

Manual:PCC

Applies to RouterOS: v3, v4



Introduction

PCC matcher will allow you to divide traffic into equal streams with ability to keep packets with specific set of options in one particular stream (you can specify this set of options from src-address, src-port, dst-address, dst-port)

Theory

PCC takes selected fields from IP header, and with the help of a hashing algorithm converts selected fields into 32-bit value. This value then is divided by a specified *Denominator* and the remainder then is compared to a specified *Remainder*, if equal then packet will be captured. You can choose from src-address, dst-address, src-port, dst-port from the header to use in this operation.

```
per-connection-classifier=  
PerConnectionClassifier ::= [!]ValuesToHash:Denominator/Remainder  
    Remainder ::= 0..4294967295      (integer number)  
    Denominator ::= 1..4294967295    (integer number)  
    ValuesToHash ::= both-addresses|both-ports|dst-address-and-port|  
        src-address|src-port|both-addresses-and-ports|dst-address|dst-port|src-address-and-port
```

Example

This configuration will divide all connections into 3 groups based on source address and port

```
/ip firewall mangle add chain=prerouting action=mark-connection \  
    new-connection-mark=1st_conn per-connection-classifier=src-address-and-port:3/0  
/ip firewall mangle add chain=prerouting action=mark-connection \  
    new-connection-mark=2nd_conn per-connection-classifier=src-address-and-port:3/1  
/ip firewall mangle add chain=prerouting action=mark-connection \  
    new-connection-mark=3rd_conn per-connection-classifier=src-address-and-port:3/2
```

Notes

PCC is available in RouterOS since v3.24. This option was introduced to address configuration issues with load balancing over multiple gateways with masquerade

Previous configurations:

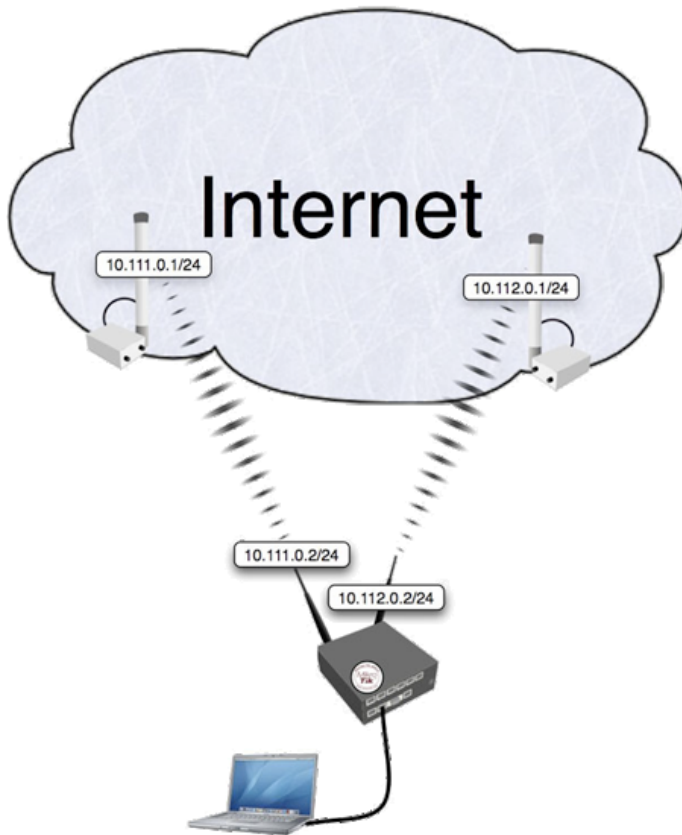
- ECMP load balancing with masquerade
- NTH load balancing with masquerade
- NTH load balancing with masquerade (another approach)



Note: PCC setups is not designed to work if **RP Filter** is enabled

Application Example - Load Balancing

Consider the following network layout:



Quick Start for Impatient

Configuration export from the gateway router:

```
/ ip address
add address=192.168.0.1/24 network=192.168.0.0 broadcast=192.168.0.255 interface=LAN
add address=10.111.0.2/24 network=10.111.0.0 broadcast=10.111.0.255 interface=ISP1
add address=10.112.0.2/24 network=10.112.0.0 broadcast=10.112.0.255 interface=ISP2

/ ip firewall mangle
add chain=prerouting dst-address=10.111.0.0/24 action=accept in-interface=LAN
add chain=prerouting dst-address=10.112.0.0/24 action=accept in-interface=LAN
add chain=prerouting in-interface=ISP1 connection-mark=no-mark action=mark-connection \
    new-connection-mark=ISP1_conn
add chain=prerouting in-interface=ISP2 connection-mark=no-mark action=mark-connection \
    new-connection-mark=ISP2_conn
add chain=prerouting in-interface=LAN connection-mark=no-mark dst-address-type=!local \
    per-connection-classifier=both-addresses:2/0 action=mark-connection new-connection-mark=ISP1_conn
add chain=prerouting in-interface=LAN connection-mark=no-mark dst-address-type=!local \
    per-connection-classifier=both-addresses:2/1 action=mark-connection new-connection-mark=ISP2_conn
add chain=prerouting connection-mark=ISP1_conn in-interface=LAN action=mark-routing \
```

```

    new-routing-mark=to_ISP1
add chain=prerouting connection-mark=ISP2_conn in-interface=LAN action=mark-routing \
    new-routing-mark=to_ISP2
add chain=output connection-mark=ISP1_conn action=mark-routing new-routing-mark=to_ISP1
add chain=output connection-mark=ISP2_conn action=mark-routing new-routing-mark=to_ISP2

/ ip route
add dst-address=0.0.0.0/0 gateway=10.111.0.1 routing-mark=to_ISP1 check-gateway=ping
add dst-address=0.0.0.0/0 gateway=10.112.0.1 routing-mark=to_ISP2 check-gateway=ping
add dst-address=0.0.0.0/0 gateway=10.111.0.1 distance=1 check-gateway=ping
add dst-address=0.0.0.0/0 gateway=10.112.0.1 distance=2 check-gateway=ping

/ ip firewall nat
add chain=srcnat out-interface=ISP1 action=masquerade
add chain=srcnat out-interface=ISP2 action=masquerade

```

Explanation

Let's assume this configuration:

IP Addresses

```

/ ip address
add address=192.168.0.1/24 network=192.168.0.0 broadcast=192.168.0.255 interface=LAN
add address=10.111.0.2/24 network=10.111.0.0 broadcast=10.111.0.255 interface=ISP1
add address=10.112.0.2/24 network=10.112.0.0 broadcast=10.112.0.255 interface=ISP2

```

The router has two upstream (ISP) interfaces with the addresses of 10.111.0.2/24 and 10.112.0.2/24. The LAN interface has IP address of 192.168.0.1/24.

Policy routing

```

/ ip firewall mangle
add chain=prerouting dst-address=10.111.0.0/24 action=accept in-interface=LAN
add chain=prerouting dst-address=10.112.0.0/24 action=accept in-interface=LAN

```

With policy routing it is possible to force all traffic to the specific gateway, even if traffic is destined to the host (other than gateway) from the connected networks. This way routing loop will be generated and communications with those hosts will be impossible. To avoid this situation we need to allow usage of default routing table for traffic to connected networks.

```

add chain=prerouting in-interface=ISP1 connection-mark=no-mark action=mark-connection \
    new-connection-mark=ISP1_conn
add chain=prerouting in-interface=ISP2 connection-mark=no-mark action=mark-connection \
    new-connection-mark=ISP2_conn

```

First it is necessary to manage connection initiated from outside - replies must leave via same interface (from same Public IP) request came. We will mark all new incoming connections, to remember what was the interface.

```

add chain=prerouting in-interface=LAN connection-mark=no-mark dst-address-type=!local \
    per-connection-classifier=both-addresses:2/0 action=mark-connection new-connection-mark=ISP1_conn
add chain=prerouting in-interface=LAN connection-mark=no-mark dst-address-type=!local \

```

```
per-connection-classifier=both-addresses:2/1 action=mark-connection new-connection-mark=ISP2_conn
```

Action mark-routing can be used only in mangle chain **output** and **prerouting**, but mangle chain **prerouting** is capturing all traffic that is going to the router itself. To avoid this we will use *dst-address-type=!local*. And with the help of the new PCC we will divide traffic into two groups based on source and destination addressees.

```
add chain=prerouting connection-mark=ISP1_conn in-interface=LAN action=mark-routing \
    new-routing-mark=to_ISP1
add chain=prerouting connection-mark=ISP2_conn in-interface=LAN action=mark-routing \
    new-routing-mark=to_ISP2
add chain=output connection-mark=ISP1_conn action=mark-routing new-routing-mark=to_ISP1
add chain=output connection-mark=ISP2_conn action=mark-routing new-routing-mark=to_ISP2
```

Then we need to mark all packets from those connections with a proper mark. As policy routing is required only for traffic going to the Internet, do not forget to specify in-interface option.

```
/ ip route
add dst-address=0.0.0.0/0 gateway=10.111.0.1 routing-mark=to_ISP1 check-gateway=ping
add dst-address=0.0.0.0/0 gateway=10.112.0.1 routing-mark=to_ISP2 check-gateway=ping
```

Create a route for each routing-mark

```
add dst-address=0.0.0.0/0 gateway=10.111.0.1 distance=1 check-gateway=ping
add dst-address=0.0.0.0/0 gateway=10.112.0.1 distance=2 check-gateway=ping
```

To enable failover, it is necessary to have routes that will jump in as soon as others will become inactive on gateway failure. (and that will happen only if check-gateway option is active)

NAT

```
/ ip firewall nat
add chain=srcnat out-interface=ISP1 action=masquerade
add chain=srcnat out-interface=ISP2 action=masquerade
```

As routing decision is already made we just need rules that will fix src-addresses for all outgoing packets. If this packet will leave via wlan1 it will be NATed to 10.112.0.2, if via wlan2 then NATed to 10.111.0.2

Article Sources and Contributors

Manual:PCC *Source:* <http://wiki.mikrotik.com/index.php?oldid=24755> *Contributors:* Janisk, Marisb, Megis, Normis

Image Sources, Licenses and Contributors

Image:Version.png *Source:* <http://wiki.mikrotik.com/index.php?title=File:Version.png> *License:* unknown *Contributors:* Normis

Image:Icon-note.png *Source:* <http://wiki.mikrotik.com/index.php?title=File:Icon-note.png> *License:* unknown *Contributors:* Marisb, Route

Image:LoadBalancing.png *Source:* <http://wiki.mikrotik.com/index.php?title=File:LoadBalancing.png> *License:* unknown *Contributors:* Normis