCMCIA adapter (ISA and PCA 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 there are a maximum for a number of PCMCIA ports − 8. If You will try to install 9 or more ports (no matter whether with one−port or two−port adapters, in any combination), no one will be recognized.

List of Drivers

The list of device drivers included in the system software package is given below:

ISA Drivers

Drivers for ISA cards should be loaded manually.

- **ne2k−isa**
  Load the driver by specifying the I/O base address. IRQ is not required.
  Driver is suitable for most of the NE2000 compatible ISA cards.
- **3c509**
  Load the driver by specifying the I/O base address. IRQ is not required.
  Driver is suitable for 3COM 509 Series ISA cards (3Com EtherLink III).

PCI Drivers

Drivers for PCI cards are loaded automatically, if the relevant interface card is installed, and it does not have hardware conflicts. The list of PCI drivers is below:

- **ne2k−pci**
  Driver is suitable for the Ethernet cards with RealTek RTL−8029 chip:
  RealTek RTL−8029
  Winbond 89C940 and 89C940F
  Compex RL2000

KTI ET32P2
NetVin NV5000SC
Via 86C926
SureCom NE34
Holtek HT80232
Holtek HT80229

- **3c95x**
  (3Com 3c590/3c900 series Vortex/Boomerang driver)
  This device driver is designed for the 3Com FastEtherLink and FastEtherLink XL, 3Com's PCI to 10/100baseT adapters. It also works with the 10Mbs versions of the FastEtherLink cards. The supported product IDs are:
  3c590, 3c592, 3c595, 3c597, 3c900, 3c905
  3c590 Vortex 10Mbps
  3c595 Vortex 100baseTx
  3c595 Vortex 100baseT4
  3c595 Vortex 100base−MII
  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 (unknown version)
  3Com Boomerang (unknown version)
- **eepro100**
  (Intel i82557/i82558/i82559ER/i82801BA−7 PCI EtherExpressPro driver)
  This device driver is designed for the Intel i82557 "Speedo3" chip, Intel's single−chip fast Ethernet controller for PCI, as used on the IntelEtherExpressPro 100 adapter
- **e1000**
  Intel PRO/1000 Desktop Adapter
  Intel PRO/1000 Server Adapter
- **tulip**
  This device driver is designed for the DECchip "Tulip", Digital's single−chip ethernet controllers for PCI. Supported members of the family are the 21040, 21041, 21140, 21140A, 21142, and 21143. Similar work−alike chips from Lite−On, Macronics, ASIX, Compex and other listed below are also supported:
  Interfaces: Digital DC21040 Tulip
  Digital DC21041 Tulip
  Digital DS21140 Tulip
  Digital DS21143 Tulip
  D−Link DFE 570TX and 580TX
  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
Xircom Tulip clone

- **rtl8139**
  This device driver is designed for the RealTek RTL8129, the RealTek Fast Ethernet controllers for PCI. This chip is used on a few clone boards:
  RealTek RTL8129 Fast Ethernet
  RealTek RTL8139 Fast Ethernet
  SMC1211TX EZCard 10/100 (RealTek RTL8139)
  Accton MPX5030 (RealTek RTL8139)
- **winbond−840**
  This driver is for the Winbond w89c840 chip:
  Winbond W89c840
  Compex RL100−ATX
- **dmfe**
  This driver is for:
  Davicom DM9102
  Davicom DM9102A
  Davicom DM9102A+DM9801
  Davicom DM9102A+DM9802

For the list of drivers included in additional feature software packages, please see the manual of the relevant software package.

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.
- *I have loaded the driver, but the interface does not show up.*
  Obtain the required software license to enable the functionality of the interface.

General Interface Settings

Document revision 23−Sep−2002
This document applies to the MikroTik RouterOS V2.6

Overview

MikroTik RouterOS supports a variety of Network Interface Cards and virtual interfaces, like VLAN interface, Bridge interface, etc. Current Manual describes general settings for MikroTik RouterOS interfaces.

Contents of the Manual

The following topics are covered in this manual:

Interface Status

Interface status can be shown using the **/interface print** command, for example

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

Here, the arguments are:

> **status** − (cannot be changed) shows the interface status. In order to use the interface, its status must be 'Running'.
> **name** − descriptive name of interface
> **type** − interface type
> **MTU** − maximum transmit unit for the interface in bytes.

You can monitor the traffic passing through any interface using the **/interface monitor** command:

```
[admin@MikroTik] interface> monitor−traffic ether6
    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>
```

You can monitor one or more interfaces at a time, for example:

```
[admin@MikroTik] interface> monitor−traffic ether2,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>
```

Interface Specific Settings

Specific interface configuration is under the **/interface _name_** submenu, for example:

```
[admin@MikroTik] interface ethernet> print detail
Flags: X − disabled, R − running
  0  R name="ether2" mtu=1500 mac−address=00:E0:C5:68:11:04 arp=enabled
       disable−running−check=yes

[admin@MikroTik] interface ethernet>
```

Argument description:

apr − 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 it, but neighbor MAC addresses will be gathered from **/ip arp** statically set table only.

disable−running−check − for 'broken' Ethernet cards it is good to disable running status checking (as default).

For almost all interfaces it is possible to monitor the interface status, for example:

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

[admin@MikroTik] interface ethernet>
```

Please see the relevant interface Manual for more information.

Atheros 5GHz 54Mbps Wireless Interface

Document revisions:
18−Jan−2003 V2.6.9 allows setting the 'supported−rate' to specific values.

This document applies to the MikroTik RouterOS V2.6

Overview

The MikroTik RouterOS supports the Atheros chipset based wireless adapter cards for working both as wireless clients (**station** mode) and wireless access points (**ap−bridge** or **bridge** mode).

For more information on the Atheros advantages, see:

- http://www.atheros.com/pt/index.html
- http://www.mt.lv/Documentation/manual_2.6/Interface/http://www.atheros.com/AtherosRangeCapacityPaper.pdf

For more information about adapter hardware please see the relevant User's Guides and Technical Reference Manuals of the hardware manufacturers.

Contents of the Manual

The following topics are covered in this manual:

Supported Network Roles

Wireless Client

The Atheros interface can be configured to act as an IEEE 802.11a wireless client (station) to associate with an access point.

Wireless Access Point

The Atheros interface can be configured to act as an IEEE 802.11a wireless access point. The access point can register wireless clients.

Wireless Bridge

This is limited version of the Access Point mode that allows only one client to be registered but does not require the 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.

Installation

The MikroTik Router should have the atheros software package installed. The software package file **atheros−2.6.x.npk** can be downloaded from MikroTik’s web page www.mikrotik.com. To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer. After successful installation the package should be listed under the installed software packages list.

License

The Atheros chipset based adapters, like 2.4GHz wireless adapters, require the 2.4GHz wireless feature license. One license is for one installation of the MikroTik RouterOS, disregarding how many cards are installed in one PC box. The wireless feature is not included in the Free Demo or Basic Software License. The 2.4GHz Wireless Feature cannot be obtained for the Free Demo License. It can be obtained only **together** with the Basic Software License.

Note! The **2.4GHz Wireless Feature License** enables only the **station** and the **bridge** modes of the Atheros card.
To enable the **access point mode**, additionally the **Wireless AP Feature License** is required.

The MikroTik RouterOS supports as many Atheros chipset based cards as many free adapter slots has your system. One license is valid for all cards on your system.

System Resource Usage

Atheros chipsets are used in PCI/miniPCI/CardBus cards and thus support IRQ sharing.

Installing the Wireless Adapter

The basic installation steps of the wireless adapter should be as follows:

1. Check the system BIOS settings and make sure you have the **PnP OS Installed** set to **Yes**.
2. The Atheros adapter should appear as **Network Adapter** in the list of by BIOS found devices during the system startup.

Note that it is recommended to use Atheros wireless cards in the systems with CPU speed higher than Celeron 600MHz or other equivalent.

Loading the Driver for the Wireless Adapter

PCI, miniPCI, PC (PCMCIA) and CardBus cards do not require a 'manual' driver loading, since they are recognized automatically by the system and the driver is loaded at the system startup.

Wireless Interface Configuration

If the driver has been loaded successfully, and you have the required Wireless Software License (same license is valid for 2.4GHz and 5GHz devices), then the Atheros Wireless interface should appear under the **/interface** list with the name atherosX, where X is 1,2,... You can change the interface name to a more descriptive one using the **set** command. To enable the interface, use the **enable** command:

```
[admin@MikroTik] > interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                 TYPE             MTU
  0  R ether1               ether            1500
  1 X  atheros1             atheros          1500
[admin@MikroTik] > interface enable 1
[admin@MikroTik] > interface print
Flags: X − disabled, D − dynamic, R − running
  #    NAME                 TYPE             MTU
  0  R ether1               ether            1500
  1  R atheros1             atheros          1500
[admin@MikroTik] >
```

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

```
[admin@MikroTik] interface atheros> print
Flags: X − disabled, R − running
  0  R name="atheros1" mtu=1500 mac−address=00:06:AB:00:37:8B
       arp=enabled mode=station root−ap=00:00:00:00:00:00
       frequency=5240MHz ssid="mikrotik"
       supported−rates=6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps
       basic−rates=6Mbps protocol=802.11−standard ack−time=100
       default−authentication=yes default−forwarding=yes
       max−clients=2007
[admin@MikroTik] interface atheros>
```

Argument description:

name − interface name (same as for other interfaces)
mtu − maximum transfer unit (same as for other interfaces)
mac−address − MAC address of card. In AP mode this will also be BSSID of BSS
arp − 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.

mode − mode of the interface:

- if **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
- if **ap−bridge**, card works as access point, i.e., it creates wireless infrastructure

root−ap − (only **ap−bridge** or **bridge**) MAC address of the root access point to register to
frequency − (only **ap−bridge** or **bridge**) frequency that AP will use to create BSS (**5180**, **5200**, **5220**, **5240**, **5260**, **5280**, **5300**, **5320**)
ssid − Service Set Identifier. In station mode − ssid to connect to, in AP − ssid to use when creating BSS (this can not be left blank)
default−authentication − (only **ap−bridge** or **bridge**) what to do with a client that wants to associate, but it is not in the access−list
default−forwarding − (only **ap−bridge** or **bridge**) what to do with a client that wants to

send packets to other wireless clients, but it is not in the access−list
max−clients − (only **ap−bridge** or **bridge**) maximum number of clients (including other access points), that is allowed to associate with this access point (1...2007)
supported−rates − Rates at which this node will work (6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps)
basic−rates − (only **ap−bridge** or **bridge**) Rates that every client that plans to connect to this AP should be able to work at
protocol − One of the:
- ♦ **802.11−standard** − timing settings as suggested by 802.11a standard
- ♦ **turbo−mode** − atheros turbo mode uses double the amount of radio frequency allowing faster speeds
- ♦ **ptp−turbo−mode** − atheros turbo mode with speed optimised timing settings to be used in ptp links

ack−time − time in microseconds to wait for ack packet for unicast transmissions, should be increased for long distance links (in standard mode 26 is fine). Maximum for 802.11a standard mode is 204 microseconds, maximum for the PTP Turbo and Turbo mode is 102 microseconds. For example, a 4km link works fine with ack−time=70

Station Mode Configuration

To set the wireless interface working with an IEEE 802.11a 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 **station**.

All other parameters can be left as default. To configure the wireless interface for registering to an AP with ssid "testing", it is enough to change the argument value of ssid to "testing" and to enable the interface:

```
[admin@MikroTik] interface atheros> set atheros1 ssid=testing
[admin@MikroTik] interface atheros> enable atheros1
[admin@MikroTik] interface atheros> pr
Flags: X − disabled, R − running
  0    name="atheros1" mtu=1500 mac−address=00:06:AB:00:37:8B
       arp=enabled mode=station root−ap=00:00:00:00:00:00
       frequency=5240MHz ssid="testing"
       supported−rates=6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps
       basic−rates=6Mbps protocol=802.11−standard ack−time=100
       default−authentication=yes default−forwarding=yes
       max−clients=2007
[admin@MikroTik] interface atheros>
```

New in V2.6.9:
You can limit the maximum data rate of a client depending on the RF link quality to, say, 36Mbps by specifying the client to work up to that rate. For that, set, for example, 'supported−rates=6Mbps,24Mbps,36Mbps'. Do not forget to include all 'basic−rates' of your access point (default is 6Mbps)!

Monitoring the Interface Status

In station mode, the atheros interface status can be monitored using the **/interface atheros monitor** command:

```
[admin@MikroTik] interface atheros> monitor atheros1
            status: connected−to−ess
         frequency: 5240MHz
           tx−rate: 36Mbps
```

```
           rx-rate: 9Mbps
              ssid: "testing"
             bssid: 00:06:AB:00:37:88
    signal-strength: 24
[admin@MikroTik] interface atheros>
```

Argument description:

status − status of the interface

- **searching−for−network** − the card has not registered to an AP and is searching for one to register to
- **authenticating** − the card is trying to authenticate on an AP
- **associating** − the card is trying to associate with an AP
- **connected−to−ess** − the card has registered to an AP

frequency − the frequency that is used for the connection
tx−rate − the actual transmitting data rate of the connection
rx−rate − the actual receiving data rate of the connection
ssid − the Service Set Identifier
bssid − the Basic Service Set Identifier (actually, the MAC address of the access point)
signal−strength − the signal strength

The monitor command does not work, if the interface is disabled, or the mode is **ap−bridge** or **bridge**.

Access Point Mode Configuration

To set the wireless interface working as an IEEE 802.11a access point (to register clients), you should set the following parameters:

- The **Service Set Identifier**. It should be unique for your system.
- The **Operation Mode** of the card should be set to **ap−bridge** or **bridge**
- 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 cover the **basic−rates** of the access point.

To configure the wireless interface for working as an access point with ssid "testing" and use the frequency 5240MHz, it is enough to enter the command:

```
[admin@MikroTik] interface atheros>
set atheros1 mode=ap-bridge frequency=5240MHz ssid=testing
[admin@MikroTik_AP] interface atheros> print
Flags: X - disabled, R - running
  0  R name="atheros1" mtu=1500 mac-address=00:06:AB:00:37:88 arp=enabled
       mode=ap-bridge root-ap=00:00:00:00:00:00 frequency=5240MHz ssid="testing"
       supported-rates=6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps
       basic-rates=6Mbps protocol=802.11-standard ack-time=100
       default-authentication=yes default-forwarding=yes max-clients=2007
[admin@MikroTik_AP] interface atheros>
```

Use the registration table to see the associated clients.

Registration Table

The registration table shows all clients currently associated with the access point, for example:

```
[admin@MikroTik_AP] interface atheros> registration-table print
  # INTERFACE   MAC-ADDRESS       TYPE      PARENT            SIGNAL     TX-...
  0 atheros1    00:40:63:C0:84:E7 local
  1 atheros1    00:06:AB:00:37:8B radio                       26         54Mbps
  2 atheros1    00:50:08:00:01:33 local
  3 atheros1    00:01:24:70:03:58 radio                       47         6Mbps
[admin@MikroTik_AP] interface atheros>
```

Argument description for the registration−table entry:

interface − interface that client is registered to
mac−address − mac address of the registered client
type − type of the client:

- **radio** − 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 − parent access point's MAC address, if forwarded from another access point
signal − current signal strength
tx−rate − the actual transmitting data rate of the connection

The **print stats** command give additional per−client statistics:

```
[admin@MikroTik_AP] interface atheros> registration-table print stats
  0 interface=atheros1 mac-address=00:40:63:C0:84:E7 type=local

  1 interface=atheros1 mac-address=00:06:AB:00:37:8B type=radio
    tx-rate=54Mbps rx-rate=54Mbps packets=182,192 bytes=17840,18642
    uptime=00:08:23.440 signal=26

  2 interface=atheros1 mac-address=00:50:08:00:01:33 type=local

  3 interface=atheros1 mac-address=00:01:24:70:03:58 type=radio tx-rate=6Mbps
    rx-rate=48Mbps packets=18,49 bytes=1764,4159 uptime=00:01:35.770
    signal=46

[admin@MikroTik_AP] interface atheros>
```

Additional argument description (only for wireless clients):

packets − number of received and sent packets
bytes − number of received and sent bytes
signal − signal strength
rx−rate − receive data rate
tx−rate − transmit data rate
uptime − time the client is associated with the access point

Access List

The access list is used 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 the following: when a new client wants to associate to the AP that is configured on interface atherosX, entry with client's MAC address and interface atherosX 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 atherosX is taken.

To add an access list entry, use the **add** command, for example:

```
[admin@MikroTik] interface atheros access−list>
add mac−address=00:06:AB:00:37:72 interface=atheros1
[admin@MikroTik] interface atheros access−list> print
Flags: X − disabled
  0   mac−address=00:06:AB:00:37:72 interface=atheros1 authentication=yes
      forwarding=yes

[admin@MikroTik] interface atheros access−list>
```

Argument description:

mac−address − MAC address of the client
interface − AP interface
authentication − accept this client when it tries to connect or not
forwarding − forward the client's frames to other wireless clients or not

If you have default authentication action for the interface set to **yes**, you can disallow this node to register at the AP's interface **atheros** by setting **authentication=no** for it. Thus, all nodes except this one will be able to register to the interface **atheros1**.

If you have default authentication action for the interface set to **no**, you can allow this node to register at the AP's interface **atheros1** by setting **authentication=yes** for it. Thus, only the specified nodes will be able to register to the interface **atheros1**.

Registering the Access Point to another Access Point

You can configure the access point to registering to another (root) access point by specifying the MAC address of the root access point:

```
[admin@MikroTik] interface atheros> set atheros1 root−ap=00:06:AB:00:37:75
[admin@MikroTik] interface atheros> print
Flags: X − disabled, R − running
  0  R name="atheros1" mtu=1500 mac−address=00:06:AB:00:37:8F arp=enabled
       mode=ap−bridge root−ap=00:06:AB:00:37:75 frequency=5180MHz ssid="testing"
       supported−rates=6−54 basic−rates=6 protocol=802.11−standard ack−time=26
       default−authentication=yes default−forwarding=yes max−clients=2007


[admin@MikroTik] interface atheros>
```

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 atheros and ethernet interfaces is used. Note, that in the station mode, bridging cannot be used between atheros and ethernet interfaces.

Troubleshooting

- *The atheros 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 atheros interface to get all nodes re−register.

- *The wireless card does not register to the AP*
  Check the cabling and antenna alignment. Check, if you have correct settings for 'supported−rates' and 'basic−rates'. The default 'supported−rates=6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps' and 'basic−rates=6Mbps' should work for all nodes on your system.
- *There is occasional packet loss when I ping the wireless client.*
  Packet loss is due to attempts to change the transmit data rate to a higher one. For lower quality RF links limit the maximum data rate of a client by specifying the 'supported−rate' argument, for example, set 'supported−rate=6Mbps,12Mbps,36Mbps'.

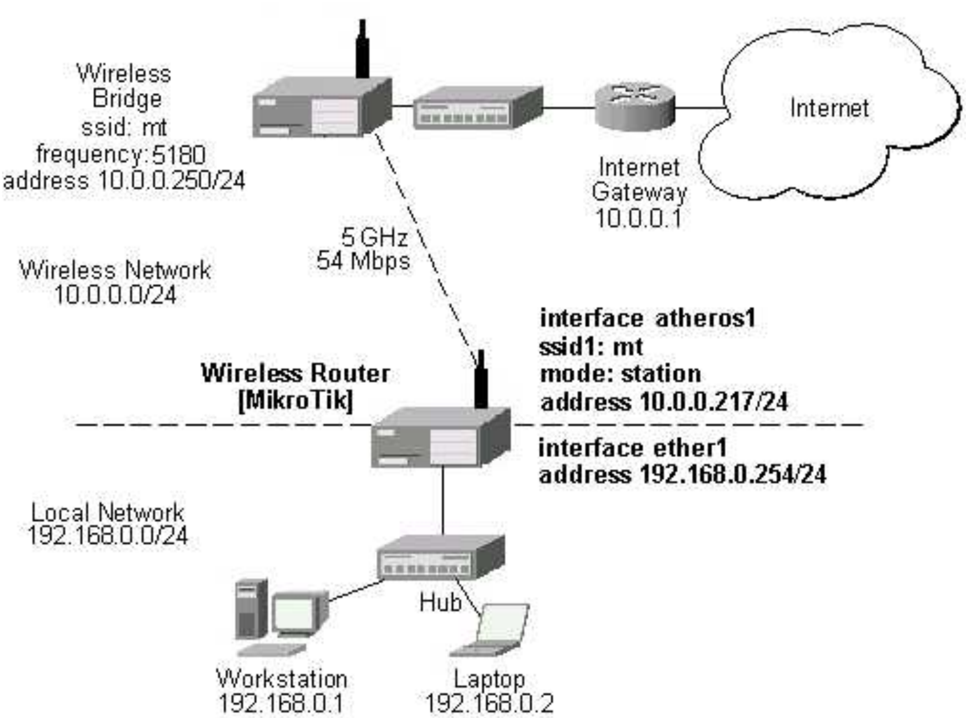
Wireless Network Applications

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

- Wireless Client
- Wireless Access Point
- Wireless Bridge

Wireless Client

Let us consider the following point−to−multipoint network setup with MikroTik with Atheros Wireless Interface in AP−bridge mode as a wireless bridge and MikroTik Wireless Router as a client:



The wireless bridge is connected to the wired network's HUB and has IP address from the network 10.0.0.0/24. See below for the wireless bridge configuration.

The minimum configuration for the MikroTik router's atheros 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 atheros> set 0 ssid=mt
[admin@MikroTik] interface atheros> monitor 0
```

```
             status: connected-to-ess
          frequency: 5180MHz
            tx-rate: 54Mbps
            rx-rate: 6Mbps
               ssid: "mt"
              bssid: 00:06:AB:00:37:8E
    signal-strength: 72

[admin@MikroTik] interface atheros>
```

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=atheros1
[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      atheros1
  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 10.0.0.250 !):

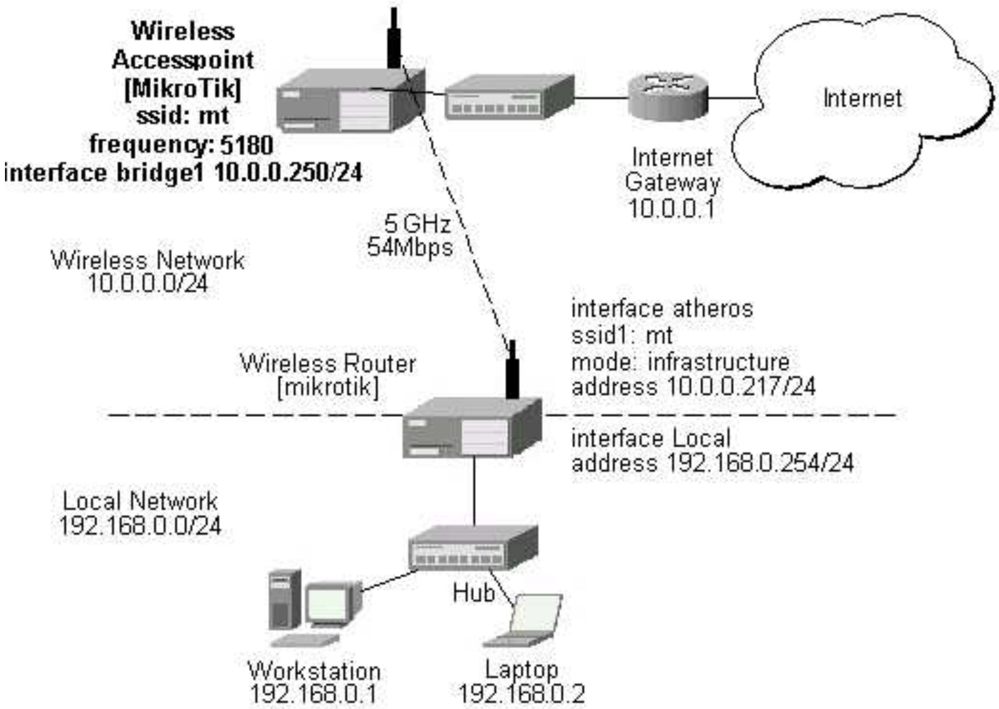
```
[admin@MikroTik] ip route> add gateway=10.0.0.1
[admin@MikroTik] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          r 10.0.0.1        1        atheros1
    1 DC 10.0.0.0/24        r 0.0.0.0         0        atheros1
    2 DC 192.168.0.0/24     r 0.0.0.0         0        ether1
[admin@MikroTik] ip route>
```

Note that you cannot use the bridging function between the atheros and ethernet interfaces, if the atheros 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:

Atheros 5GHz 54Mbps Wireless Interface



To make the MikroTik router work as an access point, the configuration of the atheros wireless interface should be as follows:

- A unique Service Set Identifier should be chosen, say "mt"
- 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 atheros interface:

```
[admin@MikroTik] interface atheros> set 0 mode=ap−bridge frequency=5180MHz ssid=mt
[admin@MikroTik] interface atheros> print
  0  R name="atheros1" mtu=1500 mac−address=00:06:AB:00:37:8E arp=enabled
       mode=ap−bridge root−ap=00:06:AB:00:37:75 frequency=5180MHz ssid="mt"
       supported−rates=6−54 basic−rates=6 protocol=802.11−standard ack−time=26
       default−authentication=yes default−forwarding=yes max−clients=2007


[admin@MikroTik] interface atheros>
```

The list of registered clients looks like follows:

```
[admin@MikroTik] interface atheros> registration−table print
  # INTERFACE   MAC−ADDRESS       TYPE      PARENT            SIGNAL     TX−...
  0 atheros1    00:06:AB:00:37:85 client                      67         6Mbps
[admin@MikroTik] interface atheros>
```

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 atheros 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 atheros interfaces. It requires different IP addresses assigned to both the Ethernet and atheros interfaces. The addresses should be from different networks as well!

To enable bridging between the ethernet and atheros interfaces, do the following:

1. Add bridge interface with the desired forwarded protocols:

```
[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>
```

2. Add the desired interfaces to the bridge interface:

```
[admin@MikroTik] interface bridge port> set "ether1,atheros1" bridge=bridge1
[admin@MikroTik] interface bridge port> print
Flags: X - disabled
  #   INTERFACE                             BRIDGE
  0   ether1                                bridge1
  1   atheros1                              bridge1
[admin@MikroTik] interface bridge port>
```

3. Enable the bridge interface:

```
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
  #    NAME                 TYPE             MTU
  0  R ether1               ether            1500
  1  R atheros1             atheros          1500
  2 X  bridge1              bridge           1500
[admin@MikroTik] interface> enable bridge1
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
  #    NAME                 TYPE             MTU
  0  R ether1               ether            1500
  1  R atheros1             atheros          1500
  2  R bridge1              bridge           1500
[admin@MikroTik] interface>
```

4. Assign an IP address to the bridge interface and specify the default gateway for the access point:

```
[admin@MikroTik] ip address> add address=10.0.0.250/24 interface=bridge1
[admin@MikroTik] 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@MikroTik] ip address> .. route add gateway=10.0.0.1
[admin@MikroTik] 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@MikroTik] ip address>
```

The client router requires the System Service Identifier 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      atheros1
  1   192.168.0.254/24   192.168.0.0     192.168.0.255   Local
[admin@mikrotik] ip address>
```

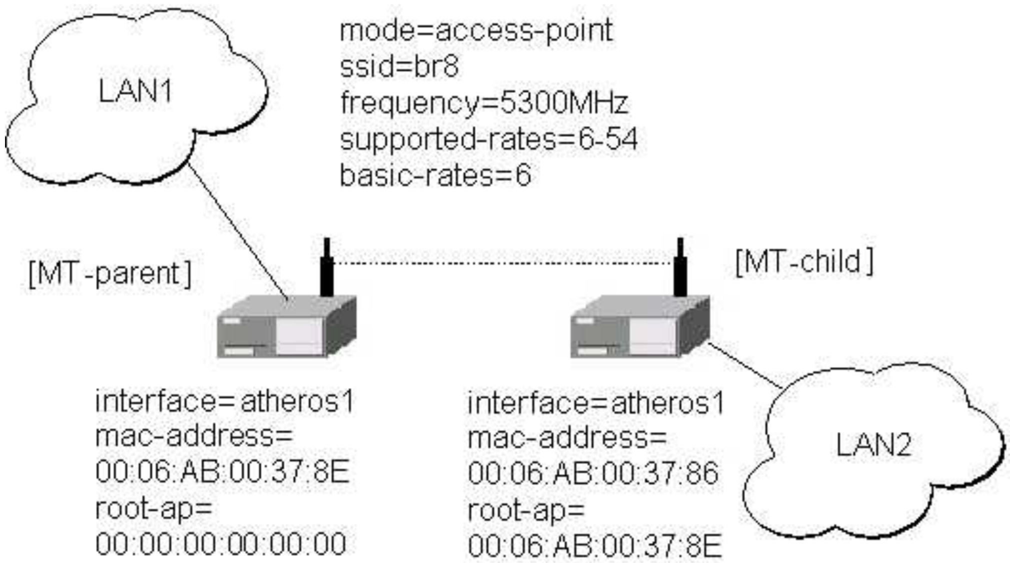
The default route should be set to gateway 10.0.0.1 for the router [mikrotik]:

```
[admin@mikrotik] ip route> add gateway=10.0.0.254
[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 10.0.0.1        1        atheros1
    1 DC 10.0.0.0/24        r 0.0.0.0         0        atheros1
    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 Atheros AP to register to another MikroTik RouterOS Atheros AP for point−to−point operation.

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 atheros1 under **/interface** list.

1. Enable the Ethernet interface ether1:

   ```
   /interface enable ether1
   ```
2. Configure atheros1 interface.
   Set mode=bridge, ssid=br8, frequency=5300MHz, and enable atheros1 interface (you can use mode=ap−bridge, if you have Atheros AP License):

   ```
   /interface atheros set atheros1 mode=bridge ssid=br8 frequency=5300MHz disabled=no
   ```
3. Add bridge interface and specify forwarded protocol list:

   ```
   /interface bridge add forward-protocols=ip,arp,other disabled=no
   ```
4. Specify ports atheros1 and ether1 that belong to bridge1:

```
/interface bridge port set ether1,atheros1 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 atheros1 under **/interface** list.

1. Enable the Ethernet interface ether1:

```
/interface enable ether1
```

2. Configure atheros1 interface.
   Here, you have to specify root−ap MAC address, so the Atheros radio registers to the root AP.
   Set mode=bridge, ssid=br8, frequency=5300MHz, root−ap=xx:xx:xx:xx:xx:xx, and enable atheros1 interface (you can use mode=ap−bridge, if you have Atheros AP License):

```
/interface atheros set atheros1 mode=bridge ssid=br8 frequency=5300MHz\
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] atheros 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 atheros interface, for example:

```
[admin@MT-child] interface atheros> registration−table print
  # INTERFACE                    MAC−ADDRESS       TYPE      PARENT
  0 atheros1                     00:06:AB:00:37:8E parent−ap
[admin@MikroTik] interface atheros>
```

4. Add bridge interface and specify forwarded protocol list:

```
/interface bridge add forward−protocols=ip,arp,other disabled=no
```

5. Specify ports atheros1 and ether1 that belong to bridge1:

```
/interface bridge port set ether1,atheros1 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.

Supported Hardware

This is the list of Atheros chipset based hardware that is tested to work with MikroTik RouterOS:

- Intel 5000 series
- Dlink DWL−A520

Bridge Interface

Document revision 12−Dec−2002
This document applies to the MikroTik RouterOS V2.6

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

Contents of the Manual

The following topics are covered in this manual:

Installation

The bridge feature is included in the 'system' package. No installation is needed for this feature.

Hardware Resource Usage

When Bridge is used, it consumes a small amount of memory. No increase of memory is suggested.

Bridge Setup

IP bridge management is accessible under the **/interface bridge** menu:

```
[admin@MikroTik] interface bridge>
Bridge interface is accessible through any interface with bridging
functionality enabled.

     print  Show bridge interfaces
```

```
       get  get value of item's property
      find  Find interfaces
       set  Change bridge interface settings
    enable  Enable interface
   disable  Disable interface
       add  create new item
    remove  remove item
    export  Export bridge interfaces settings
      port  Interface settings
      host
  firewall
[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,appletalk,ipx,ipv6,other priority=1

  1 X  name="bridge2" mtu=1500 arp=enabled mac-address=00:50:08:00:00:F7
       forward-protocols=appletalk,ipx,ipv6,other priority=1


[admin@MikroTik] interface bridge>
```

Argument description:

name − descriptive name of interface, default is bridgeX, X=1,2,...
mtu − maximum transmit unit in bytes (68...1500, default 1500)
arp − Address Resolution Protocol setting, 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, and not add dynamic entries to the arp table. If required, MAC addresses need to be added as static entries under **/ip arp** neighbor

mac−address − MAC address for the interface, cannot be changed
forward−protocols − list of forwarded protocols. 'Other' means all other protocols than appletalk, arp, ip, ipv6, or ipx, e.g., netbeui, vlan, etc.
priority − bridge interface priority (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.

Note that **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.

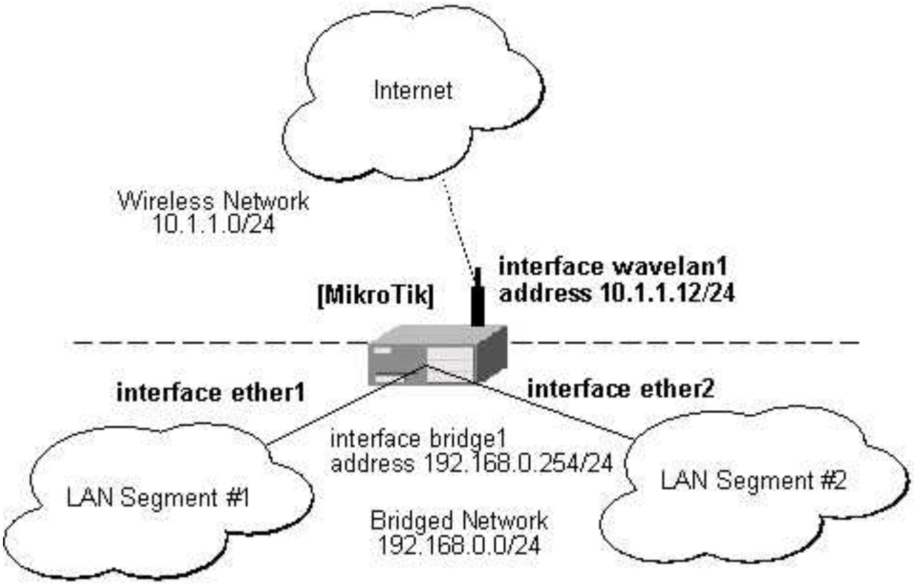
Bridge interface should be enabled and ports specified which belong to it.

Port Settings

Bridge interfaces can be associated with physical network interfaces in **port** submenu:

```
[admin@MikroTik] interface bridge port> print
Flags: X - disabled
  #   INTERFACE                                    BRIDGE
  0   ether1                                       bridge1
  1   ether2                                       bridge1
  2   ether3                                       bridge2
  3   prism1                                       bridge2
[admin@MikroTik] interface bridge port>
```

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   wavelan1                             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   wavelan1                                     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 a 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=wavelan1
[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      wavelan1
[admin@MikroTik] ip address>
```

> ***Note!***Assigning IP address to bridged interfaces 'ether1' or 'ether2' has no sense. Thus, when you assign some interface to a bridge, move its IP address to it 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).

Bridge Monitoring

The bridge can be monitored in real time. The bridging table shows the MAC address of hosts, interface which can forward packets to the host, and the age of the information shown in seconds:

```
[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 ether2             4m48s
   bridge1             00:E0:F7:7F:0A:B8 ether1             1s
[admin@MikroTik] interface bridge host>
```

Bridge Firewall

Traffic between bridged interfaces can be firewalled. The arguments used here are almost the same as for general firewalling:

action − Action to undertake if the packet matches the rule (see below).
dst−address − Destination IP address. Can be in the form address/mask, where mask is number of nonzero bits in the subnet mask, e.g., 10.0.0.204/32
in−interface − interface the packet has entered the bridge through (may be **all**)
mac−dst−address − MAC address of destination host
mac−protocol − Either **all** or the MAC protocol number of the packet. Most widely used MAC protocol numbers are: **2048** for IP, **2054** for ARP, **32821** for RARP, **32823** for IPX, **32923** for AppleTalk (EtherTalk), **33011** for AppleTalk Address Resolution Protocol (AARP), **33169** for NetBEUI, **34525** for IPv6
mac−src−address − MAC address of source host
out−interface − interface the packet is leaving the bridge through (may be **all**)
protocol − Protocol (**all**, **egp**, **ggp**, **icmp**, **igmp**, **ip−encap**, **ip−sec**, **tcp**, **udp**)
src−address − Source IP address. Can be in the form address/mask, where mask is number of bits in the subnet, e.g., 10.0.0.201/32

If the packet matches the criteria of the rule, then the performed **action** can be:

- **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.

Note that packets between bridged interfaces are also passed through the 'normal' **/ip firewall** rules, it 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.

More information about firewall−building can be found in Firewall Filters and Network Address Translation (NAT) manual.

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'.

CISCO/Aironet 2.4GHz 11Mbps Wireless Interface

Document revision 16−Sep−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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)

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

To use CISCO/Aironet PCMCIA cards, first check Notes on PCMCIA Adapters

Contents of the Manual

The following topics are covered in this manual:

Wireless Adapter Hardware and Software Installation

Software Packages

The MikroTik Router should have the aironet software package installed. The software package file **aironet−2.6.x.npk** can be downloaded from MikroTik's web page www.MikroTik.com. To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer.

After successful installation the package should be listed under the installed software packages list, for example:

```
[admin@MikroTik] > /system package pr
Flags: I - invalid
  #   NAME                   VERSION              BUILD-TIME           UNINSTALL
  0   system                 2.6beta3             jul/31/2002 14:05:02 no
  1   ppp                    2.6beta3             jul/31/2002 14:05:25 no
  2   pppoe                  2.6beta3             jul/31/2002 14:05:42 no
  3   pptp                   2.6beta3             jul/31/2002 14:05:39 no
  4   aironet                2.6beta3             jul/31/2002 14:05:45 no
[admin@MikroTik] >
```

Software License

The 2.4GHz wireless adapters require the 2.4GHz wireless feature license. One license is for one installation of the MikroTik RouterOS, disregarding how many cards are installed in one PC box. The wireless feature is not included in the Free Demo or Basic Software License. The 2.4GHz Wireless Feature cannot be obtained for the Free Demo License. It can be obtained only **together** with the Basic Software License.

System Resource Usage

Before installing the wireless adapter, please check the availability of free IRQ's and I/O base addresses:

```
[admin@MikroTik] > system resource irq print
Flags: U - unused
   IRQ OWNER
   1   keyboard
   2   APIC
 U 3
   4   sync1
 U 5
 U 6
 U 7
 U 8
 U 9
 U 10
   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
 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] >
```

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 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. Set the DIP switches on the ISA board according to the following plan:
   DIP switch #6 to 'on' (non−PnP mode)
   Use the DIP switches #1,2,3 to select the IRQ number Use the DIP switches #4,5 to select the I/O Base Address

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

Loading the Driver for the Wireless Adapter

PCI and PC (PCMCIA) cards do not require a 'manual' driver loading, since they are recognized automatically by the system and the driver is loaded at the system startup.

The ISA card requires the driver to be loaded by issuing the following command:

```
[admin@MikroTik]> driver add name=pc−isa io=0x180
[admin@MikroTik]> driver print
Flags: I − invalid, D − dynamic
  #   DRIVER                            IRQ IO         MEMORY     ISDN−PROTOCOL
  0 D PCI NE2000
  1   Aironet ISAxx00                       0x180
[admin@MikroTik] driver>
```

There can be several reasons for a failure to load the driver:

- 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.

Wireless Interface Configuration

If the driver has been loaded successfully (no error messages), and you have the required 2.4GHz Wireless Software License, then the CISCO/Aironet 2.4GHz Wireless interface should appear under the interfaces list with the name pcn, where n is 1,2,... You can change the interface name to a more descriptive one using the **set** command. To enable the interface, use the **enable** command:

```
[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] >
```

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

```
[admin@MikroTik] interface pc> pr
Flags: X - disabled, R - running
  0  R name="pc1" 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>
```

Argument description:

number − Interface number in the list
name − Interface name
mtu − Maximum Transmit Unit (256...2048 bytes). Default value is 1500 bytes.
mode − Operation mode of the card (infrastructure / ad−hoc)
rts−threshold − RTS threshold
fragmentation−threshold − Fragmentation threshold
tx−power − Transmit power in mW
rx−antenna − Receive antenna (**both**, **default**, **left**, **right**)
tx−antenna − Transmit antenna (**both**, **default**, **left**, **right**)
long−retry−limit − Long retry limit
short−retry−limit − Short retry limit
frequency − Channel frequency (**2412MHz**, **2422MHz** ... **2484MHz**)
bitrate − Data rate (**11Mbit/s**, **1Mbit/s**, **2Mbit/s**, **5.5Mbit/s**, **auto**)
ap1 − Access Point 1
ap2 − Access Point 2
ap3 − Access Point 3
ap4 − Access Point 4
ssid1 − Service Set Identifier 1
ssid2 − Service Set Identifier 2
ssid3 − Service Set Identifier 3
modulation − Modulation mode (**cck**, **default**, **mbok**)
client−name − Client name
join−net − Beaconing period
arp − 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.

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>
```

If the wireless interface card is not registered to an AP, the green status led is blinking fast.

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. 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 **bitrate** 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", 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>
```

If the wireless interface card is registered to an AP, the green status led is blinking slow.

Wireless Troubleshooting

- *The pc 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.

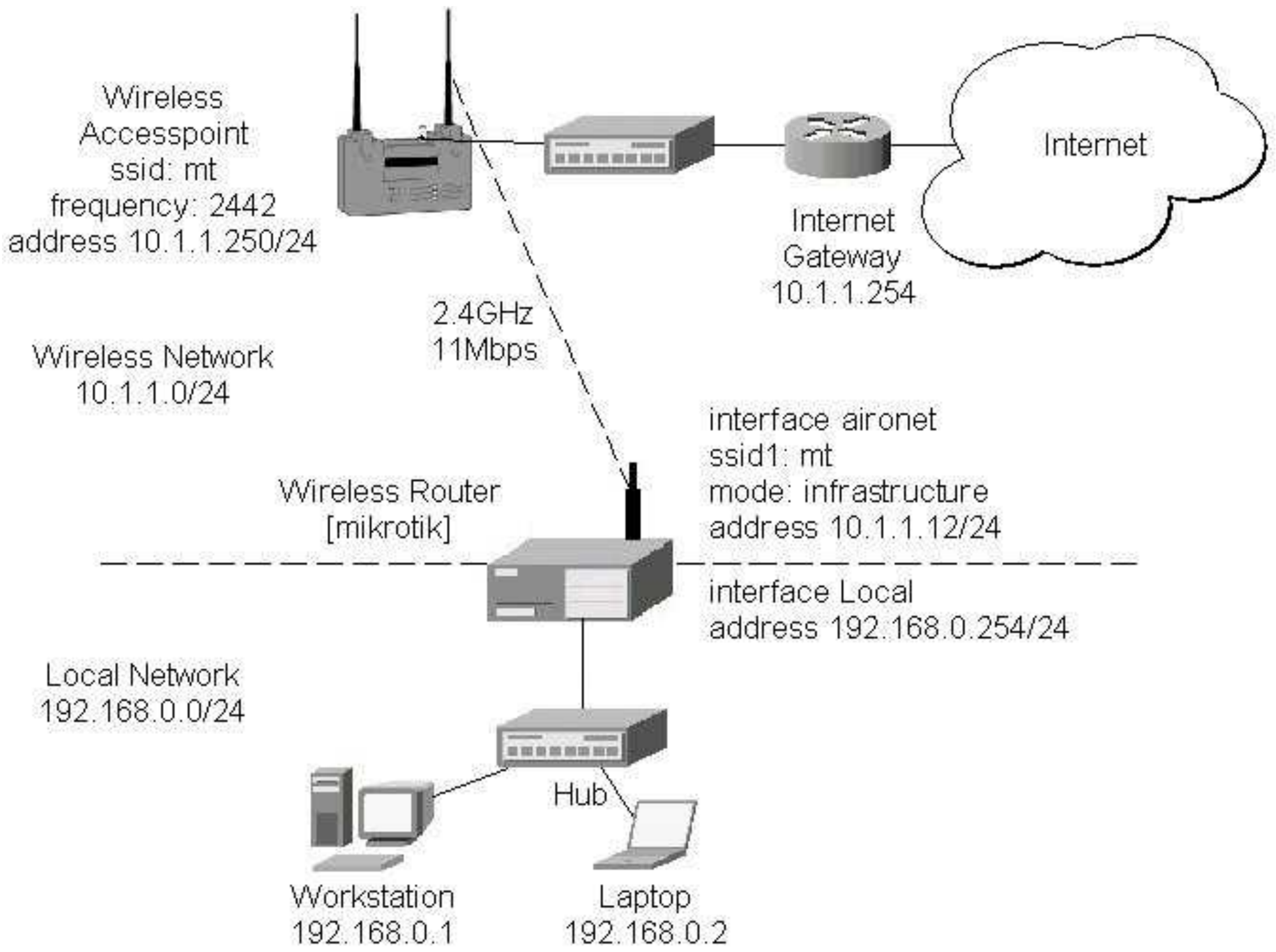
Wireless Network Applications

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:



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>
```

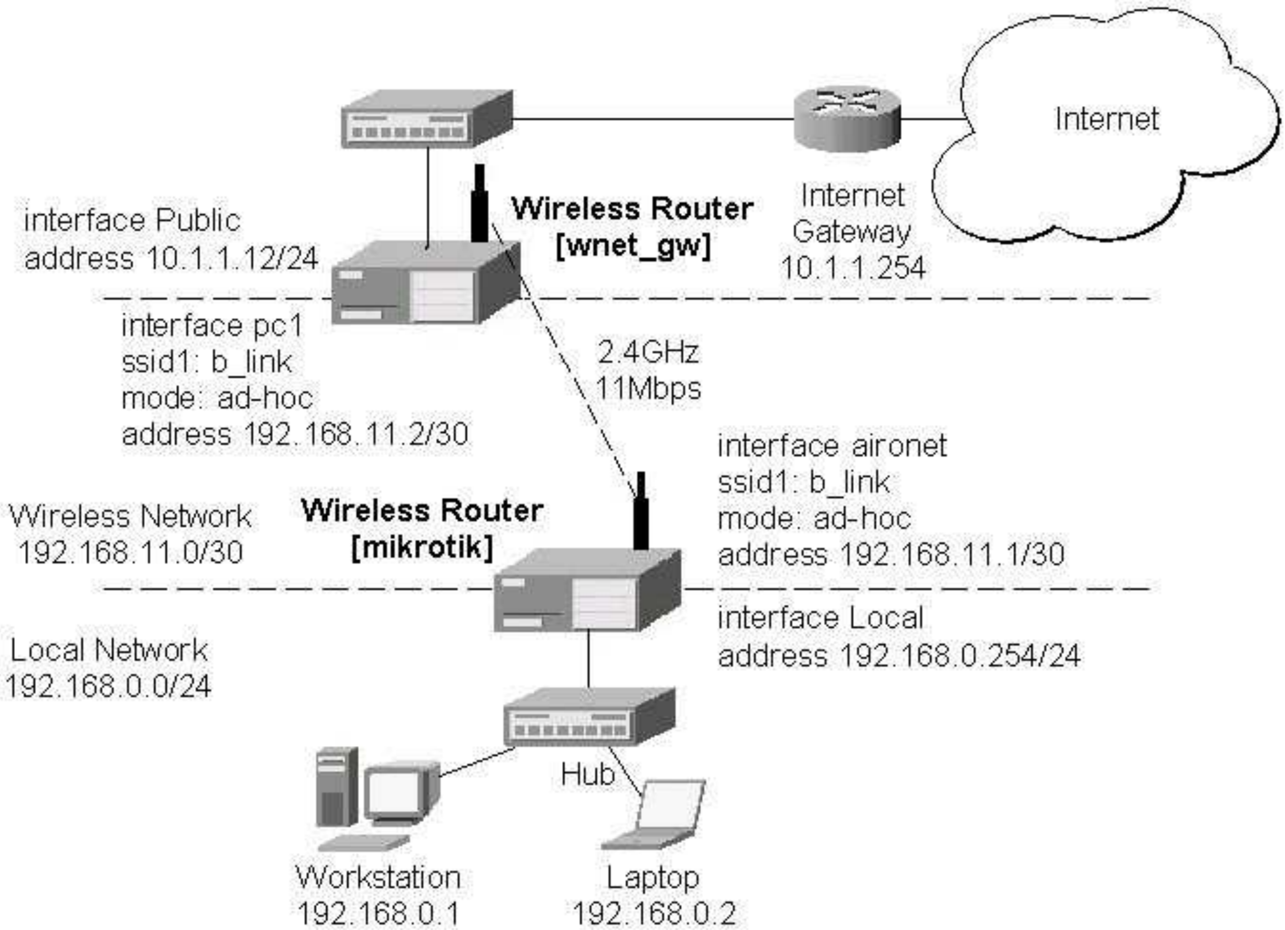
The default route should be set to the gateway router 10.1.1.254 (not the AP 10.1.1.250 !):

```
[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 is looking 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, the card 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> mo 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 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> mo 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 link 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 pc1
[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    pc1
  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>
```

Cyclades PC300 PCI Adapters

Document revision 13−Aug−2002
This document applies to the MikroTik RouterOS v2.6

Overview

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

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

Contents of the Manual

The following topics are covered in this manual:

Adapter Hardware and Software Installation

Software Packages

The MikroTik Router should have the cyclades software package installed. The software package file **cyclades−2.6.x.npk** can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer. After successful installation the package should be listed under the installed software packages list, for example:

```
[admin@MikroTik] > sys package print
Flags: I − invalid
  #   NAME                  VERSION              BUILD−TIME           UNINSTALL
  0   system                2.6beta4             aug/09/2002 20:22:14 no
  1   ppp                   2.6beta4             aug/09/2002 20:28:01 no
  2   moxa−c101             2.6beta4             aug/09/2002 20:53:57 no
  3   pppoe                 2.6beta4             aug/09/2002 20:29:18 no
  4   pptp                  2.6beta4             aug/09/2002 20:28:43 no
```

```
  5   ssh                   2.6beta4            aug/09/2002 20:25:31 no
  6   advanced-tools        2.6beta4            aug/09/2002 20:53:37 no
  7   cyclades              2.6beta4            aug/09/2002 20:52:00 no
  8   framerelay            2.6beta4            aug/09/2002 20:52:09 no
[admin@MikroTik] >
```

Software License

The Cyclades PC300 PCI Adapter requires the Synchronous Feature License. One license is for one installation of the MikroTik RouterOS, disregarding how many cards are installed in one PC box. The Synchronous Feature is not included in the Free Demo or Basic Software License. The Synchronous Feature cannot be obtained for the Free Demo License. It can be obtained only **together** with the Basic Software License.

System Resource Usage

Before installing the synchronous adapter, please check the availability of free resources:

```
[admin@MikroTik] > system resource irq print
Flags: U - unused
   IRQ OWNER
   1   keyboard
   2   APIC
 U 3
   4   serial port
 U 5
 U 6
 U 7
 U 8
   9   ether1
 U 10
   11  [Cyclades-PC300]
 U 12
 U 13
   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
 2F8-2FF             serial port
 3C0-3DF             VGA
 3F6-3F6             IDE 1
 3F8-3FF             serial port
 CF8-CFF             [PCI conf1]
 EE00-EEFF           [Realtek Semiconductor Co., Ltd. RTL-8139]
 EE00-EEFF           [8139too]
 EF80-EFFF           [Cyclades Corporation PC300 TE 1]
 EF80-EFFF           [PLX Registers]
 FC00-FC7F           [Cyrix Corporation 5530 IDE [Kahlua]]
 FC00-FC07           IDE 1
 FC08-FC0F           IDE 2
[admin@MikroTik] >
```

Installing the Synchronous Adapter

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

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.

The Cyclades PC300 PCI Adapter should be recognized by your motherboard automatically and appear on the list of PCI devices as "Simple COMM Controller" with the IRQ assigned to it.

Loading the Driver for the Cyclades PC300 PCI Adapter

The driver for the Cyclades PC300 PCI Adapter is loaded automatically at the system startup. You can check if the driver has bean loaded by issuing the following command:

```
[admin@MikroTik] >driver print
Flags: I - invalid, D - dynamic
  #   DRIVER                                IRQ IO       MEMORY   ISDN-PROTOCOL
  0 D Cyclades
  1 D RealTek 8139
[admin@MikroTik] >
```

There can be several reasons for a failure to load the driver, for example:

- The driver cannot be loaded because other device uses the requested IRQ.
  Try to set the IRQ assignment to PCI slots using the system BIOS configuration.

Interface Configuration

If the driver has been loaded successfully (no error messages), and you have the required Synchronous Software License, then the cyclades interface should appear under the interfaces list with the name cycladesX, where X is 1,2,... To enable the interface, use the **enable** command:

```
[admin@MikroTik] > interface print
Flags: X - disabled, D - dynamic, R - running
  #    NAME                 TYPE             MTU
  0  R ether1               ether            1500
  1 X  cyclades1            cyclades         1500
[admin@MikroTik] interface> enable 1
[admin@MikroTik] > interface print
Flags: X - disabled, D - dynamic, R - running
  #    NAME                 TYPE             MTU
  0  R ether1               ether            1500
  1    cyclades1            cyclades         1500
[admin@MikroTik] interface>
```

More configuration and statistics parameters can be found under the **/interface cyclades** menu. For the Cyclades PC300/RSV Synchronous PCI Adapter you should set the mtu to 1500, and have other argument values as below:

```
[admin@MikroTik] > interface prism print
Flags: X - disabled, R - running
  0  R name="cyclades1" mtu=1500 line-protocol=cisco-hdlc 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>
```

Argument description:

number − Interface number in the list
name − Interface name
mtu − Maximum Transmit Unit (68...1500 bytes). Deafault value is 1500 bytes.
line−protocol − Line protocol (**cisco−hdlc**, **frame−relay**, **sync−ppp**)
media−type − The hardware media used for this interface (**E1**, **T1**, **V24**, **V35**, **X21**)
clock−rate − The clock mode or clock rate in bps. If **0**, the external clock mode is selected. For V.35 should be set to **0** to use the external clock from the modem. Valeus greater than **0** represent the clock speed (which implies an internal clock).
clock−source − Source of the clock (**external**, **internal**, **tx−internal**)
line−code − For T1/E1 channels only. The line code (**AMI**, **B8ZS**, **HDB3**, **NRZ**)
framing−mode − For T1/E1 channels only. The frame mode (**CRC4**, **D4**, **ESF**, **Non−CRC4**, **Unframed**)
line−build−out − For T1 channels only. Line Build Out Signal Level (**0dB**, **15dB**, **22.5dB**, **7.5dB**)
rx−sensitivity − For T1/E1 channels only. Receiver sensitivity (**long−haul**, **short−haul**)

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.

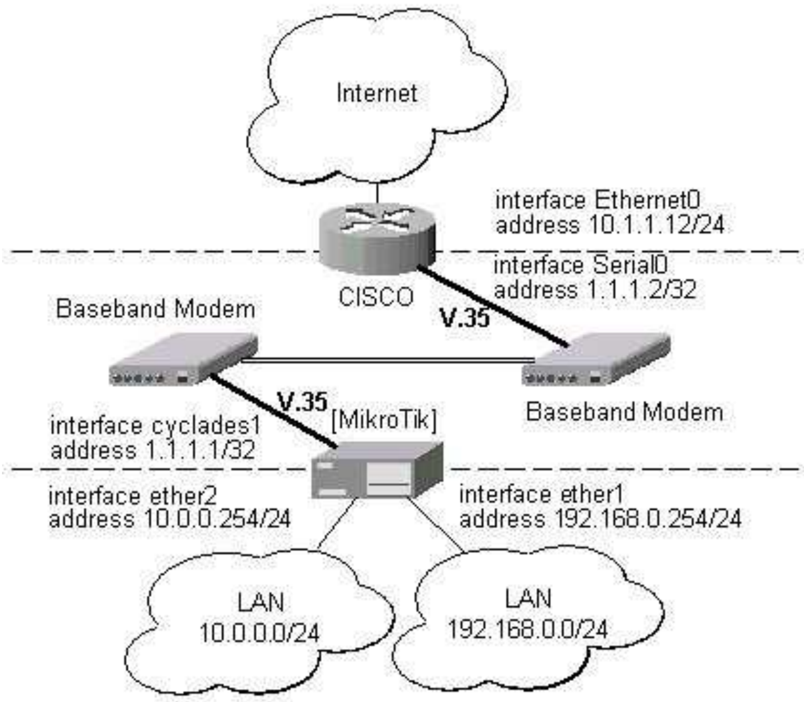
Connect a communication device, e.g., a baseband modem, to the V.35 port and turn it on. The MikroTik driver for the Cyclades Synchronous PCI Adapter allows you to unplug the V.35 cable from one modem and plug it into another modem with a different clock speed, and you do not need to restart the interface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

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   192.168.0.254   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#
```

Ethernet Interfaces

Document revision 29−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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.

Contents of the Manual

The following topics are covered in this manual:

Ethernet Adapter Hardware and Software Installation

Software Packages

The drivers for Ethernet NICs are included in the 'system' package. No installation of other packages is needed.

Software License

The license for Ethernet NICs is included in the Basic License. No additional license is needed.

System Resource Usage

Before installing the Ethernet adapter, please check the availability of free IRQ's and I/O base addresses:

```
[admin@MikroTik] > system resource irq print
Flags: U − unused
   IRQ OWNER
   1   yes  keyboard
   2   yes  APIC
 U 3   no
   4   yes  serial port
   5   yes  PCMCIA service
 U 6   no
 U 7   no
 U 8   no
```

```
 U 9   no
   10  yes  [e1000]
   11  yes  ether3
   12  yes  ether1
   13  yes  FPU
   14  yes  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
 2F8-2FF               serial port
 3C0-3DF               VGA
 3F6-3F6               IDE 1
 3F8-3FF               serial port
 9400-94FF             ether1
 F000-F007             IDE 1
 F008-F00F             IDE 2
[admin@MikroTik] >
```

Loading the Driver

PCI, PCMCIA and CardBus adapters do not require a 'manual' driver loading, since they are recognized automatically by the system and the driver is loaded at the system startup.

ISA adapters require the driver to be loaded by issuing the following command:

```
[admin@MikroTik] driver> add name=ne2k-isa io=0x300
[admin@MikroTik] driver> print
Flags: I - invalid, D - dynamic
  #   DRIVER                            IRQ IO         MEMORY     ISDN-PROTOCOL
  0 D RealTek RTL8129/8139
  1 D NationalSemiconductors 83820
  2 D Intel PRO 1000 Server Adaper
  3   ISA NE2000                            0x300
[admin@MikroTik] driver>
```

There can be several reasons for a failure to load the driver:

- The driver cannot be loaded because other device uses the requested IRQ.
  Try to free up the required IRQ, or get a different card.
- The requested I/O base address cannot be used on your motherboard.
  Get another motherboard.

Note that for some ISA cards there is an utility that configures the resources used by the card. Some other cards might have jumpers that control the same thing. If another cards use the requested resource, try changing these settings.

For more information on installing PCMCIA cards, check Notes on PCMCIA Adapters first.

Ethernet Interface Configuration

If the driver has been loaded successfully (no error messages), then the Ethernet interface should appear under the interfaces list with the name etherX, where X is 1,2,... You can change the interface name to a more descriptive one using the **set** command. To enable the interface, use the **enable** command:

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

You can monitor the traffic passing through any interface using the **/interface monitor** command:

```
[admin@MikroTik] interface> monitor−traffic ether6
    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>
```

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 LED.

In **/interface ethernet** submenu it is possible to set ethernet interface−specific parameters:

```
[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> set 0 ?
changes properties of one or several items.
                    arp  Address Resolution Protocol
  disable−running−check
               disabled
                    mtu  Maximum Trasfer Unit
                   name  New interface name
```

Parameter description:

name − interface name
arp − 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.

mtu − Maximum Transmit Unit. Default value is 1500 bytes.
disable−running−check − for 'broken' ethernet cards it is good to disable running status checking (as default).

For some Ethernet NICs it is possible to monitor the Ethernet status:

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

[admin@MikroTik] interface ethernet> monitor ether3
              status: no−link
    auto−negotiation: incomplete

[admin@MikroTik] interface ethernet> monitor ether1
              status: unknown

[admin@MikroTik] interface ethernet>
```

Please see the IP Address manual on how to add IP addresses to the interfaces.

Ethernet over IP (EoIP) Tunnel Interface

Document revision 21−Jan−2003
This document applies to the MikroTik RouterOS V2.6

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

Contents of the Manual

The following topics are covered in this manual:

Installation

The Ethernet over IP tunnel feature is included in the 'system' package. No installation is needed for this feature.

Hardware Resource Usage

There is no significant resource usage.

EoIP Interface and Protocol 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.

EoIP Setup

IP EoIP Interface management can be accessed under the **/interface eoip** submenu.

You can add an EoIP tunnel interface using the **/interface eoip add** command:

```
[admin@MikroTik] interface eoip> add
creates new item with specified property values.
             arp  Address Resolution Protocol
       copy−from  item number
        disabled
             mtu  Maximum Trasfer Unit
            name  New tunnel name
  remote−address  Remote address of tunnel
       tunnel−id
[admin@MikroTik] interface eoip> add name to_mt2 tunnel−id 1 remote−address 10.5.8.1
[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> enable 0
```

Descriptions of settings:

name − Interface name for reference
mtu − Maximum Transmit Unit. Should be the default 1500 bytes.
arp − 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.

tunnel−id − Should be a number, that is not being used for an another EoIP tunnel.
remote−address − The IP address of the other side of the EoIP tunnel – must be a MikroTik router.

You can assign an IP address to the EoIP interface.

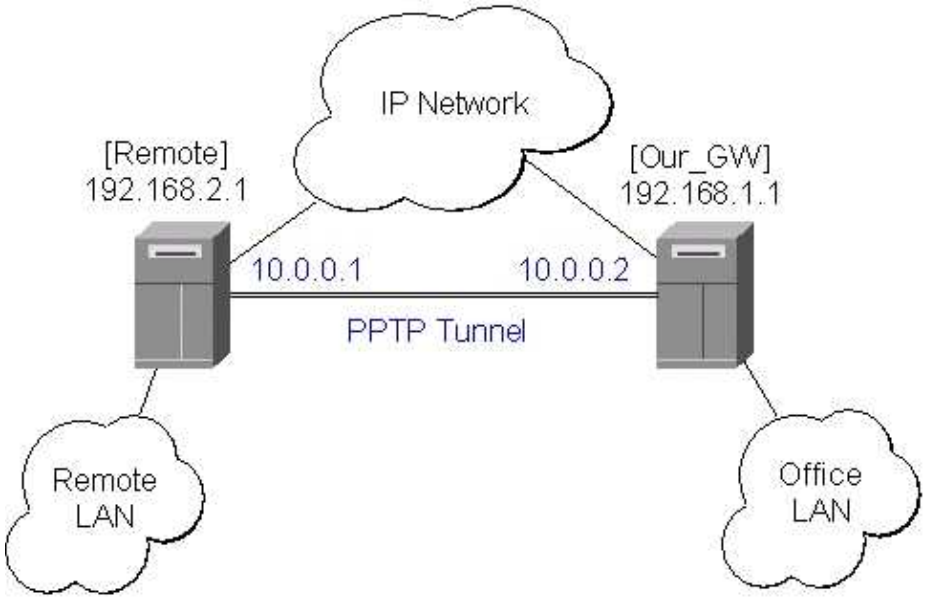
The router at the other end should have the same tunnel−id value, and should have the remote−address set to [MikroTik].

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.

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
[admin@Our_GW] 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@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 seme for the Remote:

```
[admin@Remote] interface bridge> add forward-protocols=ip,arp,other
[admin@Remote] 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@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 29−Aug−2002
This document applies to the MikroTik RouterOS v2.6

Overview

The MikroTik RouterOS supports FarSync T−Series X.21 synchronous adapter hardware. For more information about the adapter hardware please see the relevant documentation:

- http://www.farsite.co.uk/

Contents of the Manual

The following topics are covered in this manual:

Synchronous Adapter Hardware and Software Installation

Software Packages

The MikroTik Router should have the FarSync X.21 synchronous software package installed. The software package file **farsync−2.6.x.npk** (about 110 Kb) can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer. After successful installation the package should be listed under the installed software packages list.

Software License

The FarSync X.21 Synchronous Adapter requires the Synchronous Feature License. One license is for one installation of the MikroTik RouterOS, disregarding how many cards are installed in one PC box. The Synchronous Feature is not included in the Free Demo or Basic Software License. The Synchronous Feature cannot be obtained for the Free Demo License. It can be obtained only **together** with the Basic Software License.

Synchronous Interface Configuration

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

```
[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>
```

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

```
[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>
```

Argument description:

numbers − Interface number in the list
hdlc−keepalive − Cisco HDLC keepalive period in seconds (0..32767)
clock−rate − Speed of internal clock
clock−source − Clock source (**external**, **internal**)
disabled − disable or enable the interface
frame−relay−dce − Operate in DCE mode (**yes**, **no**)
frame−relay−lmi−type − Frame−Relay Local Management Interface type (**ansi**, **ccitt**)
line−protocol − Line protocol (**cisco−hdlc**, **frame−relay**, **sync−ppp**)
media−type − Type of the media (**V24**, **V35**, **X21**)
mtu − Maximum Transmit Unit (68...1500 bytes). Default value is 1500 bytes.
name − New interface name

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.

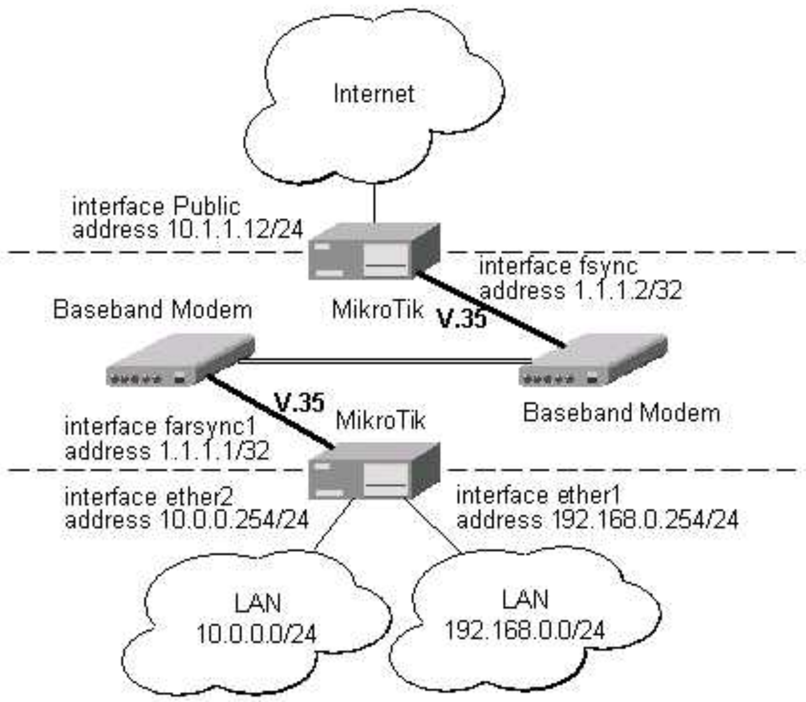
Synchronous Link Applications

One possible synchronous line configurations is discussed in the following example:

- MikroTik Router to MikroTik 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 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>
```

FrameRelay (PVC) Interfaces

Document revision 14−Aug−2002
This document applies to MikroTik RouterOS v2.6

Overview

Frame Relay is a multiplexed interface to packet switched network. Frame Relay 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.

Topics covered in this manual:

Frame Relay Installation on the MikroTik RouterOS

- Hardware part of Frame Relay installation

  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

- Software part of Frame Relay installation

  The **framerelay−2.6.x.npk**(89 KB) package is required. The package can be downloaded from MikroTik's web page www.mikrotik.com. To install this package, please upload it to the router with ftp and reboot. You may check to see if the package is installed with the command:

```
[admin@MikroTik] > system package print
Flags: I − invalid
  #   NAME                   VERSION              BUILD−TIME           UNINSTALL
  0   system                 2.6beta4             aug/09/2002 20:22:14 no
  1   ppp                    2.6beta4             aug/09/2002 20:28:01 no
  2   pppoe                  2.6beta4             aug/09/2002 20:29:18 no
  3   pptp                   2.6beta4             aug/09/2002 20:28:43 no
  4   ssh                    2.6beta4             aug/09/2002 20:25:31 no
  5   advanced−tools         2.6beta4             aug/09/2002 20:53:37 no
  6   farsync                2.6beta4             aug/09/2002 20:51:48 no
  7   framerelay             2.6beta4             aug/09/2002 20:52:09 no
[admin@MikroTik] >
```

  Line 7 shows that required package **framerelay−2.6beta4.npk** is installed.

  Package enables Frame Relay PVC (Permanent Virtual Circuit) interface, which acts as a logical network interface where endpoints and class of service are defined by network management. This logical interface is using one of supported Moxa or Cyclades synchronous adapters as a physical interface.

Configuring Frame Relay Interface

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

Cyclades PC300 interface

```
[admin@MikroTik] > interface cyclades print
Flags: X − disabled, R − running
  0  R name="cyclades1" mtu=1500 line−protocol=sync−ppp 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] >
```

Argument description:

- **name** − Assigned name of the interface
- **mtu** − Maximum Transfer Unit of an interface
- **line−protocol** − Line protocol (**cisco−hdlc**, **frame−relay**, **sync−ppp**)
- **media−type** − The hardware media used for this port (**E1**, **T1**, **V24**, **V35**, **X21**)
- **clock−rate** − Speed of the clock
- **clock−source** − Source of the clock (**external**, **internal**, **tx−internal**)
- **line−code** − The line code (For **T1**/**E1** channels only) (**AMI**, **B8ZS**, **HDB3**, **NRZ**)
- **framing−mode** − The frame mode (For **T1**/**E1** channels only) (**CRC4**, **D4**, **ESF**, **Non−CRC4**, **Unframed**)
- **line−build−out** − LBO For **T1** channels only (**0dB**, **15dB**, **22.5dB**, **7.5dB**)
- **rx−sensitivity** − The receiver sensitivity (For **T1**/**E1** channels only) (**long−haul**, **short−haul**)
- **frame−relay−lmi−type** − Type of frame relay Local Management Interface (**ansi**, **ccitt**)
- **frame−relay−dce** − Determine whether the interface will be a DCE or DTE (**yes**, **no**)
- **chdlc−keepalive** − CHDLC keepalive period (in seconds)

MOXA C101 interface

```
[admin@MikroTik] > interface synchronous print
Flags: X − disabled, R − running
  0  R name="sync1" 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] >
```

Argument description:

- **name** − Assigned name of the interface
- **mtu** − Maximum Transfer Unit of an interface
- **line−protocol** − Type of data transfer protocol (**cisco−hdlc**, **frame−relay**, **sync−ppp**)
- **clock−rate** − Speed of the clock
- **clock−source** − The clock source (**external**, **internal**, **tx−from−rx**, **tx−internal**)
- **frame−relay−lmi−type** − Type of frame relay Local Management Interface (**ansi**, **ccitt**)
- **frame−relay−dce** − Determine whether the interface will be a DCE or DTE (**yes**, **no**)
- **cisco−hdlc−keepalive−interval** − CHLDC keepalive interval (in seconds)

- **ignore−dcd** − Ignore DCD (**yes**, **no**)

Frame Relay PVC interface

To add a PVC interface, use the **/interface pvc add** command. For example, for a Cyclades interface and DLCI equal to 42, we should use the command:

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

Argument description:

- **name** − Assigned name of the interface
- **mtu** − Maximum Transfer Unit of an interface
- **dlci** − Data−Link Connection Identifier assigned to the PVC interface
- **interface** − FrameRelay interface

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 C101 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 synchronous> print
Flags: X - disabled, R - running
  0  R name="sync1" 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 ignore-dcd=no

[admin@MikroTik] interface synchronous>
```

PVC

```
[admin@MikroTik] interface pvc> print
Flags: X - disabled, R - running
  #    NAME                 MTU  DLCI INTERFACE
  0  R pvc1                 1500 42   sync1
[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 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.

IP over IP (IPIP) Tunnel Interface

Document revision 29−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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.

Network setups with IPIP interfaces:

- Possibility to tunnel Intranets over the Internet
- Possibility to avoid using source routing

Contents of the Manual

The following topics are covered in this manual:

Installation

The IP over IP tunnel feature is included in the 'system' package. No installation is needed for this feature.

Hardware Resource Usage

This protocol uses a minimum of resources.

IPIP Interface and Protocol 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.

IPIP Setup

IP over IP Interface management can be accessed under the **/interface ipip** submenu.

You can add an IPIP tunnel interface using the **/interface ipip add** command:

```
[admin@MikroTik] interface ipip> add name test_IPIP mtu 1480 local-address 10.0.0.204 \
\... remote-address 10.0.0.171
[admin@MikroTik] interface ipip> print
Flags: X - disabled, R - running
  #    NAME                 MTU   LOCAL-ADDRESS   REMOTE-ADDRESS
  0 X  test_IPIP            1480  10.0.0.204      10.0.0.171
[admin@MikroTik] interface ipip> enable 0
[admin@MikroTik] interface ipip> print
Flags: X - disabled, R - running
  #    NAME                 MTU   LOCAL-ADDRESS   REMOTE-ADDRESS
  0 R  test_IPIP            1480  10.0.0.204      10.0.0.171
[admin@MikroTik] interface ipip>
```

Descriptions of settings:

name − Interface name for reference
mtu − Maximum Transmit Unit. Should be set to 1480 bytes to avoid fragmentation of packets. May be set to 1500bytes if mtu path discovery is not working properly on links.
local−address − Local address on router which sends IPIP traffic to the remote side.
remote−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.

The router at the other end should have the remote−address set to [MikroTik].

IPIP CISCO Example

Our IPIP implementation has been tested with Cisco 1005. Sample of the Cisco 1005 configuration:

```
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
```

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 29−Nov−2002
This document applies to MikroTik RouterOS V2.6

Overview

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.

MikroTik Router OS supports following ISDN adapters (**ISA** ISDN adapters are **not** supported):

- Passive PCI adapters with Siemens chipset (Eicon.Diehl Diva, Sedlbauer Speed, ELSA Quickstep 1000, NETjet, Teles, Dr. Neuhaus Niccy, AVM, Gazel, HFC 2BDS0 based adapters, W6692 based adapters).

Topics covered in this manual:

ISDN Hardware and Software Installation

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

The **ppp−2.6.x.npk** (less than 310KB) and the **isdn−2.6.x.npk** (less than 390KB) packages are required. The packages can be downloaded from MikroTik's web page www.mikrotik.com. To install the packages, please upload them to the router with ftp and reboot. You may check to see if the packages are installed with the command:

```
[admin@MikroTik] system package> print
Flags: I − invalid
  #   NAME                   VERSION              BUILD−TIME           UNINSTALL
  0   ppp                    2.6rc4               sep/11/2002 14:43:31 no
  1   system                 2.6rc4               sep/11/2002 14:43:03 no
  2   isdn                   2.6rc4               sep/11/2002 15:06:32 no
[admin@MikroTik] system package>
```

Loading the ISDN Driver

The ISDN driver should be loaded using the **/driver add** command:

```
[admin@MikroTik] driver> add name="driver_name"
```

Argument description:

driver_name − name of the driver. The list of available drivers can be obtained by entering **/driver add name=** and pressing [Tab] twice
isdn−protocol − data channel protocol, the default is 'euro'

Complete list of all supported ISDN adapters and their driver names:

- Eicon.Diehl Diva − **diva**
- Sedlbauer Speed − **sedlbauer**
- ELSA Quickstep 1000 − **elsa**
- 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.

Check the loaded drivers by using the **/driver print** command. Example output looks like here:

```
[admin@MikroTik] driver> print
Flags: I − invalid, D − dynamic
  #   DRIVER                                IRQ IO       MEMORY   ISDN−PROTOCOL
  0 D RealTek 8139
  1   HFC 2BDS0 PCI                                               euro
[admin@MikroTik] driver>
```

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          in    data  137
  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

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 **/interface isdn−client** submenu.

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

```
[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>
```

Argument description:

name − interface name
mtu − maximum Transmit Unit
mru − maximum Receive Unit
phone − phone number to dial
msn − MSN/EAZ of ISDN line provided by the line operator
dial−on−demand − use dialing on demand
l2−protocol − level 2 protocol to be used
user − user name that will be provided to the remote server
password − password that will be provided to the remote server
add−default−route − add default route to remote host on connect
profile − profile to use when connecting to the remote server
bundle−128K − use both channels instead of just one

ISDN Server Interface Configuration

ISDN server is used to accept remote dial−in connections from ISDN clients via ISDN. To set up an ISDN dial−in connection, use the ISDN dial−in configuration menu under **/interface isdn−server** submenu.

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

```
[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>
```

Argument description:

name − Interface name
mtu − Maximum Transmit Unit
mru Maximum Receive Unit
msn − MSN/EAZ of ISDN line provided by the line operator
l2−protocol − Level 2 protocol to be used
authentication − Use authentication (**mschap2**, **chap**, **pap**)
profile − profile to use when connecting to the server
bundle−128K − Use Both channels instead of just one.

Example of a printout of configured ISDN server interface is here:

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 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 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

```
[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 set up 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=x75bui bundle-128K=no
```

Configure PPP settings and adding a user to routers 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 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

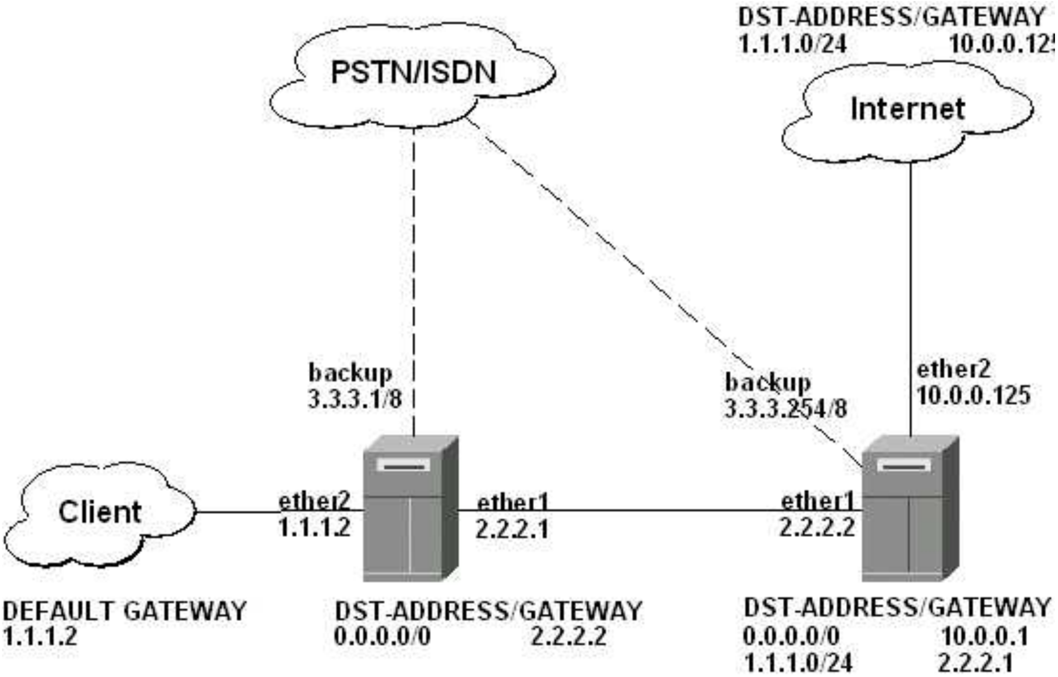
Backup systems are used in specific cases, when you need to maintain a connection, even if something fails. For example, if someone cuts the wires, the router can automatically connect to a different interface to continue its work. This backup is based on a utility that monitors the status of the connection − netwatch, and a script, which runs the netwatch.

ISDN Backup Description

This is an example of how to make a router backup system. In this example we use a ISDN connection to backup a standard ethernet connection. You can, of course, use anything instead of the ISDN connecion − PPP, for example. When the ethernet fails (the router nr.1 cannot ping the router nr.2 to 2.2.2.2 (see picture) the router establishes a ISDN connection − a so−called backup link − to continue comunicating with the nr.2 .

Note, that in our case there are just two routers, but this system can be also used to connect two or more different networks.

The backup system example is described in the following diagram:



In this case the ‘backup’ interface is a ISDN connection, but it can be anything. Follow the instructions below on how to set up the backup link:

Setting up ISDN Connection

To use ISDN, the ISDN card driver must be loaded:

```
[admin@MikroTik] driver> add name=hfc
```

The PPP connection must have the following configuration: A new user must be added to the routers one and two:

```
[admin@Mikrotik] ppp secret> add name=backup password=backup service=isdn
```

A 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
```

A 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
```

Setting 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"
```

Adding 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}
```

Setting up Netwatch

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
```

MOXA C101 Synchronous Interface

Document revision 5−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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.

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

Contents of the Manual

The following topics are covered in this manual:

Synchronous Adapter Hardware and Software Installation

Software Packages

The MikroTik Router should have the moxa c101 synchronous software package installed. The software package file **moxa−c101−2.6.x.npk** can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer. After successful installation the package should be listed under the installed software packages list, for example:

```
[admin@MikroTik] > sys package print
Flags: I − invalid
  #   NAME                   VERSION              BUILD−TIME           UNINSTALL
  0   system                 2.6beta4             aug/09/2002 20:22:14 no
  1   ppp                    2.6beta4             aug/09/2002 20:28:01 no
  2   moxa−c101              2.6beta4             aug/09/2002 20:53:57 no
  3   pppoe                  2.6beta4             aug/09/2002 20:29:18 no
  4   pptp                   2.6beta4             aug/09/2002 20:28:43 no
  5   ssh                    2.6beta4             aug/09/2002 20:25:31 no
```

```
  6   advanced-tools        2.6beta4            aug/09/2002 20:53:37 no
  7   cyclades              2.6beta4            aug/09/2002 20:52:00 no
  8   framerelay            2.6beta4            aug/09/2002 20:52:09 no
[admin@MikroTik] >
```

Software License

The MOXA C101 Synchronous Adapter requires the Synchronous Feature License. One license is for one installation of the MikroTik RouterOS, disregarding how many cards are installed in one PC box. The Synchronous Feature is not included in the Free Demo or Basic Software License. The Synchronous Feature cannot be obtained for the Free Demo License. It can be obtained only **together** with the Basic Software License.

System Resource Usage

Before installing the synchronous adapter, please check the availability of free IRQ's:

```
[admin@MikroTik] > system resource irq print
Flags: U - unused
   IRQ OWNER
   1   keyboard
   2   APIC
 U 3
   4   serial port
 U 5
 U 6
 U 7
 U 8
   9   ether1
 U 10
   11  ether2
 U 12
 U 13
   14  IDE 1
[admin@MikroTik] >
```

Installing the Synchronous Adapter

You can install up to four MOXA C101 synchronous cards in one PC box, if you have so many slots and IRQs available. For ISA variant, the basic installation steps 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. Set the jumper of the IRQ to one, which is free on your system. Usually IRQ 5 is fine.
3. Set the dip switches of the memory mapping base address. Each C101 Super−Sync Board will occupy 16KB memory window. Not all addresses might be available on your motherboard. Use, for example, switch #3 should be OFF, and 1,2,4,5 should be ON for address 0x0D0000. Consult the table in the C101 manual for these settings.
4. Set the jumper of the transmit clock direction to **in**
5. Set the jumper of the communication interface to V.35

Please note, that not all combinations of memory mapping base addresses and IRQ's may work on your motherboard. It is recommended that you choose one IRQ that is not used in your system, and then try an acceptable memory base address setting.

The PCI variant is detected automatically.

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			

Loading the Driver for the MOXA C101 Synchronous Adapter

The MOXA C101 ISA card requires the driver to be loaded by issuing the following command:

```
[admin@MikroTik] driver> add name=c101 mem=0xd0000
[admin@MikroTik] driver> print
Flags: I - invalid, D - dynamic
  #   DRIVER                                IRQ IO         MEMORY    ISDN-PROTOCOL
  0 D RealTek 8139
  1   Moxa C101 Synchronous                                0xd0000
[admin@MikroTik] driver>
```

There can be several reasons for a failure to load the driver:

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

For the MOXA C101 PCI card driver is loaded automatically:

```
[admin@MikroTik] > /driver print
Flags: I - invalid, D - dynamic
  #   DRIVER                                IRQ IO         MEMORY    ISDN-PROTOCOL
  0 D Moxa C101 PCI
  1 D RealTek 8139
[admin@MikroTik] >
```

Synchronous Interface Configuration

If the driver has been loaded successfully (no error messages), and you have the required Synchronous Software License, then the synchronous interface should appear under the interfaces list with the name syncn, where n is 0,1,2,... You can change the interface name to a more descriptive one using the **set** command. To enable the interface, use the **enable** command:

```
[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  ether3               ether            1500
  3 X  sync1                sync             1500

[admin@MikroTik] > interface
[admin@MikroTik] interface> set 3 name moxa
[admin@MikroTik] interface> enable moxa
[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  ether3               ether            1500
  3    moxa                 sync             1500

[admin@MikroTik] >
```

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

```
[admin@MikroTik] interface> synchronous
[admin@MikroTik] interface synchronous> print
Flags: X − disabled
  0   name="moxa" mtu=1500 line−protocol=cisco−hdlc clock−rate=64000
      clock−source=tx−from−rx frame−relay−lmi−type=ansi frame−relay−dce=no
      cisco−hdlc−keepalive−interval=10s ignore−dcd=no

[admin@MikroTik] interface synchronous> set ?
changes properties of one or several items.
                      <numbers>  list of item numbers
  cisco−hdlc−keepalive−interval
                     clock−rate
                   clock−source
                       disabled
                frame−relay−dce  Operate in DCE mode
           frame−relay−lmi−type
                     ignore−dcd  Ignore DCD
                  line−protocol  Line protocol
                            mtu  Maximum Transmit Unit
                           name  New interface name
[admin@MikroTik] interface synchronous> set
```

Argument description:

numbers − Interface number in the list
cisco−hdlc−keepalive−interval − Keepalive period in seconds (0..32767)
clock−rate − Speed of internal clock
clock−source − Clock source (**external**, **internal**, **tx−from−rx**, **tx−internal**)
disabled − disable or enable the interface
frame−relay−dce − Operate in DCE mode (**yes**, **no**
frame−relay−lmi−type − Frame−Relay Local Management Interface type (**ansi**, **ccitt**)

ignore−dcd − Ignore DCD (**yes**, **no**)
line−protocol − Line protocol (**cisco−hdlc**, **frame−relay**, **sync−ppp**)
mtu − Maximum Transmit Unit (68...1500 bytes). Default value is 1500 bytes.
name − New interface name

You can monitor the status of the synchronous interface:

```
[admin@MikroTik] interface synchronous> monitor 0
    dtr: yes
    rts: yes
    cts: no
    dsr: no
    dcd: no

[admin@MikroTik] interface synchronous>
```

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.

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 synchronous> monitor 0
    dtr: yes
    rts: yes
    cts: yes
    dsr: yes
    dcd: yes

[admin@MikroTik] interface synchronous>
```

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.

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.

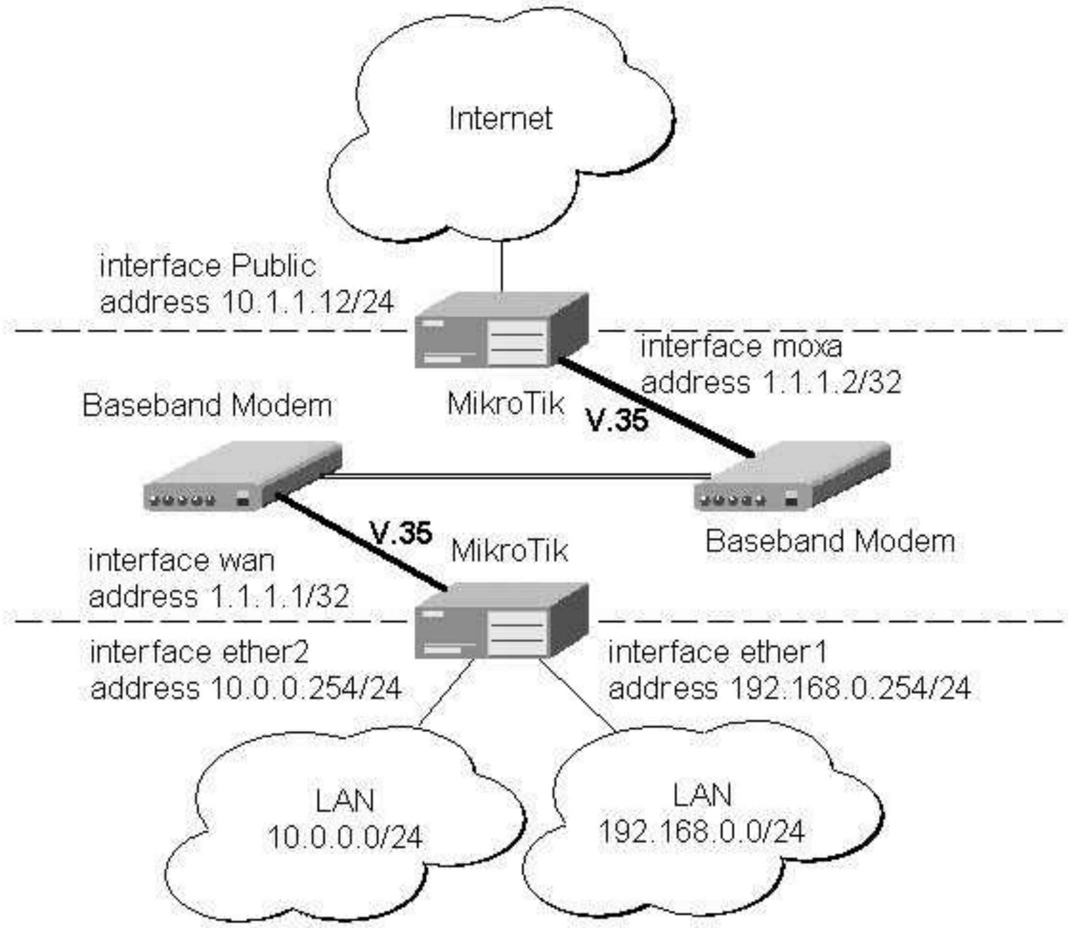
Synchronous Link Applications

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>
```

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:

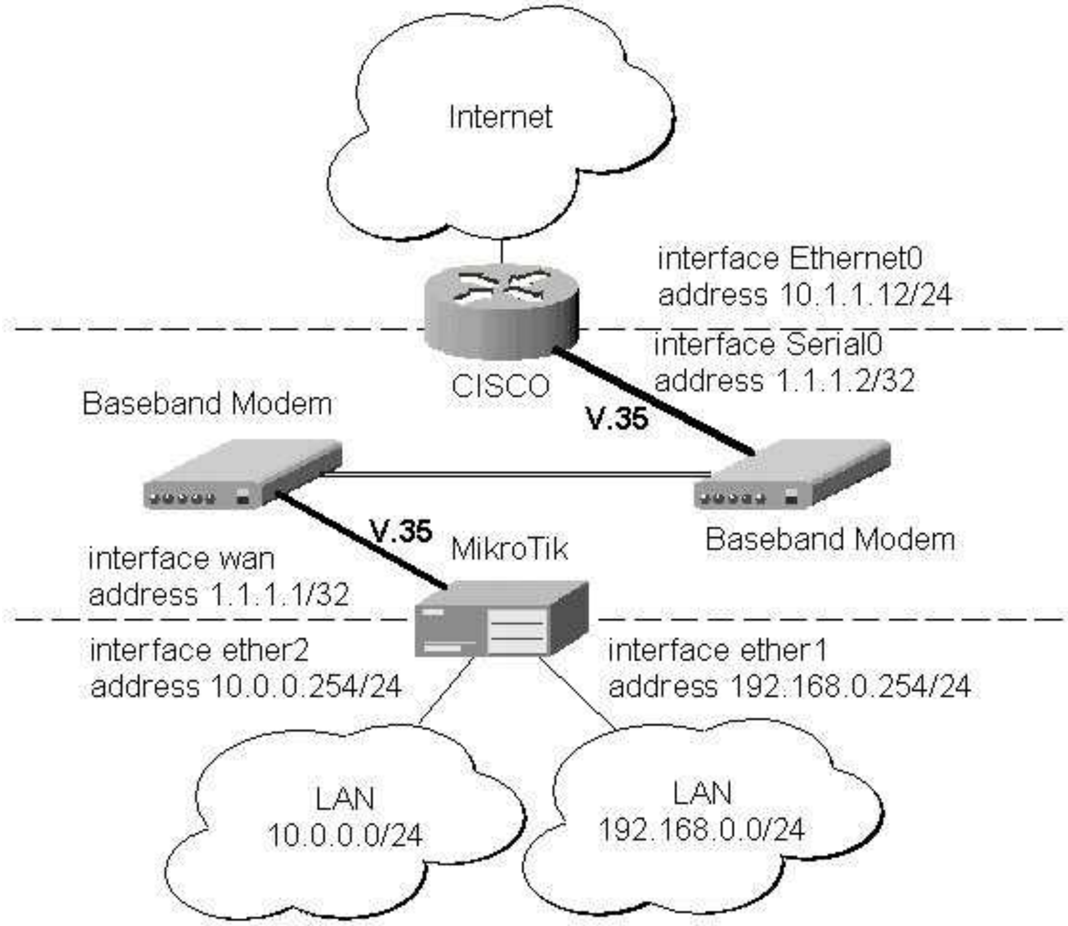
```
[admin@MikroTik] ip route> add gateway 1.1.1.2 interface wan
[admin@MikroTik] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS        G GATEWAY         DISTANCE INTERFACE
    0  S 0.0.0.0/0          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>
```

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 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#
```

MOXA C502 Synchronous Interface

Document revision 23−Sep−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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.

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
- C101 SuperSync Board User's Manual − The User's Manual in .pdf format

Contents of the Manual

The following topics are covered in this manual:

Synchronous Adapter Hardware and Software Installation

Software Packages

The MikroTik Router should have the moxa c502 synchronous software package installed. The software package file **moxa−c502−2.6.x.npk** can be downloaded from MikroTik’s web page www.mikrotik.com. To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer. After successful installation the package should be listed under the installed software packages list, for example:

```
[admin@MikroTik] > sys package print
Flags: I − invalid
  #   NAME                   VERSION              BUILD−TIME           UNINSTALL
  0   system                 2.6beta4             aug/09/2002 20:22:14 no
  1   ppp                    2.6beta4             aug/09/2002 20:28:01 no
  2   moxa−c502              2.6beta4             aug/09/2002 20:53:57 no
  3   pppoe                  2.6beta4             aug/09/2002 20:29:18 no
  4   pptp                   2.6beta4             aug/09/2002 20:28:43 no
  5   ssh                    2.6beta4             aug/09/2002 20:25:31 no
  6   advanced−tools         2.6beta4             aug/09/2002 20:53:37 no
```

```
  7   cyclades              2.6beta4              aug/09/2002 20:52:00 no
  8   framerelay            2.6beta4              aug/09/2002 20:52:09 no
[admin@MikroTik] >
```

Software License

The MOXA C502 Dual−port Synchronous Adapter requires the Synchronous Feature License. One license is for one installation of the MikroTik RouterOS, disregarding how many cards are installed in one PC box. The Synchronous Feature is not included in the Free Demo or Basic Software License. The Synchronous Feature cannot be obtained for the Free Demo License. It can be obtained only **together** with the Basic Software License.

System Resource Usage

Before installing the synchronous adapter, please check the availability of free IRQ's:

```
[admin@MikroTik] > system resource irq print
Flags: U − unused
   IRQ OWNER
   1   keyboard
   2   APIC
 U 3
   4   serial port
 U 5
 U 6
 U 7
 U 8
   9   ether1
 U 10
   11  ether2
 U 12
 U 13
   14  IDE 1
[admin@MikroTik] >
```

Installing the Synchronous Adapter

You can install up to four MOXA C502 synchronous cards in one PC box, if you have so many PCI slots available.

Loading the Driver for the MOXA C502 Synchronous Adapter

The MOXA C502 PCI card requires no manual driver loading:

```
[admin@MikroTik] > /driver print
Flags: I − invalid, D − dynamic
  #   DRIVER                                 IRQ IO       MEMORY   ISDN−PROTOCOL
  0 D Moxa C502 PCI
  1 D RealTek 8139
[admin@MikroTik] >
```

Synchronous Interface Configuration

If the driver has been loaded successfully (no error messages), and you have the required Synchronous Software License, then the two synchronous interfaces should appear under the interfaces list with the name moxaN, where N is 0,1,2,... You can change the interface name to a more descriptive one using the **set** command. To enable the interface, use the **enable** command:

```
[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  ether3               ether            1500
  3 X  moxa1                moxa             1500
  4 X  moxa2                moxa             1500

[admin@MikroTik] > interface
[admin@MikroTik] interface> set 3 name moxa
[admin@MikroTik] interface> enable moxa
[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  ether3               ether            1500
  3    moxa                 moxa             1500
  4 X  moxa2                moxa             1500

[admin@MikroTik] >
```

More configuration and statistics parameters can be found under the **/interface moxa−c502** menu:

```
[admin@MikroTik] interface> moxa-c502
[admin@MikroTik] interface moxa-c502> print
Flags: X - disabled, R - running
  0 X  name="moxa1" 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 X  name="moxa2" 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>
```

Argument description:

numbers − Interface number in the list
cisco−hdlc−keepalive−interval − Keepalive period in seconds (0..32767)
clock−rate − Speed of internal clock
clock−source − Clock source (**external**, **internal**, **tx−from−rx**, **tx−internal**)
disabled − disable or enable the interface
frame−relay−dce − Operate in DCE mode (**yes**, **no**
frame−relay−lmi−type − Frame−Relay Local Management Interface type (**ansi**, **ccitt**)
line−protocol − Line protocol (**cisco−hdlc**, **frame−relay**, **sync−ppp**)
mtu − Maximum Transmit Unit (68...1500 bytes). Default value is 1500 bytes.
name − New interface name

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>
```

The MikroTik driver for the MOXA C502 Dual−port 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.

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.

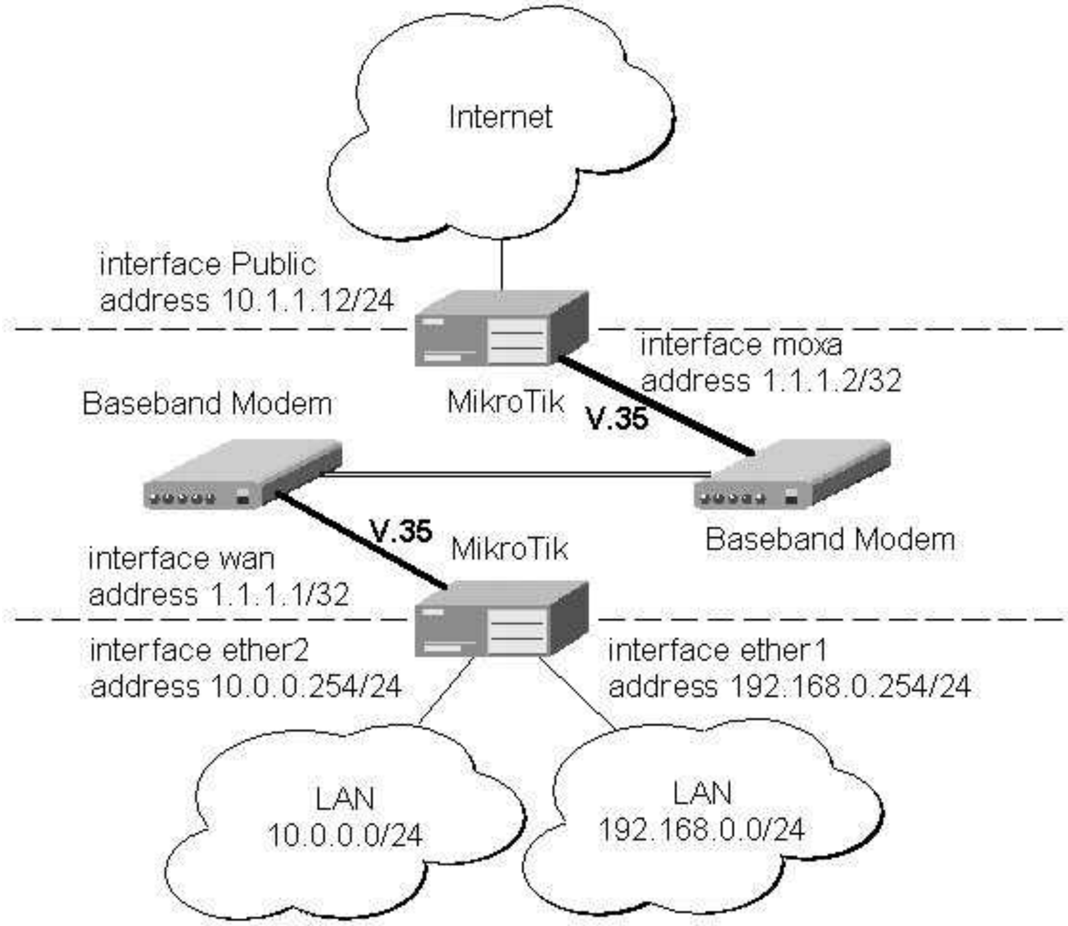
Synchronous Link Applications

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 C502 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>
```

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 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
```

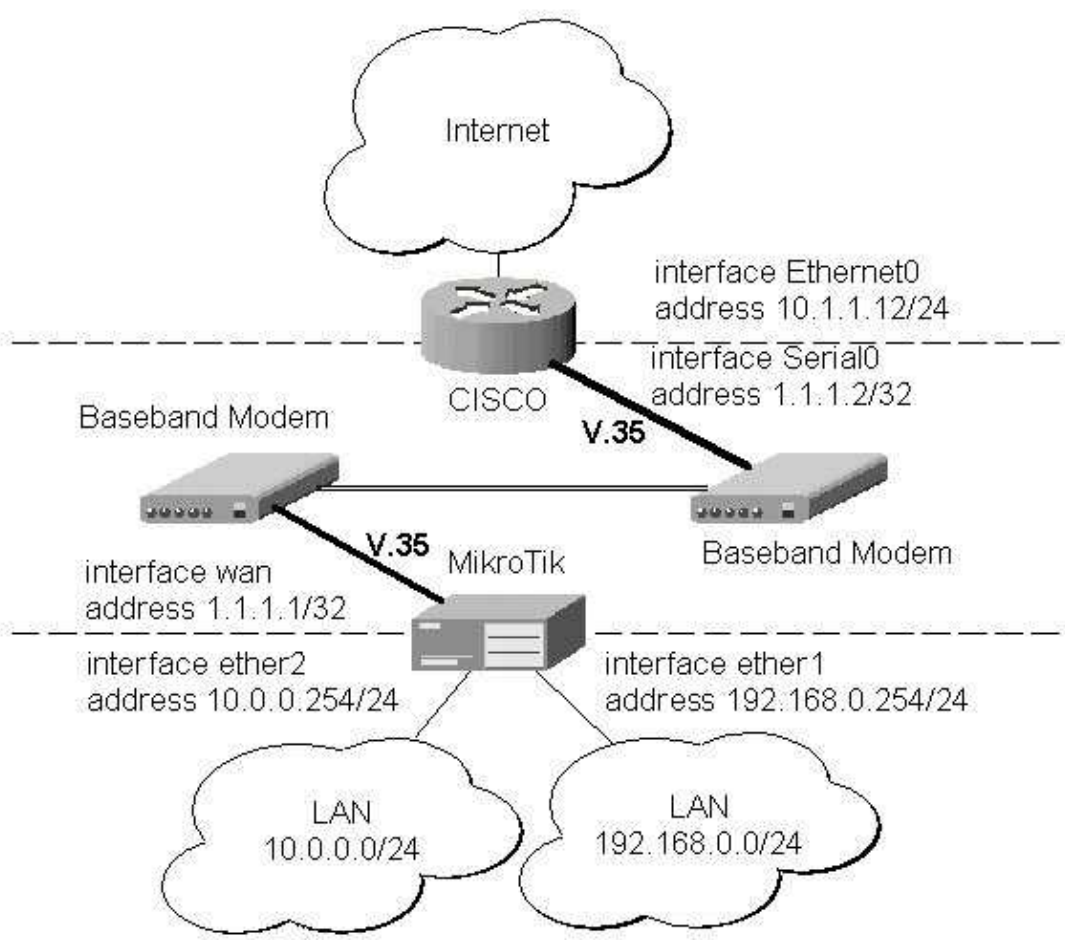
```
    1 DC 10.0.0.0/24         r 10.0.0.254      1        ether2
    2 DC 192.168.0.0/24      r 192.168.0.254   0        ether1
    3 DC 1.1.1.2/32          r 0.0.0.0         0        wan

[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 C502 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>
```

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 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#
```

General Point to Point Settings

Document revision 30−Dec−2002
This document applies to the MikroTik RouterOS V2.6

Overview

This section describes setting user configuration for Point to Point links as: PPP, PPTP, PPPoE as well as ISDN.

P2P (point to point) authentication on the MikroTik RouterOS is supported by a local authentication database or a RADIUS client. Authentication is supported for PPP asynchronous connections, PPPoE, PPTP, and ISDN PPP (local only). Authentication protocols supported are PAP, CHAP, and MS−CHAPv2. The authentication process is as follows: P2P sends a user authentication request, the user ID is first checked against the local user database for any users which have the PPP attribute, if no matching user is found then the RADIUS client (if enabled) will request authentication from the RADIUS server. Note that the users will first be checked against the local database and then only against the RADIUS server. Be careful not to have the same P2P user on the local database and the RADIUS server – the authentication will finish at the local database in this case.

Contents of the Manual

The following topics are covered in this manual:

Installation

The **ppp−2.6.x.npk** package is required. The package can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload them to the router with ftp and reboot. You may check to see if the PPP package is installed with the command:

The RADIUS client and RADIUS accounting features are included in the **PPP** package.

Hardware Resource Usage

There is no significant resource usage.

Local Authentication Overview

Local P2P authentication is part of the general user database stored on the router – this database is also responsible for administration authentication for the router. Certain attributes are supported for P2P users:

- P2P remote address set from RADIUS server
- Time limit of connections set from RADIUS server
- MAC address (PPPoE) or remote client address (PPTP) reported to RADIUS server
- System identity
- Traffic accounting (PPP style – no IP pairs)

Local Authentication Management of P2P Users

P2P users are configured in **/ppp secret** and **/ppp profile**

PPP Profile

With **PPP** installation, one default profile is created. PPP 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 one 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.

PPP profiles are configured as follows:

```
[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>
```

Argument description:

name − profile name
local−address − (either address or pool) Assigns an individual address to the PPP−Server
remote−address − (either address or pool) Assigns an individual address to the PPP−Client
session−timeout − The maximum time the connection can stay up. When set to **0**, there is no timeout
idle−timeout − The link will be terminated if there is no activity with−in the time set – in seconds. When set to **0**, there is no timeout
use−compression − defines whether compress traffic or not
use−vj−compression − use Van Jacobson header compression
use−encryption − defines whether encrypt traffic or not
require−encryption − defines whether require encryption from the client or simply prefer it

only−one − allow only one connection at a time
tx−bit−rate − Transmit bitrate in bits/s
rx−bit−rate − Receive bitrate in bits/s
incoming−filter − Firewall chain name for incoming packets. If not empty for each packet coming from client, this firewall chain will get control
outgoing−filter − Firewall chain name for outgoing packets. If not empty for each packet coming to client, this firewall chain will get control

Note that filter rules 'jumping' to the specified firewall chain are added automatically to the **ppp** firewall chain. This means that you should create **ppp** chain and pass some (or all) the packets to it in order to get filtering function.

PPP Secret

/ppp secret submenu defines P2P users ad defines owner and profile for each of them:

```
[admin@MikroTik] ppp secret> print
Flags: X − disabled
  #   NAME              SERVICE CALLER−ID       PASSWORD        PROFILE
  0   ex                any                     lkjrht          default
[admin@MikroTik] ppp secret> print detail
Flags: X − disabled
  0   name="ex" service=any caller−id="" password="lkjrht" profile=default
      local−address=0.0.0.0 remote−address=0.0.0.0 routes==""


[admin@MikroTik] ppp secret>
```

Argument description:

name − user name
service − specifies service that will use this user (**any**, **async**, **isdn**, **pppoe**, **pptp**)
caller−id − For PPTP, this may be set the IP address which a client must connect from in the form of “a.b.c.d”. For PPPoE, the MAC address which the client must connect from can be set in the form or “xx:xx:xx:xx:xx:xx”. When this is not set, there are no restrictions on from where clients may connect
password − user password
profile − profile name for the user
local−address − (either address or pool) Assigns an individual address to the PPP−Server
remote−address − (either address or pool) Assigns an individual address to the PPP−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

Active Users

Current active users can be viewed using **/ppp active print** command:

```
[admin@web−proxy] ppp active> print
Flags: R − radius
  #   NAME         SERVICE CALLER−ID        ADDRESS         UPTIME ENCODING
  0   home         pptp    10.0.0.204       10.5.0.2        40m58s MPPE12...
[admin@web−proxy] ppp active> print detail
Flags: R − radius
  0   name="home" service=pptp caller−id="10.0.0.204"
      address=10.5.0.2 uptime=40m57s encoding="MPPE128 stateless"
```

```
[admin@web-proxy] ppp active>
```

Local Accounting of PPP Users

Local authentication and accounting is enabled by default. And is used when RADIUS client is disabled. The following is an example of the local accounting when a PPPoE connection is made to the PPPoE server (access concentrator).

```
[admin@Mikrotik]> log print

 dec/09/2002 18:11:14 <pppoe-test>: authenticated
 dec/09/2002 18:11:14 <pppoe-test>: connected
 dec/09/2002 18:11:15 test logged in
 dec/09/2002 18:11:26 test logged out, 12 3760 133 15 9
 dec/09/2002 18:11:26 <pppoe-test>: terminating... - disconnected
 dec/09/2002 18:11:26 <pppoe-test>: disconnected
```

The last line is the accounting that is printed when the connection is terminated. This line indicates that the user **test** connection has terminated at **dec/09/2002 18:11:26**. The numbers following the **test logged out** entry represent the following:

```
12          session connection time in seconds
3760        bytes-in (from client)
133         bytes-out (to client)
15          packets-in (from client)
9           packets-out (to client)
```

Authentication using RADIUS Server

RADIUS Overview

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, and PPTP connections – no ISDN remote access support currently. Features supported:

- PPP remote address set from RADIUS server
- Time limit of connections set from RADIUS server
- MAC address (PPPoE) or remote client IP address (PPTP) reported to RADIUS server
- System identity
- Traffic accounting (PPP style – no IP pairs)

Note that if RADIUS server is used, then resulting settings for the client are taken from the RADIUS server and from the default profile so that settings received from the RADIUS server will always override corresponding settings taken from the default profile

RADIUS Client Setup

To use RADIUS client, enable it and set the appropriate parameters:

```
[admin@MikroTik] ppp radius-client> set enabled=yes primary-server 10.10.1.1 shared-secret us
[admin@MikroTik] ppp radius-client> print
                enabled: yes
             accounting: yes
```

```
          primary−server: 10.10.1.1
        secondary−server: 0.0.0.0
           shared−secret: "users"
     authentication−port: 1812
         accounting−port: 1813
          interim−update: 0s
[admin@MikroTik] ppp radius−client>
```

Description of the output:

enabled − (yes / no) Status of RADIUS client
accounting − (yes / no) Status of RADIUS accounting
primary−server − Primary RADIUS server
secondary−server − Secondary RADIUS server
shared−secret − corresponding text string from RADIUS server
accounting−port − accounting−port
authentication−port − default port 1645 according to RFC
interim−update − 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

RADIUS Client Monitor

The RADIUS client can be monitored using **monitor** command, for example:

```
[admin@MikroTik] ppp radius−client> monitor
             pending: 0
            requests: 2
             accepts: 1
             rejects: 0
         bad−replies: 0
    last−request−rtt: 0s

[admin@MikroTik] ppp radius−client>
```

Counters can be reset using the **reset−counters** command. Similar monitor is for HotSpot Radius client as well.

RADIUS Parameters

Authentication data sent to server (Access−Request)

```
Service−Type            always is Framed

Framed−Protocol         always is PPP

NAS−Identifier          router identity

NAS−Port−Type           Async (for async PPP)
                        Virtual (for PPTP)
                        Ethernet (for PPPoE)
                        ISDN Sync (for ISDN)

Calling−Station−Id      client MAC address (with CAPITAL letters) (for PPPoE)
                        client public IP address (for PPTP)

Called−Station−Id       service name (for PPPoE)
                        server IP address (for PPTP)
                        interface MSN (for ISDN)
```

```
NAS-Port-Id              serial port name (for async PPP)
                         ethernet interface name on which server is running (for PPPoE)

User-Name                client login name
```

Depending on authentication methods:

```
User-Password            encrypted password (used with PAP auth.)

CHAP-Password,
CHAP-Challenge           encrypted password and challenge (used with CHAP auth.)

MS-CHAP2-Response,
MS_CHAP-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. 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.

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
                         incoming or outgoing interface is client PPP, PPTP, PPPoE
                         interface. 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
MS-MPPE-Encryption-Type    use-encryption parameter. Non 0 value means use encryption

Ascend-Data-Rate         tx/rx data rate limitation (for PPPoE). If multiple
                         attributes are provided, first limits tx data rate,
                         second - rx data rate. 0 if unlimited.

MS-CHAP2-Success         auth. response if MS-CHAPv2 was used

MS-MPPE-Send-Key
  and MS-MPPE-Recv-Key   encryption keys for encrypted PPP, PPTP and PPPoE,
                         provided by RADIUS server only is MS-CHAPv2 was
                         used as authentication (for PPP, PPTP, PPPoE only)
```

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
Framed-Protocol          same as in request
```

```
NAS-Identifier          same as in request
User-Name               same as in request
NAS-Port-Type           same as in request
NAS-Port-Id             same as in request
Calling-Station-Id      same as in request
Called-Station-Id       same as in request
Acct-Authentic          authenticated by whom
Framed-IP-Address       IP address given to the user
Class                   RADIUS server cookie
```

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 Servers Suggested

MikroTik RouterOS RADIUS CLIENT should work well with all RFC compliant servers. It has been tested with:

Vircom RADIUS http://www.vircom.com/
Livingston RADIUS 2.1 http://www.livingston.com/

PPPoE Bandwidth Setting

For local authentication, this can be set in the **/ppp profile** menu with the **tx−bit−rate** and **rx−bit−rate** values (identical to bits/s). For Radius authentication, the account of each user in the radius server should be set with: Parameter: Ascend−Data−Rate (vendor id: 529, attribute id: 197 −− in bits/s).

PPP Troubleshooting

- *I am using RADIUS authentication. After abnormal connection loss between the PPP, or PPTP, or PPPoE client and MikroTik server I cannot reconnect because of wrong username/password.*
  The problem might be in the RADIUS server, which has kept the client state as 'connected'. If only one connection per client is allowed, the second connection is not authenticated.
- *My link between the PPPoE client and the MikroTik Access Concentrator not always is stable, and the Windows PPPoE clients get disconnected.*
  Set the Redialing Options of the Windows client to "Redial if line is dropped = yes" and "Time between redial attempts = 1s".

RADIUS Server Configuration Example

Below are general steps for configuring RADIUS server under UNIX. Let us assume you have downloaded a server installation, installed it, and the service is running.

1. Check what ports are used for RADIUS authentication and accounting. You can use 'netstat −l' or 'netstat −ln' command, for example:

   ```
   [root@server home]# netstat -ln
   ```

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:110             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:1812            0.0.0.0:*
udp        0      0 0.0.0.0:1813            0.0.0.0:*
...
```

2. Make sure your RADIUS clients are listed in the clients file. It should contain client's IP address or hostname, and secret key, for example:

```
[root@server raddb]# cat clients
#Client Name            Key
#----------------       -------------------
10.5.15.4               rm219pppoe-radius
10.5.6.5                a-hotspot-radius
10.0.0.100              artis-secret
[root@server raddb]#
```

3. Make sure the RADIUS attributes used are included in the dictionary file containing dictionary translations for parsing requests and generating responses. For example, for vendor specific attributes of Ascend and Mikrotik, the dictionary file should contain lines:

```
[root@server raddb]# cat dictionary
...
VENDOR      Ascend      529
VENDOR      Mikrotik    14988

#
#   Bandwidth limitation (in bits/s)
#
ATTRIBUTE   Ascend-Data-Rate        197 integer     Ascend

#
#   Traffic limitation (in bytes)
#
ATTRIBUTE   Mikrotik-Recv-Limit     1   integer     Mikrotik
ATTRIBUTE   Mikrotik-Xmit-Limit     2   integer     Mikrotik
[root@server raddb]#
```

4. All users should be listed in the 'users' file, for example:

```
[root@server raddb]# cat users
randy           Password = "w7fxc"
                Service-Type = Framed-User,
                Framed-Protocol = PPP,
                Framed-IP-Address = 10.5.13.19,
                Ascend-Data-Rate = 64000,

monica          Password = "bil"
                Service-Type = Framed-User,
                Framed-Protocol = PPP,
[root@server raddb]#
```

5. If you have changed RADIUS server settings, most probably you have to restart the RADIUS daemon (see instructions for it). For example, you have to issue command on your server:

```
[root@server raddb]# /etc/rc.d/init.d/radiusd restart
Shutting down radiusd: [  OK  ]
Starting radiusd: [  OK  ]
[root@server raddb]#
```

Remember, that users included in router's ppp secret list are not authenticated using the RADIUS server!

To troubleshoot your RADIUS server and client setup,

1. use **/ppp radius−client monitor**, or **/ip hotspot radius−client monitor** commands,
2. examine RADIUS server log files.

Point to Point Protocol (PPP) and Asynchronous Interfaces

Document revision 29−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

PPP (or Point−to−Point Protocol) provides a method for transmitting datagrams over serial point−to−point links. The 'com1' and 'com2' ports from standard PC hardware configurations will appear as **serial0** and **serial1** automatically. You can add more serial ports to use the router for a modem pool using these adapters:

- MOXA (http://www.moxa.com/) Smartio C104H 4−port PCI multiport asynchronous board with maximum of 16 ports (4 cards)
- MOXA (http://www.moxa.com/) Smartio C168H 8−port PCI multiport asynchronous board with maximum of 32 ports (4 cards)
- Cyclades (http://www.cyclades.com/) Cyclom−Y Series PCI multiport asynchronous (serial) cards
- Cyclades (http://www.cyclades.com/) Cyclades−Z Series PCI multiport asynchronous (serial) cards
- TCL (http://www.thetcl.com/) DataBooster 4 or 8 port High Speed Buffered PCI Communication Controllers

General PPP settings that are used for PPP, PPTP, and PPPoE connections are described in General Point to Point Setting manual.

Contents of the Manual

The following topics are covered in this manual:

Installation

The **ppp−2.6.x.npk** are required. The package can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload them to the router with ftp and reboot.

Hardware Resource Usage

PPP uses a minimum amount of memory.

If the devices are detected correctly, they should appear in driver list:

```
[admin@MikroTik] > driver print
Flags: I − invalid, D − dynamic
  #   DRIVER                               IRQ IO       MEMORY   ISDN−PROTOCOL
  0 D Cyclades Y/Z
```

```
  1 D RealTek 8139
  2 D TheTCL DataBooster
  3 D Intel PRO/100
[admin@MikroTik] >
```

To see the list of available serial ports, use the command **/ports print**, for 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] >
```

Serial Port Configuration

You can set parameters for each port using **/port set** command:

```
[admin@MikroTik] port> set serial0 baud-rate=57600
[admin@MikroTik] port> print detail from=serial0
  0 name="serial0" used-by="Serial Console" baud-rate=57600 data-bits=8
    parity=none stop-bits=1 flow-control=hardware


[admin@MikroTik] port>
```

Description of the printout:

name − port name
used−by − shows (cannot be changed) the user of the port. Only free ports can be used in PPP
baud−rate − maximal data rate of the port (**50** ... **4000000**)
data−bits − number of bits per character transmitted (**7**, **8**)
parity − character parity check method (**none**, **even**, **odd**)
stop−bits − number of stop bits after each character transmitted (**1**, **2**)
flow−control − flow control method (**none**, **hardware**, **xon−xoff**)

Note that **baud−rate**, **data−bits**, **parity**, **stop−bits** and **flow−control** parameters must be the same for both communicating sides.

PPP Server

The PPP server management is done in the **/interface ppp−server**submenu.

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> pr
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
        user:
      uptime: 0s
    encoding:
      status: Waiting for call...

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

Description of settings:

port − Serial port
authentication − Authentication protocol. One or more of: **mschap2**, **chap**, **pap**. Encrypted links are only supported when ms−chapv2 is selected. This is a feature of the protocol. It is suggested that only **mschap2** is selected, unless there is a special situation which requires an unencrypted link
profile − profile name for the link
mtu − Maximum Transmit Unit. Maximum packet size to be transmitted
mru − Maximum Receive Unit.
null−modem − Enable/Disable null−modem mode (when enabled, no modem initialization strings are sent). Default value is "off" (for COM1 and COM2 only). So by default null−modem is turned off.
modem−init − Modem Initialization String.
ring−count − Number of rings to wait before answering phone.
name − Interface name for reference.

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.

PPP Client Setup

PPP profiles must match at least partially (**local−address** and values connected with encryption should match) with corresponding remote server values.

The PPP client management can be accessed under the **/interface ppp−client** submenu.

You can add a PPP client using the **add** command:

```
[admin@MikroTik] interface ppp−client> add
creates new item with specified property values.
  add−default−route  Add PPP remote address as a default route
          copy−from  item number
     dial−on−demand  Enable/Disable dial on demand
           disabled
         modem−init  Modem init string
                mru  Maximum Receive Unit
                mtu  Maximum Transfer Unit
               name  New interface name
         null−modem  Enable/Disable nullmodem mode
           password
```

```
             phone  Phone number for dialout
              port  Serial port
           profile
         tone-dial  Enable/Disable tone dial
      use-peer-dns  Enable/Disable using of peer DNS
              user  User name to use for dialout
[admin@MikroTik] interface ppp-client> add name=test user=test port=setial1 \
\... 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 test2
      uptime: 0s
    encoding:
      status: Logging in to network...

[admin@MikroTik] interface ppp-client>
```

Descriptions of settings:

name − new interface name
port − serial port
user − P2P user name on the remote server to use for dialout
password − P2P user password on the remote server to use for dialout
profile − local profile to use for dialout
phone − phone number for dialout
tone−dial − defines whether use tone dial or pulse dial
mtu − Maximum Transmit Unit. Maximum packet size to be transmitted
mru − Maximum Receive Unit
null−modem − enable/disable null−modem mode (when enabled, no modem initialization strings are sent). Default value is **off** (for COM1 and COM2 only). So by default null−modem is turned off
modem−init − Modem Initialization String
dial−on−demand − enable/disable dial on demand
add−default−route − add PPP remote address as a default route
use−peer−dns − use DNS server settings from the remote server

If the PPP client is configured properly and it has established a connection to the server, you can:

1. Monitor the connection using the **/interface ppp−client monitor** command
2. See the ppp−out interface under the **/interface print** list
3. See the dynamic IP address under the **/ip address print** list
4. (Optionally) See the dynamic default route under the **/ip route print** list

Example of an established connection:

```
[admin@MikroTik] interface ppp-client> monitor test
      uptime: 4h35s
    encoding: none
      status: Connected
[admin@MikroTik] interface ppp-client>
```

Description of display:

uptime − connection time displayed in days, hours, minutes, and seconds
encoding − encryption being used in this connection
status − the status of this client may be:

- ♦ **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

Additional Resources

Links for PPP documentation:

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 23−Dec−2002
This document applies to MikroTik RouterOS V2.6

Overview

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 over 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 802.11 (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.

Our RouterOS has a RADIUS client that can be used for authentication of all PPP type connections – including PPPoE. For more information on PPP authentication, see the General Point to Point Settings manual.

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)

Topics covered in this manual:

PPPoE Installation on the MikroTik RouterOS

The **pppoe−2.6.x.npk** package and the **ppp−2.6.x.npk** are required. The packages can be downloaded from MikroTik's web page www.mikrotik.com . To install the packages, please upload them to the router with ftp and reboot.

PPPoE hardware resource usage

The PPPoE client uses a minimum amount of memory.

The PPPoE server (access concentrator) uses a minimum amount of memory for the basic setup. Each current PPPoE server connection uses approximately 100−200KB of memory. For PPPoE servers (access concentrators) designed for a large number of PPPoE connections, additional RAM should be added. In version 2.6, there is currently a maximum of 5000 connections. For example, a 1,000 user system should have 200MBs of free RAM above the normal operating RAM. For large number of clients a faster processor system is required. We recommend to use a Celeron 600MHz processor or higher. A future rewrite of parts of PPP is expected to significantly reduce the requirements.

PPPoE Client Setup

The PPPoE client supports high−speed connections. It is fully compatible with the MikroTik PPPoE server (access concentrator). Test with different ISPs and access concentrators are currently underway.

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.

An example of a PPPoE client on the MikroTik RouterOS:

```
[admin@RemoteOffice] interface pppoe−client> print
Flags: X − disabled, R − running
  0 X  name="pppoe−out1" mtu=1460 mru=1460 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
```

Descriptions of settings:

name − this settable name will appear in interface and IP address list when the PPPoE session is active.
interface − interface through which the PPPoE server can be connected. The PPPoE client can be attached to any Ethernet like interface – for example: wireless, 10/100/1000 Ethernet, and EoIP tunnels.
mtu and mru − represents the MTU and MRU when the 8 byte PPPoE overhead is subtracted from the standard 1500 byte Ethernet packet. For encryption, subtract four more bits and set the MTU and MRU to 1488
user − a user name that is present on the PPPoE server
password − a user password used to connect the PPPoE server
profile − default profile for the connection
service−name − 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 − This may be left blank and the client will connect to any access concentrator that offers the “service” name selected
add−default−route − Select yes to have a default route added automatically. Note, the dynamic default route will not be added if there is already a default route set
dial−on−demand − 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 − Sets the router default DNS to the PPP peer DNS.

PPPoE Server Setup (Access Concentrator)

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 setting below is the optimal setting to work with Windows clients such as RASPPPoE client for all versions of Windows greater than 3.x. The password authentication and encryption are set to **authentication=chap** specifically to ensure a quick login by the windows client. In the example below, the login is encrypted with PAP.

The access concentrator has a hard limit of 5000 current connections. The user setting for the connections limit is done by setting the IP pools in the **remote−address** configuration.

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.

```
[admin@MikroTik] interface pppoe−server> server print
Flags: X − disabled
  0 X service−name="office" interface=prism1 mtu=1492 mru=1492
      authentication=chap keepalive−timeout=10 default−profile=default

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

Descriptions of settings:

service−name − The PPPoE service name
mtu, mru − The default MTU nad MRU is set to 1480, but the maximum values they can be set to on the ethernet interface is 1492 because of the PPPoE overhead. For encryption, subtract four more bits and set the MTU and MRU to 1488
authentication − authentication algorithm. One or more of: **mschap2**, **chap**, **pap**
keepalive−timeout − defines the time period (in seconds) after which not responding client is proclaimed disconnected. The default 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
default−profile − default profile to use for the clients

Security issue: do not assign an IP address to the Interface you will be receiving the PPPoE requests on.

The PPPoE server will create point−to−point connection for each individual client. Each connection will have individual dynamic (virtual) P2P interface. The **local−address** will be set on its server side, and the **remote−address** will be given to the client. The addresses do not need to be from 'the same network', since the P2P connections have addresses with 32 bit netmasks anyway. What you set on the server side does not matter so much − it can be address of router's another interface, or some arbitrary address.

Please consult General Point to Point Settings manual on authorization, filtering and accounting settings.

Please see the IP Addresses and Address Resolution Protocol (ARP) Manual how to give out addresses to PPPoE clients from the same address space you are using on your local network.

PPPoE bandwidth setting

For local authentication, this can be set in the **/ppp profile** menu with the **tx−bit−rate** and **rx−bit−rate** values (identical to bits/s). For Radius authentication, the account of each user in the radius server should be set with:

```
Parameter: Ascend−Data−Rate (vendor id: 529, attribute id:197 -- in bits/s)
```

If there is one attribute sent then both tx and rx are set to that rate in b/s. If there two attributes sent then the first will be the tx and the second will be the rx (in bits/s). This means you need to add two lines to your radius attributes if you want to set tx and rx to different speeds.

PPPoE in a multipoint wireless 802.11b network

In a wireless network, the PPPoE server may be attached to our PRISMII 2.4GHz Access Point (station mode) interface. 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.

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.
- ***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.

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 28−Dec−2002
This document applies to the MikroTik RouterOS V2.6

Overview

PPTP (Point to Point Tunnel Protocol) supports encrypted tunnels over IP. The MikroTik RouterOS implementation includes a PPTP client and a PPTP server.

General usage 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)

Our RouterOS has a RADIUS client that can be used for authentication of all PPP type connections – including PPTP. For more information on PPP authentication, see the General Point to Point Settings manual.

Contents of the Manual

The following topics are covered in this manual:

Installation

The **pptp−2.6.x.npk** package and the **ppp−2.6.x.npk** are required. The package can be downloaded from MikroTik's web page www.mikrotik.com. To install the packages, please upload them to the router with ftp and reboot. You may check to see if the PPTP and PPP packages are installed with the command **system package print**

Hardware Resource Usage

PPTP uses a minimum amount of memory. RouterOS V2.6 is tested to have approximated encrypted throughput of 60Mb/s on a Celeron 600MHz CPU.

PPTP Protocol Description

Though the following may sound complex, our implementation of PPTP is easy to setup and manage. 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 1) routers and routers 2) routers and PPTP clients (clients are available for 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. There are also additional PPP configurations for management of users and connections can be found in General Point to Point Settings manual.

MPPE 40bit RC4 and MPPE 128bit RC4 encryption are supported.

PPTP traffic uses TCP port 1723 and 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

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.

The PPTP client management can be accessed under the **/interface pptp−client** submenu.

You can add a PPTP client using the **add** command:

```
[admin@MikroTik] interface pptp−client> add
creates new item with specified property values.
  add−default−route
         connect−to  PPTP server address
          copy−from  item number
           disabled
                mru  Maximum Receive Unit
                mtu  Maximum Transfer Unit
               name  New interface name
           password
            profile
               user  User name to use for dialout
[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
[admin@MikroTik] interface pptp−client> monitor test2
      uptime: 0s
    encoding:
      status: Terminated
```

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

Descriptions of settings:

name − interface name for reference
mtu − 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 − 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 − the IP address of the PPTP server to connect to
user − user name to use when logging on to the remote server
password − user password to use when logging to the remote server
profile − profile to use when connecting to the remote server
add−default−route − When the PPTP connection is up, the default route (gateway) will be added using as gateway the other side of the PPP link.

If the PPTP client is configured properly and it has established a connection to the server, you can:

1. Monitor the connection using the **/interface pptp−client monitor** command
2. See the pptp−out interface under the **/interface print** list
3. See the dynamic IP address under the **/ip address print** list
4. (if **add−default−route** is set to **yes**) See the dynamic default route under the **/ip route print** list

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>
```

Description of display:

uptime − Connection time displayed in days, hours, minutes, and seconds
encoding − Encryption being used in this connection
status − The status of this client may be:

- **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

PPTP Server Setup

The PPTP server supports unlimited connections from clients. For each current connection, a dynamic interface is created.

The PPTP server management can be accessed under the **/interface pptp−server server** submenu.

You can enable the PPTP server using the **set** command:

```
[admin@MikroTik] interface pptp-server>
Tunneling means encapsulating data of one protocol type within another
protocol and sending it over a channel that understands the encapsulating
protocol. This particular tunneling driver implements encapsulation of PPP
within IP. See also general ppp server settings.

    print  Show PPTP interfaces
      get  get value of item's property
     find  Find interfaces
      set  Change interface properties
      add  create new item
   remove  Remove interface
   enable  enables items
  disable  disables items
   server
   export
[admin@MikroTik] interface pptp-server> server
[admin@MikroTik] interface pptp-server server>

   print
     get  get value of property
     set
  export
[admin@MikroTik] interface pptp-server server> print
            enabled: no
                mtu: 1460
                mru: 1460
     authentication: mschap2
    default-profile: default
[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>
```

Descriptions of settings:

enabled − defines whether PPTP server is enabled or not
mtu − 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 − 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 − authentication algorithm. One or more from: **mschap2**, **chap**, **pap**
default−profile − default profile to use

Please consult General Point to Point Settings manual on authorization, filtering and accounting settings.

There are two types of items in PPTP server configuration − static users and dynamic connections. A dynamic connection can be established when the **default−profile** parameter is set to the profile, which have 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. Static users may be added as follows:

```
[admin@MikroTik] interface pptp-server> add
creates new item with specified property values.
  copy-from  item number
```

```
   disabled
       name  New interface name
       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>
```

Note that in both cases P2P users must be configured properly. Description of the printout:

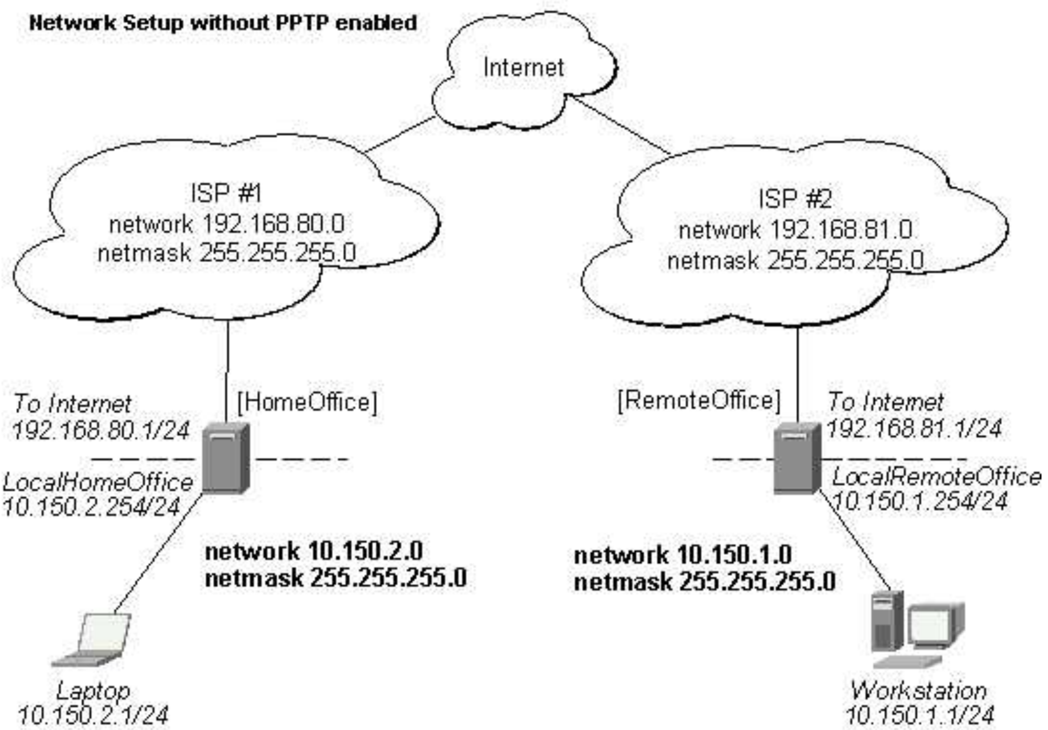
name − interface name
user − the name of the user that is configured statically or added dynamically
mtu − shows (cannot be set here) client's MTU
client−address − shows (cannot be set here) the IP of the connected client
uptime − shows how long the client is connected
encryption − shows (cannot be set here) what encryption algorithm is used for the link

If the PPTP server is configured properly and it has established connections with the clients, you can:

1. See the list of connected clients using the **/interface pptp−server print** command
2. See the pptp−in interfaces under the **/interface print** list
3. See the dynamic IP addresses under the **/ip address print** list
4. See the dynamic routes under the **/ip route print** list

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>
```

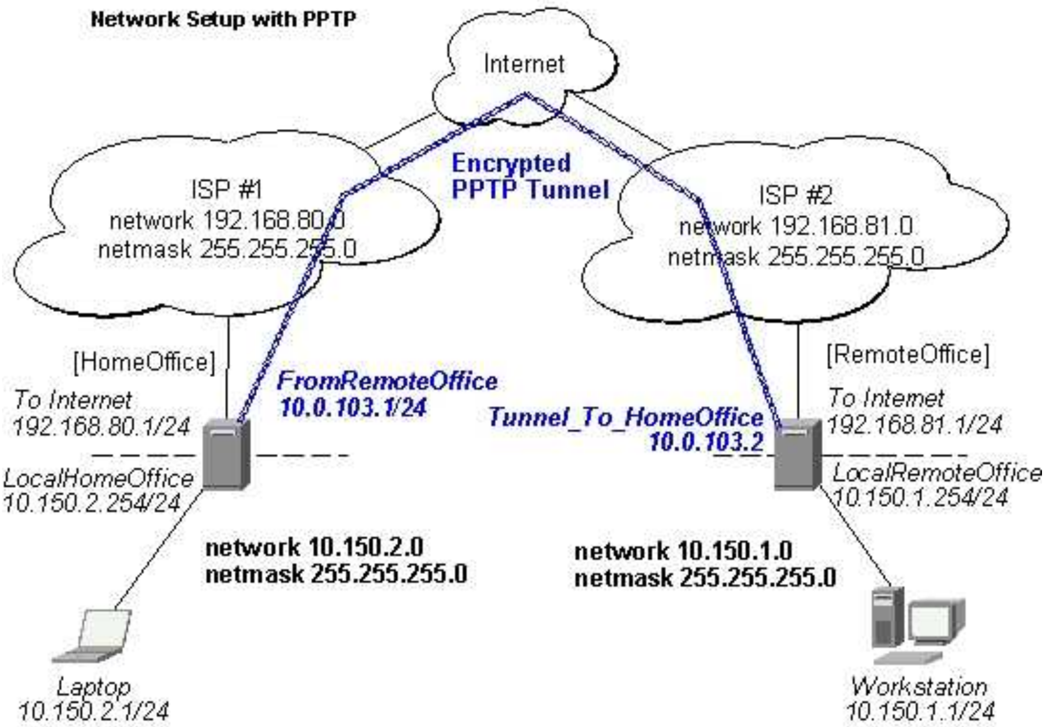
And finally, the server must be enabled:

```
[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:

```
[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
```

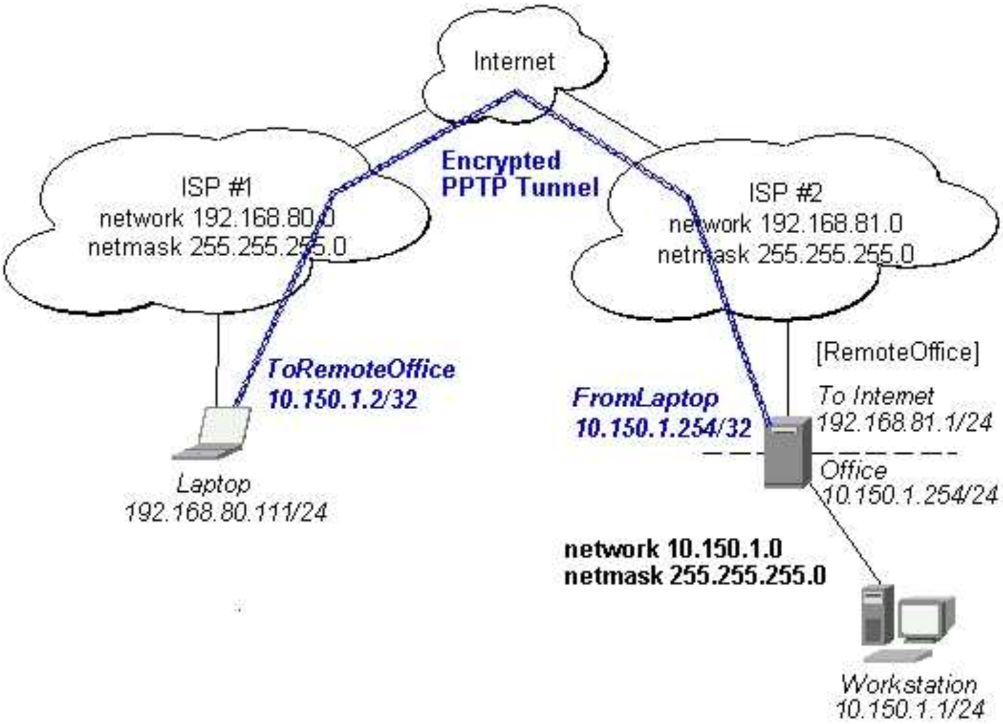
```
ping interrupted
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3/3.0/3 ms
```

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.

Links:

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 25−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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). See the list of supported Pirsm II chipset based hardware at the end of the document.

Both PCI and PCMCIA card types are supported.

For more information about adapter hardware please see the relevant User's Guides and Technical Reference Manuals of the hardware manufacturers.

Check Notes on PCMCIA Adapters for more information on PCMCIA adapters.

Contents of the Manual

The following topics are covered in this manual:

Supported Network Roles

Wireless Client

The Prism interface can be configured to act as an IEEE 802.11b wireless client (station) to associate 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

The Prism interface can be configured to act as an IEEE 802.11b wireless access point. It requires the Prism 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.

An Additional Feature License is required to enable your Access Point feature. The Wireless Client License is required as well. **AP mode** can be enabled **only** for these cards:

IEEE 802.11b 2.4GHz 11Mbps Prism II, Prism 2.5 Cards

IEEE 802.11a 5.2GHz 54Mbps Atheros Cards

The PrismII Access Point interface can register other access points. Thus, it is possible to bridge networks over wireless links.

Wireless Bridge

This is limited version of the Access Point mode which allows only one client to be registered but does not require the Prism 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.

Installation

The MikroTik Router should have the prism software package installed. The software package file **prism−2.6.x.npk** can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer. After successful installation the package should be listed under the installed software packages list.

License

The PrismII chipset based adapters, like other 2.4GHz wireless adapters, require the 2.4GHz wireless feature license. One license is for one installation of the MikroTik RouterOS, disregarding how many cards are installed in one PC box. The wireless feature is not included in the Free Demo or Basic Software License. The 2.4GHz Wireless Feature cannot be obtained for the Free Demo License. It can be obtained only **together** with the Basic Software License.

Note! The **2.4GHz Wireless Feature License** enables only the **station** or **bridge** mode of the Prism II card.
To enable the **ap−bridge mode**, additionally the **Wireless AP Feature License** is required.

The MikroTik RouterOS supports as many PrismII chipset based cards as many free resources are on your system, i.e., IRQs and adapter slots, **but not more than 6**. One license is valid for all cards on your system.

System Resource Usage

Before installing the wireless adapter, please check the availability of free IRQ's and I/O base addresses. A system with installed PrismII card and Ricoh PCMCIA−PCI adapter reports, for example, the following:

```
[admin@MikroTik] > system resource irq print
Flags: U − unused
   IRQ OWNER
   1   keyboard
   2   APIC
 U 3
   4   serial port
 U 5
 U 6
 U 7
 U 8
   9   ether1
 U 10
   11  PCMCIA service
   11  [prism2_cs]
 U 12
 U 13
   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
 100−13F            [prism2_cs]
 1F0−1F7            IDE 1
 2F8−2FF            serial port
 3C0−3DF            VGA
 3F6−3F6            IDE 1
 3F8−3FF            serial port
 CF8−CFF            [PCI conf1]
 EF00−EFFF          [Realtek Semiconductor Co., Ltd. RTL−8139]
 EF00−EFFF          [8139too]
 FC00−FC7F          [Cyrix Corporation 5530 IDE [Kahlua]]
 FC00−FC07          IDE 1
 FC08−FC0F          IDE 2
[MikroTik] >
```

Installing the Wireless Adapter

The basic installation steps of the wireless adapter should be as follows:

1. Check the system BIOS settings and make sure you have the **PnP OS Installed** set to **Yes**.
2. 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.

Loading the Driver for the Wireless Adapter

PCI and PC (PCMCIA) cards do not require a 'manual' driver loading, since they are recognized automatically by the system and the driver is loaded at the system startup. The Prism driver is not shown under the **/driver** list. If you have wireless feature license, prism interface should show up under the **/interface** list.

There can be several reasons for a failure to load the driver, for example:

- The driver cannot be loaded because there are too many PCMCIA slots on Your system (more than 8).
  Consult the driver manual: Notes on PCMCIA Adapters
- The driver cannot be loaded because other device uses the requested IRQ.
  Try to set the IRQ assignment to PCI slots using the system BIOS configuration.

Usually two consecutive beeps of high tone can be heard during the startup of the MikroTik RouterOS router with PCMCIA PrismII card. If the second beep has a lower tone, or there is only one lower tone beep, most likely there is a compatibility problem with the motherboard. Try to use another type of motherboard.

Wireless Interface Configuration

If the driver has been loaded successfully, and you have the required 2.4GHz Wireless Software License, then the Prism II 2.4GHz Wireless interface should appear under the **/interface** list with the name prismX, where X is 1,2,... You can change the interface name to a more descriptive one using the **set** command. To enable the interface, use the **enable** command:

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

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

```
[admin@MikroTik] interface prism> print
Flags: X − disabled, R − running
  0    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="mikrotik"
       default−authentication=yes default−forwarding=yes max−clients=2007
       card−type=generic tx−power=auto supported−rates=1−11 basic−rates=1

[admin@MikroTik] interface prism>
```

Argument description:

name − Interface name (same as for other interfaces)
mtu − Maximum transfer unit (same as for other interfaces)
mac−address − MAC address of card. In AP mode this will also be BSSID of BSS.
arp − 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.

mode − 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 − (only **ap−bridge** or **bridge**) MAC address of the root access point to register to.
frequency − (only **ap−bridge** or **bridge**) Frequency that AP will use to create BSS
ssid − Service Set Identifier. In station mode − ssid to connect to, in AP and P2P mode − ssid to use when creating BSS (this can not be left blank).
default−authentication − (only **ap−bridge** or **bridge**) What to do with client that wants to associate, but it is not in the access−list.
default−forwarding − (only **ap−bridge** or **bridge**) What to do with client that wants to send packets to other wireless clients, but it is not in the access−list.
max−clients − (only **ap−bridge** or **bridge**) Maximum number of clients (including other access points), that is allowed to associate with this access point (1...2007).
card−type − Card type used for power settings (**100mW**, **200mW**, **30mW**, **generic**, default is **generic**)
tx−power − Transmit power level (**0dBm−1mW**...**23dBm−200mW** / auto). Has no effect if card type is **generic**. **auto** means default setting of the card.
supported−rates − Rates at which this node will work.
basic−rates − (only **ap−bridge** or **bridge**) 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**.

Station Mode Configuration

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 **station**.
- The **Supported Rate** 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. To configure the wireless interface for registering to an AP with ssid "testing", it is enough to change the argument value of ssid to "testing" and to enable the interface:

```
[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    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

[admin@MikroTik] interface prism>
```

Note for CISCO/Aironet Wireless Bridge and Access Point users
When working with Prism II chipset based clients, the CISCO/Aironet Wireless Bridge or AP should have the following settings:
− the Proprietary Extensions should be turned 'off' under Configuration/Radio/802.11 menu
− the Encapsulation Protocol should be RFC1042 under

Configuration/Radio/802.11/Encapsulation menu

Monitoring the Interface Status

In station mode, the prism interface status can be monitored using the **/interface prism monitor** command:

```
[admin@MikroTik] interface prism> monitor 0
                      status: connected−to−ess
                   data−rate: 11Mbps
                        ssid: "testing"
                       bssid: 00:03:2F:04:25:10
              signal−quality: 92
                signal−level: 54
                 noise−level: −99

[admin@MikroTik] interface prism>
```

Argument description:

status − 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 − the actual data rate of the connection.
ssid − the Service Set Identifier.
bssid − the Basic Service Set Identifier (actually, the MAC address of the access point).
signal−quality − the signal quality (0−92).
signal−level − the average signal level (27−154).
noise−level − the average noise level (−100−0).

The monitor command does not work, if the interface is disabled, or the mode is **ap−bridge** or **bridge**.

Access Point Mode Configuration

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**. It should be unique for your system.
- The **Operation 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.

To configure the wireless interface for working as an access point with ssid "testing" and use the frequency 2442MHz, it is enough to enter the command:

```
[admin@MikroTik] interface prism> set prism1 mode=ap−bridge frequency=2442 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

[admin@MikroTik] interface prism>
```

Use the registration table to see the associated clients.

Registration Table

The registration table shows all clients currently associated with the access point, for example:

```
[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>
```

Argument description for the registration−table entry:

interface − interface that client is registered to
mac−address − mac address of the registered client
type − 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 − parent access point's MAC address, if forwarded from another access point

The **print stats** or **print detail** commands give additional per−client 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>
```

Additional argument description (only for wireless clients):

packets − number of received and sent packets
bytes − number of received and sent bytes
signal−level − min/average/max signal level
noise−level − min/average/max noise level
data−rate − min/average/max receive data rate
tx−rate − transmit data rate
last−update − time since the last update
uptime − time the client is associated with the access point

Access List

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.

To add an access list entry, use the **add** command, for example:

```
[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>
```

Argument description:

mac−address − MAC address of the client
interface − AP interface
authentication − accept this client when it tries to connect or not
forwarding − forward the client's frames to other wireless clients or not

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'.

Registering the Access Point to another Access Point

You can configure the access point to registering to another (root) access point by specifying the MAC address of the 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


[admin@MikroTik] interface prism>
```

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.

Network Scan

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.

Use the **/interface prism scan** command to scan for available networks, for example:

```
[admin@MikroTik] interface prism> scan
Scan for wireless networks
  <interface>
  frequencies  List of frequencies to scan
         time  Time to scan one frequency
[admin@MikroTik] interface prism> scan prism1
00:02:6f:01:5d:fe fequency=2412MHz ssid=waubonsie_low_ap1 signal−level=132
00:02:6f:01:63:0b fequency=2427MHz ssid=john signal−level=114
00:02:6f:01:62:ee fequency=2462MHz ssid=sales signal−level=0
[admin@MikroTik] interface prism>
```

Argument description:

<interface> − interface name to use for scanning
frequencies − list of frequencies to scan for, e.g., **2412MHz,2427MHz**
time − time to scan for one frequency. The total time used for scanning is multiplier of this value and the number of frequencies to scan.

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.

Logging of Prism Interface

The prism interface status changes can be logged locally or to a remote syslog daemon by enabling the logging facility, for example:

```
[admin@MikroTik] system logging facility> set Prism−Info logging=local
[admin@MikroTik] system logging facility> print
  # FACILITY            LOGGING PREFIX              REMOTE−ADDRESS  REMOTE−PORT
  0 Firewall−Log        none
  1 PPP−Account         none
  2 PPP−Info            none
  3 PPP−Error           none
  4 System−Info         local
  5 System−Error        local
  6 System−Warning      local
  7 Prism−Info          local
[admin@MikroTik] system logging facility>
```

The local logs can be viewed using the **/log print** command.

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 wireless card does not register to the AP*
  Check the cabling and antenna alignment.

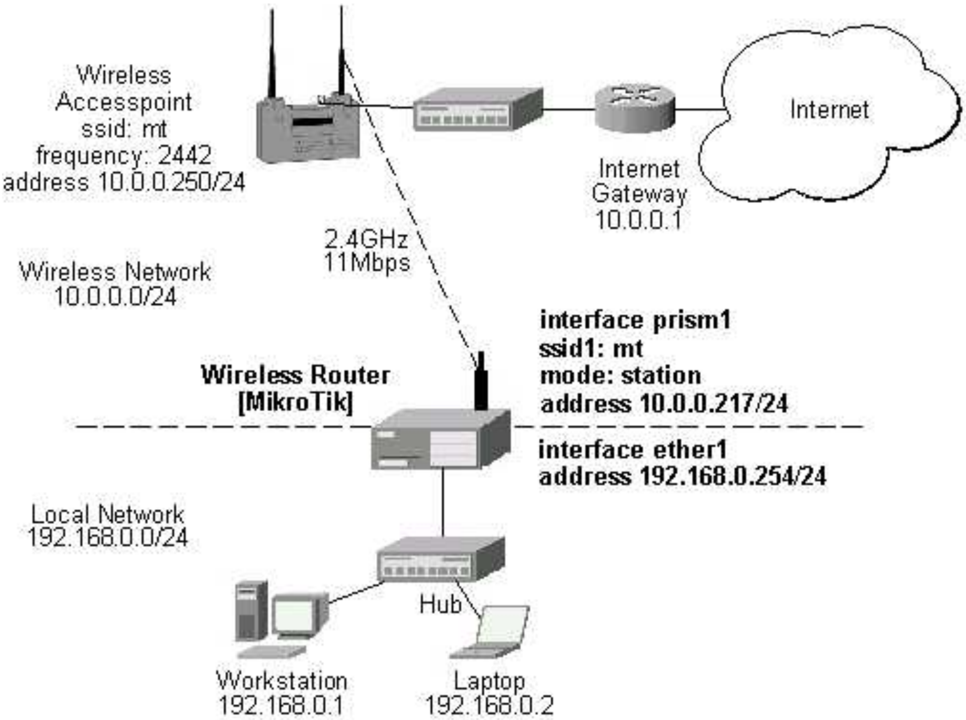
Wireless Network Applications

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

- Wireless Client
- Wireless Access Point
- Wireless Bridge

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:



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! *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!*

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> 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>
```

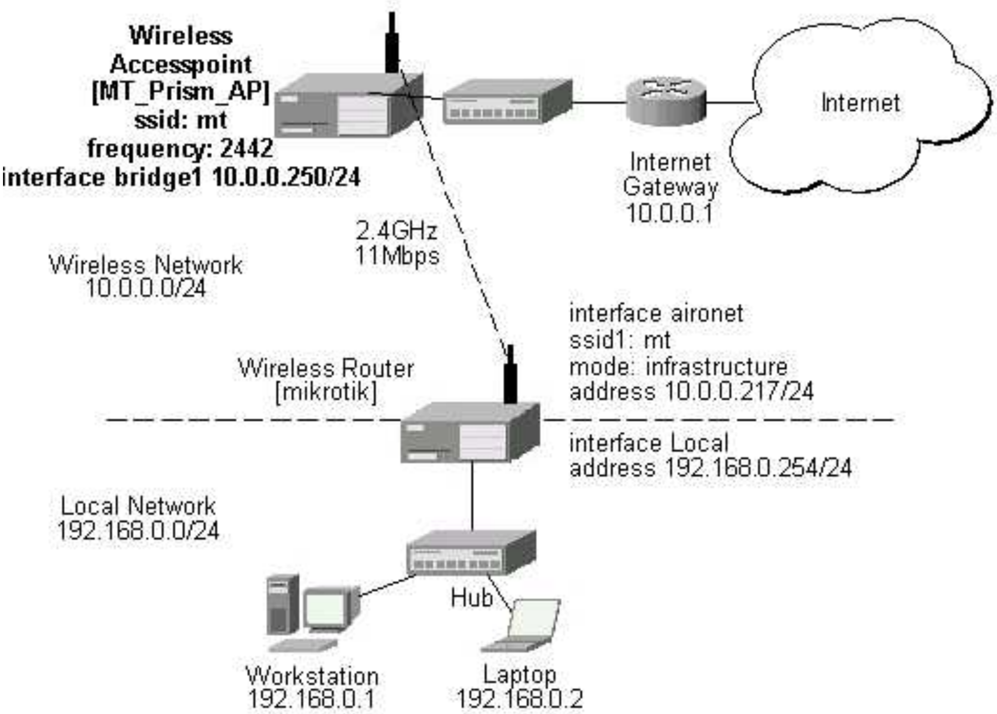
The default route should be set to the gateway router 10.0.0.1 (not to the AP 10.1.1.250 !):

```
[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

[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>
```

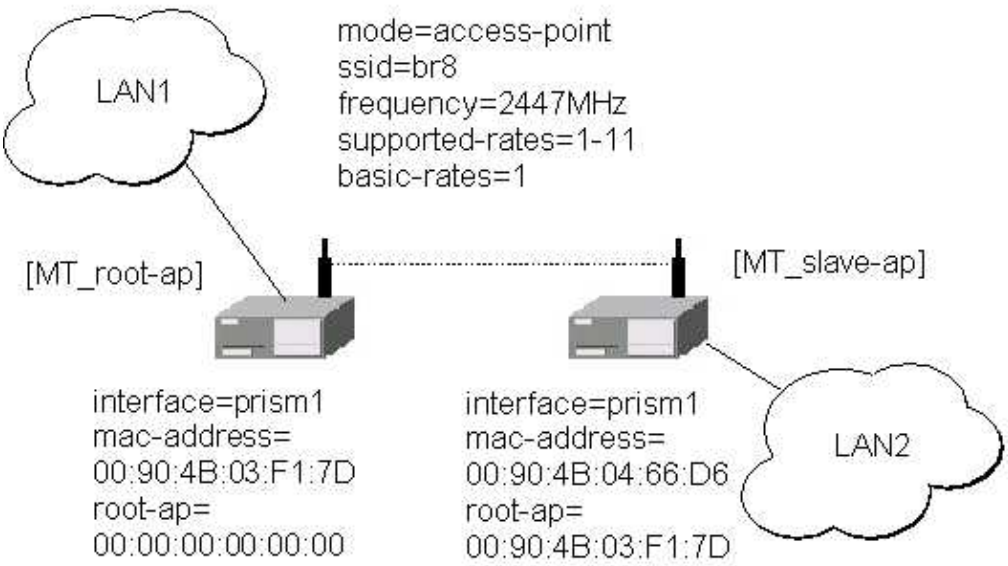
The default route should be set to gateway 10.0.0.1 for the router [mikrotik]:

```
[admin@mikrotik] ip route> add gateway=10.0.0.254
[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 10.0.0.1        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.

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@MikroTik] 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.

Supported Prism II Hardware

Many wireless cards based on the Prism 2 and above chipset use the prism reference design PCI identifier or PCI identifier of the OEM producer of the card. They do not have a unique identifier based on the brand name or company name on the PCI card. So, for many cards, it is needed to simply test and see if it is recognized.

MikroTik RouterOS supports the following PCI identifiers for the Prism 2 and above chipset based hardware:

```
card "Intersil PRISM2 Reference Design 11Mb/s 802.11b WLAN Card"
   version "INTERSIL", "HFA384x/IEEE"

card "GemTek WL−211 Wireless LAN PC Card"
   version "Wireless LAN", "11Mbps PC Card"

card "Compaq WL100/200 11Mb/s 802.11b WLzAN Card"
   manfid 0x0138, 0x0002

card "Compaq iPaq HNW−100 11Mb/s 802.11b WLAN Card"
   manfid 0x028a, 0x0002

card "Samsung SWL2000−N 11Mb/s 802.11b WLAN Card"
   manfid 0x0250, 0x0002

card "Z−Com XI300 11Mb/s 802.11b WLAN Card"
   manfid 0xd601, 0x0002

card "ZoomAir 4100 11Mb/s 802.11b WLAN Card"
   version "ZoomAir 11Mbps High", "Rate wireless Networking"

card "Linksys WPC11 11Mbps 802.11b WLAN Card"
   version "Instant Wireless ", " Network PC CARD", "Version 01.02"

card "Addtron AWP−100 11Mbps 802.11b WLAN Card"
   version "Addtron", "AWP−100 Wireless PCMCIA", "Version 01.02"

card "D−Link DWL−650 11Mbps 802.11b WLAN Card"
   version "D", "Link DWL−650 11Mbps WLAN Card", "Version 01.02"

card "SMC 2632W 11Mbps 802.11b WLAN Card"
   version "SMC", "SMC2632W", "Version 01.02"

card "BroMax Freeport 11Mbps 802.11b WLAN Card"
  version "Intersil", "PRISM 2_5 PCMCIA ADAPTER", "ISL37300P", "Eval−RevA"

card "Intersil PRISM2 Reference Design 11Mb/s WLAN Card"
   manfid 0x0156, 0x0002

card "Bromax OEM 11Mbps 802.11b WLAN Card (Prism 2.5)"
   manfid 0x0274, 0x1612

card "Bromax OEM 11Mbps 802.11b WLAN Card (Prism 3)"
   manfid 0x0274, 0x1613

card "corega K.K. Wireless LAN PCC−11"
   version "corega K.K.", "Wireless LAN PCC−11"

card "corega K.K. Wireless LAN PCCA−11"
   version "corega K.K.", "Wireless LAN PCCA−11"
```

```
card "CONTEC FLEXSCAN/FX-DDS110-PCC"
   manfid 0xc001, 0x0008

card "PLANEX GeoWave/GW-NS110"
   version "PLANEX", "GeoWave/GW-NS110"

card "Ambicom WL1100 11Mbps 802.11b WLAN Card"
   version "OEM", "PRISM2 IEEE 802.11 PC-Card", "Version 01.02"

card "LeArtery SYNCBYAIR 11Mbps 802.11b WLAN Card"
   version "LeArtery", "SYNCBYAIR 11Mbps Wireless LAN PC Card", "Version 01.02"

card "Intermec MobileLAN 11Mbps 802.11b WLAN Card"
   manfid 0x01ff, 0x0008

card "NETGEAR MA401 11Mbps 802.11 WLAN Card"
   version "NETGEAR MA401 Wireless PC", "Card", "Version 01.00"

card "Intersil PRISM Freedom 11mbps 802.11 WLAN Card"
   version "Intersil", "PRISM Freedom PCMCIA Adapter", "ISL37100P", "Eval-RevA"

card "OTC Wireless AirEZY 2411-PCC 11Mbps 802.11 WLAN Card"
   version "OTC", "Wireless AirEZY 2411-PCC WLAN Card", "Version 01.02"

card "Zcomax XI-325HP PCMCIA 200mW Card"
```

RadioLAN 5.8GHz Wireless Interface

Document revision 29−Nov−2001
This document applies to the MikroTik RouterOS V2.6

Overview

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.

Contents of the Manual

The following topics are covered in this manual:

Wireless Adapter Hardware and Software Installation

Software Packages

The MikroTik Router should have the radiolan software package installed. The software package file **radiolan−2.6.x.npk** can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer. After successful installation the package should be listed under the installed software packages list, for example:

```
[admin@MikroTik] interface> /system package print
Flags: I − invalid
  #   NAME                    VERSION              BUILD−TIME           UNINSTALL
  0   ssh                     2.6beta2             jul/05/2002 13:43:42 no
  1   radiolan                2.6beta2             jul/05/2002 13:47:46 no
  2   system                  2.6beta2             jul/05/2002 13:42:26 no
  3   vlan                    2.6beta2             jul/05/2002 14:13:43 no
  4   pptp                    2.6beta2             jul/05/2002 13:46:11 no
  5   ppp                     2.6beta2             jul/05/2002 13:45:40 no
  6   pppoe                   2.6beta2             jul/05/2002 13:46:40 no
[admin@MikroTik] interface>
```

Software License

The RadioLAN 5.8GHz wireless adapters require the RadioLAN 5.8GHz wireless feature license. One license is for one installation of the MikroTik RouterOS, disregarding how many cards are installed in one PC box. The wireless feature is not included in the Free Demo or Basic Software License. The RadioLAN 5.8GHz Wireless Feature cannot be obtained for the Free Demo License. It can be obtained only **together** with the Basic Software License.

System Resource Usage

Before installing the wireless adapter, please check the availability of free IRQ's and I/O base addresses:

```
[admin@MikroTik] interface> /system resource irq print
Flags: U - unused
   IRQ OWNER
   1   keyboard
   2   APIC
 U 3
   4   serial port
 U 5
 U 6
 U 7
 U 8
   9   ether1
 U 10
 U 11
 U 12
 U 13
   14  IDE 1
[admin@MikroTik] interface> /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
 2F8-2FF            serial port
 3C0-3DF            VGA
 3F6-3F6            IDE 1
 3F8-3FF            serial port
 CF8-CFF            [PCI conf1]
 EF00-EFFF          [Realtek Semiconductor Co., Ltd. RTL-8139]
 EF00-EFFF          [8139too]
 FC00-FC7F          [Cyrix Corporation 5530 IDE [Kahlua]]
 FC00-FC07          IDE 1
 FC08-FC0F          IDE 2
[admin@MikroTik] interface>
```

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.

For more information on installing PCMCIA cards, check Notes on PCMCIA Adapters first.

Loading the Driver for the Wireless Adapter

The ISA card requires the driver to be loaded by issuing the following command:

```
[admin@MikroTik] > driver add name=radiolan io=0x300
[admin@MikroTik] > driver print
Flags: I - invalid, D - dynamic
  #   DRIVER                             IRQ IO          MEMORY     ISDN-PROTOCOL
  0 D RealTek RTL8129/8139
  1   ISA RadioLAN                           0x300
[admin@MikroTik] >
```

There can be several reasons for a failure to load the driver:

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

Wireless Interface Configuration

If the driver has been loaded successfully (no error messages), and you have the required RadioLAN 5.8GHz Wireless Software License, then the RadioLAN 5.8GHz Wireless interface should appear under the interfaces list with the name radiolanX, where X is 1,2,... You can change the interface name to a more descriptive one using the **set** command. To enable the interface, use the **enable** command:

```
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
  #    NAME                 TYPE             MTU
  0  R ether1               ether            1500
  1 X  radiolan1            radiolan         1500
  2 X  vlan1                vlan             1500
[admin@MikroTik] interface> enable radiolan1
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
  #    NAME                 TYPE             MTU
  0  R ether1               ether            1500
  1  R radiolan1            radiolan         1500
  2 X  vlan1                vlan             1500
[admin@MikroTik] interface>
```

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

```
[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>
```

Argument description:

number − Interface number in the list
name − Interface name
mtu − Maximum Transmit Unit (68...1900 bytes). Default value is 1500 bytes.
mac−address − MAC address. Cannot be changed.
distance − distance setting for the link (0−10.2km)
rx−diversity − Receive diversity (disabled / enabled)
tx−diversity − Transmit diversity (disabled / enabled)
default−destination − default destination (**ap**, **as−specified**, **first−ap**, **first−client**, **no−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 of a host in the radio network where to send the packet, if it is for none of the radio clients.
max−retries − maximum retries before dropping the packet
sid − Service Set Identifier
card−name − Card name
arp − 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.

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.

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:

```
[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>
```

You can monitor the list of neighbours having the same sid and being within the radio 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 with two MikroTik Routers having RadioLAN interfaces:

- The Router#1 has IP address/netmask 10.1.1.12/24 on the Ethernet interface ether1, and 10.1.0.1/30 on the RadioLAN interface radiolan1.
- The Router#2 has IP address/netmask 192.168.0.254/24 on the Ethernet interface ether1, and 10.1.0.2/30 on the RadioLAN interface radiolan1.

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 29−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

VLAN is an implementation of the 802.1Q VLAN protocol for MikroTik RouterOS 2.6. 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.

Contents of the Manual

The following topics are covered in this manual:

Installation

The MikroTik Router should have the vlan software package installed. The software package file **vlan−2.6.x.npk** can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer. After successful installation the package should be listed under the installed software packages list.

Hardware Resource Usage

This protocol uses a minimum of resources.

VLAN Interface and Protocol 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

Virtual LAN interface management can be accessed under the **/interface vlan** submenu.

You can add a VLAN interface using the **/interface vlan add** command:

```
[admin@MikroTik] interface vlan> add
creates new item with specified property values.
        arp
  copy−from  item number
   disabled
  interface
        mtu
       name
    vlan−id
[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>
```

Descriptions of settings:

name − Interface name for reference
mtu − mtu − Maximum Transmit Unit. Should be set to 1500 bytes as on ethernet interfaces. Note that 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.
interface − physical interface to the network where are VLANs
arp − 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.

vlan−id − Virtual LAN identificator or tag that is used to distinguish VLANs. Must be equal for all computers in one VLAN

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

The bandwidth usage of the interface may be monitored with the **monitor−traffic** feature from the **interface** menu.

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 successefully 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)

WaveLAN/ORiNOCO 2.4GHz 11Mbps Wireless Interface

Document revision 16−Sep−2002
This document applies to the MikroTik RouterOS V2.6

Overview

Note! MikroTik does not guarantee support for Orinocco/Wavelan

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.

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

Contents of the Manual

The following topics are covered in this manual:

Wireless Adapter Hardware and Software Installation

Software Packages

The MikroTik Router should have the wavelan software package installed. The software package file **wavelan−2.6.x.npk** can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer. After successful installation the package should be listed under the installed software packages list, for example:

```
[admin@MikroTik] > /sys package print
Flags: I − invalid
  #   NAME                   VERSION              BUILD−TIME           UNINSTALL
  0   system                 2.6beta4             aug/09/2002 20:22:14 no
  1   wavelan                2.6beta4             aug/09/2002 20:31:48 no
  2   ppp                    2.6beta4             aug/09/2002 20:28:01 no
  3   pppoe                  2.6beta4             aug/09/2002 20:29:18 no
  4   pptp                   2.6beta4             aug/09/2002 20:28:43 no
  5   ssh                    2.6beta4             aug/09/2002 20:25:31 no
[admin@MikroTik] >
```

Software License

The 2.4GHz wireless adapters require the 2.4GHz wireless feature license. One license is for one installation of the MikroTik RouterOS, disregarding how many cards are installed in one PC box. The wireless feature is not included in the Free Demo or Basic Software License. The 2.4GHz Wireless Feature cannot be obtained for the Free Demo License. It can be obtained only **together** with the Basic Software License.

System Resource Usage

Before installing the wireless adapter, please check the availability of free IRQ's and I/O base addresses:

```
[admin@MikroTik] > system resource irq print
Flags: U − unused
   IRQ OWNER
   1   keyboard
   2   APIC
 U 3
   4   sync1
   5   Wavelan 802.11
 U 6
 U 7
 U 8
 U 9
 U 10
   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
 100−13F                Wavelan 802.11
```

```
 1F0-1F7           IDE 1
 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] >
```

Installing the Wireless Adapter

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.

Please note, that not all combinations of I/O base addresses and IRQ's may work on your motherboard.

> ***Special Notice for PCMCIA−PCI adapter users!*** The IRQ is not being reported back correctly on some MB for PCMCIA−PCI adapters. As a result, the wireless interface appears to be operational, but there can be no data transmitted over the wireless link. For example, when pinging the AP or GW form the router, there is no response to the ping, although the other end gets the MAC address of the WaveLAN interface of the router. To solve this, try using another MB, or use PCMCIA−ISA adapter.

Loading the Driver for the Wireless Adapter

The WaveLAN / Orinoco PC (PCMCIA) cards do not require a 'manual' driver loading, since they are recognized automatically by the system and the driver is loaded at the system startup. If the driver has loaded successfully, there should be two beeps of equal tone, which should be heard through the PC's speaker while the system startup. If the second beep has a lower tone than the first one, then the driver could not be loaded, or, there is no wavelan package installed.

> ***Note!*** The PC card can be inserted in the PCMCIA−ISA or PCI adapter when the system is running. The wavelan driver is not listed under the list of loaded drivers.

There can be several reasons for a failure to load the driver:

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

Wireless Interface Configuration

If the driver has been loaded successfully (no error messages), and you have the required 2.4GHz Wireless Software License, then the WaveLAN/ORiNOCO 2.4GHz Wireless interface should appear under the interfaces list with the name wavelanX, where X is 1,2,... You can change the interface name to a more descriptive one using the **set** command. To enable the interface, use the **enable** command:

```
[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>
```

Argument description:

number − Interface number in the list
name − Interface name
mtu − Maximum Transmit Unit (256...2296 bytes). The default value is 1500 bytes.
mac−address − MAC address of the card. Cannot be changed.
frequency − Channel frequency (**2412MHz**, **2422MHz** ... **2484MHz**)
data−rate − Data rate (**11Mbit/s**, **1Mbit/s**, **2Mbit/s**, **5.5Mbit/s**, **auto**)
mode − Operation mode of the card (**infrastructure**, **ad−hoc**)
ssid − Service Set Identifier
client−name − Client name
key1 − Encryption key #1
key2 − Encryption key #2
key3 − Encryption key #3
key4 − Encryption key #4
tx−key − Transmit key (**key1**, **key2**, **key3**, **key4**)
encryption − Encryption (**no**, **yes**)
arp − 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.

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*
  This is IRQ conflict. See the special notice for PCMCIA−PCI adapter users under the Wireless Adapter Installation instructions above.

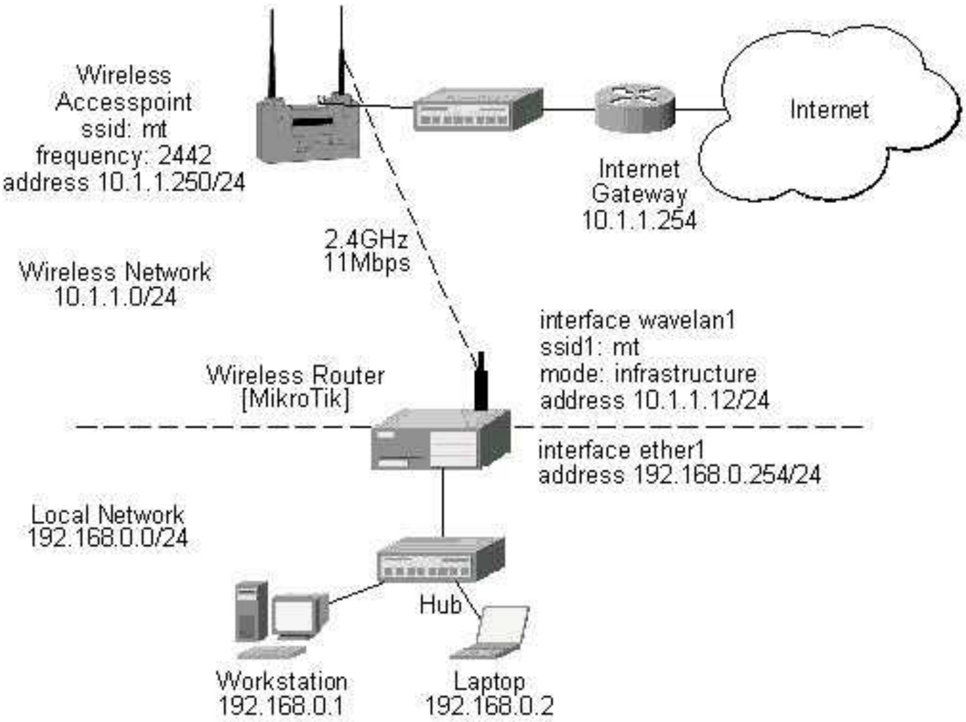
Wireless Network Applications

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 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.

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!*

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.:

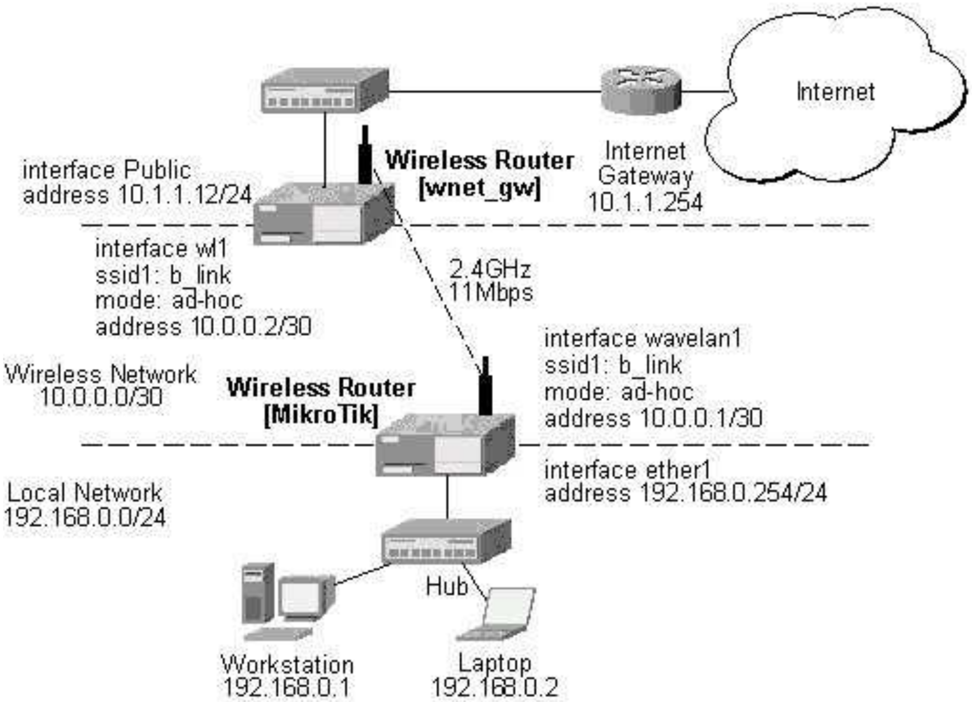
```
[admin@MikroTik] ip address> add address 10.1.1.12/24 interface wavelan1
[admin@MikroTik] ip address> add address 192.168.0.254/24 interface ether1
[admin@MikroTik] ip address> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  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>
```

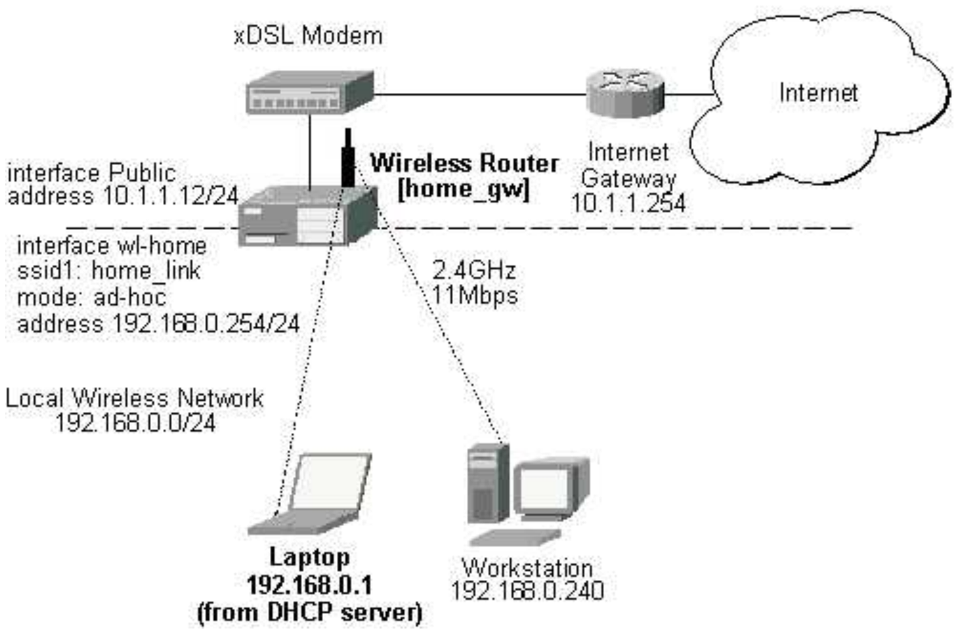
Testing the Network Connectivity

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 is 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] >
```

DHCP Client and Server

Document revision 27−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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.

Contents of the Manual

The following topics are covered in this manual:

Installation

Please download the dhcp−2.6.x.npk package from the MikroTik's web site, upload it with ftp in BINARY mode to the router and reboot.

Use the **/system package print** command to see the list of installed packages.

Hardware Resource Usage

The DHCP server does not consume any significant resources. The DHCP client may consume high resource for five to ten seconds when acquiring an address or renewing an address.

DHCP Description

The DHCP protocol gives and allocates IP addresses to IP clients. DHCP is basically insecure and should only be used on secure networks. UDP port 67 is the DHCP listen port and UDP port 68 is the DHPC transmit port.

DHCP Client Setup

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.

To enable DHCP client on Mikrotik router, specify the interface for it, for example:

```
[admin@MikroTik] ip dhcp−client> set enabled=yes interface=ether1
[admin@MikroTik] ip dhcp−client> print
              enabled: yes
            interface: ether1
            client−id: ""
    add−default−route: yes
```

Descriptions of arguments:

enabled − Enables or disables the DHCP client (**yes**, **no**)
interface − Can be set to any Ethernet−like interface – this includes wireless and EoIP tunnels
client−id − (optional) It should correspond to the settings suggested by the network administrator or ISP
add−default−route − defines whether to add the default route to the gateway specified by DHCP server (**yes**, **no**)

To show obtained leases, use **lease print** command, for example:

```
[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>
```

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).

DHCP Server Setup

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.

To use MikroTik RouterOS DHCP server feature, you should:

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
```

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=255.255.255.0 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=255.255.255.0 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>
```

Descriptions of arguments:

name − descriptive name for server
interface − All Ethernet like interfaces may run a DHCP server
lease−time − Dictates the time that a client may use an address. Suggested setting is three days. 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 − IP pool, from which to take IP addresses for clients
netmask − The netmask to be used by DHCP client
gateway − The default gateway to be used by DHCP client
src−address − 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 − 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 − The DHCP client will use this as the 'DNS domain' setting for the network adapter.
wins−server − 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 − defines whether to add dynamic ARP entry. If set to 'no', static ARP entries must be in **/ip arp** menu. See the IP Addresses and Address Resolution Protocol Manual for more details.

To monitor the leases issued to DHCP clients, use **lease print** command, for example:

```
[admin@MikroTik] ip dhcp-server> lease print
Flags: X - disabled, D - dynamic, H - hotspot
  #    ADDRESS         MAC-ADDRESS       EXPIRES-A... SERVER              STATUS
  0 D  10.0.0.202      00:04:EA:99:63:C4 1h47m24s     dhcp-office         bound
  1 D  10.5.2.90       00:04:EA:C6:0E:40 1h54m9s      switch              bound
  2 D  10.5.2.91       00:04:EA:99:63:C0 1h48m1s      switch              bound
  3 D  10.0.0.201      00:00:E8:69:68:FE 2h40m4s      dhcp-office         bound
[admin@MikroTik] ip dhcp-server>
```

Static Leases

To assign static IP address for DHCP client, static leases can be used. Static leases can be assigned to MAC addresses using **lease add** command:

```
[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>
```

Leases assigned dynamically by the DHCP server are shown as **dynamic**.

Printout description (use **print detail** to see all arguments):

address − leased IP address for the client
mac−address − MAC address of the client. It is base for static lease assignment
expires−after − time until lease expires
server − server name which serves this client
lease−time − dictates the time that a client may use an address
netmask − the netmask to be given with the IP address coming from the range of addresses that can be given out
gateway − the default gateway to be used by the DHCP client
status − 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

Note that even though client address is changed 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).

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 Cache

Document revision 16−Oct−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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 without any local items. DNS protocol is described in RFC1035 and related documents

Contents of the Manual

The following topics are covered in this manual:

Installation

The DNS cache feature is included in the **dns−cache** package. The package file **dns−cache−2.6.x.npk** can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload it with ftp in BINARY mode to the router and reboot.

Use the **/system package print** command to see the list of installed packages.

Hardware Resource Usage

The feature uses a minimum of resources. But if you plan to use larger cache then it is by default, you should monitor RAM usage.

DNS Cache Description

The MikroTik router with DNS cache feature enabled can be set as primary DNS server for any DNS−compliant clients. Moreover, MikroTik router can be specified as primary DNS server under its dhcp−server settings. When the DNS cache is enabled, the MikroTik router responds to DNS requests on TCP and UDP ports 53. Make sure you do not block this port in the firewall setup!

DNS Cache Setup

DNS cache management can be accessed under the **/ip dns−cache** submenu. DNS client configuration (accessible under **/ip dns** submenu) is not required. To enable DNS cache, use the **set** command, for example:

```
[admin@MikroTik] ip dns−cache> set enabled=yes dns−server=159.148.60.2
[admin@MikroTik] ip dns−cache> print
       enabled: yes
          size: 512
```

```
    dns−server: 159.148.60.2
[admin@MikroTik] ip dns−cache>
```

Descriptions of settings:

enabled − defines whether DNS cache (TCP and UDP port 53) is enabled or not
size − maximum number of entries in the cache
dns−server − parent DNS server that is used to resolve requests absent in the cache

Monitoring DNS Cache

Currently no monitoring of DNS cache is available. Later versions of MikroTik RouterOS will have option of DNS cache static entries, as well as cache monitoring.

Additional Resources

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

Firewall Filters and Network Address Translation (NAT)

Document revision 5−Sep−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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 they serve as security tools for preventing unauthorized access to networks.

Contents of the Manual

The following topics are covered in this manual:

Firewall Installation

The firewall feature is included in the "system" software package. No additional software package installation is needed for this feature.

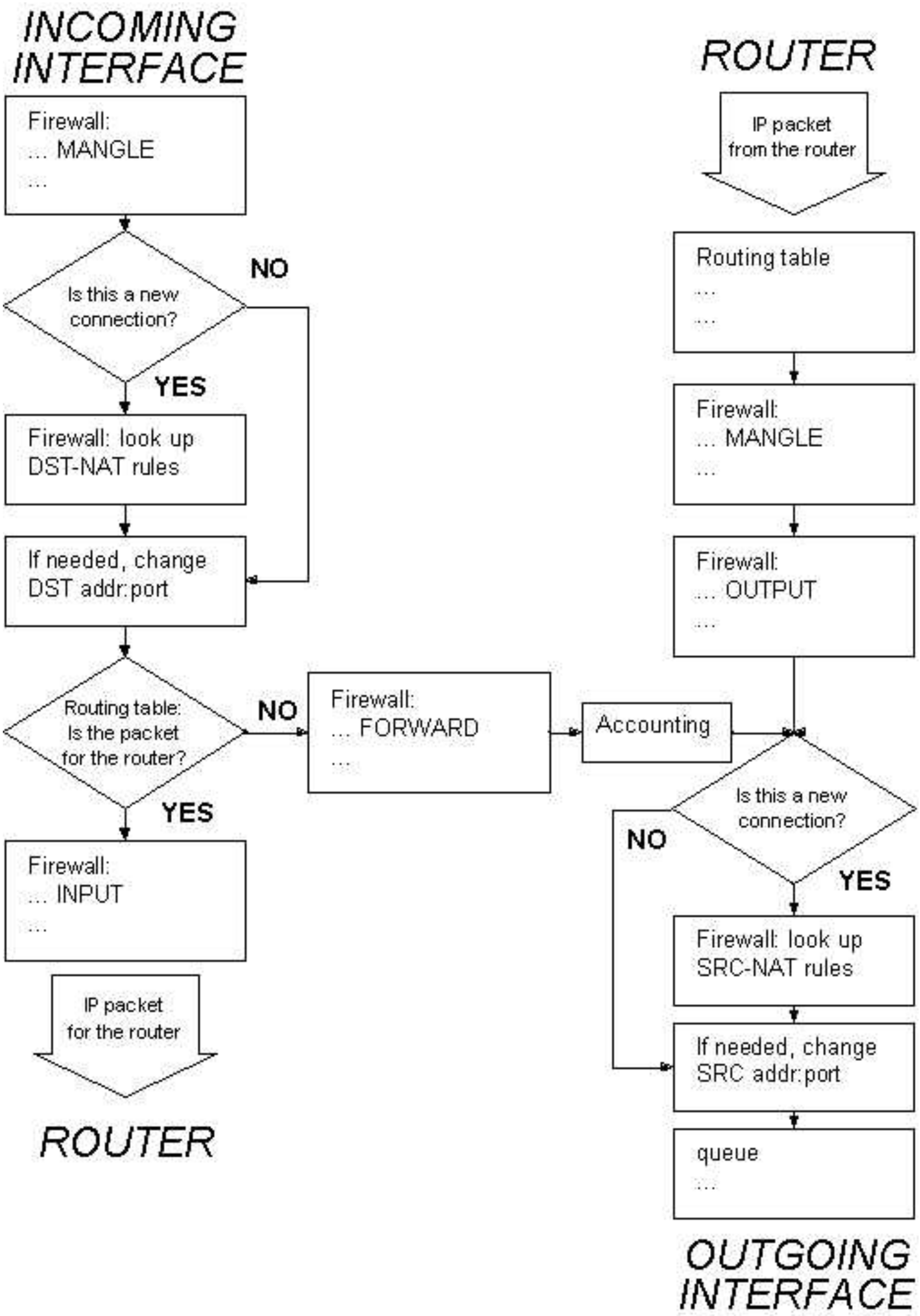
Packet Flow through the Router

The firewall rules are applied in the following order:

- When a packet arrives at an interface, 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. This is important when setting up firewall rules, since the original packets might be already modified by the NAT.

- 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.

IP packet flow through the router is given in the following diagram:



IP Firewall Configuration

The IP firewall management can be accessed under the **/ip firewall** menu. 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.

IP Firewall Common Arguments

The common arguments used in the firewall rules are:

action − Action to undertake if the packet matches the rule (see below). The choice of the available action is different for firewall filter, mangle and NAT rules.
mark−flow − (MANGLE only) Flow mark string.
dst−address − Destination IP address. Can be in the form address/mask:ports, where mask is number of bits in the subnet, and ports is one port, or range of ports, e.g., x.x.x.x/32:80−81
dst−netmask − Destination netmask in decimal form x.x.x.x
dst−port − Destination port number or range (0−65535). 0 means all ports 1−65535.
icmp−options − "any:any". ICMP options.
out−interface − interface the packet is leaving the router. If the default value 'all' is used, it may include the local loopback interface for packets with destination to the router.
limit−burst − allowed burst regarding the limit−count/limit−time
limit−count − how many times to use the rule during the **limit−time** period
limit−time − time interval, used in **limit−count**
protocol − Protocol (all / egp / ggp / icmp / igmp / ip−encap / ip−sec / tcp / udp). **all** cannot be used, if you want to specify ports.
src−address − Source IP address. Can be in the form address/mask:ports, where mask is number of bits in the subnet, and ports is one port, or range of ports, e.g., x.x.x.x/32:80−81
src−mac−address − host's MAC address the packet has been received from.
src−netmask − Source netmask in decimal form x.x.x.x
src−port − Source port number or range (0−65535). 0 means all ports 1−65535.
in−interface − 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.
tcp−mss − (MANGLE only) The new TCP Maximum Segment Size (MSS) value, MTU minus 40, or **dont−change**.
tcp−options − (all / syn−only / non−syn−only). **non−syn−only** is for all other options than **syn−only**.
connection−state − (any / established / invalid / new / related). The connection state.
flow − Flow mark to match. Only packets marked in the MANGLE would be matched.
jump−target − Name of the target chain, if the action=jump is used.
log − Log the action (yes / no).

To view the byte and packet counters, use commands **print bytes**, **print packts**. To reset the counters, use the command **reset−counters**.

If the packet matches the criteria of the rule, then the performed ACTION can be:

- **accept** − Accept the packet. No action, i.e., 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.
- **passthrough** − (MANGLE only) mark the packet for further processing against some rule, and go on processing the next rule.
- **masquerade** − (SRC−NAT only) 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.
- **redirect** − (DST−NAT only) 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** − (SRC−NAT and DST−NAT only) Perform Network Address Translation. For source NAT, the **to−src−address** should be specified (not required with 'action=masquerade'). For destination NAT, the **to−dst−address** should be specified (not required with 'action=redirect').

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
  0 Firewall−Log        local
  1 PPP−Account         none
  2 PPP−Info            none
  3 PPP−Error           none
  4 System−Info         local
  5 System−Error        local
  6 System−Warning      local
  7 Prism−Info          local
[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 detail without−paging
...
 time=feb/24/2002 19:37:08
    message=router−>REJECT, in:ether1, out:(local), src−mac \
             00:30:85:95:67:2b, prot TCP (SYN), \
             213.67.20.9:4164−>195.13.162.195:21, len 60

(The format of the log is:
DATE TIME Chain −> ACTION, in:interface, out:interface, \
             src−mac ADDRESS, protocol (protocol option), \
             src−address:port−>dst−address:port, packet_length )
```

Marking the Packets (Mangle) and Changing the MSS

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. Use the **/ip firewall mangle** to manage the packet marking. 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
  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="" 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

[admin@MikroTik] ip firewall mangle>
```

Note, that the packets originated from the router cannot be mangled!

To change the TCP Maximum Segment Size (MSS), set the 'tcp−mss' argument to a value which is your desired MTU value less 40, for example, if your connection MTU is 1500, you can set 'tcp−mss=1460' or lower. The MSS can be set only for TCP SYN packets.

For example, if you have encrypted PPPoE link with MTU=1492, 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
  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=1448

[admin@MikroTik] ip firewall mangle>
```

Firewall Chains

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 **Input Chain** 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 **Forward Chain** is used to process packets passing through the router.

The **Output Chain** 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.

> Note, that the packets passing through the router are not processed against the rules of neither the input, nor output chains!

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 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>
```

These three chains cannot be deleted. 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.

> ***Note!*** 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.

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.

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>
```

The policy of user added chains is 'none', and it cannot be changed. Chains cannot be removed, if they contain rules (are not empty).

Firewall Rules

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. In the terminal console, use the **/ip firewall rule** command with the argument value that specifies a chain, for example:

```
[admin@MikroTik] ip firewall> rule input
[admin@MikroTik] ip firewall rule input>
```

To add a rule, use the **add** command, for example:

```
[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=""
      src-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>
```

Here, the available values for the argument **action** are: **accept**, **drop**, **jump**, **passthrough**, **reject**, **return**. See the argument description above.

Masquerading and Source NAT

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.

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=10.5.91.0/24:0−65535 \
\... out−interface=ether1 action=masquerade
[admin@MikroTik] ip firewall src−nat> print
Flags: X − disabled, I − invalid
  0   src−address=10.5.91.0/24:0−65535 dst−address=0.0.0.0/0:0−65535
      out−interface=ether1 protocol=all icmp−options=any:any flow=""
      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 'masquerading' 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**:

```
[admib@MikroTik] ip firewall src−nat> add src−address=192.168.0.1/32 \
\... out−interface=ether1 action=nat to−src−address=10.0.0.217
[admin@MikroTik] ip firewall src−nat> print
Flags: X − disabled, I − invalid
  0   src−address=192.168.0.1/32:0−65535 dst−address=0.0.0.0/0:0−65535
      out−interface=ether1 protocol=all icmp−options=any:any flow=""
      limit−count=0 limit−burst=0 limit−time=0s action=nat
      to−src−address=10.0.0.217 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

Redirection and destination NAT should be used when you need to give access to services located on a private network from the outside world. 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
  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=""
      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.

Connection Tracking

Connections through the router and their states can be monitored at 'ip firewall connection', for 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>
```

Connection timeouts are as follows:

- 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.

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.

Additional Resources

Read about connection tracking at
http://www.cs.princeton.edu/~jns/security/iptables/iptables_conntrack.html

IP Firewall Applications

Further on, the following examples of using firewall rules are given:

Basic Firewall Building Principles
Example of Firewall Filters
Example of Source NAT (Masquerading)
Example of Destination NAT

Basic Firewall Building Principles

Assume we have 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**.

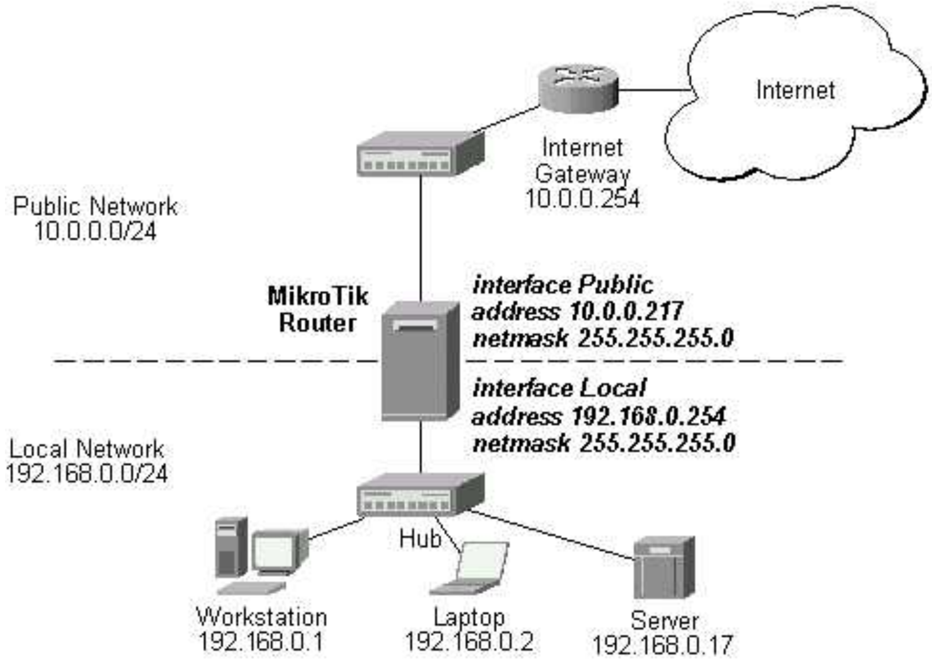
Examples of setting up firewalls are discussed below.

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      10.0.0.217      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] >
```

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 what is 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 of ours"
[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
  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="" 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=""
      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=""
      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=""
      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=""
      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 the 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="" 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=""
      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=""
      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=""
      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=""
      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=""
      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=""
      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=""
      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=""
      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=""
      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=""
      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=""
      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.

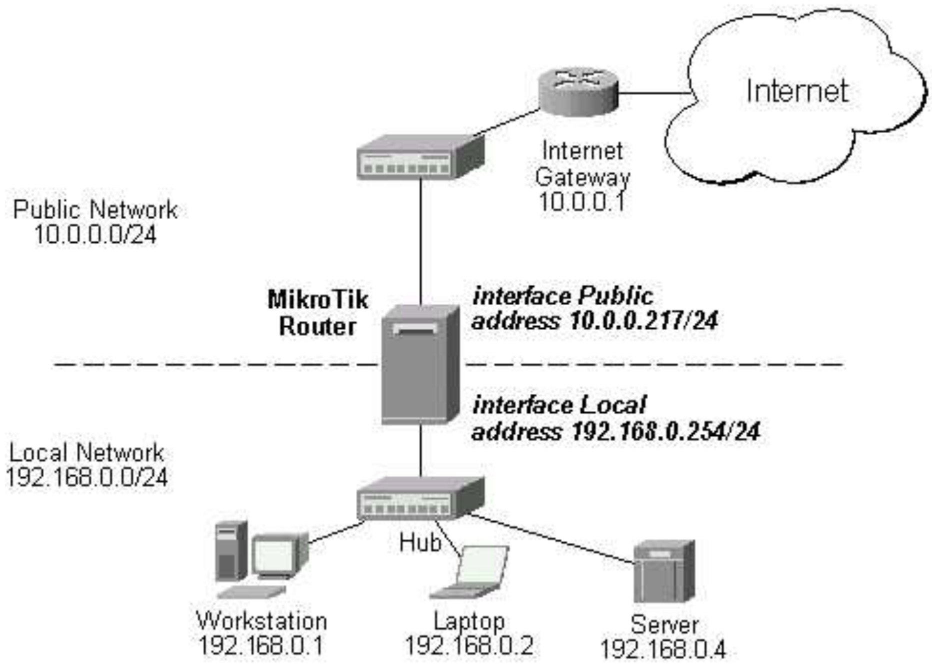
To use masquerading, a source NAT rule with action 'masquerade' should be added to the firewall configuration:

```
[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=""
      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 Static 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's address 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
  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=""
    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>
```

HotSpot Gateway

Document revision 21−Jan−2003
This document applies to the MikroTik RouterOS v2.6

Overview

The MikroTik HotSpot Gateway enables provision of public network access for clients using wireless or wired network connections.

HotSpot Gateway features:

- uses DHCP server to assign temporary (not valid in outer networks) IP addresses to clients prior to authentication;
- authentication of clients using local client database, or RADIUS server;
- after successful authentication the DHCP server assigns address to client from different pool.

It is recommended that you read General Point to Point Setting manual first since the authentication configuration is very similar.

Contents of the Manual

The following topics are covered in this manual:

Installation

The MikroTik HotSpot is included in the **HotSpot** package. This also requires **DHCP** package. Please download the **hotspot−2.6.x.npk** and **dhcp−2.6.x.npk** packages from MikroTik's web site, upload them using ftp BINARY mode to router and reboot.

Use the **/system package print** command to see the list of installed packages.

Software License

The Hotspot limits active user count to 4 for **Demo** Software License and Software Licenses without additional features enabled (i.e. **Basic** Software License without additional packages purchased). For unlimited number of users, any additional (wireless, synchronous) license is required. Thus, if you plan to use wireless hotspot, the license is already there.

Hardware Resource Usage

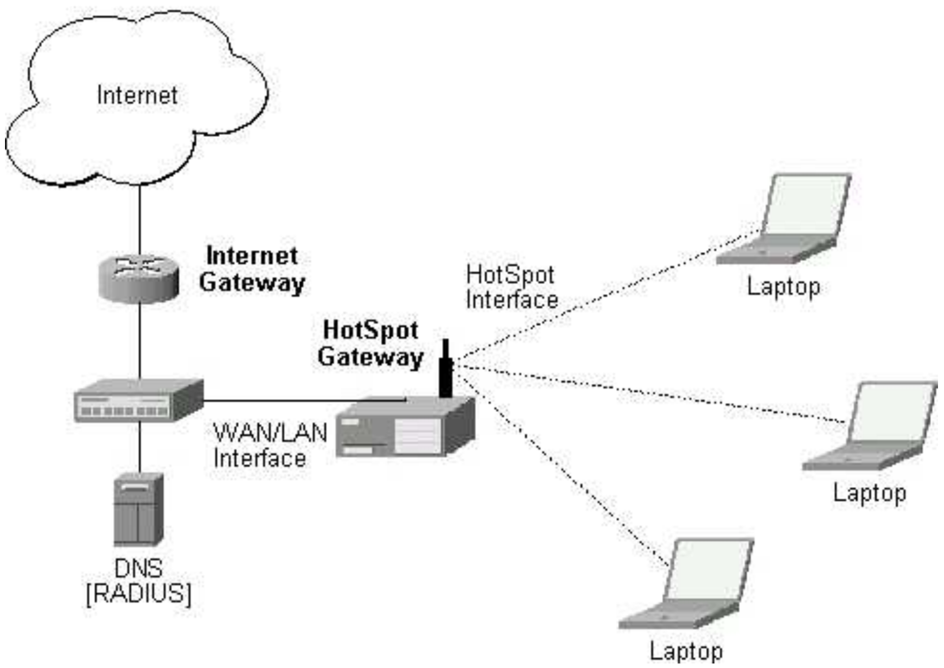
There is no significant resource usage.

How MikroTik HotSpot Gateway Works

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 two IP addresses assigned to it: 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).

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 register to AP), or wired (the NIC card should be connected to HUB).

The Initial Contact

MikroTik HotSpot Gateway's DHCP server assigns IP addresses from the temporary address pool with a very short lease time (approx. 10s), 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 servlet (port 8080 by default). This brings up the HotSpot Welcome/Login page.

It may be useful to have port 80 for HotSpot servlet because the users might want to see status and log out. If this is impossible, you may redirect requests to a virtual IP address to the servlet.

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 described in detailes later on.

Authentication

After client computer receives temporary IP address from HotSpot DHCP server, 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 servlet. So, password is never sent in plain text over ip network.

HotSpot can authenticate users using local user database or some RADIUS server. Which option is used is determined by **/ip hotspot radius−client** enabled parameter. If radius client is enabled, RADIUS authentication is used, otherwise local user authentication is done. 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 when 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

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 required, 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://*virtual_HotSpot_ip*/logout Going to this page will logout user. After that logout page (**logout.html**) will be shown to user.

MikroTik HotSpot Gateway Setup

MikroTik HotSpot Gateway setup is under **/ip hotspot** submenu:

```
[admin@MikroTik] ip hotspot>
HotSpot management
         active  HotSpot active user list
           user  HotSpot local user list
        profile  HotSpot user profile management
         server  HotSpot DHCP profile management
  radius−client  RADIUS client configuration
         cookie  HotSpot active HTTP cookie list
          print  Print current configuration and status
            get  Get value of configuration property
            set  Change hotspot configuration
         export  Export hotspot settings
[admin@MikroTik] ip hotspot> print
         hotspot−address: 0.0.0.0
      status−autorefresh: 1m
                auth−mac: no
       auth−mac−password: no
        auth−http−cookie: no
    http−cookie−lifetime: 1d
```

These are general parameters for HotSpot:

auth−http−cookie − defines whether HTTP authentication by cookie is enabled
auth−mac − defines whether authentication by ethernet MAC address is enabled
auth−mac−password − uses MAC address as password if MAC authorization is enabled
hotspot−address − IP address for HotSpot www access
http−cookie−lifetime − validity time of HTTP cookies
status−autorefresh − WWW status page autorefresh time

HotSpot RADIUS Client Setup

Here is RADIUS client configuration. If it is disabled, users are authorized locally:

```
[admin@MikroTik] ip hotspot radius−client> print
```

```
                enabled: no
             accounting: yes
         primary-server: 10.0.0.96
       secondary-server: 0.0.0.0
          shared-secret: "secret"
    authentication-port: 1812
        accounting-port: 1813
          interim-update: 5m
[admin@MikroTik] ip hotspot radius-client>
```

All parameters are the same as for ppp (**/ppp radius−client**):

accounting − enable or disable RADIUS accounting
accounting−port − IP port on RADIUS server for accounting
authentication−port − IP port on RADIUS server for authentication
enabled − defines whether RADIUS client is enabled
interim−update − Interim−Update time interval
primary−server − IP address of primary RADIUS server
secondary−server − IP address of secondary RADIUS server
shared−secret − shared secret of RADIUS server

RADIUS Parameters

Authentication data sent to server (Access−Request)

```
NAS-Identifier          router identity

NAS-Port-Type           for HotSpot is Ethernet

Calling-Station-Id      client MAC address (with CAPITAL letters)

Called-Station-Id       Hotspot server name (from version 2.6.9)

NAS-Port-Id             Hotspot server name

User-Name               client login name

CHAP-Password,
CHAP-Challenge          encrypted password and challenge
```

Data received from server (Access−Accept)

```
Framed-IP-Address       IP address given to client. 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-Pool             IP pool name (on the router) from which to get IP address
                        for the client

Idle-Timeout            idle-timeout parameter

Session-Timeout         session-timeout parameter

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 if come 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 (used only if RADIUS
                        client does not have local interim-update setting).

Ascend-Data-Rate        tx/rx data rate limitation (for PPPoE). If multiple
                        attributes are provided, first limits tx data rate,
                        second - rx data rate. 0 if unlimited.

Mikrotik-Recv-Limit     total recv limit in bytes for the client

Mikrotik-Xmit-Limit     total transmit limit in bytes for the client

Framed-IP-Netmask       client network netmask

Ascend-Client-Gatway    client gateway
```

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        Satrt, Stop, or Interim-Update
Acct-Session-Id         accounting session ID
NAS-Identifier          same as in request
User-Name               same as in request
NAS-Port-Type           same as in request
NAS-Port-Id             same as in request
Calling-Station-Id      same as in request (from version 2.6.9)
Called-Station-Id       same as in request (from version 2.6.9)
Framed-IP-Address       IP address given to the user
```

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)
```

HotSpot Profiles

The HotSpot profiles are similar to PPP profiles:

```
[admin@MikroTik] ip hotspot profile> print
Flags: * - default
  0 * name="default" session-timeout=0s idle-timeout=0s only-one=no
      tx-bit-rate=0 incoming-filter="" outgoing-filter=""
[admin@MikroTik] ip hotspot profile>
```

Most of these parameters are exactly the same as for **/ppp profile**:

name − profile name
session−timeout − session timeout for client
idle−timeout − idle timeout for client
only−one − only one simultaneous login per user (**yes**, **no**)
tx−bit−rate − transmit bitrate. '0' means no limitation
incoming−filter − firewall chain name for incoming packets

outgoing−filter − firewall chain name for outgoing packets

Default profile will be used in case of RADIUS authentication as well. RADIUS server argument for limiting the data rate (transmited to the client) is Ascend−Data−Rate (vendor id: 529, attribute id:197).

Note that filter rules 'jumping' to the specifies firewall chain are added automatically to the **hotspot** firewall chain. This means that you should create **hotspot** chain and pass some (or all) the packets to it in order filtering to function.

HotSpot Server Settings

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.

```
[admin@MikroTik] ip hotspot server> print
  0 name="dhcp1" dhcp−server=hotspot_dhcp lease−time=1m login−delay=10s
    address−pool=hotspot netmask=0.0.0.0 gateway=0.0.0.0

[admin@MikroTik] ip hotspot server>
```

Description of parameters:

address−pool − IP pool name, from which HotSpot client will get IP address if it is not given some static already
gateway − default gateway
lease−time − DHCP lease time for logged in user
login−delay − Time required to log in user
name − DHCP profile name, is sent as NAS−Port−Id by RADIUS client
netmask − network mask
dhcp−server − DHCP server with which to use this profile

HotSpot User Database

The local user database is manages in **/ip hotspot user** submenu:

```
[admin@MikroTik] ip hotspot user> print
Flags: X − disabled
  #   NAME          PASSWORD      ADDRESS         PROFILE       UPTIME
  0   ax            ex            10.0.0.3        default       29m40s
[admin@MikroTik] ip hotspot user> print detail
Flags: X − disabled
  0   name="ax" password="ex" address=10.0.0.3 profile=default routes=""
      limit−uptime=0s limit−bytes−in=0 limit−bytes−out=0 uptime=29m40s
      bytes−in=187476 packets−in=683 bytes−out=327623 packets−out=671
[admin@MikroTik] ip hotspot user>
```

Parameter description:

name − user name
password − user password
address − 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
profile − user profile
routes − user routes. Usage and meaning is exactly the same as for ppp

limit−bytes−in − maximum amount of bytes user can receive
limit−bytes−out − maximum amount of bytes user can transmit
limit−uptime − total uptime limit for user (pre−paid time)

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. Else, the username and the password should be equal. When client is connecting, it's MAC address is checked first. If there is a user with that MAC address, the client is authorized as this user. If there is no match, client is asked for username and password.

The RADIUS attributes for limit−bytes−in and limit−bytes−out are Mikrotik−Recv−Limit (14988, 1) and Mikrotik−Xmit−Limit (14988, 2). These 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.

All these limits (limit−uptime, limit−bytes−in, limit−bytes−out) can be used for pre−paid solutions. Probably 'quota' is a good name for such limits.

Along with these parameters, some statistics are available for each user:

```
[admin@MikroTik] ip hotspot user> print stats
Flags: X − disabled
  #   NAME         UPTIME          BYTES−IN   BYTES−OUT  PACKETS−IN PACKETS−OUT
  0   ax           29m40s          187476     327623     683        671
[admin@MikroTik] ip hotspot user>
```

Statistics include:

uptime − total time user has been logged in
bytes−in − total bytes received from user
bytes−out − total bytes sent to user
packets−in − total packets received from user
packets−out − total packets sent to user

Note that these stats are updated each time user logs out and RADIUS accounting is disabled (or RADIUS is disabled). 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.

The active user list shows the list of currently logged in users. Nothing can be changed here, except user can be removed with the **remove** command.

```
[admin@MikroTik] ip hotspot active> print
  # USER           ADDRESS         UPTIME          SESSION−TIMEOUT IDLE−TIMEOUT
  0 ex             10.0.0.204      6m10s
[admin@MikroTik] ip hotspot active>
```

Description of the printout:

user − name of user logged in
address − IP address of logged in user
uptime − current session time (logged in time) for this IP address

session−timeout − how much time it is left for IP address until it will be automatically logged out
idle−timeout − how much idle time it is left for IP address until it will be automatically logged out

Statistics about logged in user are available too:

```
[admin@MikroTik] ip hotspot active> print stats
  # USER           UPTIME          BYTES−IN   BYTES−OUT  PACKETS−IN PACKETS−OUT
  0 ax             12m53s          1237091    1222130    4062       4241
[admin@MikroTik] ip hotspot active>
```

HotSpot Cookies

HotSpot Cookies can be managed within **ip hotspot cookie** submenu:

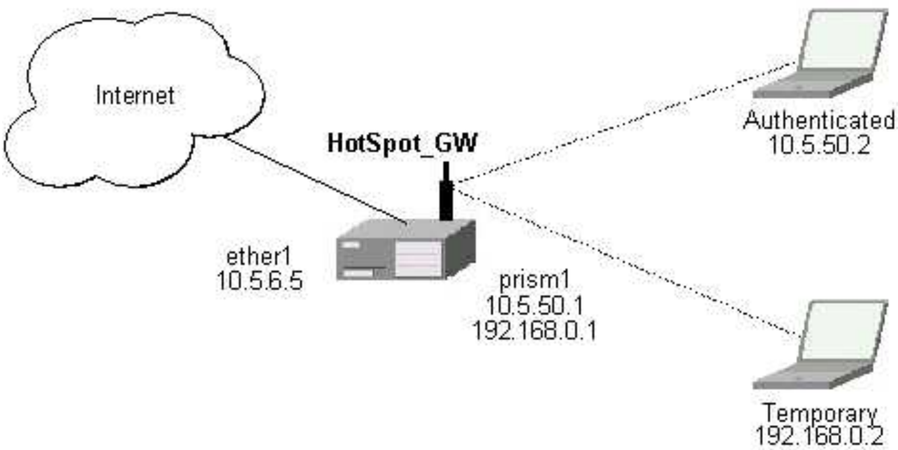
```
[admin@MikroTik] ip hotspot cookie>
HotSpot active HTTP cookie list
    find  Find active HTTP cookie
   print  Show active HTTP cookie list
  remove  Remove active HTTP cookie
     get  Get active HTTP cookie properties
[admin@MikroTik] ip hotspot cookie> print
  # USER                                MAC−ADDRESS       EXPIRES−IN
  0 ex                                  00:30:4F:13:BF:EF 2d23h56m56s
[admin@MikroTik] ip hotspot cookie>
```

Cookies can be listed and removed. They can not be changed or added manually.

HotSpot Step−by−Step User Guide

Planning the Configuration

First of all, make sure you have MikroTik RouterOS 2.6.2 or higher with hotspot and dhcp packages installed. 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 dhcp 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
   /ip address add address=10.5.50.1/24 interface=prism1
   ```
5. add 2 IP pools:

   ```
   /ip pool add name=temp ranges=192.168.0.2−192.168.0.254
   /ip pool add name=hspot 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
   ```

   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_temp" interface=prism1 lease−time=12s \
       address−pool=temp netmask=255.255.255.0 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 logged in IP addresses):

   ```
   /ip hotspot server add name=hs_dhcp dhcp−server=hs_temp address−pool=hspot \
       netmask=255.255.255.0 gateway=10.5.50.1
   ```
9. Add local hotspot user:

   ```
   /ip hotspot user add name=ax 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 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 rule forward add src-address=192.168.0.0/24 protocol=icmp
/ip firewall rule forward add src-address=192.168.0.0/24 protocol=udp\
    dst-port=53
/ip firewall rule forward add src-address=192.168.0.0/24 action=reject\
    comment="reject access for unauthorized hotspot clients"
```

13. Add hotspot chain:

```
/ip firewall add name=hotspot
```

14. Pass all through going traffic to hotspot chain:

```
/ip firewall rule forward add action=jump jump-target=hotspot
```

If client has obtained temporary address, its lease is shown as:

```
[admin@HotSpot_GW] > ip dhcp-server lease print
Flags: X - disabled, D - dynamic, H - hotspot
  #    ADDRESS         MAC-ADDRESS       EXPIRES-A... SERVER                STATUS
  0 D  192.168.0.254   00:40:96:13:B3:47 8s           hs_temp               bound
[admin@HotSpot_GW] >
```

After successful authentication its DHCP address is changed, and it is listed under active hotspot users:

```
[admin@HotSpot_GW] > ip dhcp-server lease print
Flags: X - disabled, D - dynamic, H - hotspot
  #    ADDRESS         MAC-ADDRESS       EXPIRES-A... SERVER                STATUS
  0 DH 10.5.50.2       00:40:96:13:B3:47 56s          hs_temp               bound
[admin@HotSpot_GW] > ip hotspot active print
  # USER                ADDRESS         UPTIME          SESSION-TIMEOUT IDLE-TIMEOUT
  0 ax                  10.5.50.2       2m25s
[admin@HotSpot_GW] > /ip hotspot active print stats
  # USER                UPTIME          BYTES-IN   BYTES-OUT  PACKETS-IN PACKETS-OUT
  0 ax                  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   ax                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** 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 forward 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 forward chain: accept requests going to your web server (this rule has to be before "reject access for unauthorized hotspot clients" rule!)

   ```
   /ip firewall rule forward add dst−address=x.x.x.x/32 dst−port=80 protocol=tcp\
       action=accept
   ```
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 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 transparent web−proxy"
   /ip firewall rule output add src−port=3128 protocol=tcp\
       out−interface=prism1 action=jump jump−target=hotspot\
           comment="account traffic from transparent web−proxy back to hotspot client"
   ```
5. You may want to prevent multiple logins using the same username/password. Set the argument value of 'only−one' to 'yes' in hotspot profile, for example:

   ```
   /ip hotspot profile set default only−one=yes
   ```

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 dns−server=159.148.60.2 enabled=yes
/ip dhcp−server set hs_temp dns−server=10.5.50.1,159.148.108.1
```

Customizing the Servlet

There are many possibilities to customize what the authorization servlet pages look like:

- The pages are easily modifiable. They are stored on the router's FTP server in **hotspot** directory.
- Changing the variables client is sending to the HotSpot gateway 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.

Servlet Page 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)

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/")

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 ("01:02:03:04:05:06")
    - **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 JavaScript for password encoding ("name=login onSubmit=...")
    - **main** − MD5 encryption JavaScript and form for encrypted password

  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 ethernet in plain text. And of course, that login will fail in this case, too.
- 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 ("01:02:03:04:05:06")
    - **uptime** − logged in time ("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, 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, 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 to:

   ```
   https://www.server.serv/register.html?mac=%mac%
   ```

   (you should correct the link to point to your server)

IP Addresses and Address Resolution Protocol (ARP)

Document revision 16−Sep−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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. It is possible to add multiple IP addresses to an interface or to leave the interface without 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.

Contents of the Manual

The following topics are covered in this manual:

Assigning IP Addresses

IP address management can be accessed under the **/ip address** submenu:

```
[admin@MikroTik] ip address>
IP addresses are given to router to access it remotely and to specify it as a
gateway for other hosts/routers.

    print  Show IP addresses
      get  get value of item's property
     find  Find addresses
      set  Change IP address properties
      add  Add IP address
   remove  Remove IP address
   enable  Enable IP address
  disable  Disable IP address
  comment  Set comment for IP address
   export  Export list of IP addresses
[admin@MikroTik] ip address>
```

Use the **/ip address add** command to add an IP address to an interface. 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, for example:

```
[admin@MikroTik] ip address> add
creates new item with specified property values.
    address  Local IP address
  broadcast  Broadcast address
    comment  short description of the item
  copy−from  item number
   disabled
  interface  Interface name
    netmask  Network mask
    network  Network prefix
[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   192.168.0.254/24   192.168.0.0     192.168.0.255   Local
[admin@MikroTik] ip address>
```

Description of the arguments:

address − local IP address, can be in the form address/mask, where mask is number of bits in the subnet mask.
netmask − network mask to be used with the network prefix. Must be in the decimal form a.b.c.d
network − (optional) network prefix to be used with the address. It shows what network can be reached through the interface with the given IP address. If not specified, will be calculated from local address and network mask. For point−to−point links should be the address of the remote end.
broadcast − (optional) broadcast address to be used with the address. If not specified, will be calculated from local address and network mask.
interface − name of the interface the address will be used with

Address Resolution Protocol (ARP)

Address Resolution Protocol is used to map IP address to MAC layer address. Router has a table of currently used ARP entries. Normally table is built dynamically, but to increase network security, static entries can be added.

The ARP management can be accessed under the **/ip arp** submenu:

```
[admin@MikroTik] ip arp>
Address Resolution Protocol is used to map IP address to MAC layer address.
Router has a table of currently used ARP entries. Normally table is built
dynamically, but to increase network security, static entries can be added.

    print  Show ARP entries
      set  Change ARP entry properties
     find  Find ARP entries
      get  get value of item's property
  comment  Set comment for ARP entry
   enable  Enable static ARP entry
  disable  Disable static ARP entry
      add  Add static ARP entry
   remove  Remove ARP entry
   export  Export list of ARP entries
[admin@MikroTik] ip arp>
```

To view the list of arp entries, use the **/ip arp print** command:

```
[admin@MikroTik] ip arp> print
```

```
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS         MAC−ADDRESS       INTERFACE
  0 D 10.1.1.254      00:80:C8:C9:B0:45 Public
  1 D 10.5.8.214      08:00:46:04:33:17 Local
  2 D 10.5.9.202      00:00:E8:69:65:5F sales
  3 D 10.5.9.204      00:00:E8:69:69:9F sales
  4 D 10.5.8.204      00:60:52:0B:B4:80 Local

[admin@MikroTik] ip arp>
```

If static arp entries are used for network security on an interface, you should set arp to 'replay−only' on that interface. Do it under the relevant **/interfaces** menu:

```
[admin@MikroTik] ip arp> /interface ethernet set Local arp=replay−only
[admin@MikroTik] ip arp> add address=10.5.8.214 mac−address=08:00:46:04:33:17 \
\... interface=Local
[MikroTik] ip arp> print
Flags: X − disabled, I − invalid, D − dynamic
  #   ADDRESS         MAC−ADDRESS       INTERFACE
  0 D 10.1.1.254      00:80:C8:C9:B0:45 Public
  1   10.5.8.214      08:00:46:04:33:17 Local
  2 D 10.5.9.202      00:00:E8:69:65:5F sales
  3 D 10.5.9.204      00:00:E8:69:69:9F sales

[MikroTik] ip arp>
```

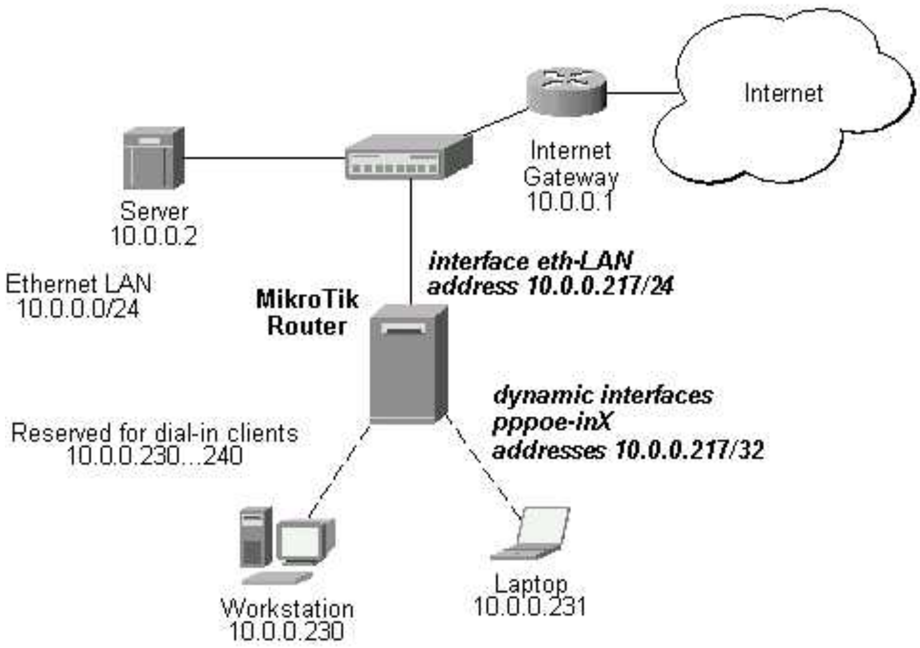
If arp feature is turned off on interface, i.e., 'arp=disabled' is used, ARP requests from clients are not answered by the router. Therefore, static arp entry should be added to the clients as well. For example, the router's IP and MAC addresses should be added to the windows workstations using the **arp** command, for example:

```
C:\> arp −s 10.5.8.254  00−aa−00−62−c6−09
```

See the relevant documentation on how to manage static arp entries on your system.

Using the Proxy−ARP Feature

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. 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

The unnumbered interfaces can be used on serial point−to−point links, e.g., MOXA C101, 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). For 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 totally becomes 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.

IP Pool Management

Document revision 16−Dec−2002
This document applies to the MikroTik RouterOS v2.6

Overview

IP pools are used to define range of IP addresses that is used for DHCP server and **/ppp**

Contents of the Manual

The following topics are covered in this manual:

Installation

The IP pool feature is included in the **system** package. No installation is needed for this feature.

Hardware Resource Usage

There is no significant resource usage.

IP Pool 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

IP Pool Setup

IP Pool management can be accessed under the **/ip pool** submenu:

```
[admin@MikroTik] ip pool>

   print  print values of item properties
    find  finds items by value
     get  get value of item's property
     set  change item properties
     add  create new item
  remove  remove item
  export
    used
[admin@MikroTik] ip pool> print
  # NAME                                        RANGES
  0 a                                           10.0.0.0−10.0.0.255
[admin@MikroTik] ip pool>
```

Argument description:

name – name of the pool
ranges − 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**

To see the existent pools use **print** command:

```
[admin@MikroTik] ip pool> print
  # NAME                                            RANGES
  0 a                                               10.0.0.0−10.0.0.255
  1 b                                               10.0.0.1−10.0.0.27
[admin@MikroTik] ip pool>
```

RADIUS settings

The IP pool **name** can be specified in a RADIUS server with FRAMED_POOL attribute (id: 88, RFC2869)

Monitoring Used IP Addresses

To see, what addresses are currently used, use **used print** command:

```
[admin@MikroTik] ip pool> used print
 POOL                   ADDRESS          OWNER                    INFO
 b                      10.0.0.27        DHCP                     00:e0:c5:6e:23:1d
[admin@MikroTik] ip pool>
```

IPsec

Document revision 30−Dec−2002
This document applies to the MikroTik RouterOS V2.6

Overview

IPsec (IP Security) supports secure (encrypted) communications over IP networks.

Contents of the Manual

The following topics are covered in this manual:

Installation

Please download the **ipsec−2.6.x.npk** package from the MikroTik's web site, upload the package to the router and reboot.

Note that you cannot install IPsec package without SSH installed.

Use the **/system package print** command to see the list of installed packages.

Hardware Resource Usage

IPsec consumes a lot of CPU time, so it needs powerful processor. Intel Pentium MMX or AMD K6 suggested as minimal configuration.

How IPsec Works

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.

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.

IPsec Setup

```
[admin@MikroTik] ip ipsec>

             policy
       installed−sa
          manual−sa
               peer
  pre−shared−secret
           proposal
           counters
             export
[admin@MikroTik] ip ipsec>
```

Descriptions of settings:

policy − set up security policies
installed−sa − look at currently installed security associations
manual−sa − templates for manual security associations
peer − IKE peer configuration
pre−shared−secret − to authenticate with IKE peers

proposal − phase2 IKE proposal settings
counters − counters

To get IPsec to work with automatic keying you will have to configure **policy**, **peer**, **pre−shared−secret** and **proposal** entries. For manual keying you will have to configure **policy** and **manual−sa** entries.

Policy Settings

To define new policy, use **/ip ipsec policy add** command:

```
[admin@MikroTik] ip ipsec policy> add sa−src−address=10.0.0.205 sa−dst−address=10.0.0.201 \
\... action=encrypt
[admin@MikroTik] ip ipsec policy> print
Flags: X − disabled, I − invalid
  0   src−address=10.0.0.205/32:any dst−address=10.0.0.201/32:any
      protocol=all action=encrypt level=require ipsec−protocols=esp tunnel=no
      sa−src−address=10.0.0.205 sa−dst−address=10.0.0.201 proposal=default
      manual−sa=none dont−fragment=clear


[admin@MikroTik] ip ipsec policy>
```

Argument description:

src−address − Source IP address. Can be in the form address/mask:ports
dst−address − Destination IP address. Can be in the form address/mask:ports
protocol − name or number of protocol **action** − What to do with packet that matches policy. Choices are:

- **accept** − pass the packet. This is default action when no policies are configured.
- **drop** − drop the packet.
- **encrypt** − apply transormations specified by this policy and it's security associations.

dont−fragment − default value works OK. It is good to have **dont−fragment** cleared because encrypted packets are always bigger than original and thus they may need fragmentation.
tunnel − **yes** if you want to use tunnel mode. 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.
ipsec−protocols − One of **ah**, **esp**, **ah,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.
level − 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.

sa−src−address − SA source
sa−dst−address − SA destination
manual−sa − Name of manual−sa template that will be used to create SAs for this policy, or **none** if you don't want to set up any manual keys.
proposal − Name of proposal info that will be sent by IKE daemon to establish SAs for

this policy.

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.

Statistics can be printed out 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=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>
```

Description of the printout:

ph2−state − progress of key establishing. 'expired' means there are some leftovers from previous phase2, and is similar to 'no−phase2', which means nothing has happened. 'established' means SAs are in place and everything should be working. Anything else falls between these last two states.
in−accepted − how many incoming packets were passed through by policy without attempting decryption.
in−dropped − how many incoming packets were dropped by policy without attempting decryption.
out−accepted − how many outgoing packets were passed through by policy without encryption.
out−dropped − how many outgoing packets were dropped by policy without attempting encryption.
encrypted − how many outgoing packets were encrypted and passed on successfully.
not−encrypted − how many outgoing packets policy attempted to encrypt, but discarded for any reason.
decrypted − how many incoming packets policy decrypted and passed on successfully.
not−decrypted − how many incoming packets policy tried to decrypt, but discarded for any reason.

See global counters for more specific conditions.

Peer

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. These parameters won't affect the established SAs in any way. To define new peer configuration, use **/ip ipsec peer add** command:

```
[admin@MikroTik] ip ipsec peer> add address=10.0.0.201
[admin@MikroTik] ip ipsec peer> print
Flags: X − disabled
  0   address=10.0.0.201:500 exchange−mode=main send−initial−contact=yes
      proposal−check=strict hash−algorithm=md5 enc−algorithm=3des
      dh−group=modp1024


[admin@MikroTik] ip ipsec peer>
```

Argument description:

address − address of the remote peer
dh−group − Diffie−Hellman (DH) key exchange protocol allows two parties without any initial shared secret to create one. This value defines cipher strength. Allowed values: **modp768**, **modp1024**, **modp1536**, **modp2048**, **modp3072**, **modp4096**, **modp8192**. First three (768, 1024 and 1536) are standard, others might be incompatible with similarly named groups in other implementations
enc−algorithm − Encryption algorithm. Valid algorithms are: **des**, **3des**, **aes−128**, **aes−192** and **aes−256** in strength (and computation time) increasing order
exchange−mode − Valid values are: **main**, **aggressive** or **base**. See RFC 2408 for an overview of ISAKMP phase 1 exchange modes. Currently only **main** mode is tested
hash−algorithm − Hashing algorithm. Valid algorithms are **md5** and **sha** in strength (and computation time) increasing order
proposal−check − Lifetime check logic. This is for phase 2 lifetimes (you cannot configure lifetimes for phase 1 proposals yet). One of:

- **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. This is default value

send−initial−contact − **yes**

Note that both peers MUST have the same encryption and authentication algorithms, dh−group and exchange−mode. Some legacy hardware may support only DES and MD5.

Statistics can be printed out using **print stats** command.

For not yet established connections:

```
[admin@MikroTik] ip ipsec peer> print stats
Flags: X − disabled
  0   address=10.0.0.201:500 exchange−mode=main send−initial−contact=yes
      proposal−check=strict hash−algorithm=md5 enc−algorithm=3des
      dh−group=modp1024 ph1−state=no−phase1


[admin@MikroTik] ip ipsec peer>
```

For running connection:

```
[admin@MikroTik] ip ipsec peer> print stats
Flags: X − disabled
  0   address=10.0.0.201:500 exchange−mode=main send−initial−contact=yes
      proposal−check=strict hash−algorithm=md5 enc−algorithm=3des
      dh−group=modp1024 ph1−state=established ph1−side=initiator
      ph1−established=nov/19/2008 17:13:24 ph2−active=0 ph2−total=1


[admin@MikroTik] ip ipsec peer>
```

Description of the printout:

ph1−state − state of phase 1 negotiation with this peer. **established** is the normal working state
ph1−side − who spoke first. **initiator** means that phase 1 negotiation was started by this

router. **responder** − by peer
ph1−established − when current phase 1 between router and peer was established
ph2−active − how many phase 2 negotiations with this peer are currently taking place
ph2−total − how many phase 2 negotiations with this peer took place

Pre−shared−secret

For IKE peers to know each other they must have same pre−shared−secret configuration. It's kind of like passwords. So if there are two routers: 10.0.0.205 and 10.0.0.201, then on the first (10.0.0.205) it should look like this:

```
[admin@MikroTik] ip ipsec pre−shared−secret> print
Flags: X − disabled
  #   ADDRESS         SECRET
  0   10.0.0.201      gwejimezyfopmekun
[admin@MikroTik] ip ipsec pre−shared−secret>
```

And on the second (10.0.0.201) − like this:

```
[admin@MikroTik] ip ipsec pre−shared−secret> print
Flags: X − disabled
  #   ADDRESS         SECRET
  0   10.0.0.205      gwejimezyfopmekun
[admin@MikroTik] ip ipsec pre−shared−secret>
```

Parameter description:

address − address of remote peer
ident−string − identity string of remote peer
secret − secret string. If it starts with '0x', it is parsed as a hexadecimal value

Manual SA

To add manual−sa entry, use **ip ipsec manual−sa add** command:

```
[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>
```

Command parameters are:

ah−algorithm − Authentication Header encryption algorithm, one of the following:

- **md5** − 128 bit key
- **null** − any key length
- **sha1** − 160 bit key

esp−auth−algorithm − Encapsulating Security Payload authentication encryption algorithm, one of the following:

- **md5** − 128 bit key
- **null** − any key length
- **sha1** − 160 bit key

esp−auth−algorithm − Encapsulating Security Payload encryption algorithm, one of the following:
- **md5** − 128 bit key
- **null** − any key length
- **sha1** − 160 bit key

ah−key − incoming−authentication−key/outgoing−authentication−key
ah−spi − incoming−SA−SPI/outgoing−SA−SPI, in hexadecimal. May be equal
esp−auth−key − incoming−authentication−key/outgoing−authentication−key
esp−enc−key − incoming−encryption−key/outgoing−encryption−key
esp−spi − incoming−SA−SPI/outgoing−SA−SPI, in hexadecimal. May be equal
name − name of item for reference from policies

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.

Proposal

To add proposal, use **ip ipsec proposal add** command. There is a default proposal:

```
[admin@MikroTik] ip ipsec proposal> print
Flags: X − disabled
  0   name="default" auth−algorithms=sha1 enc−algorithms=3des lifetime=30m
      lifebytes=0 pfs−group=modp1024


[admin@MikroTik] ip ipsec proposal>
```

Command parameters are:

auth−algorithms − allowed algorithms for authorization:

- **md5** − 128 bit key
- **null** − any key length
- **sha1** − 160 bit key

enc−algorithms − allowed algorithms and key lengths to use for SAs that will be acquired from IKE daemon by policy that references this proposal (**3des**, **aes−128**, **aes−192**, **aes−256**, **des**, **null**) **lifebytes** − how many bytes to encrypt using SA before throwing it out and making new one. **0** means SA won't expire based on byte count (default)
lifetime − how long to use SA before throwing it out. See also proposal−check in peer config
name − name of proposal for referencing it from policy
pfs−group − Diffie−Helman group used for Perfect Forward Secrecy

Proposals on both peers must (at least partially) match. The more they match the better.

Installed SA

Prints a lot of information about each installed SA (including keys):

```
[admin@MikroTik] ip ipsec installed−sa> print
Flags: A − AH, E − ESP, P − pfs, M − manual
  0 E   spi=21237B07 direction=out src−address=10.0.0.204
```

```
        dst−address=10.0.0.201 auth−algorithm=sha1 enc−algorithm=3des
        replay=4 state=mature
        auth−key="3c1f4a3f5d2014e565f9f3fb671bab89056febb5"
        enc−key="725d43ed2742530a257d19dd36702259ea7a50060aa760a3"
        add−lifetime=24m/30m use−lifetime=0s/0s lifebytes=0/0
        current−addtime=nov/24/2008 14:28:42
        current−usetime=jan/01/1970 00:00:00 current−bytes=0

  1 E   spi=FAACF20D direction=in src−address=10.0.0.201
        dst−address=10.0.0.204 auth−algorithm=sha1 enc−algorithm=3des
        replay=4 state=mature
        auth−key="acb0c8c3dc81f3ff5f92cbc15c49c7a710f9efa5"
        enc−key="a50c04b44904c07009c3e218760f3827493579172b29bcfd"
        add−lifetime=24m/30m use−lifetime=0s/0s lifebytes=0/0
        current−addtime=nov/24/2008 14:28:42
        current−usetime=jan/01/1970 00:00:00 current−bytes=0


[admin@MikroTik] ip ipsec installed−sa>
```

Description of the printout:

spi − SPI value of SA, in hexadecimal
replay − size of replay window, in bytes
state − **larval**, **mature**, **dying** or **dead**
auth−algorithm − **none**, **md5** or **sha1**
enc−algorithm − **none**, **des**, **3des**, **aes**
src−address − source of SA from policy configuration
dst−address − destination of SA from policy configuration
auth−key − authentication key, as hex string
enc−key − encryption key, as hex string (only used by ESP SAs)
direction − **in** or **out**
current−addtime − time when this SA was installed
current−usetime − time when this SA was first used
current−bytes − amount of data processed by this SA's crypto algorithms
add−lifetime − expiration time counted from installation of SA. soft/hard
use−lifetime − expiration time counter from the first use of SA. soft/hard>
lifebytes − expiration threshold for amount of processed data. soft/hard

Counters

Prints miscellaneous counters:

```
[admin@MikroTik] ip ipsec> counters print
                    out−accept: 2298
                      out−drop: 0
                   out−encrypt: 4
                     in−accept: 3497
                       in−drop: 0
                  in−decrypted: 4
              in−accept−isakmp: 20
             out−accept−isakmp: 12
    in−drop−encrypted−expected: 0
[admin@MikroTik] ip ipsec>
```

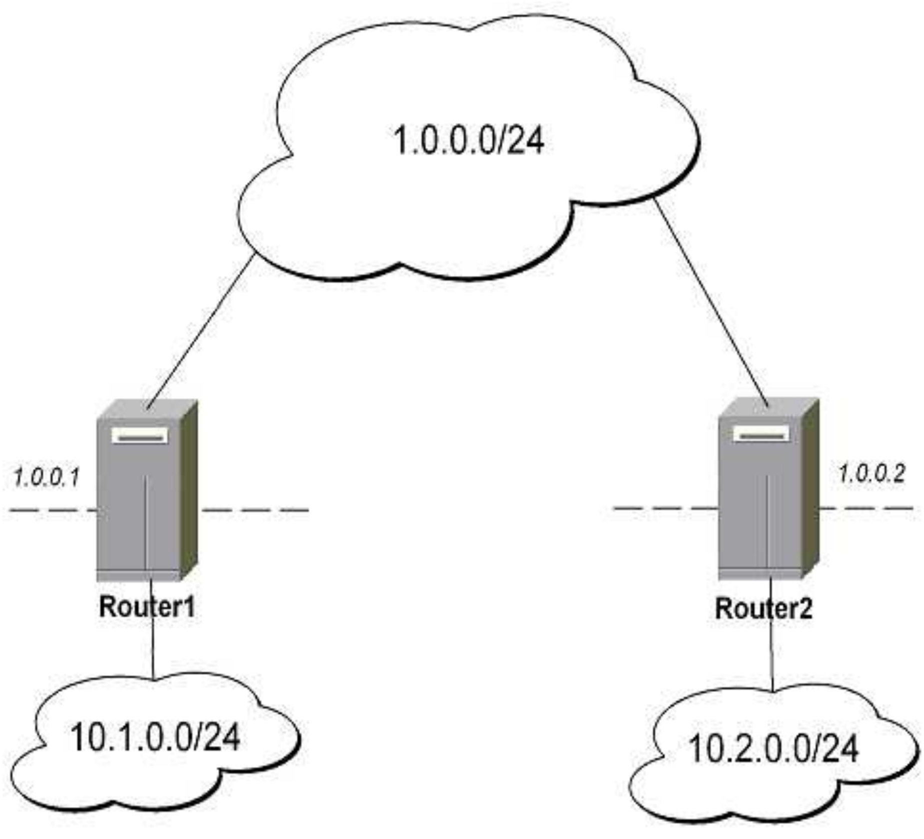
Description of the printout:

out−accept − how many outgoing packets were matched by 'accept' policy (including the default "accept all" case)

out−accept−isakmp − how many locally originated UDP packets on source port 500 (which is how ISAKMP packets look) were let through without policy matching
out−drop − how many outgoing packets were matched by **drop** policy (or **encrypt** policy with level=require that doesn't have all SAs)
out−encrypt − how many outgoing packets were encrypted successfully
in−accept − how many incoming packets were matched by **accept** policy
in−accept−isakmp − how many incoming UDP packets on port 500 were let through without policy matching
in−drop − how many incoming packets matched **drop** policy. (or **encrypt** policy with level=require that didn't have all SAs)
in−decrypted − how many incoming packets were successfully decrypted
in−drop−encrypted−expected − how many incoming packets were matched by **encrypt** policy and dropped because they were not encrypted

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
ip ipsec pre−shared−secret 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
ip ipsec pre−shared−secret add address=1.0.0.1 secret="roberkenon"
```

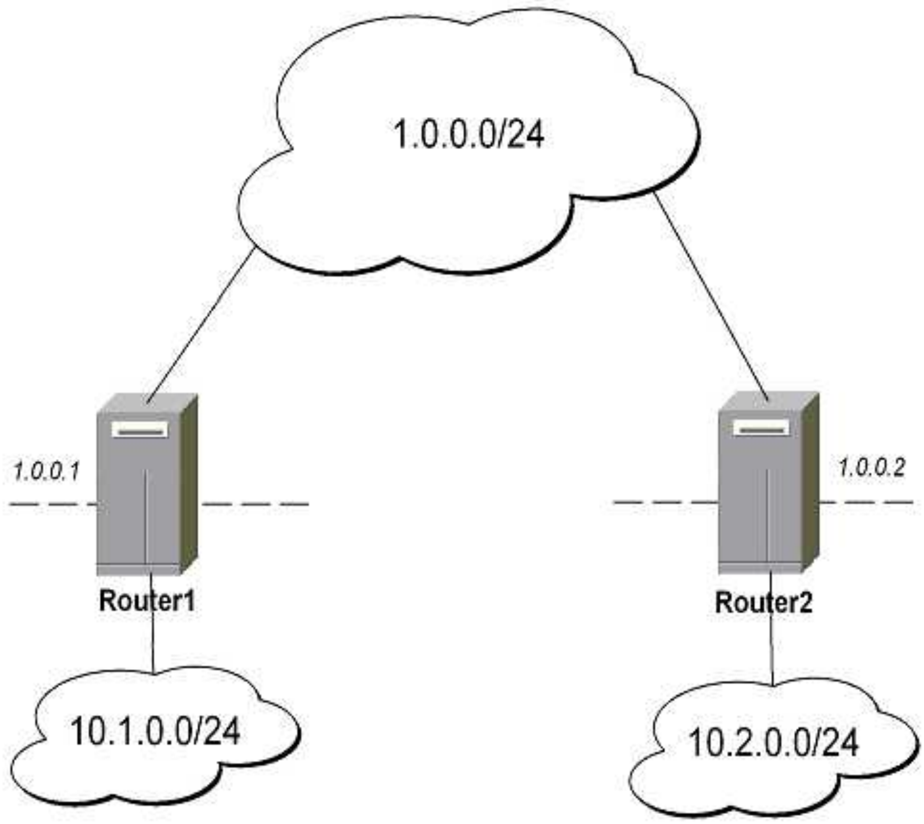
Minimal config example for tunnel mode AH with manual keying on Router 1:

```
ip ipsec key add key algorithm=sha1 length=160 \
   key=00000000000000000000000000000000000000000
ip ipsec manual-sa add ah-key=auth-key1 ah-spi=0x101/0x100
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 key add key algorithm=sha1 length=160 \
\... key=00000000000000000000000000000000000000000
ip ipsec manual-sa add ah-key=auth-key1
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
/ip ipsec pre-shared-secret add address=1.0.0.2 secret="sviestapika"
```

On Router2:

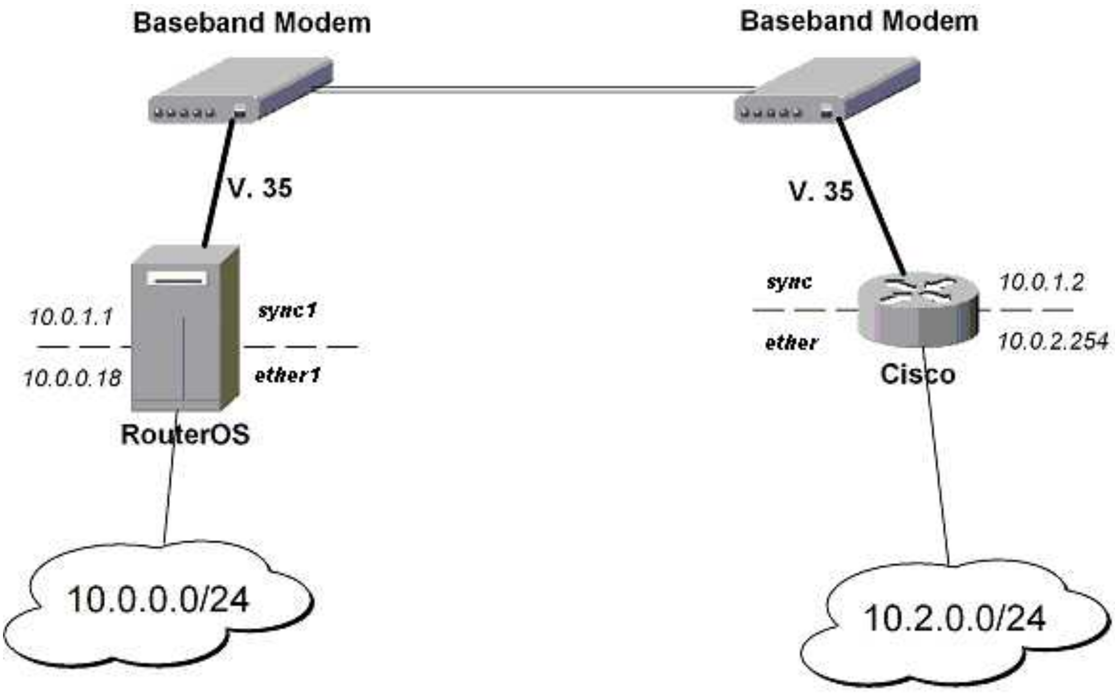
- Add accept and masquerading rules in SRC−NAT:

```
/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
/ip ipsec pre-shared-secret add address=1.0.0.1 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

Add encryption proposal (phase2 proposal − settings that will be used to encrypt actual data), we will use DES to encrypt data and SHA1 to authenticate:

```
[admin@MikroTik] ip ipsec proposal> add name=to_cisco pfs-group=none
algorithms=enc-des,auth-sha1
```

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 enc-algorithm=des
auth-method=pre-shared-key hash-algorithm=sha dh-group=modp1024
```

Add preshared secret to use when talking to Cisco:

```
[admin@MikroTik] ip ipsec pre-shared-secret> add secret=test_key
address=10.0.1.2
```

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 protocol=all action=encrypt ipsec-protocols=esp
level=require tunnel=yes sa-src="IP/10.0.1.1 sa-dst=10.0.1.2 proposal=to_cisco
"
```

Configuring Cisco

Parts from Cisco configuration with comments follow...

```
! 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 10
 authentication pre-share
 group 2

! Add preshared key to be used when talking to RouterOS
crypto isakmp key test_key address 10.0.1.1

! 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

! 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
 match address 101

! And finally apply crypto map to serial interface:
interface Serial1
 crypto map mymap
```

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
```

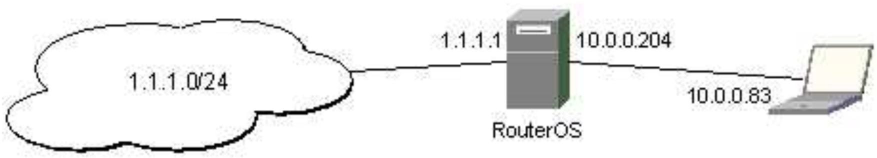
```
[admin@MikroTik] ip ipsec installed-sa>
```

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
   ```
2. Add pre−shared secret to identify remote client.

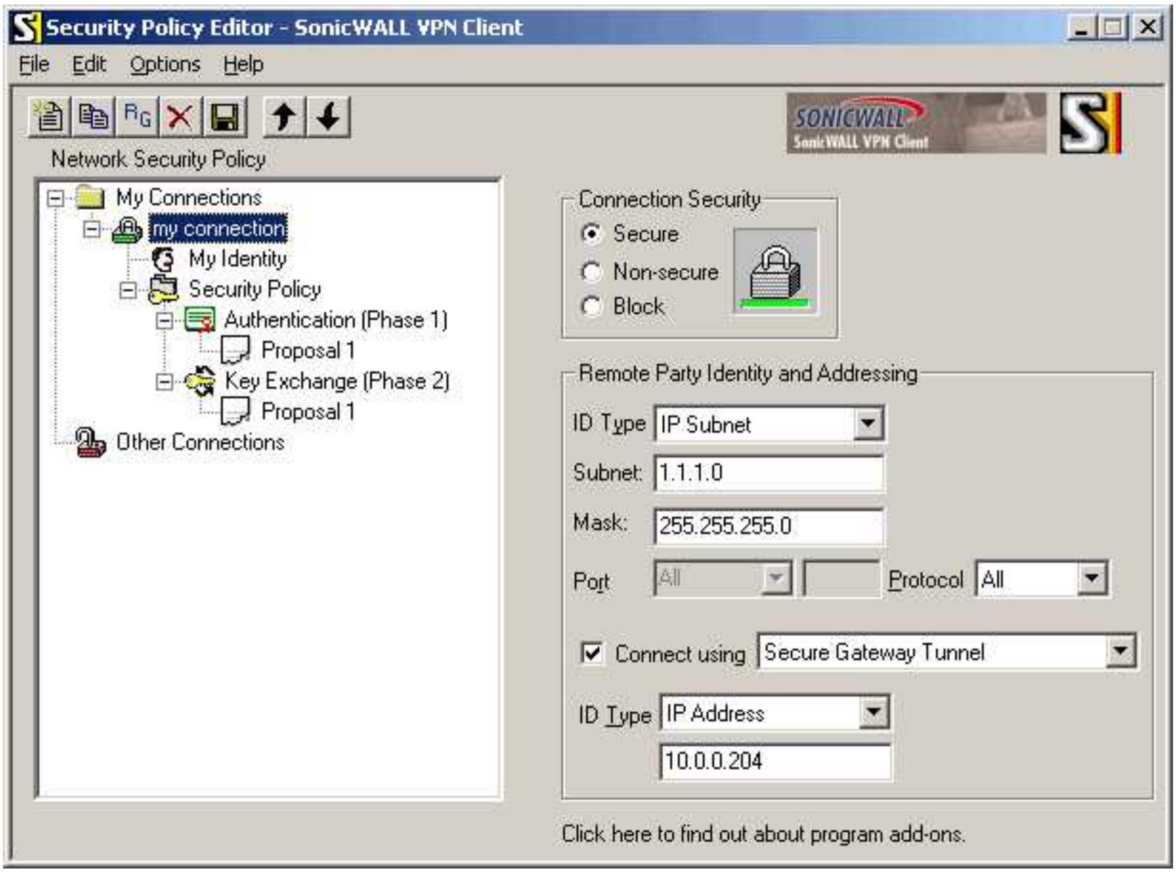
   ```
   / ip ipsec pre-shared-secret add address=10.0.0.81 secret="********"
   ```
3. Add encryption proposal. Use MD5, DES and Diffie−Hellman Group 1 for Perfect Forward Secrecy.

   ```
   / ip ipsec proposal add name=sw-client auth-algorithms=md5 enc-algorithms=des \
   lifetime=30m pfs-group=modp768
   ```
4. 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

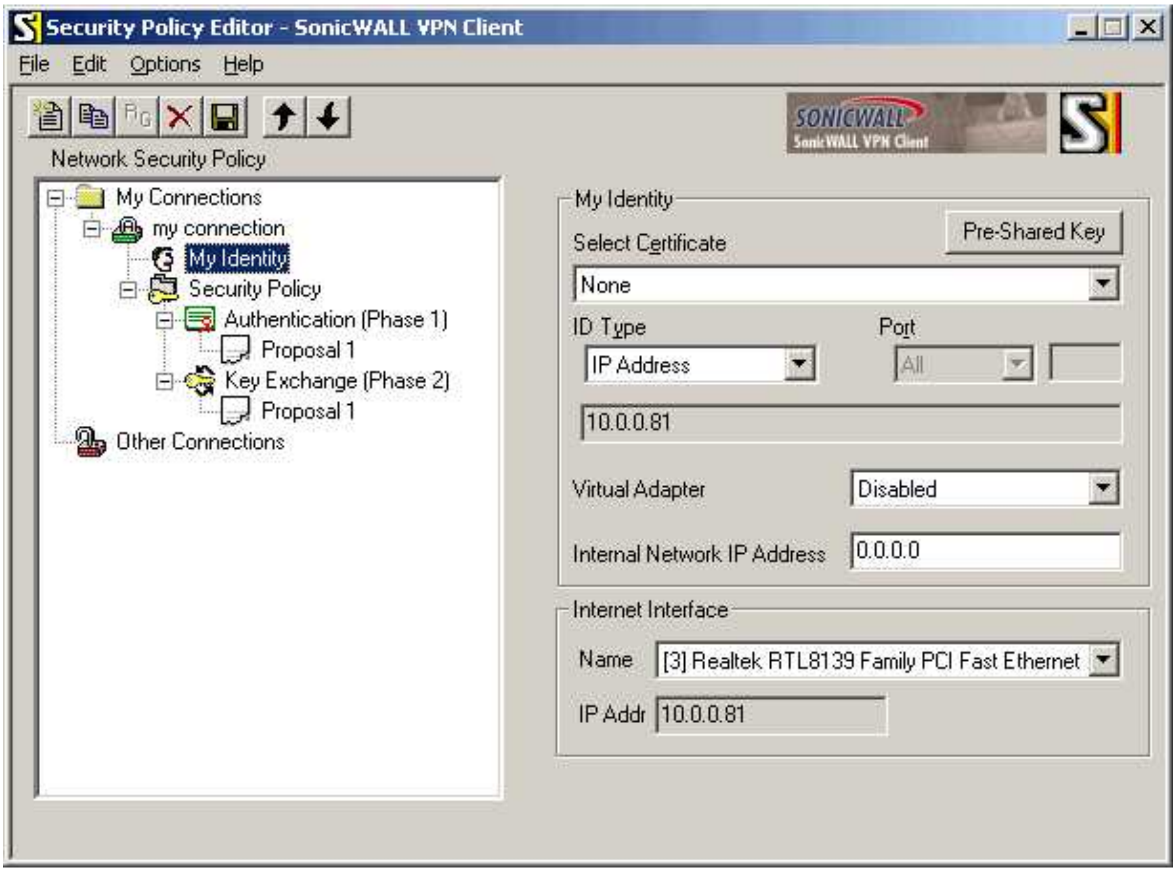
Here you create IPSec policy that should match all traffic between 10.0.0.81 host and 1.1.1.0/24 network. You also specify the address of remote IPSec peer:




```
"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:

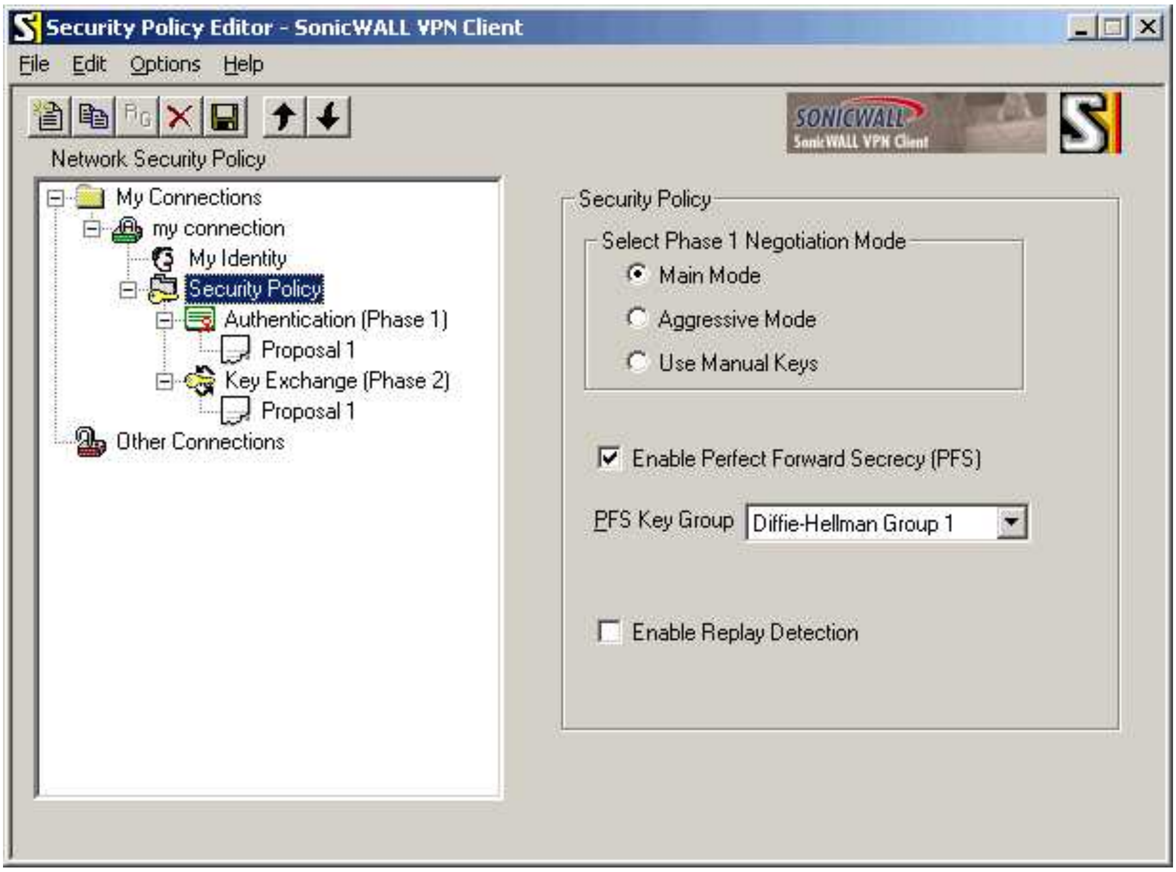



```
in "My Identity" box:
  "Select Certificate": select "None"
  click "Pre−Shared Key"

"Pre−Shared Key" pops up:
```

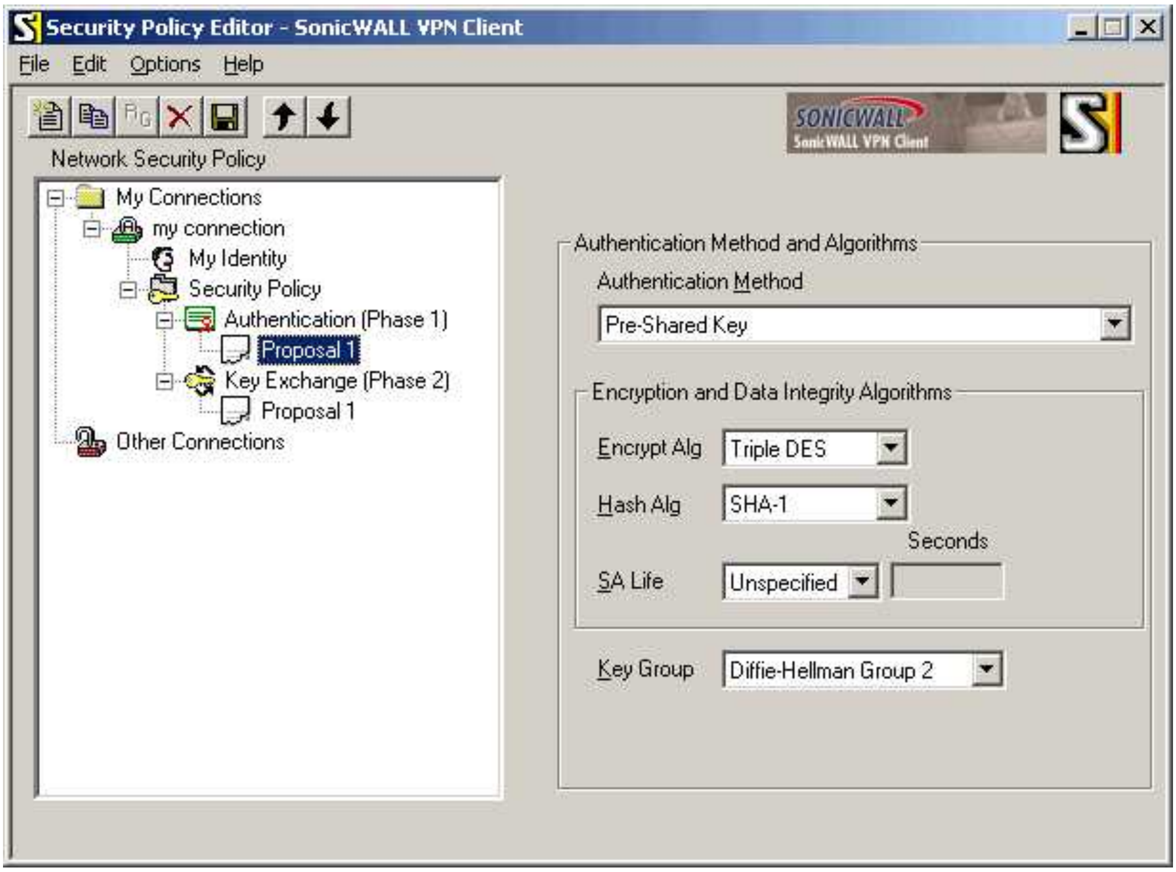

```
    click "Enter Key"
    type ********, click "OK"
```

```
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:

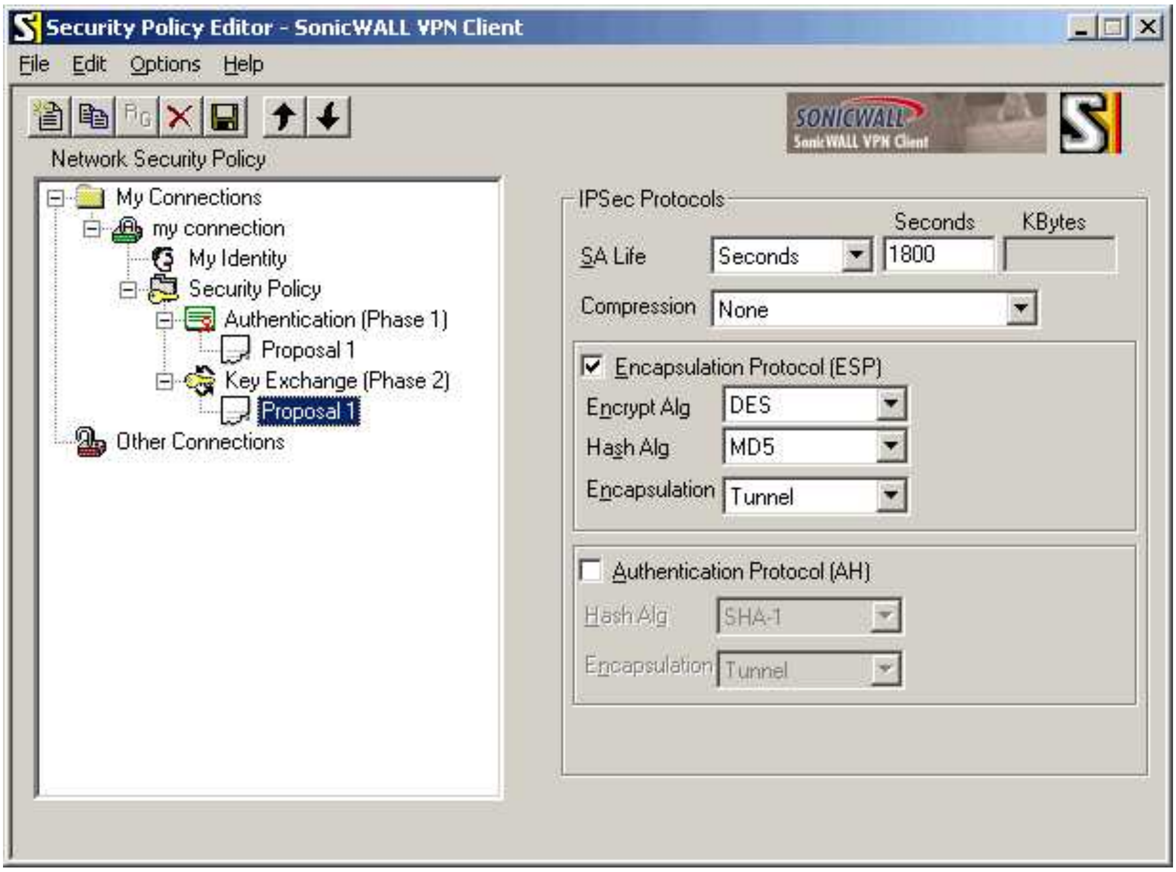



```
"Select Phase 1 Negotiation Mode": select "Main Mode"
check "Enable Perfect Forward Secrecy (PFS)"
"PFS Key Group": select "Diffie-Hellman Group 1"
clear "Enable Replay Detection"
```




```
"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)
```

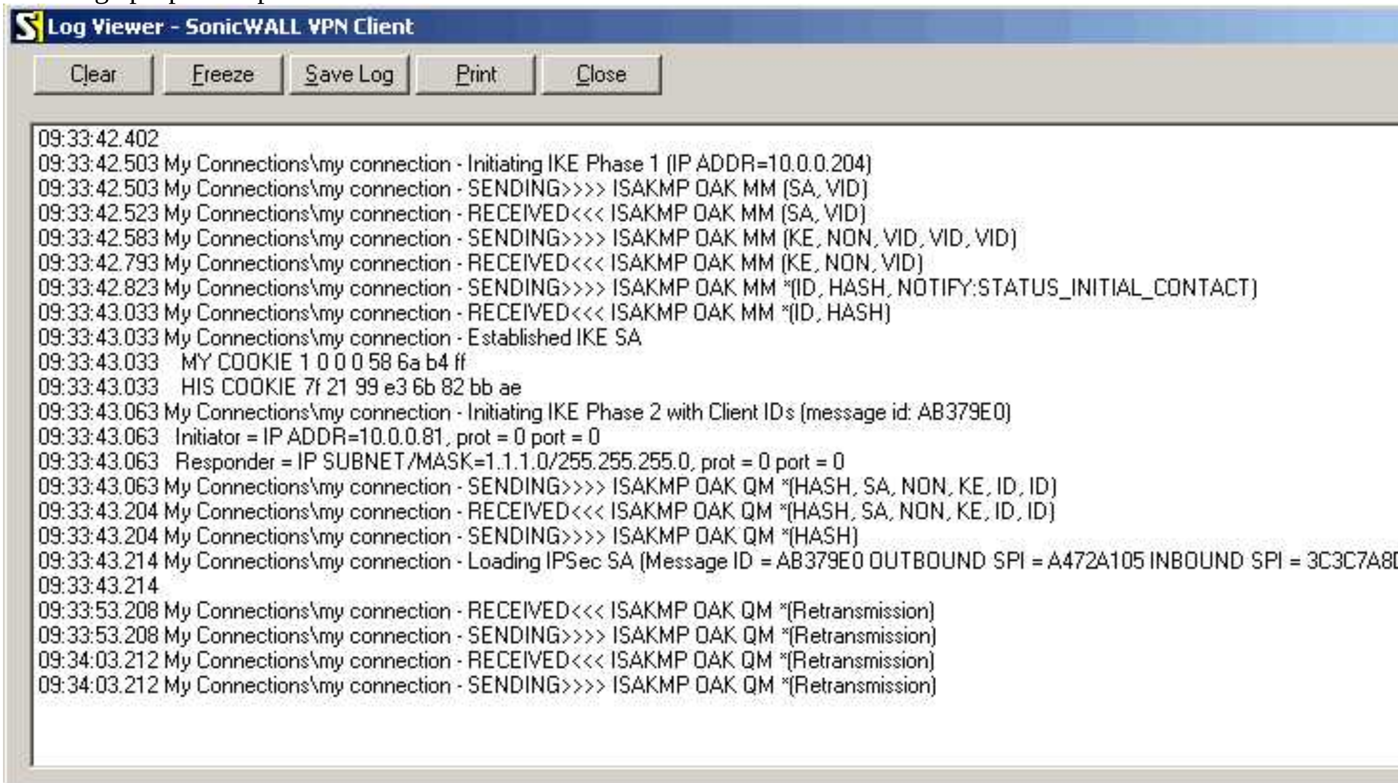
Testing

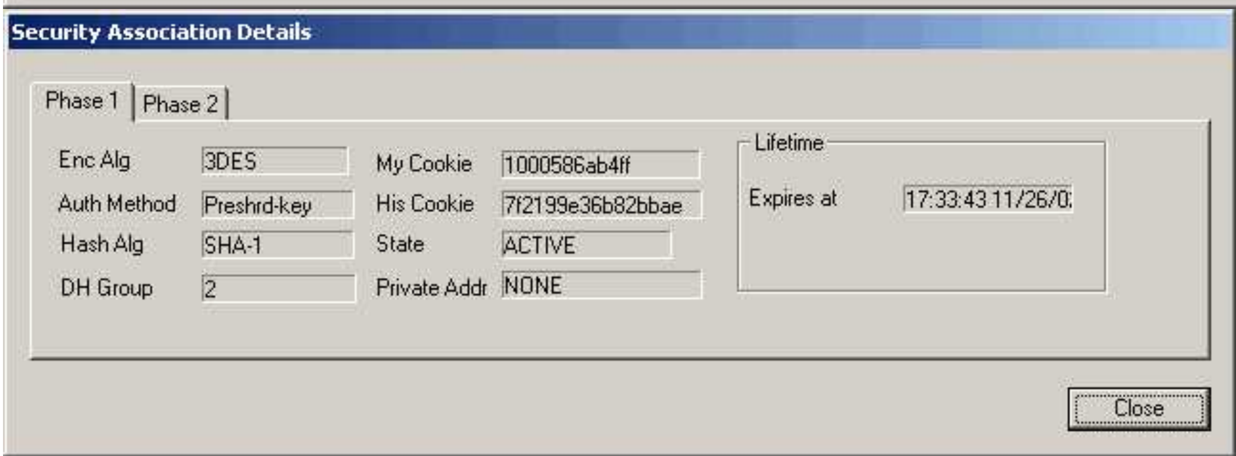
Try accessing some host on 1.1.1.0/24 network from 10.0.0.81 box. After some time IPSec tunnel will be established and data will start to pass through.

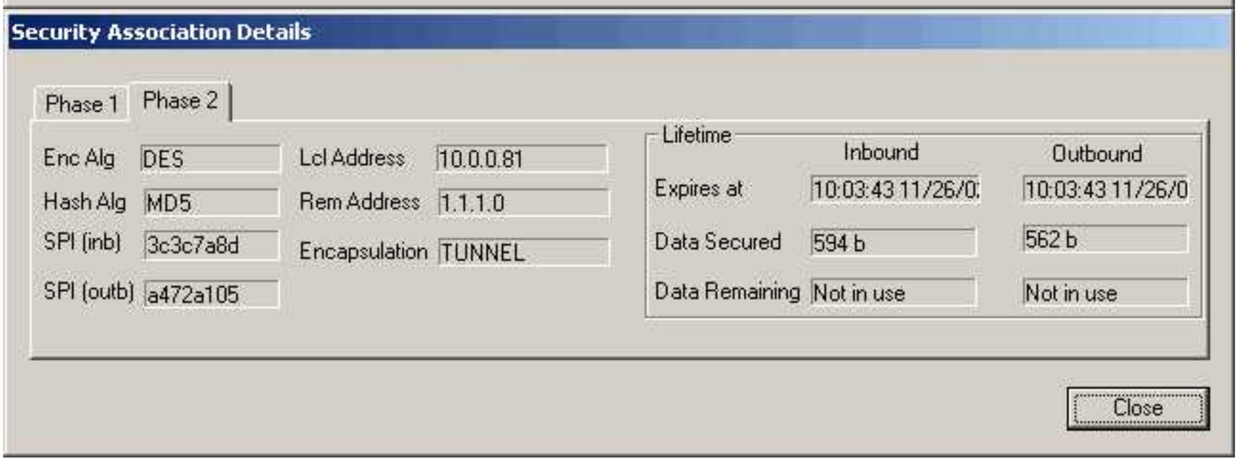
On RouterOS side you can see the statistics for established SAs:

```
[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:







IP Telephony

Document revision 29−Nov−2002
This document applies to the MikroTik RouterOS V2.6

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

Topics covered in this manual:

IP Telephony Specifications

Supported Hardware

The MikroTik RouterOS V2.6 supports following telephony cards from Quicknet Technologies, Inc. (http://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 V2.6 supports the Voicetronix OpenLine4 card for connecting four (4) analog telephone lines telephony cards from Voicetronix, Inc. (http://www.voicetronix.com.au/)

The MikroTik RouterOS V2.6 supports the Zaptel Wildcard X100P IP telephony card for connecting one analog telephone line from Linux Support Services (http://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 and Software Installation

Software Packages

The MikroTik Router should have the telephony package installed. To install the package, please upload it to the router and reboot. The package can be downloaded from MikroTik's web page www.mikrotik.com

The software package size is 1.7MB, after installation it requires 5MB of additional HDD space and 6MB of additional RAM. Please make sure you have the required capacity. Use **/system resource print** command to see the amount of available resources:

```
[admin@MikroTik] > system resource print
           uptime: 7m17s
     total-memory: 61240
      free-memory: 32756
         cpu-type: AMD-K6(tm)
    cpu-frequency: 300
        hdd-total: 46474
         hdd-free: 20900
[admin@MikroTik] >
```

You may want to increase the amount of RAM from 32MB to 48/64MB if you use telephony. Use the **/system package print** command to see the list of installed packages.

Pesase 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.

Software License

The telephony does not require any additional Software License. It works with the Basic License.

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 or Zaptel card.

Please consult the ISDN Manual for more information about installing the ISDN adapters.

IP Telephony Configuration

The IP Telephony requires IP network connection and configuration. The basic IP configuration can be done under the **/ip address** and **/ip route** menus.

Configuration of the IP telephony can be accessed under the **/ip telephony** menu:

```
[admin@MikroTik] ip> telephony
IP Telephony interface
  gatekeeper  Gatekeeper client configuration
  accounting  Accounting configuration
```

```
     numbers  Telephone numbers management
       codec  Audio compression capability management
  voice-port  Telephony voice port management
      region  Telephony voice port regional setting management
      export
[admin@MikroTik] ip> telephony
```

Telephony Voice Ports

The management of all IP telephony voice ports (**linejack**, **phonejack**, **isdn**, **voip**, **voicetronix**, **zaptel**) can be accessed under the **/ip telephony voice−port** menu. Use the **print** command to view the list of available telephony voice ports and their configuration.

```
[admin@MikroTik] ip telephony voice-port> print
Flags: X - disabled
  #   NAME                          AUTODIAL                     TYPE
  0   PBX_Line                                                   linejack
  1   ISDN_GW                                                    isdn
  2   VoIP_GW                                                    voip
[admin@MikroTik] ip telephony voice-port>
```

Description of arguments:

name − name assigned to the voice port by user.
type − type of the installed telephony voice port **linejack**, **phonejack**, **isdn**, **voip**, **voicetronix**, **zaptel**
autodial − number to be dialed automatically, if call is coming in from this voice port.

Note that 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

Use the **monitor** command under the corresponding menu to view the current state of the port, for example:

```
[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>
```

Note that monitoring feature is not available for VoIP ports

Argument description:

status − 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 − (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 − 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 − (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 − the number which is being dialed
remote−party−name − name and IP address of the remote party
codec − CODEC used for the audio connection
duration − duration of the audio call

Voice−Port Statistics

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. For example:

```
[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

[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 minimize the number of lost packets. 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

All commands relating the Quicknet, Voicetronix and Zaptel Wildcard cards are listed under the **/ip telephony voice−port** submenus. For example:

```
[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>
```

Argument descriptions:

name − name given by the user or the default one
type − (only for phonejack) type of the card (phonejack, phonejack−lite or phonejack−pci), cannot be changed
autodial − 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 − playback volume in dB, 0dB means no change, possible values are −48...48dB.
record−volume − recording volume in dB, 0dB means no change, possible values are −48...48dB.
ring−cadence − (only for quicknet cards) a 16−symbol ring cadence for the phone, each symbol is 0.5 seconds, **+** means ringing, **−** means no ringing.
region − 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 − echo detection and cancellation. Possible values are **yes** and **no**. If the echo cancellation is on, then the following parameters are used:

> **aec−tail−length** − size of the buffer of echo detection. Possible values are: **short**, **medium**, **long**.
> **aec−nlp−threshold** − level of cancellation of silent sounds. Possible values are 'off/low/medium/high'.
> **aec−attenuation−scaling** − factor of additional echo attenuation. Possible values are 0...10.
> **aec−attenuation−boost** − level of additional echo attenuation. Possible values are 0 ... 90dB
> **software−aec** − software echo canceller (experimental, for most of the cards)

agc−on−playback − automatic gain control on playback (can not be used together with hardware voice codecs)
agc−on record − automatic gain control on record (can not be used together with hardware voice codecs)
detect−cpt − automatically detect call progress tones

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 under several linejack cards.

Voice Port for ISDN

All commands relating the ISDN voice ports are listed under the **/ip telephony voice−port isdn** menu. 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.

```
[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>
```

Argument descriptions:

name − Name given by the user or the default one.
msn − Telephone number of the ISDN voice port (ISDN MSN number).
lmsn − 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 − 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 − playback volume in dB, 0dB means no change, possible values are −48...48dB.
record−volume − recording volume in dB, 0dB means no change, possible values are −48...48dB.
region − regional setting for the voice port (for tone generation only).
aec − 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 − size of the buffer of echo detection. Possible values are: **short** (8 ms), **medium** (16 ms), **long** (32 ms).
software−aec − software echo cancellation (experimental)
agc−on−playback − automatic gain control on playback
agc−on−record − automatic gain control on record

Voice Port for Voice over IP (voip)

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.

```
[admin@MikroTik] ip telephony voice-port voip> print detail
Flags: X - disabled
  0   name="VoIP_GW" autodial="" remote-address=10.5.8.12
      jitter-buffer=50ms prefered-codec=none silence-detection=no
      fast-start=yes

[admin@MikroTik] ip telephony voice-port voip>
```

Argument description:

name − Name given by the user or the default one.
remote−address − 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 with the address 0.0.0.0 is used. If there is no default record, then default values are used.
autodial − 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 − 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 − if **yes**, then silence is detected and no audio data is sent over the IP network during the silence period.
prefered−codec − the preferred codec to be used for this voip voice port. If possible, the specified codec will be used.
fast−start − 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.

Numbers

This is the so−called "routing table" for voice calls. This table assigns numbers to the voice ports.

```
[admin@MikroTik] ip telephony numbers> print
Flags: I - invalid, X - disabled
  #    DST-PATTERN      VOICE-PORT             PREFIX
  0    26               VoIP_GW                26
[admin@MikroTik] ip telephony numbers>
```

Argument description:

dst−pattern − 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 − voice port to be used when calling the specified telephone number.
prefix − 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.

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.

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.

Let us consider the following example for the number table:

```
[admin@MikroTik] ip telephony numbers> print
Flags: I − invalid, X − disabled
  #    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
  #    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
  #    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
  #    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

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.

Regional settings are managed under the /ip telephony region menu:

```
[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

  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
```

```
  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

  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

  ...
```

Argument 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** − Busy tone cadence in ms (0 − end of cadence)
> **busy−tone−frequency** − Frequency and volume gain of busy tone Hz x dB
> **data−access−arrangement** − ring voltage, impedance setting for line−jack card (**australia**, **france**, **germany**, **japan**, **uk**, **us**)
> **dial−tone−frequency** − Frequency and volume gain of dial tone Hz x dB
> **ring−tone−cadence** − Ring tone cadence in ms (0 − end of cadence)
> **ring−tone−frequency** − Frequency and volume gain of ring tone Hz x dB

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.

Sometimes it is necessary to add an additional regional setting matching the properties of a particular PBX. Use the **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 (0 − end of cadence)
      busy−tone−frequency  Frequency and volume gain of busy tone Hz x dB
                copy−from  item number
  data−access−arrangement  Ring voltage, impendance setting for line−jack card
      dial−tone−frequency  Frequency and volume gain of dial tone Hz x dB
                     name  New regional setting name
        ring−tone−cadence  Ring tone cadence in ms (0 − end of cadence)
      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 'office', you need to enter the command:

```
[admin@MikroTik] ip telephony region> set office dial−tone−frequency=350x−6,440x−6
```

Audio CODEC

The available Audio Coding and Decoding Protocols (CODEC) are listed under **/ip telephony codec** menu:

```
[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>
```

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.

IP Telephony Accounting

The RADIUS accounting feature can be configured under **/ip telephony accounting** menu:

```
[admin@MikroTik] ip telephony accounting> print
                    enabled: no
              radius−server: 0.0.0.0
              shared−secret: ""
    secondary−radius−server: 0.0.0.0
    secondary−shared−secret: ""
    interim−update−interval: 0s
[admin@MikroTik] ip telephony accounting>
```

Argument description:

enabled − defines whether RADIUS client is enabled or not
radius−server − IP address of accounting RADIUS server
shared−secret − secret shared with RADIUS server
secondary−radius−server − IP address of secondary RADIUS server
secondary−shared−secret − secret shared with secondary RADIUS server
interim−update−interval − 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 CDR (Call Detail Record) messages are sent to the main RADIUS server. If the main server does not respond, then these records are sent to the secondary RADIUS server. If the secondary RADIUS server does not respond neither, an error is sent to the Telephony−Error log. The router tries each server for three times waiting 0.7 seconds between the tries.

The contents of the CDR 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)

Note that all the parameters, which names begin with **h323**, are CISCO vendor specific Radius attributes

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"
```

Description of parameters:

gatekeeper − Select which gatekeeper 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 of remote gatekeeper to use. If set to 0.0.0.0, broadcast gatekeeper discovery is used
remote−id − 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
registered − shows whether local H.323 endpoint is registered to any gatekeeper
registered−with − name of gatekeeper to which local H.323 endpoint is registered

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.

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 gatkeeper. 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.

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

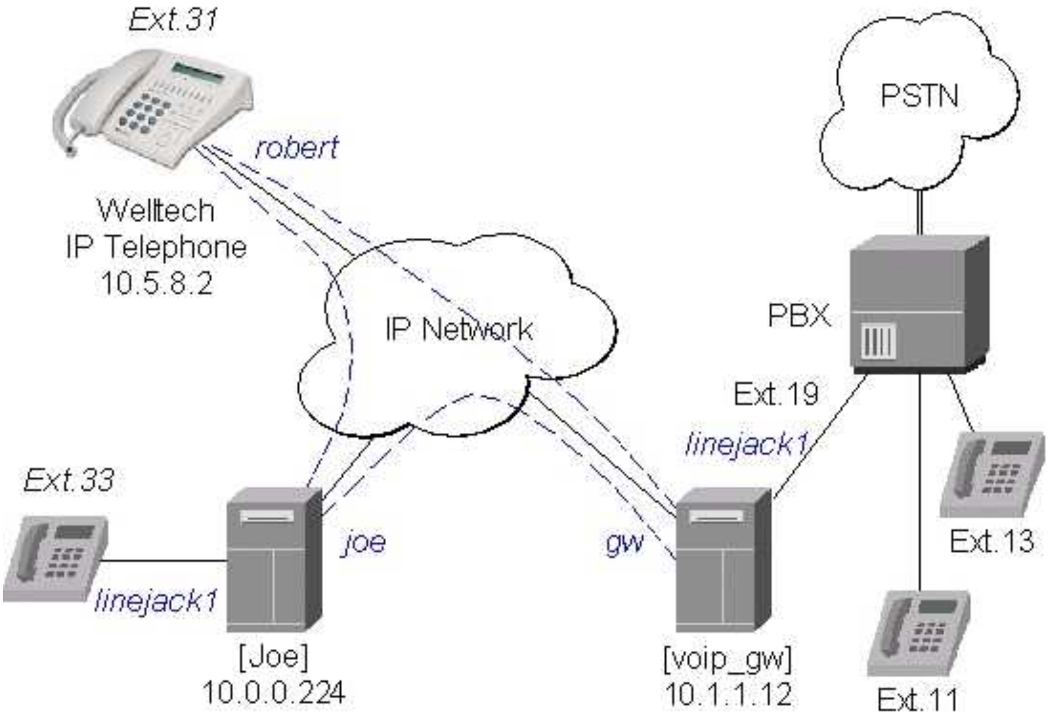
- **The IP Telephony does not work after upgrading from 2.5.x version**
  You need to completely reinstall the router using any installation procedure. You may keep the configuration using either the installation program option or the backup file.
- **The IP Telephony gateway does not detect the drop 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.

IP Telephony Applications

The following describes examples of some useful IP telephony applications using the MikroTik RouterOS Quicknet telephony cards or ISDN cards.

Setting up the MikroTik IP Telephone
Setting up the IP Telephony Gateway
Setting up the Welltech IP Telephone
Setting up the MikroTik Router and CISCO Router

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) 10.0.0.22. 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=robert 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    robert             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   robert                        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=robert
[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                      robert
  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=robert 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    robert             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=robert 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                        robert                   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 http://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     : Robert
    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
```

IP Traffic Accounting

Document revision 22−Nov−2002
This document applies to the MikroTik RouterOS v2.6

Overview

The IP Traffic Accounting feature enables administrators to keep an accurate record of traffic passed through the router (even through the bridged interfaces) between IP level hosts. ISPs or network administrators can use this for traffic based billing or detailed monitoring of network activity. This feature generates simple traffic data. Additional utilities are required for useful analysis and calculation of the traffic data. Information on utilities and examples of scripts for collecting data are provided in this manual.

The MikroTik RouterOS supports:

- Cisco **IP pairs** and **snapshot** image traffic data output style
- Collection of **snapshot** image with standard Unix/Linux utilities
- Collection of **snapshot** image with MT_Syslog utility
- Viewing of **snapshot** image from the console

Topics covered in this manual:

Installation

The Traffic Accounting feature is included in the 'system' package. No installation is needed for this feature.

Hardware Resource Usage

The maximum number [threshold] of **IP pairs** stored may require additional RAM installation. Each IP pair uses approximately 100 bytes. The system uses a **current** table which accounts for current data. The system also keeps the **snapshot** table for retrieval. Therefore, the memory usage for the IP pairs can be calculated with **number of IP pairs** x 100 bytes x 2 (for the two tables). The default threshold of IP pairs is set to 256 (50KB). When using the default threshold setting of 256, no additional memory is suggested. For threshold settings higher than 6500 (1MB), memory usage estimates should be made, system resources should be monitored, and RAM should be increased accordingly. The maximum setting is 8192 IP pairs.

Traffic accounting setup

```
[admin@MikroTik] ip accounting> set enabled yes
[admin@MikroTik] ip accounting> print
    threshold: 256
      enabled: yes
```

Description of arguments:

enabled − Traffic accounting is disabled by default
threshold − The threshold setting sets the maximum number of IP pairs for the traffic accounting table – see **Threshold settings** for more information on the optimal settings. The default setting is for 256 IP pairs.

Traffic data description

Only IP traffic is accounted. As each packet passes through the router, the packet source and destination is matched to an IP pair in the accounting table and the traffic for that pair is increased. User data for PPP, PPTP, PPPoE and ISDN connections are accounted too. If no matching IP or user pair exists, a new entry to the table will be created. Both the number of packets and number of bytes are accounted. Only packets that enter and leave the router are counted. 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 masqueraded with 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.

See Traffic Display and collection for a printout of a snapshot.

For example, a TCP connection between two computers with traffic going through the router will cause two IP pairs to be added to the traffic accounting table. One IP pair will have computer A as the source and computer B as the destination. Another IP pair will have computer B as the source and computer A as the destination.

Threshold settings

The threshold setting limits the maximum number of IP pairs in the accounting table. When the limit is reached, no new IP pairs will be added to the accounting table. Each packet that is not accounted for 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:

```
[MikroTik] ip accounting uncounted> print
    packets: 0
      bytes: 0
```

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. It is suggested that traffic data be collected every 15 minutes.

Traffic data display and collection

The traffic data can be viewed by both the telnet/terminal console and WinBox. The traffic data can be collected manually or by using standard Unix/Linux utilities and MikroTik's shareware MT_Syslog Daemon and Traffic Counter . This manual section will cover:

- Snapshots
- Web report setup

The traffic accounting system consists of a **current** accounting table and a **snapshot** image. When the **snapshot** image is made of the **current** accounting table, the **current** accounting table is cleared and starts accounting data anew. The **snapshot** image can be made in two ways.

An image of traffic data can be made manually by issuing the **/ip accounting snapshot take** command from the terminal/console or WinBox. The **snapshot** can then be viewed with the **/ip accounting snapshot print** command. The traffic data from the telnet/terminal console will appear:

```
[admin@MikroTik] ip accounting snapshot> print
  # SRC−ADDRESS     DST−ADDRESS     PACKETS    BYTES      SRC−USER   DST−USER
  0 10.0.0.4        159.148.147.198 6589       517850
  1 10.7.2.250      10.0.0.161      307403     19673792
  2 10.0.0.161      10.7.2.250      307403     19673792
  3 159.148.147.198 10.0.0.4        6589       680894
  4 10.0.0.99       159.148.147.194 213        12700
```

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. If the web report is enabled and the web page is viewed, the **snapshot** will be made when the wget (or standard browser) 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

Note that ip.cgi has different value order: **src−address**, **dst−address**, **bytes**, **packets**, **src−user**, **dst−user**

```
[admin@MikroTik] ip accounting web−access> print
    accessible−via−web: yes
               address: 0.0.0.0/0
[admin@MikroTik] >
```

For security purposes, an IP address or IP subnet can be limited to the collection of the web report. The above example of address: 0.0.0.0/0 allows all IP hosts to access the web reports. With the settings address: 10.1.0.3/32, only IP host 10.1.0.3 is allowed to access the web reports.

A simple script can be run with crond and wget to periodically collect traffic data. Timestamps can be added to the traffic data file as well as other features.

<u>MikroTik Download Utilities Page</u>

Traffic data analysis

There are many tools and systems to analyze traffic data. Useful common tools are:

- Microsoft Excel
- Grep – Unix/Linux utility

- Perl scripts

Additional Resources

Links for documentation:

http://www.gnu.org/manual/wget/
http://www.gnu.org/manual/grep−2.4/

IP Packet Packer Protocol (M3P)

Document revision 9−Aug−2002
This document applies to the MikroTik RouterOS v2.6

Overview

The MikroTik Packet Packer Protocol (M3P) optimizes the bandwidth 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

Contents of the Manual

The following topics are covered in this manual:

Installation

The MikroTik Packet Packer Protocol feature is included in the “system” package. No installation is needed for this feature.

Hardware Resource Usage

There is no significant resource usage.

MikroTik Packet Packer Protocol Description

The wireless protocol IEEE 802.11 and, to a lesser extent, Ethernet protocol have a high overhead per packet because for each packet it is necessary to access the media, check for errors, resend in case of errors, 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 useful 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 enabled by default for all new wireless 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

IP MikroTik Packet Packer Protocol is working only between MikroTik routers, which are discovered with MikroTik Neighbor Discovery Protocol. So you should enable MNDP in order to get M3P to work. Consult MNDP manual on how to do it.

IP MikroTik Packet Packer Protocol management can be accessed under the **/ip packing** submenu:

```
[admin@MikroTik] ip packing>
  interface  Interface settings
      print  Show packing settings
        get  get value of property
        set
     export  display the configuration as a set of commands
[admin@MikroTik] ip packing> print
    enable−unpacking: yes
       expected−size: 28
     aggregated−size: 1500
[admin@MikroTik] ip packing>
```

Argument description:

> **enable−unpacking** – enables unpacking feature of M3P for all Ethernet like interfaces on the router – should be enabled if you have any interface set to send M3P packets.
> **expected−size** – the average size packet you expect for aggregation, i.e., if your VoIP generates 100 byte packets – this would be the expected size. This is used by the protocol to determine if it should wait for another packet to complete the aggregated packet – determined by the **aggregated−size** setting – or send an aggregated packet immediately even though it has not reached the size of the **aggregated−size** setting.
> **aggregated−size** – the maximum size of the aggregated packet – the suggested setting is 1000 bytes and the maximum setting is the MTU size of the interface (generally 1500 bytes)

To see the interface settings use:

```
[admin@MikroTik] ip packing interface> print
Flags: X − disabled
  #   INTERFACE
  0 X bridge1
  1 X ether1
  2 X Local219
  3   wireless
[admin@MikroTik] ip packing interface>
```

© Copyright 1999−2002, MikroTik

MikroTik Neighbor Discovery Protocol (MNDP))

Document revision 9−Aug−2002
This document applies to the MikroTik RouterOS v2.6

Overview

The MikroTik Neighbor Discovery Protocol (MNDP) eases configuration and management by enabling each MikroTik router to discover other connected MikroTik routers and learn information about the system and features which are enabled. The MikroTik routers can then automatically use set 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

Contents of the Manual

The following topics are covered in this manual:

Installation

The MikroTik Discovery Protocol feature is included in the 'system' package. No installation is needed for this feature.

Hardware Resource Usage

There is no significant resource usage.

MikroTik Discovery Protocol Description

MNDP basic function is to assist with automatic configuration of features that are only available between two 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. The MikroTik routers must be connected by an Ethernet like interface.

Specific Properties:

- works on interfaces that support IP protocol and have least one IP address
- is enabled by default for all new Ethernet−like interfaces −− Ethernet, radio, 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
- a 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 Discovery Protocol Setup

IP MikroTik Packet Packer Protocol management can be accessed under the **/ip neighbor** submenu:

```
[admin@MikroTik] ip neighbour>

      print  print values of item properties
       find  finds items by value
        get  get value of item's property
  interface  interfaces
     export
[admin@MikroTik] ip neighbour> print
  # INTERFACE  ADDRESS         MAC−ADDRESS       UNPACKING AGE
  0 Public     10.5.8.196      00:E0:C5:BC:12:07 yes       23s
  1 Public     10.5.8.167      00:E0:4C:39:23:31 yes       0s
  2 Public     10.5.8.1        00:80:C8:C9:B0:45 yes       3s
[admin@MikroTik] ip neighbor>
```

Argument description:

interface – local interface to which the neighbor is connected
address – IP address of the neighbor router
mac−address – MAC−address of the neighbor router
unpacking – identifies if the interface of the neighbor router is unpacking 'Packed Packets'
age – a counter (in seconds) that shows the age of the information

To see the interface settings use:

```
[admin@MikroTik] ip neighbor interface> print
  # NAME                DISCOVER
  0 Public              yes
  1 Local               yes
[admin@MikroTik] ip neighbor interface>
```

To change the interface settings, use **/ip neighbor interface set** command:

```
[admin@MikroTik] ip neighbor interface> set Public discover=no
[admin@MikroTik] ip neighbor interface> print
  # NAME                DISCOVER
  0 Public              no
  1 Local               yes
[admin@MikroTik] ip neighbor interface>
```

© Copyright 1999−2002, MikroTik

IP Route Management

Document revision 16−Oct−2002
This document applies to the MikroTik RouterOS V2.6

Overview

The following Manual discusses managing the IP routes. 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.
- About **OSPF**, **RIP** and **BGP** dynamic routing protocols, see respective manuals

Contents of the Manual

The following topics are covered in this manual:

Adding Static Routes

Any static route can be added using the **add** command under the **/ip route** menu. 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. For example, we can add two static routes to networks 192.168.0.0/16 and 0.0.0.0/0 (the default destination address) of a router with two interfaces and two IP addresses:

```
[admin@MikroTik] ip route> /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
[admin@MikroTik] ip route> add
creates new item with specified property values.
           comment  short description of the item
         copy−from  item number
          disabled
          distance
       dst−address  Destination
           gateway  Gateway
           netmask  Network mask
  preferred−source  Source address of packets leaving the router
[admin@MikroTik] ip route> add dst−address=192.168.0.0/16 gateway=10.0.0.2
[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  S 192.168.0.0/16     r 10.10.10.2      1        Public
    2 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  S dst−address=0.0.0.0/0 preferred−source=0.0.0.0 gateway=10.0.0.1
         gateway−state=reachable distance=1 interface=Public

    1  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 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>
```

Description of the printout:

number − number assigned to the item in the list
flag − shows the status of the item
dst−address/netmask − destination address and network mask, where mask is number of bits in the subnet mask.
gateway − gateway host, that can be reached directly through some of the interface. You can specify multiple gateways separated by comma "," for equal cost multipath routes. See more information on that below.
gateway−state − shows the status of the next hop. Can be **r** (reachable) or **u** (unreachable).
preferred−source − 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, where the packet leaves. Default value is 0.0.0.0, i.e., it is determined at the time of sending the packet out through the interface.
interface − interface through which the gateway can be reached. If **(unknown)**, then the gateway cannot be reached directly, or the route has been disabled.
distance − administrative distance of the route. When forwarding a packet the router will use the route with the lowest administrative distance and reachable gateway.

Equal Cost Multipath Routing

Equal cost multipath routing feature can be used for load balancing.

New gateway is chosen for new source/destination IP pair. This means that, for example, one FTP connection will use only one link, but new connection to 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. 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.

To create a static multipath route, specify the gateway argument in the form "gateway=x.x.x.x,y.y.y.y", for example:

```
[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 192.168.1.0/24     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 route> set 0 gateway=192.168.0.50,192.168.0.51,10.0.0.17
[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 192.168.1.0/24     r 192.168.0.50    1        Local
                            r 192.168.0.51             Local
                            r 10.0.0.17                Public
    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 route>
```

Note that you can specify more than 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.

Policy Routing

Policy routing is implemented using multiple routing tables and list of rules that specify how these tables should be used.

The Policy Routing is implemented in the MikroTik RouterOS based on source and destination addresses of the packet and on the interface the packet arrives at the router.

> ***Note!*** 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.

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 the routing table does not have a route for the packet, next rule after the one that directed to current table is examined, until either route is found, end of rule list is reached, or some rule with action drop or unreachable is hit.

This way 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).

Action for the rule can be one of:

- **drop** − silently drop packet
- **unreachable** − reply that destination host is unreachable
- **lookup** − lookup route in given routing table

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.

Policy routing rules are configured in /ip policy−routing rule menu

```
[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
[admin@MikroTik] ip policy-routing rule>
```

After installation, there is one default rule, which says that routes for all packets should be looked up in the "main" table. Argument description:

> **src−address/mask** − Source IP address/mask, where mask is number of bits in the subnet. For example, x.x.x.x/32 for the address x.x.x.x and the 32−bit netmask 255.255.255.255
> **dst−address/mask** − Destination IP address/mask, where mask is number of bits in the subnet.
> **interface** − 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** − flow mask of the packet to be matched by this rule. The flow masks are set using '/ip firewall mangle'.

Routing tables can be created/deleted in the **/ip policy−routing** menu:

```
[admin@MikroTik] ip policy-routing> print
Flags: D - dynamic
  #   NAME
  0 D main
[admin@MikroTik] ip policy-routing>
```

There is always the table "main" − this one can not be deleted and its name can not be changed. The "main" table is routing table that can be changed by issuing commands in the **/ip route** menu.

A new table can be added:

```
[admin@MikroTik] ip policy-routing> add name=mt
[admin@MikroTik] ip policy-routing> print
Flags: D - dynamic
  #   NAME
  0   karlis
  1 D main
[admin@MikroTik] ip policy-routing>
```

Routes in a routing table can be added/removed/changed in **/ip policy−routing table _table−name_** menu, where **_table−name_** is name of the 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>
```

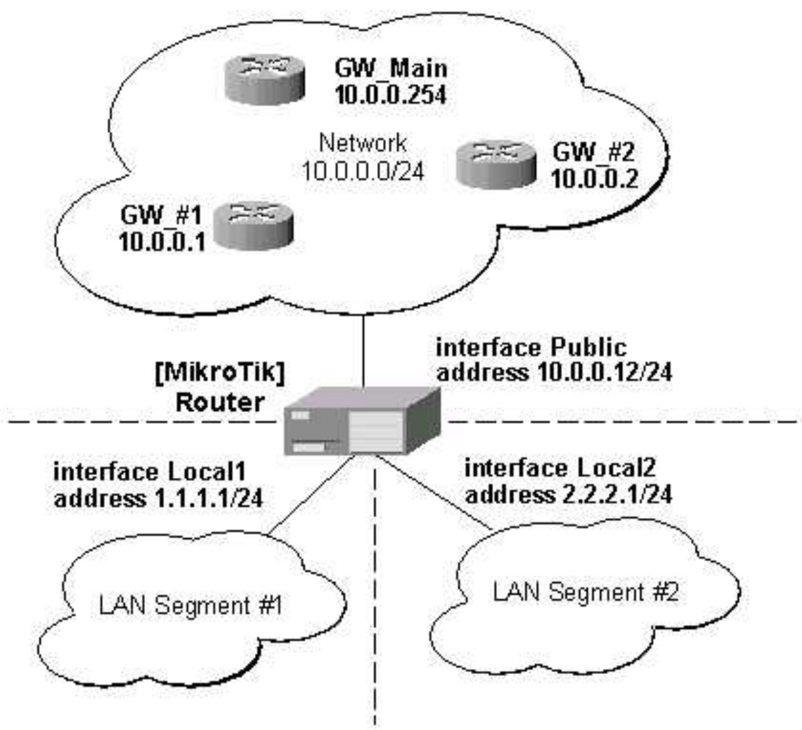
The "main" table is the same as one in **/ip route**:

```
[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>
```

Application Example for Policy Routing

We want packets coming from 1.1.1.0/24 use gateway 10.0.0.1 and packets from 2.2.2.0/24 use gateway 10.0.0.2. And the rest of packets use gateway 10.0.0.254 (assuming we already have it so):



Commands 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: X - disabled
  #   NAME
  0   from_net1
  1   from_net2
  2   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        NEXTHOP-S... GATEWAY     DISTANCE INTERFACE
  0    static  0.0.0.0/0          A            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        NEXTHOP-S... GATEWAY     DISTANCE INTERFACE
  0    static  0.0.0.0/0          A            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        INTERFACE   ACTION      TABLE
  0   0.0.0.0/0          0.0.0.0/0          all         lookup      main
[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        INTERFACE   ACTION      TABLE
  0   0.0.0.0/0          0.0.0.0/0          all         lookup      main
  1   1.1.1.1/32         0.0.0.0/0          all         lookup      main
  2   2.2.2.1/32         0.0.0.0/0          all         lookup      main
  3   1.1.1.0/24         0.0.0.0/0          all         lookup      from_net1
  4   2.2.2.0/24         0.0.0.0/0          all         lookup      from_net2
[admin@MikroTik] ip policy-routing rule> move 0 4
[admin@MikroTik] ip policy-routing rule> print
Flags: X - disabled, I - invalid
  #   SRC-ADDRESS        DST-ADDRESS        INTERFACE   ACTION      TABLE
  0   1.1.1.1/32         0.0.0.0/0          all         lookup      main
  1   2.2.2.1/32         0.0.0.0/0          all         lookup      main
  2   1.1.1.0/24         0.0.0.0/0          all         lookup      from_net1
  3   2.2.2.0/24         0.0.0.0/0          all         lookup      from_net2
  4   0.0.0.0/0          0.0.0.0/0          all         lookup      main
[admin@MikroTik] ip policy-routing rule>
```

Here the rules #0 and #1 are needed to process correctly connections from the local networks to the local addresses of the router. Namely, the 'connected' routes from the main table should be used instead of using the default routes from table from_net1 or from_net2. Rules #2 and #3 will handle packets with destination other than locally connected networks.

Additional Resources

Recommended readings for guidelines on routing issues:

- http://www.ietf.org/rfc/rfc2328.txt

Services, Protocols, and Ports

Document revision 23−Oct−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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.

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

Some service settings can be changed under **/ip service** menu. You can specify IP addresses from which the service is accessible, for example:

```
[admin@MikroTik] ip service> set www port=8081 address=10.5.0.0/16
[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                                                         8081  10.5.0.0/16
[admin@MikroTik] ip service>
```

Below is list of protocols and ports used by MikoTik RouterOS services. Some services require additional package to be installed, as well as enabling them, 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
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
```

WEB Proxy

Document revision 22−Oct−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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, and URL.
- Cache access list.

Contents of the Manual

The following topics are covered in this manual:

Installation

The MikroTik Web Proxy feature is included in the **web−proxy** package. To install the web−proxy package, upload it to the router and reboot. After successful install of the web−proxy package it should be listed under the **/system package print** list.

Software License

The web−proxy does not require any additional Software License. It works with the Basic License. **Note** that web−proxy does not work with Demo License.

Hardware Resource Usage

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 reserves for the system.

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 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!

MikroTik Web Proxy Setup

The Web Proxy management can be accessed under the **/ip web−proxy** submenu:

```
[admin@MikroTik] > ip web-proxy
HTTP proxy
  clear-cache  Clear http cache
       access  Access list
        cache  Cache access list
       direct  Direct access list
      monitor  Monitor proxy status and usage
        print  Print current configuration and status
          get  Get value of configuration property
          set  Change proxy configuration
       export  Export web proxy settings
[admin@MikroTik] > ip web-proxy
```

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 by the command **/ip web−proxy clear−cache**.

Monitoring the Web Proxy

Use the command **/ip web−proxy print** to see the current web proxy status:

```
[admin@MikroTik] ip web-proxy> print
                enabled: yes
                address: 0.0.0.0:3128
               hostname: "proxy.mt.lv"
      transparent-proxy: yes
           parent-proxy: 10.5.5.1:8080
    cache-administrator: "support@mt.lv"
        max-object-size: 10000 kB
                 status: running
     reserved-for-cache: 2633728 kB
[admin@MikroTik] ip web-proxy>
```

Description of the parameters:

enabled − whether web−proxy is enabled or not
address − IP address (**0.0.0.0** for any) and port (mandatory) on which proxy will listen for

requests
hostname − hostname (DNS or IP address) of the web proxy
transparent−proxy − use transparent mode
parent−proxy − upper−level proxy. Use **0.0.0.0:0** to disable parent−proxy
max−object−size − objects larger than this size will not be saved on disk. The value is specified in kilobytes, and the default is **4096**. 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
status − displays status of the proxy server. Can be one of the following:

- **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.

reserver−for−cache − maximal cache size, that is accessible to web−proxy

Access logs are sent to Web−Proxy−Access logging facility. These logs can be disabled, logged locally or sent to remote address. To log locally:

```
/system logging facility set Web−Proxy−Access logging=local
```

In this case logs can be viewed using **/log print** command.

Some more statistics details can be monitored with **/ip web−proxy monitor** command:

```
[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] >
```

Printout description:

status − the same as for **/ip web−proxy print**
uptime − uptime of the proxy server
clients − number of present and past proxy clients (in current uptime)
requests − total number of requests to the proxy (in current uptime)

hits − number of requests satisfied with proxy's cache (in current uptime)
cache−size − current cache size in kilobytes
received−from−servers − how many kilobytes did proxy receive from remote servers (in current uptime)
sent−to−clients − how many kilobytes did proxy send to the clients to resolve their requests (in current uptime)
hits−sent−to−clients − how many kilobytes of sent traffic were taken from the cache (in current uptime)

Access List

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 or substring of requested url. 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.

```
[admin@MikroTik] ip web−proxy access> print
Flags: X − disabled
  #   SRC−ADDRESS         DST−ADDRESS         DST−PORT     URL            ACTION
  0   0.0.0.0/0           0.0.0.0/0           0−65535      .mp3           deny
  1   10.0.0.1/32         0.0.0.0/0           0−65535                     allow
  2   0.0.0.0/0           0.0.0.0/0           0−65535      ftp://         deny
  3   10.0.0.0/24         10.9.9.128/28       0−65535                     allow
  4   0.0.0.0/0           0.0.0.0/0           0−65535                     deny
[admin@MikroTik] ip web−proxy access>
```

Argument description:

src−address − source address of the request
dst−address − destination address of the request
dst−port − destination port of the request
url − the URL of the request. Can be regular expression
action − action to take (**allow**, **deny**)

Access list, shown above, disables access to any mp3 files for everyone.
Local gateway 10.0.0.1 has access to everything else (excluding mp3 files).
All other local network (10.0.0.0/24) users have access to servers located at 10.9.9.128/28, but, ftp protocol is not allowed for them.
Any other request is denied.

Details about regular expressions used in **url** field can be found here:
http://www.cs.utah.edu/dept/old/texinfo/regex/regex_toc.html

Security Notice
If you have web−proxy running, someone is probably using you as a relay. You have to use access rules in the web−proxy setting denying all IP addresses except those behind the router. Also, consult examples in Firewall Manual on how to protect your router.

Direct Access List

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.

Description of the action argument values:

- **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**.

Default action (if no rules specified or request did not match any) is **deny**.

Managing the Cache

Cache access list specifies, which requests (domains, servers, pages) have to be cached locally by web proxy, and which not. The Web Proxy cache access list is located under the **/ip web−proxy cache** submenu.

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). By default, one cache access rule is already added:

```
[admin@MikroTik] ip web−proxy cache> print
Flags: X − disabled
  #   SRC−ADDRESS         DST−ADDRESS         DST−PORT     URL            ACTION
  0   0.0.0.0/0           0.0.0.0/0           0−65535      cgi−bin \?     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.

Note: Objects, which are larger than max−object−size, are not cached.

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: 0.0.0.0:3128
               hostname: "proxy.mt.lv"
      transparent−proxy: yes
           parent−proxy: 10.5.5.1:8080
    cache−administrator: "support@mt.lv"
        max−object−size: 10000 kB
                 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!

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
  ```
- Enable the proxy service:

  ```
  /ip web-proxy set enabled=yes
  ```

Now it is possible to use this proxy, by setting it as proxy for IE, Netscape, Opera, etc.

Troubleshooting

- **Can I use transparent proxy feature on a MikroTik router with bridged interfaces?**
  No. Transparent proxy requires redirection of IP packets by firewall destination NAT. NAT is not involved when packets are passed from one bridged interface to another. But packets have to be translated by firewall destination NAT for transparent web−proxy to work. So, web−proxy is not going to work in transparent mode between bridge interfaces.
- **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 src−nat add dst−address=address/32:80 protocol=tcp action=accept
BEFORE the redirect rule. Alternatively, you can use just one rule
/ip firewall src−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 Bandwidth Management

Document revision 17−Jan−2003
This document applies to the MikroTik RouterOS V2.6

Overview

Queuing is a mechanism that control bandwidth 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

The queuing can be used for limiting the bandwidth for certain IP addresses, protocols or ports. The queuing is performed for packets leaving the router through a physical interface. It means that the queues should always be configured on the outgoing interface regarding the traffic flow. If there is a desire to limit the traffic arriving at the router, then it should be done at the outgoing interface of some other router. But in some cases you can use firewall rule that simply drop packets when traffic matching this rule exceeds some value.

Contents of the Manual

The following topics are covered in this manual:

Installation

The queue management feature is included in the 'system' software package. No additional software package installation is needed for this feature.

How Queues Work

There are four types of simple queues implemented in RouterOS: PFIFO, BFIFO, SFQ and RED. This chapter explains difference between these types and introduces queue trees.

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 are very useful if you have different kinds of traffic which should have differing 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 physical interface) Class Based Queue (CBQ) 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 bandwidth 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 CBQ, but with upper limit.

Each queue represents a virtual interface with the allowed bandwidth. It can be borrowed from sibling queues (queues that are children of one queue) if we set **bounded** to **no**. If we set **bounded** to **yes**, the queue can not occupy bandwidth of other queues. If set to **no**, the queue would use over the allocated bandwidth whenever possible. Only when other queues are getting too long and a connection is not to be satisfied, then the 'not−bounded' queues would be limited at their allocated bandwidth. When the 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 there are some queues with the same **priority** value, they are asked in Weighted Round Robin (WRR) fashion. In each WRR round the queue can send the amount of data equal to **weight*****allot**, where **allot** is the amount of data sent in one turn, and **weight** shows the number of allowed transmittings in one Weighted Round Robin round (for example, if there are two queues, but **weight** for the second is two times higher then for the first, then the second queue gets its data sent two times in a round, while the first queue − only one time). That is why **allot** should be bigger than interface MTU (MTU+14 works fine in most cases).

max−burst parameter specifies the maximal number of packets that can burst when there are no packets in the queue. In other words, when current data rate is below the limit, **max−burst** packets may spillover before the actual limiting will be applied. CBQ algorithm obviates the possibility of exceeding the allowed average data rate.

Configuring Simple Queues

Simple queues can be used to set up bandwidth 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.

To add simple queues, use the **/queue simple add** command:

```
[admin@MikroTik] queue simple> add dst−address=192.168.0.0/24 interface=ether1 \
limit−at=128000
[admin@MikroTik] queue simple> print
Flags: X − disabled, I − invalid
  0   name="" src−address=0.0.0.0/0 dst−address=192.168.0.0/24
      interface=ether1 limit−at=128000 queue=default priority=8 bounded=yes

[admin@MikroTik] queue simple>
```

Argument description:

name − descriptive name for the queue
src−address − Source IP address. Can be set in the form a.b.c.d/n, where n is network mask
src−netmask − Source netmask in decimal form a.b.c.d
dst−address − Destination IP address. Can be in the form a.b.c.d/n
dst−netmask − Destination netmask in decimal form a.b.c.d
interface − Outgoing interface of the traffic flow
limit−at − Maximum stream bandwidth (bits/s). **0** means no limit (default for the interface).
queue − 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**. See the **/queue type** for available types.
priority − Flow priority (1..8), **1** is the highest.
bounded − Queue is bounded.

To track how the rules are processed, see the bytes and packets counters for the queues:

```
[admin@MikroTik] queue simple> .. tree print
Flags: X − disabled, I − invalid, D − dynamic
  0  D name="" parent=ether1 flow="" limit−at=128000 max−burst=20
       queue=default priority=8 weight=1 allot=1514 bounded=yes


[admin@MikroTik] queue simple>
```

Queue rules are processed in the order they appear in the **/queue tree print** 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.

Queue Types

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. You can add your own queue types by specifying the argument values, for example:

```
[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>
```

Argument description:

name − name for the queue type
kind − 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
none − (same as **default**) The queue type as it is by default for the specific interface.
bfifo−limit − BFIFO queue limit. Maximum packet number that queue can hold.
pfifo−limit − PFIFO queue limit. Maximum byte number that queue can hold.
red−limit − RED queue limit
red−min−threshold − RED minimum threshold
red−max−threshold − RED maximum threshold
red−burst − RED burst
sfq−perturb − amount of data in bytes that can be sent in one round−robin round
sfq−allot − how often to change hash function

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.

Setting Default Queue Type for the Interface

To change the default queue type for the interface, use the **/queue interface set** command, e.g.:

```
[admin@MikroTik] queue interface> print
  # INTERFACE                              QUEUE
  0 ether1                                 ethernet−default
  1 prism1                                 default
[admin@MikroTik] queue interface> set prism1 queue=wireless−default
[admin@MikroTik] queue interface> print
  # INTERFACE                              QUEUE
  0 ether1                                 ethernet−default
```

```
  1 prism1                              wireless-default
[admin@MikroTik] queue interface>
```

Configuring Queue Trees

The queue trees should be used when you want to use sophisticated bandwidth allocation based on protocols, ports, groups of IP addresses, etc. If you have added a simple queue, it is listed as dynamic one under the **/queue tree print**, e.g.:

```
[admin@MikroTik] queue simple> print
Flags: X - disabled, I - invalid
  0   name="A_Simple" src-address=0.0.0.0/0 dst-address=192.168.0.0/24
      interface=ether1 limit-at=128000 queue=default priority=8 bounded=yes


[admin@MikroTik] queue simple> .. tree
[admin@MikroTik] queue tree> print
Flags: X - disabled, I - invalid, D - dynamic
  0  D name="A_Simple" parent=ether1 flow="" limit-at=128000 max-burst=20
       queue=default priority=8 weight=1 allot=1514 bounded=yes


[admin@MikroTik] queue tree>
```

Argument description:

name − descriptive name for the queue
parent − name of the parent queue. The top−level parents are the available interfaces (actually, main CBQ). Lower level parents can be other queues. Dynamic queues (created with the simple queue tool) cannot be used as parents.
flow − 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 − Maximum stream bandwidth (bits/s). **0** means no limit (default for the interface).
max−burst − Maximal number of packets allowed for bursts of packets when there are no packets in the queue. Set to **0** for no burst.
queue − queue type. See the **/queue type** for available types.
priority − Flow priority (1..8), **1** is the highest.
weight − Flow weight in the Weighted Round Robin process
allot − Number of bytes allocated for the bandwidth. Should not be less than the MTU for the interface.
bounded − Queue is bounded.

To apply queues on flows, the mangle feature should be used first to mark incomming packets:

```
[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>
```

See the Firewall Filters and Network Address Translation (NAT) Manual for details on how to mark the packets.

You can add queue using the **/queue tree add** command:

```
[admin@MikroTik] queue tree> add name=HTTP parent=ether1 flow=abc-http \
limit-at=128000 max-burst=0 bounded=yes
[admin@MikroTik] queue tree> print
Flags: X - disabled, I - invalid, D - dynamic
  0  D name="A_Simple" parent=ether1 flow="" limit-at=128000 max-burst=20
       queue=default priority=8 weight=1 allot=1514 bounded=yes

  1    name="HTTP" parent=ether1 flow=abc-http limit-at=128000 max-burst=0
       queue=default priority=8 weight=1 allot=1514 bounded=yes


[admin@MikroTik] queue tree> print brief
Flags: X - disabled, I - invalid, D - dynamic
  #    NAME          PARENT      FLOW        LIMIT-AT   PACKETS    BYTES
  1  D A_Simple      ether1                  128000     0          0
  0    HTTP          ether1      abc-http    128000     0          0
[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.
- *The traffic is not limited, when the **bounded** parameter is not set to **yes**.*
  Use the **bounded** flag for the queue, if you do not want to exceed the set limit when other queues are not using the available bandwidth for the interface.
- *Queuing does not work for the start of the file transfer. It starts limiting the bandwidth only after the first x packets have been downloaded.*
  Do not use the **burst** parameter value greater than **0**, if you do not want to allow any traffic bursts.

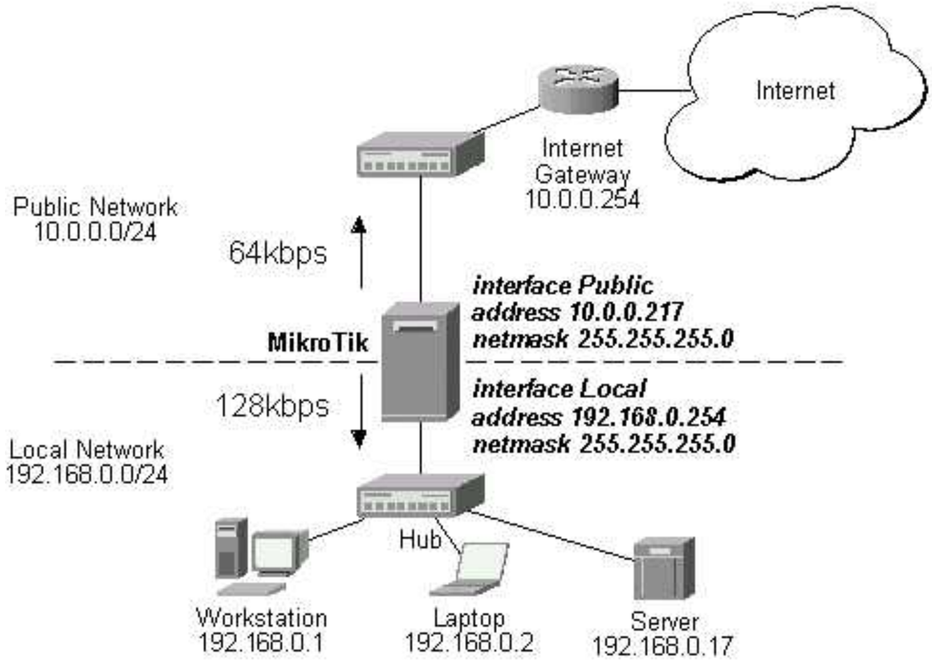
Queue Applications

One of the ways to avoid network traffic ‘jams’ is usage of traffic shaping in large networks. Traffic shaping and bandwidth allocation is implemented in the MikroTik RouterOS as queuing mechanism. Thus, the network administrator is able to allocate a definite portion of the total bandwidth and grant it to a particular network segment or interface. Also the bandwidth of particular nodes can be limited by using this mechanism.

Further on, several examples of using bandwidth management are given arranged according to complexity:

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

  1   name="UP" src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Public
      limit-at=64000 queue=default priority=8 bounded=yes
```

```
[admin@MikroTik] queue simple> .. tree print
Flags: X - disabled, I - invalid, D - dynamic
  0  D name="Down" parent=Local flow="" limit-at=128000 max-burst=20
       queue=default priority=8 weight=1 allot=1514 bounded=yes


  1  D name="UP" parent=Public flow="" limit-at=64000 max-burst=20
       queue=default priority=8 weight=1 allot=1514 bounded=yes


[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 **limit−at=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 limit-at=0
[admin@MikroTik] queue simple> add name=Serv_U interface Public \
\... src-address=192.168.0.17/32 limit-at=0
[admin@MikroTik] queue simple> print
Flags: X - disabled, I - invalid
  0   name=Down src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Local
      limit-at=128000 queue=default priority=8 bounded=yes

  1   name=UP src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Public
      limit-at=64000 queue=default priority=8 bounded=yes

  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 bounded=yes

  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 bounded=yes

[admin@MikroTik] queue simple> move 2 0
[admin@MikroTik] queue simple> move 3 1
```

```
[admin@MikroTik] queue simple> print
Flags: X - disabled, I - invalid
  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 bounded=yes

  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 bounded=yes

  2   name=Down src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Local
      limit-at=128000 queue=default priority=8 bounded=yes

  3   name=UP src-address=0.0.0.0/0 dst-address=0.0.0.0/0 interface=Public
      limit-at=64000 queue=default priority=8 bounded=yes

[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 \
\... limit-at=64000 bounded=yes max-burst=0
[admin@MikroTik] queue tree> print
Flags: X - disabled, I - invalid, D - dynamic
  0    name=Server parent=Public flow=Serv_Up limit-at=0 max-burst=20
       queue=default priority=8 weight=1 allot=1514 bounded=no

  1    name=Workst parent=Public flow=Local-all limit-at=64000 max-burst=0
```

```
       queue=default priority=8 weight=1 allot=1514 bounded=yes

[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 bandwidth 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 speed of 32k for each flow direction).

First of all, you should limit the interface speed:

```
[admin@MikroTik] queue tree> add name=Up parent=Public limit-at=64000 \
\... max-burst=0 bounded=yes
[admin@MikroTik] queue tree> print
Flags: X - disabled, I - invalid, D - dynamic
  0    name="Up" parent=Public flow="" limit-at=64000 max-burst=0
       queue=default priority=8 weight=1 allot=1514 bounded=yes

[admin@MikroTik] queue tree>
```

Next, mark the traffic from the FTP server. We will mark only TCP port 20 because that port is used to send and receive FTP data.

```
[admin@MikroTik] ip firewall mangle> add src-address=192.168.0.17/32:20 \
\... 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 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=0.0.0.0/0 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 \
\... max−burst=0 bounded=no flow=Server_Up
[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=64000 max−burst=0
       queue=default priority=8 weight=1 allot=1514 bounded=yes

  1    name="Server_Up" parent=Up flow="Server_Up" limit−at=32000 max−burst=0
       queue=default priority=8 weight=1 allot=1514 bounded=no

  2    name="Local_Up" parent=Up flow="Local_Up" limit−at=0 max−burst=0
       queue=default priority=8 weight=1 allot=1514 bounded=yes

[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)

Additional Resources

Links on Class−Based Queuing (CBQ):

http://www.aciri.org/floyd/cbq.html

Links on Random Early Detection (RED):

http://www.aciri.org/floyd/papers/red/red.html

More Complete Informatin about Traffic Cotrol:

http://www.linuxdoc.org/HOWTO/Adv−Routing−HOWTO.html

Open Shortest Path First (OSPF) Routing Protocol

Document revision 18−Jan−2003
This document applies to the MikroTik RouterOS V2.6

Overview

MikroTik RouterOS implements OSPF Version 2 (RFC 2328). The OSPF protocol is on 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.

OSPF distributes routing information between routers belonging to a single autonomous system (AS). An AS is a group of routers exchanging routing information via a common routing protocol.

Contents of the Manual

The following topics are covered in this manual:

Installation

The OSPF feature is included in the **ospf** package. The package file **ospf−2.6.x.npk** can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload it to the router with ftp and reboot.

Hardware Resource Usage

There is no significant resource usage.

OSPF Description

For OSPF description and deployment guidelines please refer to list of Additional Resources. Current document discusses OSPF configuration for MikroTik RouterOS.

When deploy the OSPF, all routers should be configured in a coordinated manner. 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.

OSPF Setup

The OSPF management can be accessed under the **/routing ospf** submenu.

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**

Setting the Basic OSPF Argument Values

To view the argument settings for OSPF, use the **/routing ospf print** command, for example:

```
[admin@MikroTik] routing ospf>
OSPF is a shortest path first or link-state protocol. OSPF is an interior
gateway protocol that distributes routing information between routers in a
single autonomous system. OSPF is described in RFC1583.

     interface  OSPF interface settings
       network  OSPF networks
          area  OSPF areas
      neighbor
  virtual-link  OSPF virtual links
         print  Show OSPF settings
           get  get value of property
           set  Change OSPF settings
        export  Export OSPF settings
[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>
[admin@MikroTik] routing ospf> set redistribute−static=as−type−2 \
\... redistribute−connected=as−type−1
```

Argument description:

router−id – the Router ID. If not specified (default 0.0.0.0), OSPF uses the largest IP address configured on the interfaces as its router ID
redistribute−connected – 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 (**as−type−1**, **as−type−2**, **no**)
redistribute−static – 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 of the router (**as−type−1**, **as−type−2**, **no**)
redistribute−rip – If set, the router will redistribute the information about all routes learned by the RIP protocol (**as−type−1**, **as−type−2**, **no**)
redistribute−bgp – If set, the router will redistribute the information about all routes learned by the BGP protocol (**as−type−1**, **as−type−2**, **no**)
distribute−default – Controls how to propagate the default route to other routers:

- ♦ **never** − do not send own default route to other routers
- ♦ **if−installed** (as **type 1** or **type 2**) − send the default route only if it has been installed (a static default route, or route added by DHCP, PPP, etc.)
- ♦ **always** (as **type 1** or **type 2**)− always send the default route

metric−default − cost of the default route
metric−connected − cost of connected routes
metric−static − cost of static routes
metric−rip − cost of the routes learned by the RIP protocol
metric−bgp − cost of the routes learned by the BGP protocol

Note that 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.

Note on metrics − OSPF protocol will try to use the shortest path (path with the least total cost) if available.

Note on types − 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

Usually you want to redistribute connected and static routes, if any. Therefore change the settings for these arguments and proceed to the OSPF areas and networks.

OSPF Areas

The area management can be accessed under the /routing ospf area submenu. There is one area that is configured by default − the backbone area (area ID 0.0.0.0):

```
[admin@MikroTik] routing ospf area> print detail
Flags: X − disabled
  0 name=backbone area−id=0.0.0.0 stub−area=no default−cost=0
    authentication=none

[admin@MikroTik] routing ospf area>
```

To define additional OSPF area(s) for the router, use the **/routing ospf area add** command:

```
[admin@MikroTik] routing ospf area> add area-id=0.0.10.5 name=local_10
[admin@MikroTik] routing ospf area> print
Flags: X - disabled
  0 name=backbone area-id=0.0.0.0 stub-area=no default-cost=0
    authentication=none

  1 name=local_10 area-id=0.0.10.5 stub-area=no default-cost=0
    authentication=none

[admin@MikroTik] routing ospf area>
```

Argument description:

name − area name. Cannot be changed for the backbone area.
area−id − area ID, must be in IP address notation. Cannot be changed for the backbone area.
default−cost − Cost for the default summary route used for a stub area. Only for area boundary router.
stub − (yes / no) Sets the area type.
authentication − (md5 / none / simple) authentication method for OSPF

- **none** − no authentication
- **simple** − clear text authentication
- **md5** − Keyed Message Digest 5 (MD5) authentication

OSPF Network

To start the OSPF protocol, you have to define the networks on which OSPF runs and the area ID for those networks. Use the **/routing ospf network add** command:

```
[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>
```

Argument description:

area − Area to be associated with the address range
network − the network address/mask that is 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

Note that 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**

OSPF Interfaces

To run OSPF you don't have to configure interfaces. **/routing ospf interface** command level is only for additional configuration of OSPF specific interface parameters.

```
[admin@MikroTik] routing ospf> interface add interface=ether2
[admin@MikroTik] routing ospf> interface print
  0 interface=ether2 cost=1 priority=1 authentication-key=""
    retransmit-interval=5s transmit-delay=1s hello-interval=10s
```

```
    dead−interval=40s


[admin@MikroTik] routing ospf>
```

Argument description:

interface − interface on which rus OSPF. **all** sets the defaults, that will be used for all the interfaces not having specific settings
authentication−key − Authentication key to be used by neighboring routers that are using OSPF's simple password authentication
cost − Interface cost (1..65535) expressed as the link state metric.
dead−interval − 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.
hello−interval − 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.
priority − Router priority (0..255). It helps 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.
retransmit−interval − Time between retransmitting lost link state advertisements (3..65535 seconds). 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 − Link state transmit delay (1..65535 seconds) is the estimated time it takes to transmit a link state update packet on the interface

OSPF Virtual Links

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.

To add a virtual link use the **/routing ospf network add** command:

```
[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>
```

Argument description:

neighbor−id − **router−id** of the neighbour
transit−area − non−backbone area the two routers have in common

Note that virtual links cannot be established through stub areas

OSPF Neighbours

To see list of OSPF neighbors for router, with brief statistics, use "/routing ospf neighbor print" command.

It also shows the router itself in this list. The next is printed 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>
```

Description of the printout:

router−id − **router−id** parameter of the OSPF neighbour
address − appropriate IP address of the OSPF neighbor
priority − priority of neighbor which is used in designated router elections on this network
state − state of connection:

- **Down** − the connection is down
- **Attempt** − sending Hallo packet
- **Init** − Hallo 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 − number of state changes of the connection
ls−retransmits −number of Link State retransmits
ls−requests − number of Link State requests
db−summaries − number of records in link−state database advertised by the neighbour
dr−id − router id of designated router for this neighbor
backup−dr−id − router id of backup designated router for this neighbor

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**.

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

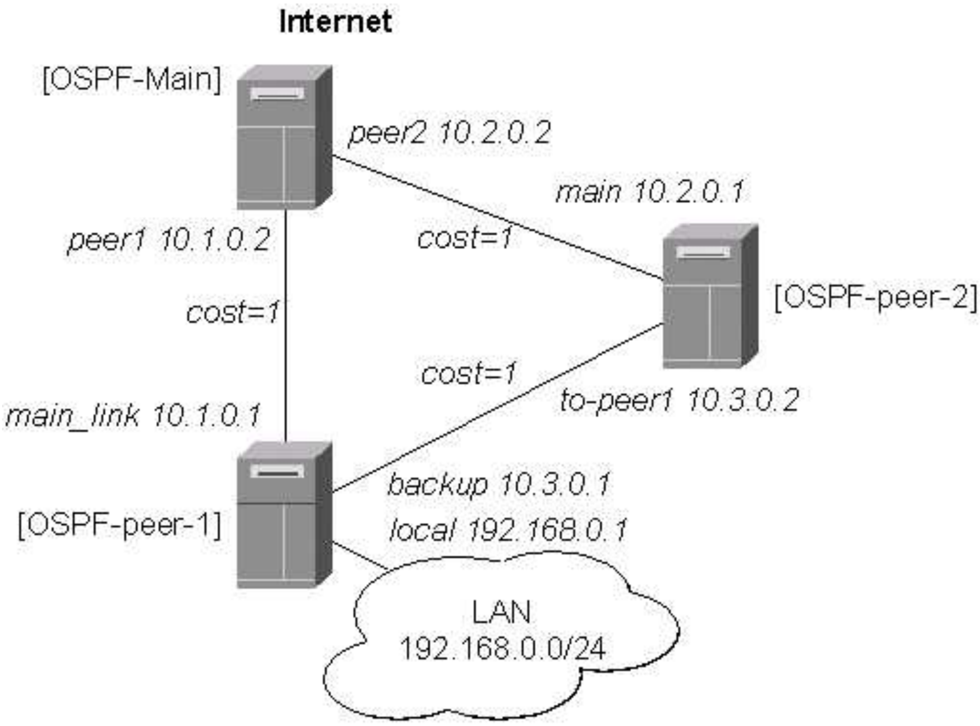
OSPF Application Examples

Let us consider the following examples of OSPF protocol used for backup links:

- OSPF Backup without using Tunnel
  The example is for the situation, when OSPF is running both on the main and the backup routers.
- OSPF Backup using Encrypted Tunnel through a Third Party
  The example is for situation, when a third party link and routers are involved for backup, and you do not have control over the involved routers.

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 links 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.

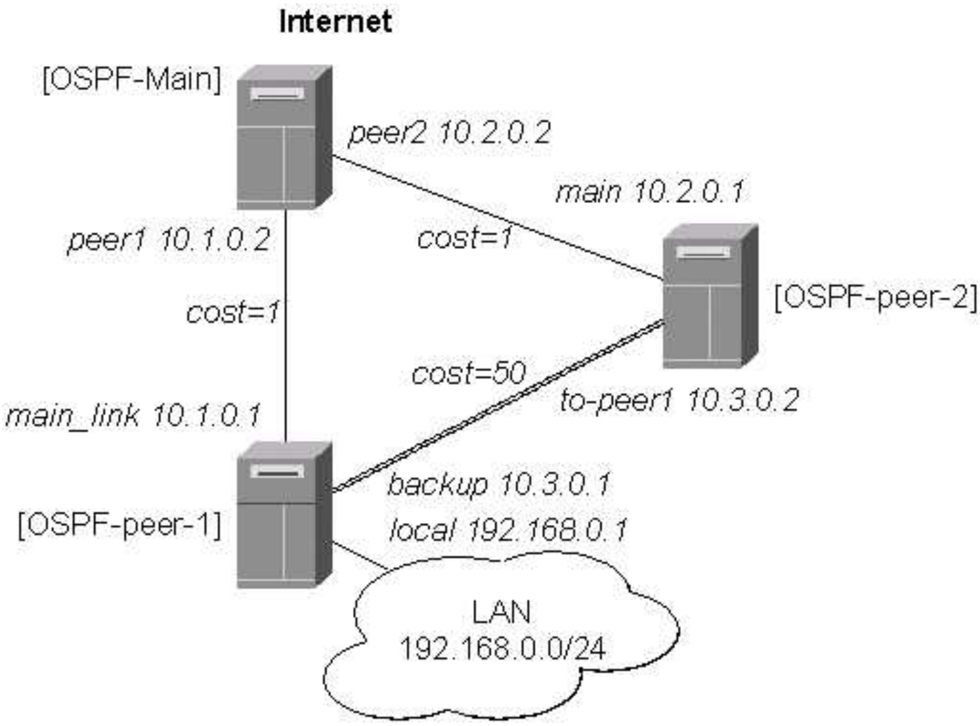
> The cost is calculated as the sum of costs over each hop to the destination. Unless this is not specially desired, we may want to avoid such situations, i.e., and adjust the cost settings for the interfaces (links) accordingly.

Routing Tables with Revised Link Cost

Let as 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] >
```

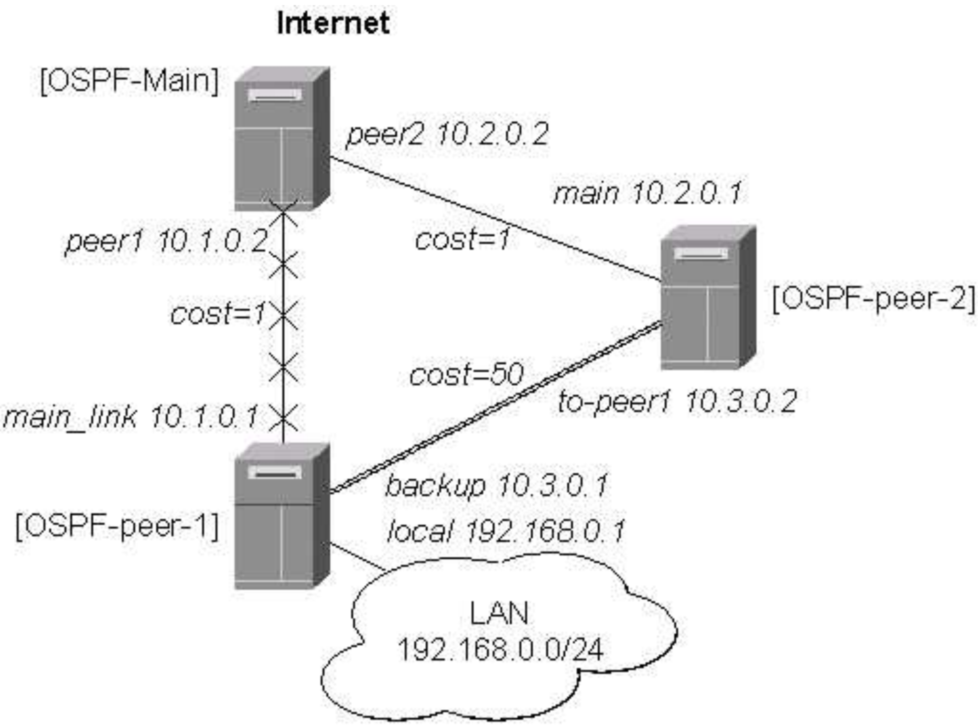
On the Peer 2:

```
[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] >
```

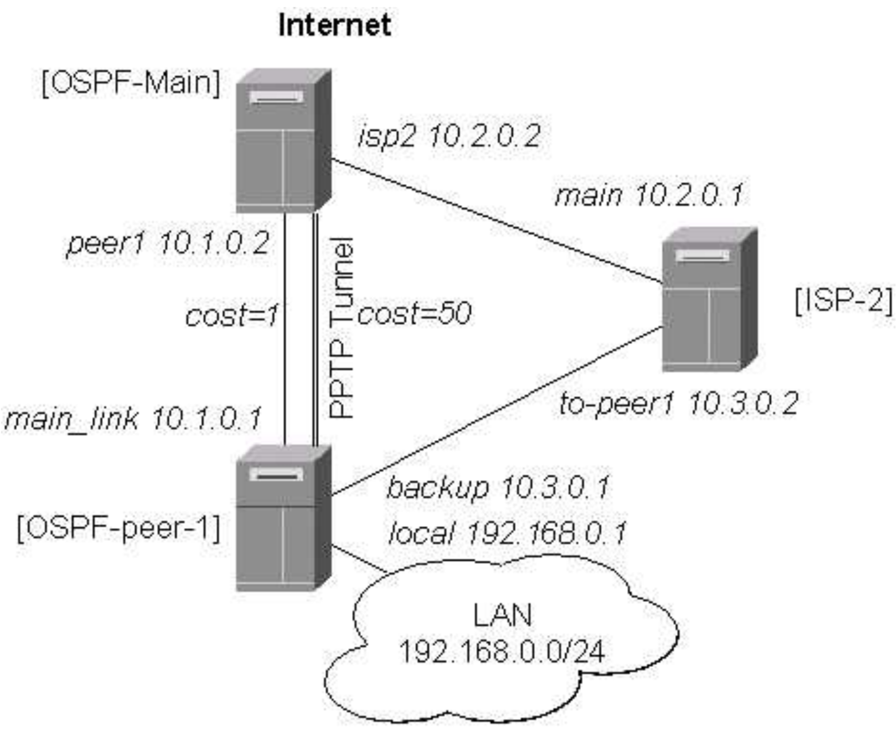
On the Peer 2:

```
[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:

```
[OSPF−Main] >
/ip route add dst−address=10.3.0.1/32 gateway=10.2.0.1
/ppp secret add name=ospf service=pptp password=asdf4 \
     local−address=10.4.0.2 remote−address=10.4.0.1
/interface pptp−server add name=pptp−in1 user=ospf
/interface pptp−server server set enabled=yes
/interface pptp−server print
Flags: X − disabled, D − dynamic, R − running
  #     NAME                 USER         MTU   CLIENT−ADDRESS  UPTIME   ENC...
  0     pptp−in1             ospf
[OSPF−Main] >
```

The IP address configuration of the [OSPF_Main] router is as follows:

```
[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
[OSPF−Main] >
```

OSPF settings:

```
[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
[OSPF−Main] routing ospf> interface add interface=pptp−in1 cost=50
[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

[OSPF−Main] routing ospf> area print
Flags: X − disabled, I − invalid
```

```
  #   NAME                                        AREA-ID         STUB DEFAULT-COST AUTHENTIC
  0   backbone                                    0.0.0.0                           none
[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
[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:

```
[OSPF-peer-1] >
/ip route add dst-address=10.2.0.2/32 gateway=10.3.0.2
/interface pptp-client add name=pptp-out1 user=ospf \
    connect-to=10.2.0.2 password=asdf4 mtu=1500 mru=1500
/interface pptp-client enable pptp-out1
/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

/interface pptp-client monitor pptp-out1
      status: "connected"
      uptime: 39m46s
    encoding: "none"

[OSPF-peer-1] >
```

The IP address configuration of the [OSPF−peer−1] router is as follows:

```
[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
[OSPF-peer-1] >
```

OSPF settings:

```
[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
[OSPF-peer-1] routing ospf> interface add interface=pptp-out1 cost=50
[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
```

```
[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

[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
[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:

```
[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
[OSPF−Main] >
==============================================================================
[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
[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:

```
[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
[OSPF-Main] >
==========================================================
[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
[OSPF-peer-1] >
```

As we see, all routing goes through the PPTP tunnel now.

Routing Prefix Lists

Document revision 21−Aug−2002
This document applies to MikroTik RouterOS V2.6

Overview

Prefix lists are used to filter routes received from or sent to other routers.

Topics covered in this manual:

- Prefix List Installation on the MikroTik RouterOS
- Prefix List Setup

Prefix List Installation on the MikroTik RouterOS

The **plist−2.6.y.npk** package is required. The package can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload one to the router with ftp and reboot. You may check to see if the package is installed with the command:

```
[admin@MikroTik] > system package print
Flags: I − invalid
  #   NAME                   VERSION              BUILD−TIME           UNINSTALL
  0   system                 2.6beta4             aug/09/2002 20:22:14 no
  1   rip                    2.6beta4             aug/09/2002 20:33:41 no
  2   ppp                    2.6beta4             aug/09/2002 20:28:01 no
  3   plist                  2.6beta4             aug/09/2002 20:32:58 no
  4   pppoe                  2.6beta4             aug/09/2002 20:29:18 no
  5   pptp                   2.6beta4             aug/09/2002 20:28:43 no
  6   ssh                    2.6beta4             aug/09/2002 20:25:31 no
  7   advanced−tools         2.6beta4             aug/09/2002 20:53:37 no
  7   bgp                    2.6beta4             aug/09/2002 20:34:22 no
  9   ipsec                  2.6beta4             aug/09/2002 20:24:51 no
 10   ospf                   2.6beta4             aug/09/2002 20:34:08 no
[admin@MikroTik] >
```

Prefix List Setup

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**. An empty prefix list permits all prefixes.

To add a prefix list, use the **/routing prefix−list add** command, for example:

```
[admin@MikroTik] routing prefix−list> add name=cybernet
[admin@MikroTik] routing prefix−list> print
  # NAME                                                            DEFAULT−ACTION
  0 cybernet                                                        accept
[admin@MikroTik] routing prefix−list>
```

Argument description:

> **name** − Name for the prefix list
> **default−action** − Default action for all members of this list (**accept**, **reject**)

The list members can be added using the **/routing prefix−list list _listname_ add** command, for example:

```
[admin@MikroTik] routing prefix-list> list cybernet
[admin@MikroTik] routing prefix-list list cybernet> add prefix=172.16.0.0 \
\... prefix-length=16
[admin@MikroTik] routing prefix-list list cybernet> print
  # PREFIX             PREFIX-LENGTH ACTION
  0 172.16.0.0/0       16            accept
[admin@MikroTik] routing prefix-list list cybernet>
```

Argument description:

prefix − network prefix, e.g., 198.168.0.0
prefix−length − length (range) of the network prefix in bits, e.g., 16−24
action − action for the list member (**accept**, **reject**)

You can add as many members to the list as required.

Note that 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.

Routing Information Protocol (RIP)

Document revision 14−Jan−2003
This document applies to MikroTik RouterOS V2.6

Overview

Routing Information Protocol (RIP) is one protocol in a series of routing protocols based on Bellman−Ford (or distance vector) algorithm. This interior routing protocol 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).

Topics covered in this manual:

RIP Installation on the MikroTik RouterOS

The **rip−2.6.y.npk** package is required. The package can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload one to the router with ftp and reboot.

RIP Routing Setup

RIP general settings are under the **/routing rip** menu:

```
[admin@MikroTik] routing rip>
RIP is interior gateway protocol based on distance vector algorithm. Route
which has the smallest number of hops (gateways) to destination is used. RIP
is described in RFC1058 and RIPv2 in RFC2453.

  interface  RIP interface settings
   neighbor
      route
    network
      print  Show RIP settings
        get  get value of property
        set  Change RIP settings
     export  Export RIP settings
[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>
```

Argument description:

- **redistribute-static** - redistribution of static routes to neighbor routers
- **redistribute-connected** - redistribution of connected routes to neighbor routers
- **redistribute-ospf** - redistribution of routes learned by OSPF to neighbor routers
- **redistribute-bgp** - redistribution of routes learned by BGP to neighbor routers
- **metric-static** - metric, the distance to the destination for static routes
- **metric-connected** - metric, the distance to the destination for connected routes
- **metric-ospf** - metric, the distance to the destination for OSPF routes
- **metric-bgp** - metric, the distance to the destination for BGP routes
- **update-timer** - time period for RIP update to start
- **timeout-timer** - time period after route is not valid more
- **garbage-timer** - time period after dropped out route is dropped from neighbor router table

Set the desired argument values to **yes** for redistributing the routing information to other routers, for example:

```
[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>
```

Note that 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.

RIP Interface Setup

To run RIP you don't have to configure interfaces. **/routing rip interface** command level is only for additional configuration of RIP specific interface parameters.

```
[admin@MikroTik] routing rip> interface add interface=ether1
[admin@MikroTik] routing rip> interface print
Flags: I - inactive
  0   interface=ether1 receive=v2 send=v2 authentication=none
      authentication-key="" prefix-list-in=none prefix-list-out=none


[admin@MikroTik] routing rip>
```

Argument description:

interface − physical network to access the first router. **all** sets the defaults, that will be used for all the interfaces not having specific settings
send − distributed RIP protocol versions. One of: **v1**, **v1−2**, **v2**
receive − RIP protocol versions the router can receive. One of: **v1**, **v1−2**, **v2**
authentication − authentication method for RIP messages:

- **none** − no authentication
- **simple** − clear text authentication
- **md5** − Keyed Message Digest 5 (MD5) authentication

authentication−key − authentication key for RIP messages
prefix−list−in − Name of the filtering prefix list for receiving routes
prefix−list−out − Name of the filtering prefix list for advertising routes

The prefix lists should be defined under the **/routing prefix−list**. See corresponding manual for the details on using prefix lists.

Security issue: it is recommended not to use RIP version 1 when it is possible.

RIP Networks

To start the RIP protocol, you have to define the networks on which RIP runs. Use the **/routing rip network add** command:

```
[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>
```

Argument description:

address − the network address/mask that is associated with the area. It allows defining one or multiple interfaces RIP to be run on. Only directly connected networks of the router may be specified
network − specifies the network mask of the **address** (if it is not specified in the **address** argument)

Note that 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**

RIP Neighbors

To define a neighboring router with which to exchange routing information, use the **/routing rip neighbour add** command, for example:

```
[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>
```

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.

RIP Routes

The routes installed by RIP and other routing protocols can be viewed using the **/routing rip route print** command:

```
[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>
```

Printout description:

dst−address − destination network address and netmask
gateway − last gateway to destination address
metric − distance vector length to the network
from − from which router this route was received

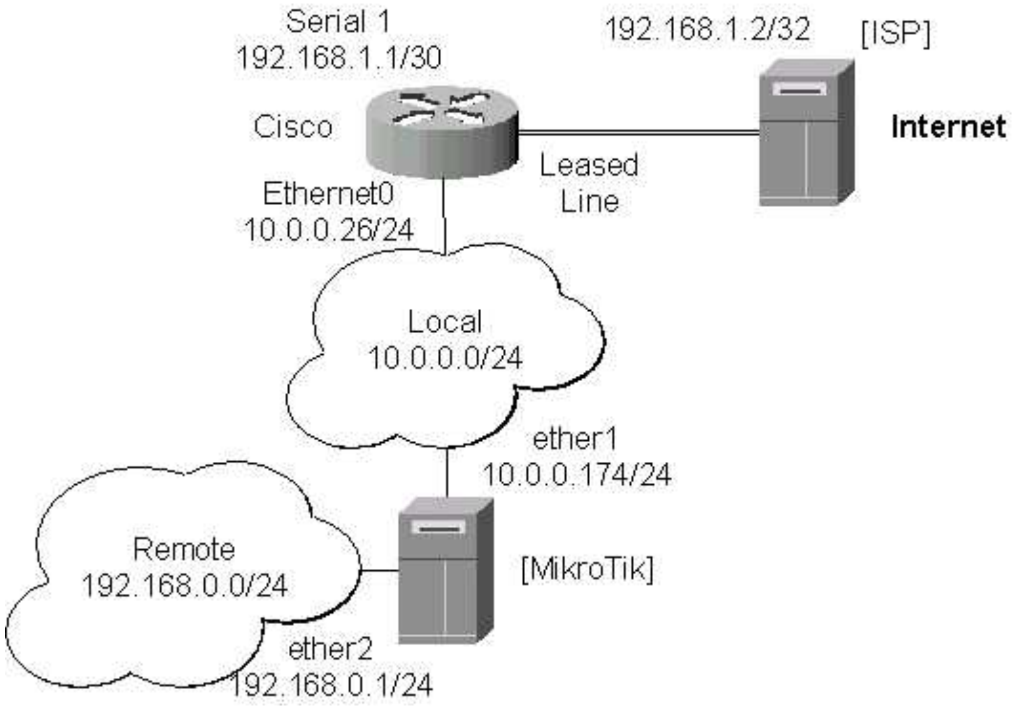
Additional Resources

Links for RIP documentation:

- http://www.ietf.org/rfc/rfc1058.txt
- http://www.ietf.org/rfc/rfc2453.txt
- Cisco Systems RIP protocol overview

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 ether1:

```
[admin@MikroTik] routing rip interface> add interface=ether1
[admin@MikroTik] routing rip interface> print
Flags: I - inactive
  0   interface=ether1 receive=v2 send=v2 authentication=none
      authentication-key="" prefix-list-in=none prefix-list-out=none


[admin@MikroTik] routing rip interface>
```

Note, that the ether2 does not need to be enabled, if no propagation of RIP information is required into the Remote network. The routes obtained by RIP can be viewed in the **/routing rip route** menu:

```
[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).

Border Gateway Protocol (BGP) Routing Protocol

Draft

Document revision 5−Sep−2002
This document applies to the MikroTik RouterOS 2.6

Overview

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.

MikroTik RouterOS supports BGP Version 4, as defined in RFC1771.

The MikroTik RouterOS implementation of the BGP has the following features:

- Filtering using prefix lists

Contents of the Manual

The following topics are covered in this manual:

Installation

The BGP feature is included in the **bgp** package. The package file **bgp−2.6.y.npk** can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload it to the router with ftp and reboot. You may check to see if the routing package is installed with the command:

```
[admin@MikroTik] > system package print
Flags: I − invalid
  #   NAME                    VERSION              BUILD−TIME           UNINSTALL
  0   system                  2.6beta4             aug/09/2002 20:22:14 no
  1   rip                     2.6beta4             aug/09/2002 20:33:41 no
  2   ppp                     2.6beta4             aug/09/2002 20:28:01 no
  3   plist                   2.6beta4             aug/09/2002 20:32:58 no
  4   pppoe                   2.6beta4             aug/09/2002 20:29:18 no
  5   pptp                    2.6beta4             aug/09/2002 20:28:43 no
  6   ssh                     2.6beta4             aug/09/2002 20:25:31 no
  7   advanced−tools          2.6beta4             aug/09/2002 20:53:37 no
  7   bgp                     2.6beta4             aug/09/2002 20:34:22 no
  9   ipsec                   2.6beta4             aug/09/2002 20:24:51 no
 10   ospf                    2.6beta4             aug/09/2002 20:34:08 no
```

```
[admin@MikroTik] >
```

Hardware Resource Usage

The BGP requires additional RAM for storing the routing information. It is recommended to have 128MB or more RAM.

BGP Description

For BGP description and implementation guidelines please refer to the readings mentioned in the list of Additional Resources. Current document discusses BGP configuration for MikroTik RouterOS.

BGP Setup

The BGP management can be accessed under the **/routing bgp** submenu.

Setting the Basic BGP Configuration

To enable the BGP and set the AS number, use the **/routing bgp set** command, for example:

```
[admin@MikroTik] routing bgp> print
                   enabled: no
                        as: 0
                 router−id: 0.0.0.0
       redistribute−static: no
    redistribute−connected: no
          redistribute−rip: no
         redistribute−ospf: no
                     state: disabled
[admin@MikroTik] routing bgp> set as=65002 router−id=159.148.147.206 enabled=yes \
\... 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>
```

Argument description:

enabled − enable or disable the BGP
as − autonomous system number
router−id – the Router ID
redistribute−connected – if set to **yes**, then 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 – if set to **yes**, then 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 of the router
redistribute−rip – if set to **yes**, then the router will redistribute the information about all routes learned by the RIP protocol

state − status of the BGP:

- **disabled** − not working, has been disabled
- **running** − working

Usually you want to redistribute connected and static routes, if any. Therefore change the settings for these arguments and proceed to the BGP networks.

BGP Network

To tell the BGP router which networks to advertise, use the **/routing bgp network add** command:

```
[admin@MikroTik] routing bgp network> add network=159.148.150.192/27
[admin@MikroTik] routing bgp network> print
  # NETWORK
  0 159.148.150.192/27
[admin@MikroTik] routing bgp network>
```

Here, the **network** argument is used to specify the network/mask to advertise. You can add to the list as many networks as required. Also, you can use 0.0.0.0/0 to advertise all networks.

Note, that the OSPF uses network list for different purpose − to determine where to send updates.

BGP Peers

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, for example:

```
[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   217       no       no            none          none
[admin@MikroTik] routing bgp> peer print status
  # REMOTE−ADDRESS  REMOTE−AS STATE          ROUTES−RECEIVED
  0 192.168.0.254   217       connected      1
[admin@MikroTik] routing bgp>
```

Argument description:

remote−address − address of the remote peer
remote−as − AS number of the remote peer
multihop − if set to **yes**, 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).
route−reflect − defines whether to further redistribute 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 of the filtering prefix list for receiving routes
prefix−list−out − Name of the filtering prefix list for advertising routes
state − Shows the status of the BGP connection to the peer. Can be **not−connected** or **connected**
routes−received − Shows the number of received routes from this peer

The prefix lists should be defined under the **/routing prefix−list**. See corresponding manual for the details on using prefix lists.

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** commend. 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

BGP Application Examples

(Not complete yet)

Export and Import

Document revision 16−Sep−2002
This document applies to MikroTik RouterOS v2.6

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.

Note that it is impossible to import the hole 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.

Topics covered in this manual:

Installation

The Export and Import features are included in the 'system' package. No installation is needed for this feature

Hardware Resource Usage

There is no significant resource usage.

Export and Import Description

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. The **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.

Export and Import Examples

```
[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 use the following command:

```
[admin@MikroTik] ip address> export file=address
[admin@MikroTik] ip address>
```

To make an export file from only one item use the following command:

```
[admin@MikroTik] ip address> export file=address1 from=1
[admin@MikroTik] ip address>
```

To see the files stored on the router use the following command:

```
[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>
```

To load the saved export file use the following command:

```
[admin@MikroTik] > import
file-name: address1.rsc
[admin@MikroTik] >
```

Backup and Restore

Document revision 19−Nov−2002
This document applies to MikroTik RouterOS v2.6

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.

Topics covered in this manual:

Installation

The Backup and Restore features are included in the 'system' package. No installation is needed for this feature

Hardware Resource Usage

There is no significant resource usage.

Backup and Restore Description

Backup and Restore feature can be found under **system backup** submenu. This function is used to store the entire router configuration in a backup file. The file is stored in the **file** folder under **[admin@MikroTik] file**>. You can download this file 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.

Backup and Restore Examples

To make a backup file use the following command:

```
[admin@MikroTik] system backup> save name=test
Configuration backup saved
[admin@MikroTik] system backup>
```

To see the files stored on the router use the following command:

```
[admin@MikroTik] > file print
  # NAME                            TYPE         SIZE       CREATION−TIME
  0 MikroTik−12082002−2107.backup   backup       12567      aug/12/2002 21:07:50
```

```
[admin@MikroTik] >
```

To load the saved backup file use the following command:

```
[admin@MikroTik] system backup> load name=test
Restore and reboot? [y/N]:
```

The restored configuration is loaded and the router is rebooted.

Liquid Crystal Display (LCD) Manual

Document revision 02−Dec−2002
This document applies to the MikroTik RouterOS V2.6

Overview

The MikroTik RouterOS supports the following LCD hardware:

- Crystalfontz (http://www.crystalfontz.com/) Intelligent Serial LCD Module 632 (16x2 characters) and 634 (20x4 characters)
- Powertip (http://www.powertip.com.tw/) Character LCD Modules

Contents of the Manual

The following topics are covered in this manual:

Installation

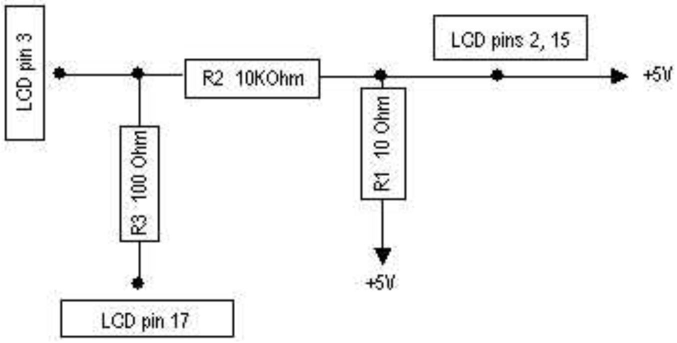
The MikroTik Router should have the LCD software package installed. The software package file **lcd−2.6.x.npk** can be downloaded from MikroTik's web page www.mikrotik.com To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer. After successful installation the package should be listed under the installed software packages list.

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	7
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 MikroTikls 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

Hardware Resource Usage

Before connecting the LCD, please check the availability of ports, their configuration, and free the desired port resource, if required. For serial LCD:

```
[admin@MikroTik] system lcd> /port print
  # NAME                                 USED−BY                            BAUD−RATE
  0 serial0                              Serial Console                     9600
  1 serial1                                                                 9600
[admin@MikroTik] system lcd>
```

Please install the LCD module hardware into the PC accordingly the instructions provided by the module manufacturer.

The basic installation steps should be as follows:

- Connect the LCD's serial connector to the COM1 or COM2 port of the router.
- Connect the LCD's power cable to the router's power supply (+5V and ground).
- Turn on the router and configure the LCD settings.

Configuring the LCD's Settings

The LCD configuration can be accessed under the menu **system lcd**

Use the **/system lcd set** command to configure the **type** and to **enable** the LCD.

For Powertip parallel port LCDs:

```
[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>
```

For Crystalfontz serial LCDs:

```
[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                          LCP Panel                       9600
[admin@MikroTik] system lcd> print
        enabled: yes
           type: crystalfontz
    serial−port: serial1
[admin@MikroTik] system lcd>
```

Note 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.

Argument description:

> **enabled** − turns the LCD on or off
> **type** − sets the type of the LCD (**powertip**, **crystalfontz**) **serial−port** − name of the port where the LCD is connected

LCD Information Display Configuration

The **system lcd page** menu is used for configuring the LCD information display. Use the **system lcd page print** command to see the configuration of the information display. Example output of the **print** command:

```
[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>
```

The output of the **print** command shows the number, time, and short description of the displayed information items. Use the **enable** command to enable the specified item, or the **disable** command to disable it.

Use the **/system lcd page set** command to set the display time for specified item.

```
[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 9−Aug−2002
This document applies to the MikroTik RouterOS v2.6

Overview

MikroTik RouterOS software has a licensing system where Software License (Software Key) is issued for each individual installation of the RouterOS. 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.

Contents of the Manual

The following topics are covered in this manual:

Managing the License

License management can be accessed under the **/system license** menu:

```
[admin@MikroTik] system license> print
          software−id: M61X−UPT
                  key: 7CJH−BD6−UXK
    upgradeable−until: apr/01/2002
[admin@MikroTik] system license> ?

      set  Set the new Software Key
  feature  Unlocked router features
    print  Show license information
      get  get value of property
[admin@MikroTik] system license>
```

Here, the **upgradeable−until** means the date until which software can be upgraded to higher versions.

To see the software features that are enabled with the current license use the following command:

```
[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.

Obtaining Additional License Features

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 should be supplied to the router and the system should be rebooted:

```
[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: dec/01/2002
[admin@MikroTik] system license>
```

Log Management

Document revision 19−Nov−2002
This document applies to MikroTik RouterOS v2.6

Overview

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.

Topics covered in this manual:

Installation

The Log Management feature is included in the 'system' package. No installation is needed for this feature.

Hardware Resource Usage

There is no significant resource usage.

Log Management 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 for logging facility can be configured in the **/system logging** menu:

```
[admin@MikroTik] system logging> print
    default−remote−address: 10.5.13.11
       default−remote−port: 514
              buffer−lines: 100
```

General logging parameters:

buffer−lines − 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.
default−remote−address − Remote log server IP address. Used when remote logging is enabled but no IP address of the remote server is specified (IP=0.0.0.0).
default−remote−port − Remote log server UDP port. Used when remote logging is enabled but no UDP port of the remote server is specified (UDP=0).

Individual settings for various logging facilities are in the **/system logging facility** menu:

```
[admin@MikroTik] system logging> facility print
  # FACILITY            LOGGING PREFIX              REMOTE−ADDRESS  REMOTE−PORT
  0 Firewall−Log        none
  1 PPP−Account         none
  2 PPP−Info            remote                      10.5.13.10      514
  3 PPP−Error           none
  4 System−Info         remote                      10.5.13.11      514
  5 System−Error        remote                      10.5.13.11      514
  6 System−Warning      local
```

Logging facility parameters:

facility − (Read−only) Name of the log group.
logging − Type of logging.
prefix − Local log prefix.
remote−address − Remote log server IP address. Used when logging type is remote. If not set, default log server IP address is used
remote−port − Remote log server UDP port. Used when logging type is remote. If not set, default log server UDP port is used.

Types 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.

Log Management Examples

Use the **/log print** command to view the local logs:

```
[admin@MikroTik] > log print
 TIME                 MESSAGE
 aug/12/2002 16:42:05 user admin logged in via console
 aug/12/2002 16:42:32 user admin logged in from 10.0.0.250 via ftp
 aug/12/2002 16:42:57 user admin logged out from 10.0.0.250 via ftp
 aug/12/2002 16:50:49 user admin logged in from 10.0.0.250 via telnet
 aug/12/2002 19:20:53 user admin logged in via web
 aug/12/2002 19:23:10 route changed by admin
 aug/12/2002 19:23:22 route changed by admin
 aug/12/2002 19:26:11 route changed
 aug/12/2002 19:26:28 route changed
 aug/12/2002 19:37:13 added prefix−list by admin
 aug/12/2002 19:38:48 pool a added
 aug/12/2002 19:39:00 pool a removed
 aug/12/2002 19:39:11 pool a added
−− more
```

To view complete (not truncated) log lines, use the **/log print detail** command:

```
[admin@MikroTik] > log print detail

 time=aug/12/2002 16:42:32
    message="user admin logged in from 10.0.0.250 via ftp"
```

```
 time=aug/12/2002 16:42:57
    message="user admin logged out from 10.0.0.250 via ftp"

...
```

Network Time Protocol (NTP)

Document revision 19−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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.htm

Contents of the Manual

The following topics are covered in this manual:

NTP Installation on the MikroTik RouterOS

The **ntp−2.6.x.npk** package ir required. The package can be downloaded from MikroTik's web page www.mikrotik.com . To install the package, please upload it to the router via ftp and reboot. You may check to see if the packages are installed with the **/system package print** command.

NTP Client

The NTP Client setup is under **/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] >
```

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 sent (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.

Status of NTP client can be monitored by looking at status parameter. There are several possible statuses:

- **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)

NTP Server

The NTP Server setup is under **/system ntp server**

```
[admin@MikroTik] > system ntp server print
      enabled: no
    broadcast: no
    multicast: no
     manycast: yes
[admin@MikroTik] >
```

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.

If **broadcast** is enabled, NTP broadcast message is sent to 255.255.255.255 every 64s.

If **multicast** is enabled, NTP multicast message is sent to 224.0.1.1 every 64s.

If **manycast** is enabled, NTP server listens for multicast messages sent to 239.192.1.1 and responds to them.

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.

TIMEZONE

NTP changes local clock to UTC (GMT) time by default. To specify different time zone, time−zone parameter under **/system clock** has to be changed.

```
[admin@MikroTik] > system clock print
```

```
         time: aug/12/2002 18:31:20
    time−zone: +00:00
[admin@MikroTik] >
```

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=3
[admin@MikroTik] > system clock print
         time: aug/12/2002 21:31:57
    time−zone: +03:00
[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] >
```

Serial Console

Document revision 12−Aug−2002
This document applies to the MikroTik RouterOS v2.6

Overview

The Serial Console 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. Alternatively, another MikroTik router can be used as terminal, if its communication port is configured as serial terminal. See the relevant manual for details.

Contents of the Manual

The following topics are covered in this manual:

Installation

The Serial Console feature is included in the 'system' package. No installation is needed for this feature.

Hardware Resource Usage

There is no significant resource usage.

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:

```
1 --- 1
2 --- 3
3 --- 2
4 --- 4
5 --- 5
6 --- 6
7 --- 8
8 --- 7
9 n/c 9
```

After installation of the MikroTik RouterOS the serial console is configured to use port serial0 (COM1 on the motherboard), if available. To check the Serial Console settings use:

```
[admin@MikroTik] system serial−console> print
    enabled: no
       port: serial0
[admin@MikroTik] system serial−console>
```

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 change port:

```
[admin@MikroTik] system serial-console> set port=serial1
[admin@MikroTik] system serial-console> print
    enabled: yes
       port: serial1
[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="" baud-rate=9600 data-bits=8 parity=none stop-bits=1
    flow-control=none

  1 name=serial1 used-by=Serial Console baud-rate=9600 data-bits=8 parity=none
    stop-bits=1 flow-control=none

[admin@MikroTik] system serial-console>
```

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**).

Serial Terminal

Document revision 16−Sep−2002
This document applies to the MikroTik RouterOS v2.6

Overview

The **/system serial−terminal** command is used to communicate with devices and other systems that are connected to router via serial port. The serial terminal may be used to monitor and configure many devices – including modems, network devices, and any device that can be connected to a serial−terminal.

Contents of the Manual

The following topics are covered in this manual:

Installation

The Serial Terminal feature is included in the 'system' package. No installation is needed for this feature.

Hardware Resource Usage

There is no significant resource usage.

Serial Terminal Description

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. It is not possible to send "Ctrl−Q" key to serial port as it is intercepted and the serial−terminal is closed. 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.

Serial Terminal Usage

The serial−terminal is invoked with one argument − the name of serial port:

```
[admin@MikroTik] system> serial−terminal port=serial0
[Type Ctrl−Q to return to console]
```

Serial Terminal Examples

Several customers have described situations where the serial−terminal feature would be useful. One situation is described as a mountaintop where a MikroTik wireless installation sits next to equipment that also includes switches and Cisco routers that can not be managed in−band (by telnet through an IP network). Another situation describes a need to monitor weather−reporting equipment through a serial−console. Another situation described a 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, one to thirty−four devices can be monitored and controlled (using serial expansion cards from more than two devices).

The serial−console was tested and found working with:

- PLANET FNSW−1600S Ethernet Smart Switch
- Cisco 1005
- US Robotics Courier V.Everything Modem
- MikroTik RouterOS

Support Output File

Document revision 12−Aug−2002
This document applies to MikroTik RouterOS v2.6

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.

Topics covered in this manual:

Installation

The Support file feature is included in the 'system' package. No installation is needed for this feature

Hardware Resource Usage

There is no significant resource usage.

Support File Description

Support file feature can be found under **/system** submenu. The file is stored in the **file** folder under **[admin@MikroTik] file**>. You can download this file through ftp to send it to the MikroTik Support.

Example of Making Support Output File

To make a Support Output File use the following command:

```
[admin@MikroTik] > system sup−output
creating supout.rif file, might take a while
Accomplished!
[admin@MikroTik] >
```

To see the files stored on the router use the following command:

```
[admin@MikroTik] > file print
  # NAME                           TYPE         SIZE       CREATION−TIME
  0 supout.rif                     unknown      38662      aug/12/2002 21:51:04
[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 19−Nov−2002
This document applies to the MikroTik RouterOS v2.6

Overview

MikroTik RouterOS offers several features for monitoring and managing the system resources. Most of the system resource management tools are grouped under the **/system** menu. The user management, logging feature and some other system features are described in separate manuals.

Contents of the Manual

The following topics are covered in this manual:

System Resource Monitor

System Resource Monitor can be accessed under the **/system resource** menu:

```
[admin@MikroTik] system resource>
System resources
  monitor  Monitor CPU and memory usage
      irq  Interrupt Request usage information
       io  Input/Output ports usage information
    print  Print basic system resources information
      get  get value of property
[admin@MikroTik] system resource>
```

Basic System Resources

Use the **print** command 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>
```

The argument values are self−explanatory.

System Resource Monitoring

The current system CPU usage and free memory can be viewed using the **monitor** command:

```
[admin@MikroTik] system resource> monitor
       cpu-used: 3
    free-memory: 1112

[admin@MikroTik] system resource>
```

The values for cpu usage and free memory are in percentage and megabytes, respectively.

IRQ and IO Usage Monitor

The IRQ and IO addresses can be viewed using the **/irq print** and **io 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>
```

Reboot and Shutdown

The system reboot is required when upgrading or installing new software packages. The packages are installed during the system shutdown. Use the **/system reboot** command to reboot the router:

```
[admin@MikroTik] system> reboot
Reboot, yes? [y/N]: y
system will reboot shortly
```

Only users which are members of groups with reboot privileges can reboot the router or shutdown. The reboot process sends termination signal to all running processes, unmounts the file systems, and reboots the router.

Before turning the power off for the router, the system should be brought to halt using the **/system shutdown** command:

```
[admin@MikroTik] system> shutdown
Shutdown, yes? [y/N]: y
system will shutdown promptly
```

For most systems, it is necessary to wait approximately 30 seconds for a safe power down.

Configuration Reset

The **reset** command clears all configuration of the router and sets it to the default including the login name and password ('admin' and no password):

```
[admin@MikroTik] system> reset
Dangerous! Reset anyway? [y/N]:
```

The router is rebooted after the reset command.

Router Identity

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. The router identity can be set using the **/system identity set** command:

```
[admin@MikroTik] system identity> print
    name: "MikroTik"
[admin@MikroTik] system identity> set name=Our_GW
[admin@Our_GW] system identity>
```

Date and Time Settings

The system Date and Time settings are managed under the **/system clock** menu:

```
[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 use the **set** command:

```
[admin@MikroTik] system clock> set
```

```
Set new system date or time
       date  New system date [month/DD/YYYY]
       time  New system time [HH:MM:SS]
  time−zone  Local time zone
[admin@MikroTik] system clock> set
[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>
```

Date and time settings become permanent and effect BIOS settings.

Configuration Change History

The history of system configuration changes is held until the next router shutdown. The invoked commands can be 'undone' using the **/undo** command. By invoking the command several times, the configuration changes can be 'undone' in reverse order they have been invoked. Use the **/system history print** command to see the list of performed actions:

```
[admin@MikroTik] system history> print
Flags: U − undoable, R − redoable
   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 list is printed with the newest actions at the top.

```
[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>
```

Tip: If you accidentally removed some item, or set wrong argument value, just execute the **/undo** command to undo previously done action. The **/redo** would do the opposite − redo the previous undo action.

System Scheduler Manual

Document revision 19−Nov−2002
This document applies to the MikroTik RouterOS V2.6

Overview

The scheduler is used to execute scripts at certain times. It has an ordered list of tasks

For details on scripting, consult respective manual

Contents of the Manual

The following topics are covered in this manual:

Installation

The System Scheduler features are included in the 'system' package. No installation is needed for this feature

Hardware Resource Usage

Hardware resource usage depends on the script that is run using the System Scheduler feature.

Using System Scheduler

To add a task, use the **add** command. For example, we 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>
```

Argument description:

name − name of the task
start−time and start−date − time and date of first execution
interval − 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 − to monitor script usage, this counter is incremented each time the script is executed, it can be reset to zero. **Note** that rebooting the router will reset this counter
script − name of the script. The script must be present at **/system script**.

System Scheduler Examples

Here are two scripts 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 setup 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>
```

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 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>
```

Telnet Client

Document revision 12−Aug−2002
This document applies to the MikroTik RouterOS v2.6

Overview

MikroTik RouterOS has a build−in Telnet Client. It is used to communicate with other systems over a network.

Contents of the Manual

The following topics are covered in this manual:

Installation

The Telnet client feature is included in the 'system' package. No installation is needed for this feature.

Hardware Resource Usage

There is no significant resource usage.

Telnet Client Description

```
[admin@MikroTik] system> telnet ?
Run telnet session to remote host.

  <host>  IP address of host
[admin@MikroTik] system> telnet
```

Telnet Client Examples

A simple example of using Telnet:

```
[admin@MikroTik] > system telnet 10.0.0.100
Trying 10.0.0.100...
Connected to 10.0.0.100.
Escape character is '^]'.

MikroTik v2.5.12
Login:
```

Telnet using Telnet command mode:

```
[Mikrotik] > system telnet
telnet> open 10.0.0.100
Trying 10.0.0.100...
```

```
Connected to 10.0.0.100.
Escape character is '^]'.

MikroTik v2.5.12
Login:
```

UPS Monitor

Document revision 1.0 (20−Jan−2003)
This document applies to the MikroTik RouterOS v2.6

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*

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

When enabled, additional properties appear (that cannot be changed):

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

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 (only shown when the unit is on−battery) − the reason for the most recent transfer to on−battery operation:

- 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

Users and Groups

Document revision 19−Nov−2002
This document applies to the MikroTik RouterOS v2.6

Overview

MikroTik RouterOS has a local user database. Permissions and user rights are granted to groups. Users belong to groups and receive all the permissions and user rights assigned to that group.

Contents of the Manual

The following topics are covered in this manual:

User Management

User management can be accessed under the **/user** menu:

```
[admin@MikroTik] user> print
Flags: X − disabled
  0   ;;; system default user
      name="admin" group=full address=0.0.0.0/0


[admin@MikroTik] user>
```

Use the **add** command to add a user to the user database:

```
[admin@MikroTik] user> add
creates new item with specified property values.
    address  Network address part of addresses user is allowed to use
    comment  short description of the item
  copy−from  item number
   disabled
      group  Permissions group for user
       name  New user name
    netmask  Netmask part of addresses user is allowed to use
   password  User password
[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>
```

Argument description:

> **name** − User name. Must start with an alphanumeric character and may contain alphanumeric characters, "*", "_", ".", "@".

group − Name of the group the user belongs to. The system default groups are **full**, **write**, **read**. See below on how to manage user groups.
password − 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 form which the user is allowed to log in.
netmask − Network mask of addresses assigned to the user

List of active users can be viewed using the **/user active print** command:

```
[admin@MikroTik] user> active print
  0 when=aug/09/2002 21:46:13 name="admin" address=0.0.0.0 via=console

  1 when=aug/09/2002 15:54:36 name="admin" address=0.0.0.0 via=web

  2 when=aug/09/2002 14:23:44 name="admin" address=10.0.0.250 via=telnet

[admin@MikroTik] user>
```

When the user has logged on 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.

User Groups

User group management can be accessed under the **/user group** menu:

```
[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>
```

There are three system groups which cannot be deleted. Use **add** command to add a user group:

```
[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>
```

Here, the argument **name** is the name of the group, and **policy** contains the list of policies assigned to the group:

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 http

Note: if there is exclamation sign (**!**)right before policy name, it means **not**.

Bandwidth Test

Document revision 19−Nov−2002
This document applies to MikroTik RouterOS v2.6

Overview

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’.

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.

Topics covered in this manual:

Installation

The Bandwidth Test feature is included in the 'system' package. No installation is needed for this feature

Hardware Resource Usage

!Caution! Bandwidth Test uses all available bandwidth (by default) and may impact network usability.

There is no other significant resource usage.

Bandwidth Test Description

Bandwidth Test Server Configuration

```
[admin@MikroTik] tool> bandwidth−server
Configure network bandwidth tester service. Use authentication for disabling
unwanted bandwidth wasting. Note that remote router must be MikroTik router in
order to run the test.
```

```
  session
    print
      get  get value of property
      set
   export
[admin@MikroTik] tool> bandwidth−server print
                   enabled: yes
              authenticate: no
    allocate−udp−ports−from: 2000
              max−sessions: 10
[admin@MikroTik] tool>
```

Setting description:

enable − enable client connections for bandwidth test
authenticate − communicate only with authenticated (by valid username and password) clients
allocate−udp−ports−from − allocate UDP ports from
max−sessions − maximal number of bandwidth−test clients

The list of current connections can be get in **session** submenu:

```
[admin@MikroTik] tool> bandwidth−server session

   print  print values of item properties
  remove  remove item
[admin@MikroTik] tool> bandwidth−server session print
  # FROM             PROTOCOL DIRECTION USER
  0 10.0.0.202       tcp      send
[admin@MikroTik] tool>
```

Bandwidth Test Client Configuration

Bandwidth Test uses TCP or UDP protocol for test. The test tries to use maximum or partial amount of bandwidth to test link speed. Be aware that default test uses all available bandwidth and may impact network usability.

```
[admin@MikroTik] tool> bandwidth−test
Run TCP or UDP bandwidth test. Tries to use maximum or partial amount of
bandwidth to test link speed. Note that remote router must be MikroTik router
in order to run the test. Be aware that default test uses all available
bandwidth and may impact network usability.

         <address>
  assume−lost−time
         direction  Direction of data flow
                do
          duration
          interval
    local−tx−speed
              once  print statistics once and quit
          password  Password for remote user
          protocol  Protocol to use for test
   remote−tx−speed
              size  UDP packet size or TCP segment size
              user
[admin@MikroTik] tool> bandwidth−test
```

Descriptions of arguments:

address − IP address of destination host
assume−lost−time − If Bandwidth Server is not responding for that time, assume that connection is lost
direction − specify the direction of the test (**receive**, **transmit**, **both**, default is **transmit**)
do − Script source
duration − Duration of the test
interval − Delay between messages (in seconds). Default is 1 second. Can be **20ms**...**5s**
local−tx−speed − Transfer test maximum speed (given in bits per second)
password − Password for remote user
protocol − Type of protocol to use (**UDP** or **TCP**, default **TCP**)
remote−tx−speed − Receive test maximum speed (given in bits per second)
size − Packet size in bytes (**50**..**1500**, default **512**). Works only with UDP protocol
user − Remote user

Bandwidth Test Example

```
[admin@MikroTik] tool> bandwidth−test 10.0.0.202 user=admin direction=both protocol=udp \
\... size=1500 duration=14s
                  status: done testing
              tx−current: 11.49Mbps
    tx−10−second−average: 10.05Mbps
        tx−total−average: 7.96Mbps
              rx−current: 12.55Mbps
    rx−10−second−average: 10.33Mbps
        rx−total−average: 8.14Mbps

[admin@MikroTik] tool>
```

Dynamic DNS (DDNS) Update Tool

Document revision 20−Aug−2002
This document applies to the MikroTik RouterOS V2.6

Overview

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. Dynamic DNS Update protocol is described in RFC2136, RFC3007 and related documents

Contents of the Manual

The following topics are covered in this manual:

Installation

The Dynamic DNS Update feature is included in the **ddns** package. The package file **ddns−2.6.x.npk** can be downloaded from MikroTik's web page www.mikrotik.com. To install the package, please upload it to the router with ftp and reboot.

Hardware Resource Usage

The feature uses a minimum of resources.

Dynamic DNS Update Description

Dynamic DNS Update is a tool that should be manually run to update dynamic DNS server

Note that you have to have dns server that supports dns updates and that it is properly configured

Dynamic DNS Update tool can be accessed with the **/tool dns−update** command:

```
[admin@MikroTik] tool> dns−update

     address
  dns−server
         key
    key−name
        name
         ttl
        zone
[admin@MikroTik] tool> dns−update
```

Descriptions of arguments:

address − defines IP address associated with the domain name
dns−server − DNS server to send update to
key − authorization key (password of a kind) to access the server
key−name − authorization key name (username of a kind) to access the server
name − name to attach with the IP address
ttl − time to live for the item (in seconds)
zone − DNS zone where to update the domain name in

Dynamic DNS Update Example

```
[admin@MikroTik] tool> dns−update address=12.23.34.45 dns−server=23.34.45.56 \
\... name=mydomain zone=myzone.com ttl=3600 key−name=dns−update−key key=sviests
```

Additional Resources

Links to Dynamic DNS Update documentation:

http://www.zoneedit.com/doc/rfc/
http://www.faqs.org/rfcs/rfc2136.html

ICMP Bandwidth Test

Document revision 19−Nov−2002
This document applies to MikroTik RouterOS v2.6

Overview

The ICMP Bandwidth Tester (Ping Speed) can be used to approximately evaluate (algorithm is not very precise) the throughput to **any** remote computer and thereby help to discover network ‘bottlenecks’.

The ICMP test uses two standard echo−requests per second. Time between these pings can be changed. As ping packet size can be varied, it is possible to evaluate connection parameters and speed approximately with different packet sizes. Statistics for throughput are calculated using the entire size of the ICMP packets, interval between ICMP echo−request and echo−reply and differences between parameters of the first packets and the second.

Topics covered in this manual:

Installation

The ICMP Bandwidth Test feature is included in the **advanced−tools** package. The software package file **advanced−tools−2.6.x.npk** can be downloaded from MikroTik’s web page www.MikroTik.com. To install the package, please upload the correct version file to the router and reboot. Use BINARY mode ftp transfer.

Hardware Resource Usage

There is no other significant resource usage.

ICMP Bandwidth Test Description

```
[admin@MikroTik] tool> ping−speed

           <address>
                  do
     first−ping−size
            interval
                once  print statistics once and quit
    second−ping−size
  time−between−pings
[admin@MikroTik] tool> ping−speed
```

Setting description:

do − scription feature
first−ping−size − Size of the first ICMP packet (default value=**32**)
second−ping−size − Size of the second ICMP packet (default value=**1500**)
time between pings − time between these two ICMP echo−requests in seconds. New ICMP−packet pair will never be sent before previous pair is completely sent and the algorithm will never send more than two requests in one second

Bandwidth Test Example

Correct showings:

```
[admin@MikroTik] tool> ping−speed 10.0.0.202 first−ping−size=750 second−ping−size=760
    current: 4.32Mbps
    average: 5.32Mbps

[admin@MikroTik] tool>
```

Incorrect showings:

```
[admin@MikroTik] tool> ping−speed 10.0.0.202 first−ping−size=1000
    current: 2666.66Mbps
    average: 764.46Mbps

[admin@MikroTik] tool>
```

Note that you should know approximate connection speed to a remote host and do not pay attention to overt erroneous showings and change **ping size** values until you get what you want to get. Besides, you should look only on **average** value as it is more informative.

Ping

Document revision 19−Nov−2002
This document applies to MikroTik RouterOS v2.6

Overview

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.

Topics covered in this manual:

Installation

The Ping feature is included in the 'system' package. No installation is needed for this feature

Hardware Resource Usage

There is no significant resource usage.

Ping Description

Ping utility shows Time To Live value of the received packet (ttl) and Roundtrip time (time) in ms. The console Ping session may be stopped when the Ctrl + C is pressed.

```
[admin@MikroTik] > ping
Send ICMP Echo packets. Repeat after given time interval.

        <address>
            count  Number of packets
  do−not−fragment
         interval  Delay between messages
             size  Packet size
              ttl
[admin@MikroTik] > ping
```

Descriptions of arguments:

address − IP address for the host you want to ping
size − Size of the IP packet (in bytes, including the IP and ICMP headers). Can be **36** ...**4096**
do−not−fragment − if added, packets will not be fragmented
interval − Delay between messages (in seconds). Can be **10ms**...**5s**. Default is 1 second
count − How many time ICMP packets will be sent. If not specified, ping continues till CTRL+C is pressed
ttl − Time To Live (TTL) value of ICMP packet. Can be **1**...**255**

Ping 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] >
```

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**.

```
[admin@MikroTik] > ping www.lv count=5 interval=100ms size=64
159.148.95.5 64 byte pong: ttl=247 time=71 ms
159.148.95.5 64 byte pong: ttl=247 time=48 ms
159.148.95.5 64 byte pong: ttl=247 time=33 ms
159.148.95.5 64 byte pong: ttl=247 time=33 ms
159.148.95.5 64 byte pong: ttl=247 time=73 ms
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 33/51.6/73 ms
[admin@MikroTik] >
```

Traceroute

Document revision 19−Nov−2002
This document applies to MikroTik RouterOS v2.6

Overview

Traceroute is a TCP/IP protocol−based utility, which allows the 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.

Topics covered in this manual:

- Installation
- Hardware Resource Usage
- Traceroute Description
- Traceroute Example

Installation

The Traceroute feature is included in the 'system' package. No installation is needed for this feature

Hardware Resource Usage

There is no significant resource usage.

Traceroute Description

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 session may be stopped when the Ctrl + C is pressed.

```
[admin@MikroTik] tool> traceroute
Trace route to host by increasing Time To Live value in sent packets and
waiting for "TTL expired" messages from routers.

  <address>
       port  UDP port number
   protocol  Protocol of sent packets
       size  Packet size
    timeout  Response wait timeout
        tos  Type of service
    use−dns
[admin@MikroTik] tool> traceroute
```

Descriptions of arguments:

address − IP address of the host you are tracing route to
port − UDP Port number. Values are in range **0−65535**
protocol − Type of protocol to use (**UDP** or **ICMP**). If one fails (for example, it is blocked

by a firewall) try the other
size − Packet size in bytes (**28**..**1428**, default **64**)
timeout − Response waiting timeout, i.e. delay between messages. Can be **1s**..**5s**, default **1s**
tos − Type Of Service – parameter of IP packet. Can be **0**..**255** , default **0**
use−dns − specifies whether to use DNS server, which can be set in **/ip dns** menu (**yes**, **no**, default is **no**)"

Traceroute Example

```
[admin@MikroTik] tool> traceroute 216.239.39.101 size=64 timeout=4s tos=0 protocol=icmp
     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>
```

Traffic Monitor

Document revision 6−Sep−2002
This document applies to MikroTik RouterOS v2.6

Overview

The traffic monitor tool is used to execute console scripts on when interface traffic crosses some given thresholds.

For details on scripting, consult respective manual

Contents of the Manual

The following topics are covered in this manual:

Installation

Traffic monitor feature is included in the 'system' package. No installation is needed for this feature

Hardware Resource Usage

There is no other significant resource usage.

Traffic Monitor Description

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.

Events (monitor items) are managed under **tool traffic−monitor** submenu:

```
[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
```

Argument description for traffic monitoring tool:

name − Name of traffic monitor item
interface − Interface to monitor
threshold − Traffic threshold, in bits per second
trigger − Condition on which to execute script (**above**, **always**, **below**)

traffic − Type of traffic to monitor (**transmitted**, **received**)
on−event − Script source. Must be present under **/system script**

You should specify the **interface** on which to monitor the traffic, the type of **traffic** to monitor (**transmitted** or **received**), the **threshold** (bits per second). The script is started, when traffic exceeds the threshold in direction given by the **trigger** argument. **above** means that script will be run each time traffic exceeds the threshold, i.e. goes from being less than threshold to being more than threshold value. **below** triggers script in the opposite condition, when traffic drops under the threshold. **always** triggers script on both **above** and **below** conditions.

Traffic Monitor Examples

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>
```

SNMP Service

Document revision 01−Oct−2002
This document applies to the MikroTik RouterOS V2.6

Overview

SNMP is a network protocol that allows managing many network devices from one location. MikroTik RouterOS supports SNMPv2 (Simple Network Management Protocol version 2) as defined by RFC 1592. Installation of the SNMP package makes the router an SNMP agent.

The MikroTik RouterOS supports:

- SNMPv2 only;
- Read−only access is provided to the NMS (network management system);
- User defined communities are supported;
- No Trap support.

Contents of the Manual

The following topics are covered in this manual:

Installation

The 'snmp−2.6.x.npk' (less than 150KB) package for installation of SNMP is required. The package can be downloaded from MikroTik’s web page www.mikrotik.com. To install the package, please upload it to the router with ftp and reboot. See if you have the required software package installed using the **/system package print** command

Hardware Resource Usage

When the SNMP is enabled, it uses approximately 2MB of RAM. When using SNMP, memory usage estimates should be made, system resources should be monitored, and RAM should be increased accordingly.

SNMP Setup

SNMP management can be accessed under the **/snmp** menu. Use the **set** command to configure it and enable the service:

```
[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>
```

Description of arguments:

contact, location − Informative only settings for the NMS.
enabled − SNMP service is disabled by default.

SNMP Communities

Community management can be accessed under the **/snmp community** menu. The default community for the SNMP is "public":

```
[admin@MikroTik] snmp> community
[admin@MikroTik] snmp community> print
  # NAME                                                                  READ−ACCESS
  0 public                                                                yes
[admin@MikroTik] snmp community>
```

Argument description:

name − Community name.
read−access − Enables or disables the read access for the community.

You can add new communities and change the read access type, for example:

```
[admin@MikroTik] snmp community> set public read−access=no
[admin@MikroTik] snmp community> add name=private
[admin@MikroTik] snmp community> print
  # NAME                                                                  READ−ACCESS
  0 public                                                                no
  1 private                                                               no
[admin@MikroTik] snmp community>
```

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 2.6.x 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/rfc1592.txt

http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm