

Routes, Equal Cost Multipath Routing, Policy Routing

Document revision 1.6 (Mon Mar 22 09:10:18 GMT 2004)

This document applies to V2.8

Table of Contents

[Table of Contents](#)

[Summary](#)

[Specifications](#)

[Related Documents](#)

[Description](#)

[Additional Documents](#)

[Static Routes](#)

[Property Description](#)

[Notes](#)

[Example](#)

[Routing Tables](#)

[Description](#)

[Property Description](#)

[Example](#)

[Policy Rules](#)

[Property Description](#)

[Notes](#)

[Example](#)

[Application Examples](#)

[Standard Policy-Routing Setup](#)

General Information

Summary

The following manual surveys the IP routes management, equal-cost multi-path (ECMP) routing technique, and policy-based routing, which gives the opportunity to select routes in order to restrict the use of network resources to certain classes of customers.

Specifications

Packages required: *system*

License required: *level1*

Home menu level: */ip route, /ip policy-routing*

Standards and Technologies: [IP \(RFC 791\)](#)

Hardware usage: *Not significant*

Related Documents

- [Package Management](#)
- [IP Addresses and ARP](#)
- [Firewall Filters](#)
- [Network Address Translation](#)

Description

MikroTik RouterOS has following types of routes:

- **Connected Routes** are created automatically when adding address to an interface. These routes specify networks, which can be accessed directly through the interface.
- **Static routes** are user-defined routes that specify the router that can forward traffic to the specified network. They are useful for specifying the default gateway.

You do not need to add routes to networks directly connected to the router, since they are added automatically when adding the IP addresses. However, unless you use some routing protocol (RIP or OSPF), you may want to specify static routes to specific networks, or the default route.

More than one gateway for one destination network may be used. This approach is called 'Equal-Cost Multi-Path Routing' and is used for load balancing (**Note** that this does not provide failover). With ECMP, a router potentially has several available next hops towards any given destination. A new gateway is chosen for each new source/destination IP pair. This means that, for example, one FTP connection will use only one link, but new connection to a different server will use other link. This also means that routes to often-used sites will always be over the same provider. But on big backbones this should distribute traffic fine. Also this has another good feature - single connection packets do not get reordered and therefore do not kill TCP performance.

Equal cost multipath routes can be created by routing protocols (RIP or OSPF), or adding a static route with multiple gateways (in the form **gateway=x.x.x.x,y.y.y.y**) The routing protocols may create routes with equal cost automatically, if the cost of the interfaces is adjusted properly. For more information on using the routing protocols, please read the corresponding section of the Manual.

Note! In routing process, the router decides which route it will use to send out the packet. Afterwards, when the packet is masqueraded, its source address is taken from the **preferred-source** field.

Additional Documents

- [RFC 2328](#)
- [RFC 2992](#)
- [RFC 1102](#)

Static Routes

Home menu level: */ip route*

Property Description

dst-address (*IP address/mask*; default: **0.0.0.0/0**) - destination address and network mask, where netmask is number of bits which indicate network number

netmask (*IP address*) - network mask

gateway (*IP address*) - gateway host, that can be reached directly through some of the interfaces. You can specify multiple gateways separated by comma ",", for ECMP routes. See more information on that below

preferred-source (*IP address*; default: **0.0.0.0**) - source address of packets, leaving the router via this route. Must be a valid address of the router, which is assigned to the router's interface, through which the packet leaves

- **0.0.0.0** - determined at the time of sending the packet out through the interface

distance (*integer*; default: **1**) - administrative distance of the route. When forwarding a packet, the router will use the route with the lowest administrative distance and reachable gateway

gateway-state (*read-only: r | u*) - shows the status of the next hop. Can be r (reachable) or u (unreachable)

- **(unknown)** - the gateway cannot be reached directly, or the route has been disabled

Notes

You can specify more than one or two gateways in the route. This is called Equal-cost multipath routing. Moreover, you can repeat some routes in the list several times to do a kind of cost setting for gateways.

Note also that if there are more than one public interface and more than one address on any of these interfaces, then policy-routing and equal-cost multipath routing may not work correctly if masquerading is used. To avoid this problem, use `action=nat` instead of `action=masquerade`.

Example

To add two static routes to networks 192.168.0.0/16 and 0.0.0.0/0 (the default destination address) on a router with two interfaces and two IP addresses:

```
[admin@MikroTik] ip route> add dst-address=192.168.0.0/16 gateway=10.10.10.2
[admin@MikroTik] ip route> add gateway 10.10.10.1
[admin@MikroTik] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, r - rip, o - ospf, b - bgp
#   DST-ADDRESS      G GATEWAY      DISTANCE  INTERFACE
0   S 192.168.0.0/16  r 10.10.10.2   1         Local
1   S 0.0.0.0/0       r 10.10.10.1   1         Public
2   DC 10.10.10.0/24  r 0.0.0.0      0         Public

[admin@MikroTik] ip route> print detail
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, r - rip, o - ospf, b - bgp
0   S dst-address=192.168.0.0/16 preferred-source=0.0.0.0
    gateway=10.10.10.2 gateway-state=reachable distance=1
    interface=Local

1   S dst-address=0.0.0.0/0 preferred-source=0.0.0.0 gateway=10.10.10.1
    gateway-state=reachable distance=1 interface=Public

2   DC dst-address=10.10.10.0/24 preferred-source=10.10.10.1
    gateway=0.0.0.0 gateway-state=reachable distance=0 interface=Public
[admin@MikroTik] ip route>
```

To set the **192.168.0.0/16** network is reachable via both **10.10.10.2** and **10.10.10.254** gateways:

```
[admin@MikroTik] ip route> set 0 gateway=10.10.10.2,10.10.10.254
[admin@MikroTik] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, r - rip, o - ospf, b - bgp
#   DST-ADDRESS      G GATEWAY      DISTANCE  INTERFACE
0   S 192.168.0.0/16  r 10.10.10.2   1         Local
   r 10.10.10.254   1         Local
1   S 0.0.0.0/0       r 10.10.10.1   1         Public
2   DC 10.10.10.0/24  r 0.0.0.0      0         Public
[admin@MikroTik] ip route>
```

Routing Tables

Home menu level: */ip policy-routing*

Description

Policy routing allows to select routes in order to variate the use of network resources to certain classes of users (in other words, you can set different routes to the same networks depending on some classifiers). This is implemented using multiple routing tables and a list of rules specifying how these tables should be used.

The Policy Routing is implemented in the MikroTik RouterOS based on source and destination addresses of a packet, the interface the packet arrives to the router and the firewall mark that may be associated with some packets.

When finding the route for a packet, the packet is matched against policy routing rules one after another, until some rule matches the packet. Then action specified in that rule is executed. If no rule matches the packet, it is assumed that there is no route to given host and appropriate action is taken (packet dropped and ICMP error sent back to the source).

If a routing table does not have a route for the packet, next rule after the one that directed to the current table is examined, until the route is found, end of rule list is reached or some rule with action drop or unreachable is hit. Thus it is good to have last rule say "from everywhere to everywhere, all interfaces, lookup main route table", because then gateways can be found (connected routes are entered in the main table only).

Note that the only way for packet to be forwarded is to have some rule direct to some routing table that contains route to packet destination.

Note also that if there are more than one public interface and more than one address on any of these interfaces, then policy-routing and equal-cost multipath routing may not work correctly if masquerading is used. To avoid this problem, use action=nat instead of action=masquerade.

Routing Tables

Routing tables is a way to organize routing rules into groups for a purpose of easy management. These tables can be created/deleted in the */ip policy-routing* menu.

The routes in the routing tables are managed the same way as the static routes described above, but in the submenu */ip policy-routing table name* submenu, where **name** is the name of the table.

Property Description

name (*name*) - table name

Example

There is always a table called **main**, this table cannot be deleted and its name cannot be changed. The **main** table can be managed in the **/ip route** submenu as well:

```
[admin@MikroTik] ip policy-routing> table main
[admin@MikroTik] ip policy-routing table main> print
Flags: X - disabled, I - invalid, D - dynamic, R - rejected
#   TYPE      DST-ADDRESS      G GATEWAY      DISTANCE  INTERFACE
0   static    192.168.1.0/24   r 192.168.0.50  1          Local
1   static    0.0.0.0/0       r 10.0.0.1     1          Public
2   D connect  192.168.0.0/24   r 0.0.0.0     0          Local
3   D connect  10.0.0.0/24     r 0.0.0.0     0          Public
[admin@MikroTik] ip policy-routing table main>
[admin@MikroTik] ip policy-routing table main> /ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
#   DST-ADDRESS      G GATEWAY      DISTANCE  INTERFACE
0   S 192.168.1.0/24   r 192.168.0.50  1          Local
1   S 0.0.0.0/0       r 10.0.0.1     1          Public
2   DC 192.168.0.0/24   r 0.0.0.0     0          Local
3   DC 10.0.0.0/24     r 0.0.0.0     0          Public
[admin@MikroTik] ip policy-routing table main>
```

To add a new table named **mt**:

```
[admin@MikroTik] ip policy-routing> add name=mt
[admin@MikroTik] ip policy-routing> print
Flags: D - dynamic
#   NAME
0   mt
1   D main
[admin@MikroTik] ip policy-routing>
```

To add the route to the **10.5.5.0/24** network via **10.0.0.22** gateway to the **mt** table:

```
[admin@MikroTik] ip policy-routing> table mt
[admin@MikroTik] ip policy-routing table mt> add dst-address=10.5.5.0/24 \
\d... 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
[admin@MikroTik] ip policy-routing table mt>
```

Policy Rules

Home menu level: **/ip policy-routing rule**

Property Description

src-address (*IP address/mask*) - source IP address/mask

dst-address (*IP address/mask*) - destination IP address/mask

interface (*name | all*; default: **all**) - interface name through which the packet arrives. Should be 'all' for the rule that should match locally generated or masqueraded packets, since at the moment of processing the routing table these packets have interface name set to loopback

flow (*name*; default: **''**) - flow mask of the packet to be matched by this rule. To add a flow, use **/ip firewall mangle** commands

action (*drop* | *unreachable* | *lookup*; default: **unreachable**) - action to be processed on packets matched by this rule:

- **drop** - silently drop packet
- **unreachable** - reply that destination host is unreachable
- **lookup** - lookup route in given routing table

Notes

You can use policy routing even if you use masquerading on your private networks. The source address will be the same as it is in the local network. In previous versions of RouterOS the source address changed to **0.0.0.0**

It is impossible to recognize peer-to-peer traffic from the first packet. Only already established connections can be matched. That also means that in case source NAT is treating Peer-to-Peer traffic differently from the regular traffic, Peer-to-Peer programs will not work (general application is policy-routing redirecting regular traffic through one interface and Peer-to-Peer traffic - through another). A known workaround for this problem is to solve it from the other side: making not Peer-to-Peer traffic to go through another gateway, but all other useful traffic go through another gateway. In other words, to specify what protocols (HTTP, DNS, POP3, etc.) will go through the gateway A, leaving all the rest (so Peer-to-Peer traffic also) to use the gateway B (it is not important, which gateway is which; it is only important to keep Peer-to-Peer together with all traffic except the specified protocols)

Example

To add the rule specifying that all the packets from the 10.0.0.144 host should lookup the **mt** routing table:

```
[admin@MikroTik] ip policy-routing rule> add src-address=10.0.0.144/32 \  
\... table=mt action=lookup  
[admin@MikroTik] ip policy-routing rule> print  
Flags: X - disabled, I - invalid  
# SRC-ADDRESS DST-ADDRESS INTE... FLOW ACTION TABLE  
0 0.0.0.0/0 0.0.0.0/0 all lookup main  
1 10.0.0.144/32 0.0.0.0/0 all lookup mt  
[admin@MikroTik] ip policy-routing rule>
```

Application Examples

Standard Policy-Routing Setup

Suppose we want packets coming from 1.1.1.0/24 to use gateway 10.0.0.1 and packets from 2.2.2.0/24 to use gateway 10.0.0.2. And the rest of packets will use gateway 10.0.0.254:

Command sequence to achieve this:

1. Add 3 new routing tables. One for local network **1.1.1.0/24**, one for network **2.2.2.0/24** and the rest for all other networks (**0.0.0.0/0**):

```
[admin@MikroTik] ip policy-routing> add name=from_net1; add name=from_net2; add  
name=rest  
[admin@MikroTik] ip policy-routing> print  
Flags: D - dynamic  
# NAME  
0 from_net1
```

```

1   from_net2
2   rest
2 D main
[admin@MikroTik] ip policy-routing>

```

2. Create the default route in each of the tables:

```

[admin@MikroTik] ip policy-routing> table from_net1 add gateway=10.0.0.1
[admin@MikroTik] ip policy-routing> table from_net2 add gateway=10.0.0.2
[admin@MikroTik] ip policy-routing> table rest add gateway=10.0.0.254
[admin@MikroTik] ip policy-routing> table from_net1 print
Flags: X - disabled, I - invalid, D - dynamic, R - rejected
#   TYPE   DST-ADDRESS      G GATEWAY      DISTANCE  INTERFACE
0   static 0.0.0.0/0        u 10.0.0.1      1         Public
[admin@MikroTik] ip policy-routing>
[admin@MikroTik] ip policy-routing> table from_net2 print
Flags: X - disabled, I - invalid, D - dynamic, R - rejected
#   TYPE   DST-ADDRESS      G GATEWAY      DISTANCE  INTERFACE
0   static 0.0.0.0/0        u 10.0.0.2      1         Public
[admin@MikroTik] ip policy-routing>
[admin@MikroTik] ip policy-routing> table rest print
Flags: X - disabled, I - invalid, D - dynamic, R - rejected
#   TYPE   DST-ADDRESS      G GATEWAY      DISTANCE  INTERFACE
0   static 0.0.0.0/0        u 10.0.0.254    1         Public
[admin@MikroTik] ip policy-routing>

```

3. Create rules that will direct traffic from sources to given tables, and arrange them in the desired order:

```

[admin@MikroTik] ip policy-routing> rule
[admin@MikroTik] ip policy-routing rule> print
Flags: X - disabled, I - invalid
#   SRC-ADDRESS      DST-ADDRESS      INT... FLOW      ACTION
0   0.0.0.0/0         0.0.0.0/0        all         lookup
[admin@MikroTik] ip policy-routing rule> add src-address=1.1.1.0/24 \
\... 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> add src-address=0.0.0.0/0 \
\... action=lookup table=rest
[admin@MikroTik] ip policy-routing rule> print
Flags: X - disabled, I - invalid
#   SRC-ADDRESS      DST-ADDRESS      INTERFACE  FLOW      ACTION
0   0.0.0.0/0         0.0.0.0/0        all         all         lookup
1   1.1.1.0/24        0.0.0.0/0        all         all         lookup
2   2.2.2.0/24        0.0.0.0/0        all         all         lookup
3   0.0.0.0/0         0.0.0.0/0        all         all         lookup
[admin@MikroTik] ip policy-routing rule>

```

Here the rule #0 is needed to reach directly connected networks. Note that there (in table main) is only directly connected routes! The rules #1 and #2 process local networks **1.1.1.0/24**, which is routed through the gateway **10.0.0.1**, and **2.2.2.0/24**, which is routed through the gateway **10.0.0.2**. Rule #3 handles packets originated from other networks (**0.0.0.0/0**).