MikroTik RouterOS - Dual WAN failover routing without scripting (2 methods)

Help and Support

Help Desk

Homepage

SERVER MANAGEMENT

Managed Service Provider | Network Solutions | Information Systems | IT Projects | www.serman.ee

Overview

> This article is written by Server Management Inc. IT systems administrator Timo Puistaja - 03.03.2017, Tartu, Estonia.

Need help configuring your device? Contact us info@serman.ee

In many cases you need dual uplinks (from 2 different ISP´s) to be protected against potential downtime in your office. Sometimes 30min downtime can be devastating for the company. In this tutorial im going to show you how to achieve simple failover without any scripting. Mikrotik RouterOS is capable to do this in the layer of routing (/ip route) using built-in tool **"check-gateway=ping"**.

The already is a great tutorial in MikroTik Wiki, but it´s ***NOT WORKING for my ROS 6.38.1:***

1. *Routing Marks (ISP1 and ISP2) break the connection - removing them allows me and my RB to access internet. I don't know why.*
2. *Default routes (dst-address=0.0.0.0/0) are messed up - there are 4 routes with distances*

   1,2,2,1 - we actually only need 2 of them because we have 2 uplinks, everything comes clear if you follow the the tutorial.
3. *For PPP connections (PPPoE; PPtP etc.) YOU MUST USE your PPP gateway address as gateway - NOT just the interface name.*

In this tutorial **NO** load balancing OR dual remote access WAN´s are configured - only failover for LAN clients to access internet any time ISP fails.

There are several methods to do the failover with dual WAN/ISP/UPLINK (or name it whatever you link):

1. **Method 1 - Dual WAN failover routing with SINGLE external host ping check (ISP1 - Static IP; ISP2 - PPPoE)**
2. **Method 2- Dual WAN failover routing with MULTIPLE external host ping check (ISP1 - Static IP; ISP2 - PPPoE)**
3. **Method 3 - Dual WAN failover with Distance´s**

This tutorial also covers **testing you failover** to be 100% sure everything is as it should be.

First and second methods are better because you will check some kind of external webpage/server (**host**) to ensure your connection is running.

Sometimes the GW (usually the GW for your MikroTik will be you ISP router) of your connection is still working and pingable but there is no connetion - the line is broken. In this case your MikroTik still thinks ISP1/WAN1 is working but it actually isnt - thats why I recommend using first or second method for failover.

Method 2 is the best because your failover configuration doesn't have SPoF (Single Point of Failure) - for example your external host for checking is Google DNS (8.8.8.8) - **but what happens when 8.8.8.8 is NOT reachable (pingable) - yes Google´s services are sometimes DOWN too.** So we need failover host (another host) for our failover conf. Then we are going to use 2 or more (if you still feel unsure about the quality of your host) to be absolutely sure the ISP line is DOWN, not the host we are checking.

NB! Note that all the public IP´s im using are made up and actually configured in my other Mikrotik just for testing purposes.

NB! You CANNOT test failover with continious #ping google.com from your PC - because when jumping from ISP1 to ISP2 the packet flow needs to be restarted. For testing just open different webpages OR check Interface traffic from MikroTik. Good idea is to watch Interface traffic on Mikrotik while you are doing speedtest on internet.

Prerequisities/metainfo

1. This tutorial applies to **Router OS v6 (other verisions are not tested)**
2. Im using **RB3011UiAS as end device** in this example
3. My **RouterOS is version 6.38.1** at the moment
4. I have generated two "virtual ISP" connections on another MikroTik RB - so all the IP´s im using are made up.

MikroTik basic configuration BEFORE we are going to setup dual WAN failover

NB! Note that all the public IP´s im using are made up and actually configured in my other Mikrotik just for testing purposes.

BRIDGE - ports ether1 and ether2 are for my ISP´s, everything else is bridged:

```
/interface bridge
add admin-mac=00:BB:01:32:00:00 auto-mac=no name=bridge1
/interface bridge port
add bridge=bridge1 interface=ether3
add bridge=bridge1 interface=ether4
add bridge=bridge1 interface=ether5
add bridge=bridge1 interface=ether6
add bridge=bridge1 interface=ether7
add bridge=bridge1 interface=ether8
add bridge=bridge1 interface=ether9
add bridge=bridge1 interface=ether10
```

DNS - im using my ISP´s DNS servers for MikroTik:

```
/ip dns
set allow-remote-requests=yes servers=194.126.115.18,192.98.49.8
```

DHCP - simple 192.168.0.0/24 network. **Note** that im using Mikrotik itself (192.168.1.254) as DNS server for clients.

```
/ip dhcp-server
add address-pool=pool disabled=no interface=bridge1 lease-time=8h10m name=\
server1
/ip dhcp-server network
add address=192.168.1.0/24 dns-server=192.168.1.254 domain=SERMAN gateway=\
192.168.1.254
```

NAT - we have to masquerade our LAN network to default route(0.0.0.0/0) - thats how i prefer doing it:

```
/ip firewall nat
add action=masquerade chain=srcnat dst-address=0.0.0.0/0 src-address=\
192.168.1.0/24
```

FIREWALL - i have been using this firewall since i found out about THIS peresentation:

```
/ip firewall filter
add action=drop chain=input comment="Drop DNS requests from public" \
connection-state=new dst-port=53 in-interface=ether1 protocol=tcp
add action=drop chain=input comment="Drop DNS requests from public" \
connection-state=new dst-port=53 in-interface=ether1 protocol=udp
add action=drop chain=input comment="Disallow weird packets" connection-state=\
invalid
add action=accept chain=input comment="Allow LAN access to router and Internet" \
connection-state=new in-interface=bridge1
add action=accept chain=input comment="Allow connections that originated from LAN" \
connection-state=established
add action=accept chain=input comment="Allow ping ICMP from anywhere" protocol=\
icmp
add action=accept chain=input comment="Allow WAN access to router" dst-port=8291 \
protocol=tcp
add action=drop chain=input comment=\
"Disallow anything from anywhere on any interface"
add action=drop chain=forward comment="Disallow weird packets" connection-state=\
invalid
```

ETHER1 and ETHER2 - they are going to be changed based on the situation Im explaining.

Method 1 - Dual WAN failover with SINGLE remote host ping check (ISP1 - Static IP; ISP2 - PPPoE)

We have two uplinks: **MAIN (GW1 IP - 88.196.6.185)** and **BACKUP (PPoE GW IP: 10.10.1.1)** - usually those gateways are ISP routers.

So we need to monitor our gateways connectivity with external Hosts (thos Hosts can be whatever you think is stable enough - some webpage/server addresses. I´m using Google DNS serverver for this checking (Host1 = 8.8.8.8; Host2 = 8.8.4.4))

Host1 via **GW1** and **Host2** via **GW2**

1. First we create routes to thoe Hosts via corresponding gateways:

```
/ip route
add dst-address=Host1(8.8.8.8) gateway=GW1(88.196.6.185) scope=10
add dst-address=Host2(8.8.4.4) gateway=GW2(10.10.1.1) scope=10
```

2. Create default routes with different Distances

```
/ip route
add distance=1 gateway=Host1(8.8.8.8) check-gateway=ping
add distance=2 gateway=Host2(8.8.4.4) check-gateway=ping
```

Here is a picture of Route List window:

These routes will be resolved recusively and will be active only if **HostN** is pingable.

Thats it, now your failover is woking as needed. If your GW1 cant ping 8.8.8.8 - MikroTik will switch connection to GW2. If GW1 becomes back UP - MikroTik will switch connection back to the MAIN (GW1) connetion because its Distance is smaller (routing is more important).

Method 2 - Dual WAN failover with MULTIPLE remote host ping check (ISP1 - Static IP; ISP2 - PPPoE)

We have two uplinks: **MAIN (GW1 IP - 88.196.6.185)** and **BACKUP (GW2 - PPoE IP: 10.10.1.1)** - usually those gateways are ISP routers.

So we need to monitor our gateways connectivity with external Hosts (those Hosts can be whatever you think is stable enough - some webpage/server addresses. I´m using Google DNS and OpenDNS servers as hosts.

GoogleDNS:

Host1A: 8.8.8.8

Host1B: 208.67.220.220

OpenDNS:

Host2A: 8.8.4.4

Host2B: 208.67.222.222

We are going to check **Host1A** and **Host1B** via **GW1**

Host2A and **Host2B** via **GW2**

1. First we create routes to thoe Hosts via corresponding gateways:

```
/ip route
add dst-address=Host1A(8.8.8.8) gateway=GW1(88.196.6.185) scope=10
add dst-address=Host1B(208.67.220.220) gateway=GW1(88.196.6.185) scope=10
add dst-address=Host2A(8.8.4.4) gateway=GW2(10.10.1.1) scope=10
add dst-address=Host2B(208.67.222.222) gateway=GW2(10.10.1.1) scope=10
```

2. Create "virtual" hops for those hosts (IP addresses of virual hops arent really important, im using 10.1.1.1 and 10.2.2.2 as example)

```
/ip route
add dst-address=10.1.1.1 gateway=Host1A(8.8.8.8) scope=10 target-scope=10 check-gateway=ping
add dst-address=10.1.1.1 gateway=Host1B(208.67.220.220) scope=10 target-scope=10 check-gateway=ping
add dst-address=10.2.2.2 gateway=Host2A(8.8.4.4) scope=10 target-scope=10 check-gateway=ping
add dst-address=10.2.2.2 gateway=Host2B(208.67.222.222) scope=10 target-scope=10 check-gateway=ping
```

3. Add default routes for our clients:

```
/ip route
add distance=1 gateway=10.1.1.1
add distance=2 gateway=10.2.2.2
```

Here is a picture of Route List window:

These routes will be resolved recusively and will be active only if **HostN** is pingable.

Thats it, now your failover is woking as needed. If your GW1 cant ping 8.8.8.8 and *208.67.220.220* **AT THE SAME TIME (**connection switch only applies when both hosts are down) - MikroTik will switch connection to GW2. If any of GW1 hosts becomes back UP - MikroTik will switch connection back to the MAIN (GW1) connetion because its Distance is smaller (routing is more important).

Method 3 - Dual WAN failover with GW ping check

WAN1 (MAIN; ether1; ISP1; STATIC):

IP: 88.196.6.186

Mask: 255.255.255.248 (/29)

GW: 88.196.6.185

WAN2 (BACKUP; ether2; ISP2; STATIC):

IP: 194.204.36.11

Mask: 255.255.255.192 (/26)

GW: 194.204.36.62

LAN: (ether 3-10)

IP: 192.168.70.1

Mask: 255.255.255.0

DNS; DHCP: 192.168.70.1

When GW1 fails to answer the second gateway takes over. **Distance and check-gateway=ping** parameter are used for that:

```
/ip route
add check-gateway=ping distance=1 gateway=88.196.6.185
add distance=2 gateway=194.204.36.62
```

Thats it. When GW (88.196.5.185) does´t answer to ping, Mikrotik changes connection to other GW. When it comes back up, connetion is returned to the first gateway since the Distance is smaller (1) than other gateways (distance 2).

Note: As told in the beginnig this is very simple method for failover but in many cases your GW (ISP router) is pingable but connetion is actually down. So your failover conf FAILS to do the important job.

Testing the failover

For testing you can use MikroTik OR your PC.

You can just disable Interface.

For eg. diasble ether1 (WAN1; ISP1) and see if the traffic starts to flow through ether2 (WAN2; ISP2) and other way - **note: do not disable both of your WAN interfaces (especially when you are configuring it remotely). You wont be able to access the router from outside if you disable both interfaces.**

You can to the same testing physically - just disconnect cable going to ether1 (WAN1: ISP1) and see if you pc (in LAN) can access internet and other way.

NB! You CANNOT test failover with continious #ping google.com from your PC - because when jumping from ISP1 to ISP2 the packet flow needs to be restarted. For testing just open different webpages OR check Interface traffic from MikroTik. Good idea is to watch Interface traffic on Mikrotik while you are doing speedtest on internet.

Need help configuring your device? Contact us info@serman.ee

References

1. MikroTik Wiki - Advanced routing failover without scripting

2. MikroTik Wiki - Two gateways failover
3. MikroTik forum - Topic: check-gateway=ping isnot works, but netwatch - works

Last update: 22/05/2019