Load Balance with Masquerade Network on RouterOS

Prepared by:
Janis Megis (Mikrotik)
Valens Riyadi (Citraweb)

Copyrights 2010
About Me

• Jānis Meģis, MikroTik
• Jānis (Tehnical, Trainer, NOT Sales)
  – Support & Training Engineer for almost 6 years
  – Specialization: QoS, PPP, Firewall, Routing
  – Teaching MikroTik RouterOS classes since 2005
About Me

• Valens Riyadi - valens@mikrotik.co.id

• Company: Citraweb Nusa Infomedia
  – Wireless ISP - www.citra.net.id
  – Web Developer - www.citra.web.id

• Head of National Internet Resources of Indonesian ISP Association / IDNIC

• Founder and Volunteer of Airputih Foundation, an IT Emergency Task Force on Disaster Area
Basic Concept

• Load Balance
  – How to share traffic into 2 or more gateways

• Fail Over
  – How to choose one link as primary link, and automatically swing to another link if the primary link fail
Load Balance

• Load Balancing is a technique to distribute workload across two or more network links in order to maximize throughput, minimize response time, and avoid overload.

• Using multiple network links with load balancing, instead of a single network links, may increase reliability through redundancy.
Load Balance

$1 + 1 = 2$

$1 + 1 = 1 + 1$

$1 + 1 = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2}$

$1 + 1 = \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4}$

The more users, more connections, the load balance will be more balance
Load Balance

- The traffic distributed base on probability.
- We have to know how big is each link, and distributed traffic accordingly.
- If we have 2 gateways… A & B
  - A has 1 mbps, and B has 2 mbps
  - We will divide traffic to 3 flow, and send 1 flow to A, and 2 flows to B
RouterOS Features

• We need to use:
  – Static route and policy route
  – Firewall Mangle
  – Firewall src-nat

• For more advanced setting, we can use also OSPF and BGP
Key of Load Balance

• UPLINK
  – In simple network, we can choose which gateway we want to use for each uplink flow, using static route/policy route
Key of Load Balance

- **DOWNLINK**
  - In natted network, we choose downlink gateway using src-nat/masq. Traffic will return from internet according to IP Address we use in NAT for each flow.
  - In non natted network, we have to use BGP advertisement to control the routing from internet to our network.
Key Load Balance

• Traffic src-natted to IP Address located on gateway A, will return from internet through gateway A.

• If we use plain masquerade for each flow on all gateways, traffic will return from internet on the same gateway when leaving the network.
Static Route

• You can specify IP Address for the gateway in static route, if the interface is a static interface and has a static IP config.
Static Route

- For dynamic interface (ex: PPTP, PPPoE) you can choose interface as the gateway
Load Balance Method

- Static Route with Address List
- ECMP (equal cost multi path)
- NTH
- PCC
- BGP
Static Route

• Base on destination address
  – Gateway A for internasional
  – Gateway B for local/domestic traffic
    • Using address-list of IP Address on domestic network/local internet exchange
Static Route

- Base on source address
  - Client IP Address: 192.168.0.0/24
    - 192.168.0.0-127 → gateway A
    - 192.168.0.128-255 → gateway B
ECMP

- Equal Cost Multi Path
- The easiest way to do load balance for several gateways is using ECMP.
- ECMP will balance traffic to several gateways randomly
ECMP

- With 2 gateways with same capacity.
ECMP

• 2 gateway, capacity of gateway A is twice than gateway B
ECMP

- 3 gateway, gateway C is using gateway interface
ECMP Drawback

• As forwarding database is rebuilt every 10min in Linux Kernel, there is a chance that connection will jump to other gateway

• In case of masquerade this jump results in change of source address and in eventual disconnect

• More info at:
  – http://marc.info/?m=105217616607144
  – http://lkml.indiana.edu/hypermail/linux/net/0305.2/index.html#19
Configuration Setup

GW: 11.11.11.254

Internet

ISP1
/24

ISP2
/24

GW: 12.12.12.254

Masquerade

192.168.88.0/24
Basic Configuration

```bash
[admin@MikroTik] > /interface set 1 name=to_ISP1
[admin@MikroTik] > /interface set 2 name=to_ISP2
[admin@MikroTik] > /interface set 3 name=Local

[admin@MikroTik] /ip address> add address=192.168.88.254/24 interface=Local
[admin@MikroTik] /ip address> add address=11.11.11.1/24 interface=to_ISP1
[admin@MikroTik] /ip address> add address=12.12.12.1/24 interface=to_ISP2

[admin@MikroTik] /ip route> add gateway=11.11.11.254 distance=2
[admin@MikroTik] /ip route> add gateway=12.12.12.254 distance=3

[admin@MikroTik] /ip firewall nat> add chain=srcnat out-interface=to_ISP1 action=masquerade
[admin@MikroTik] /ip firewall nat> add chain=srcnat out-interface=to_ISP2 action=masquerade
```
Policy Routing

- Policy routing is a method that allow to create separate routing polices for different traffic by creating custom routing tables.

- In RouterOS these routing tables are created:
  - For every table specified in /ip route rule
  - For every routing-mark in mangle facility

- Marked traffic is automatically assigned to the proper routing table (no need for lookup rules)
Routing-mark

• RouterOS attribute assigned to each packet
• Routing-mark can be changed in firewall mangle facility just before any routing decision:
  – chain Prerouting – for all incoming traffic
  – chain Output – for outgoing traffic from router
• Every new routing mark have its own routing table with the same name
• By default all packets have “main” routing mark
Traffic to Connected Networks

• As connected routes are available only in “main” routing table, it is necessary that traffic to connected networks will stay in “main” routing table

• This will also allow proper communication between locally and remotely connected

```
/ip firewall mangle> add chain=prerouting src-address=192.168.88.0/24
dst-address=11.11.11.0/24 action=accept
/ip firewall mangle> add chain=prerouting src-address=192.168.88.0/24
dst-address=12.12.12.0/24 action=accept
/ip firewall mangle> add chain=prerouting src-address=192.168.88.0/24
dst-address=192.168.88.0/24 action=accept
```
Remote Connections

• In case when connection is initiated from public interface it is necessary to ensure that these connections will be replied via the same interface (from the same public IP)

• First we need to capture these connections (you can either use default connection mark “no-mark” or connection state “new” here)

```
/ip firewall mangle> add chain=prerouting connection-mark=no-mark in-interface=to ISP1
          action=mark-connection new-connection-mark=ISP1_conn
/ip firewall mangle> add chain=prerouting connection-mark=no-mark in-interface=to ISP2
          action=mark-connection new-connection-mark=ISP2_conn
```
Custom Policy Routing

• Now we need to create a default route for every routing table (or else it will be resolved by main routing table)

/ip route> add gateway=11.11.11.254 routing-mark=ISP1_traffic
/ip route> add gateway=12.12.12.254 routing-mark=ISP2_traffic

• Lets create a jump rule to your custom policy routing here

/ip firewall mangle> add chain=prerouting in-interface=Local connection-mark=no-mark action=jump jump-target=policy_routing
Mark Routing

• Mark routing rules in mangle chain “output” will ensure that router itself is reachable via both public IP addresses

• Mark routing rules in mangle chain “prerouting” will ensure your desired load balancing

```
ip firewall mangle> add chain=prerouting connection-mark=ISP1_conn src-address=192.168.88.0/24 action=mark-routing new-routing-mark=ISP1_traffic

/ip firewall mangle> add chain=prerouting connection-mark=ISP2_conn src-address=192.168.88.0/24 action=mark-routing new-routing-mark=ISP2_traffic

/ip firewall mangle> add chain=output connection-mark=ISP1_conn action=mark-routing new-routing-mark=ISP1_traffic

/ip firewall mangle> add chain=output connection-mark=ISP2_conn action=mark-routing new-routing-mark=ISP2_traffic
```
Mangle configuration

<table>
<thead>
<tr>
<th>#</th>
<th>Action</th>
<th>Chain</th>
<th>Src. Address</th>
<th>Dst. Address</th>
<th>In. Interface</th>
<th>Connection Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>accept</td>
<td>prerouting</td>
<td>192.168.88.0/24</td>
<td>11.11.11.0/24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>accept</td>
<td>prerouting</td>
<td>192.168.88.0/24</td>
<td>12.12.12.0/24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>accept</td>
<td>prerouting</td>
<td>192.168.88.0/24</td>
<td>192.168.88.0/24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>mark connection</td>
<td>prerouting</td>
<td></td>
<td></td>
<td>to_ISP1</td>
<td>no-mark</td>
</tr>
<tr>
<td>4</td>
<td>mark connection</td>
<td>prerouting</td>
<td></td>
<td></td>
<td>to_ISP2</td>
<td>no-mark</td>
</tr>
<tr>
<td>5</td>
<td>jump</td>
<td>prerouting</td>
<td></td>
<td>Local</td>
<td></td>
<td>no-mark</td>
</tr>
<tr>
<td>6</td>
<td>mark routing</td>
<td>prerouting</td>
<td>192.168.88.0/24</td>
<td></td>
<td>ISP1_conn</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>mark routing</td>
<td>prerouting</td>
<td>192.168.88.0/24</td>
<td></td>
<td>ISP2_conn</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>mark routing</td>
<td>output</td>
<td></td>
<td></td>
<td>ISP1_conn</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>mark routing</td>
<td>output</td>
<td></td>
<td></td>
<td>ISP2_conn</td>
<td></td>
</tr>
</tbody>
</table>
Custom Policy Routing

• There are no best way that we can suggest for load balancing you can either:
  – Balance based on client IP address (address list)
  – Balance based on traffic type (p2p, layer-7, protocol, port)
  – Use automatic balancing (PCC)

• We do not suggest to use “nth” for policy routing of typical user traffic.
Per-address-pair Load Balancing

- In many situations communication between two hosts consist of more than one simultaneous connection.
- If those connections are taking different routing path they might have different latency, drop rate, fragmentation or source address (NAT) – this way making multi-connection communications impossible.
- That is why instead of per-connection load balancing we should think about per-address-pair load balancing.
Per Connection Classifier

• PCC is a firewall matcher that allows you to divide traffic into equal streams with ability to keep packets with specific set of options in one particular stream

• You can specify set of options from src-address, src-port, dst-address, dst-port

• More info at: http://wiki.mikrotik.com/wiki/PCC
PCC Configuration

• We just need to add 2 rules to our “policy_routing” chain to ensure automatic per-address-pair load balancing

```
/ip firewall mangle> add chain=policy_routing dst-address-type=!local per-connection-classifier=both-addresses:2/0 action=mark-connection new-connection-mark=ISP1_conn
/ip firewall mangle> add chain=policy_routing dst-address-type=!local per-connection-classifier=both-addresses:2/1 action=mark-connection new-connection-mark=ISP2_conn
```
Usual Problems

• Be careful about using “no-mark” connection mark if you have other mangle configuration in different chain

• ISP specified DNS servers might block request from non-ISP public IPs, so we suggest to use public (ISP independent) DNS servers.

• If you would like to ensure fail-over – enable “check-gateway” option in all default routes.
Thank you!

• Q&A........
• Or email to:
  – support@mikrotik.com
  – valens@mikrotik.co.id