

Bandwidth-based load-balancing with failover. The easy way.

We need more bandwidth.

Presenter information

Tomas Kirnak

Network design

Security, wireless

Servers, Virtualization

Mikrotik Certified Trainer

Atris, Slovakia

Established 1991



Complete IT solutions

Networking, servers

Virtualization

IP security systems

Load-balancing, why?

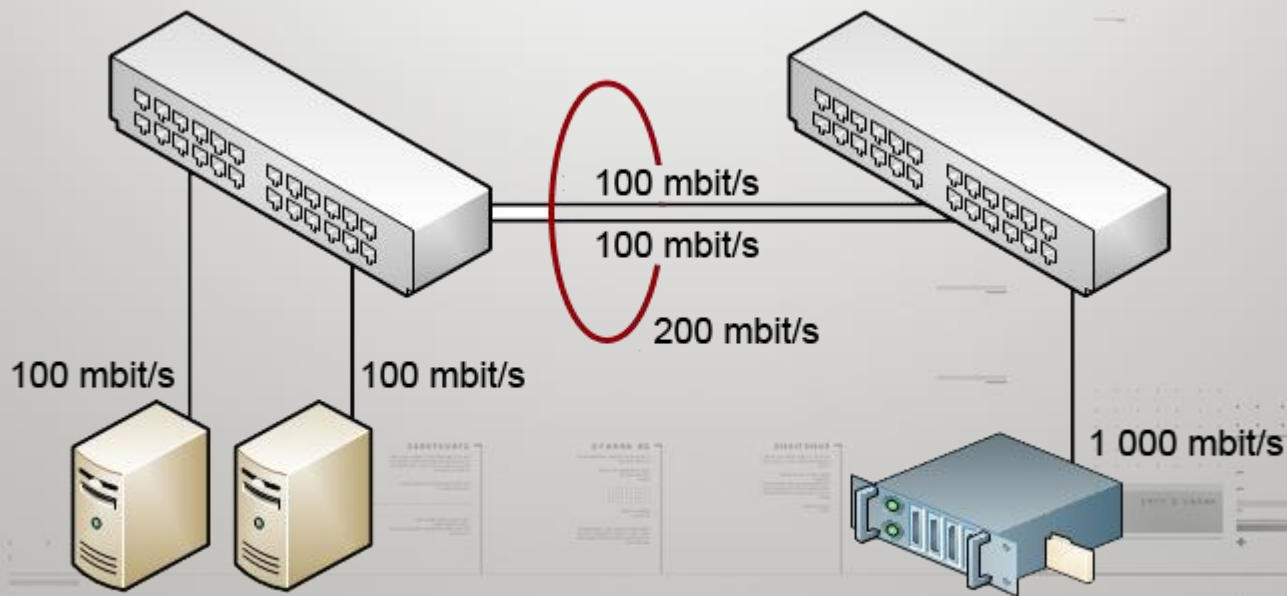
- Distributing workload to multiple network links to maximize throughput and minimize latency.
- Using multiple network links, when properly configured, will also provide redundancy.

Load balancing types

- Bonding
- Policy routing
- PCC
- Bandwidth based

Load balancing types

Bonding - 802.3ad LACP



Bonding

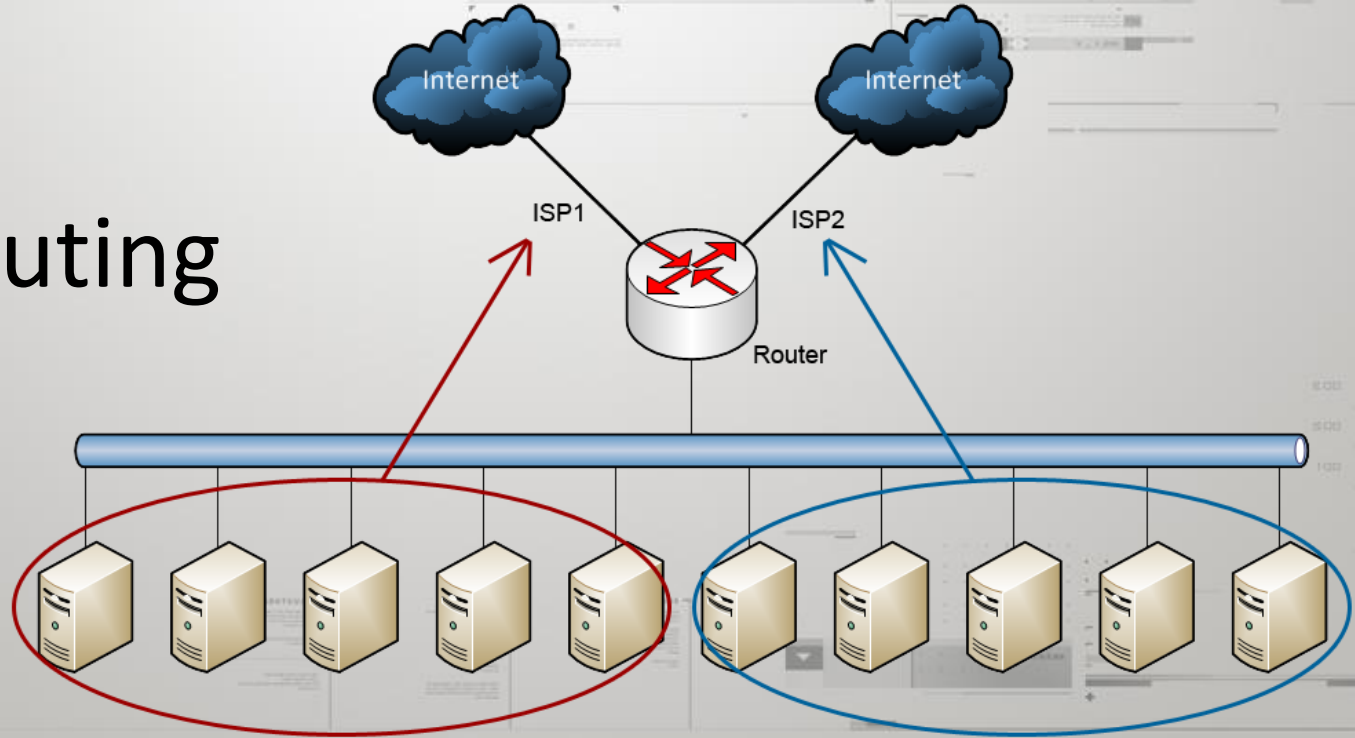
+ Easy to implement

Automatic redundancy with fail-over

- You need to control of both ends of the link

Load balancing types

Policy routing



Policy routing

+ Easy to implement

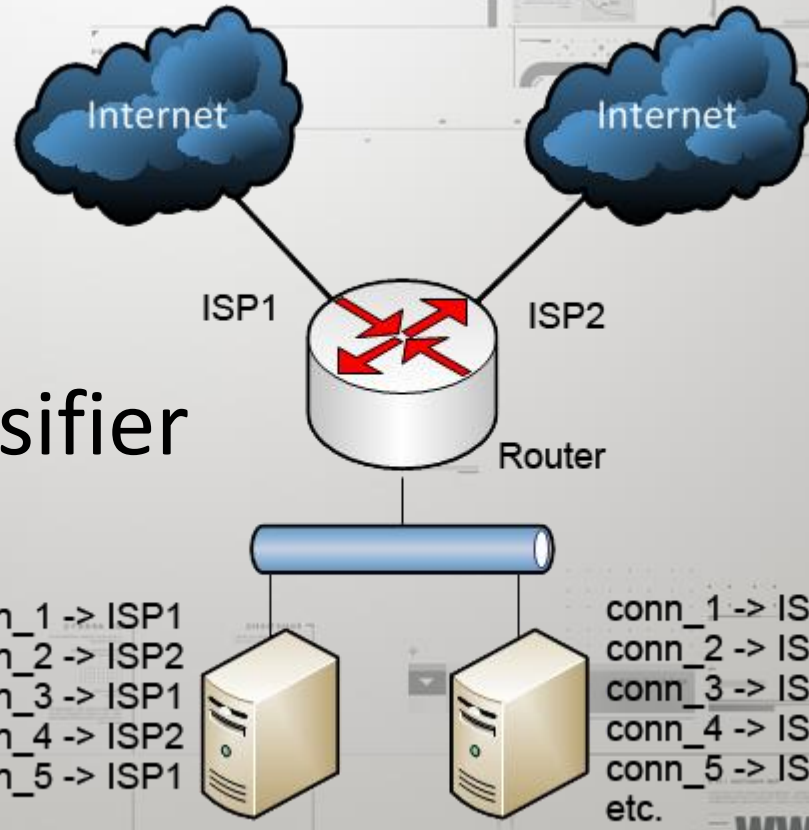
You have exact control of traffic

- Not dynamic

Scalability problems

Load balancing types

PCC
per connection classifier



PCC

- + Easy to configure
- Good scalability

- Not aware of link state (bandwidth wise)

Not so great with very un-similar links (4:1)

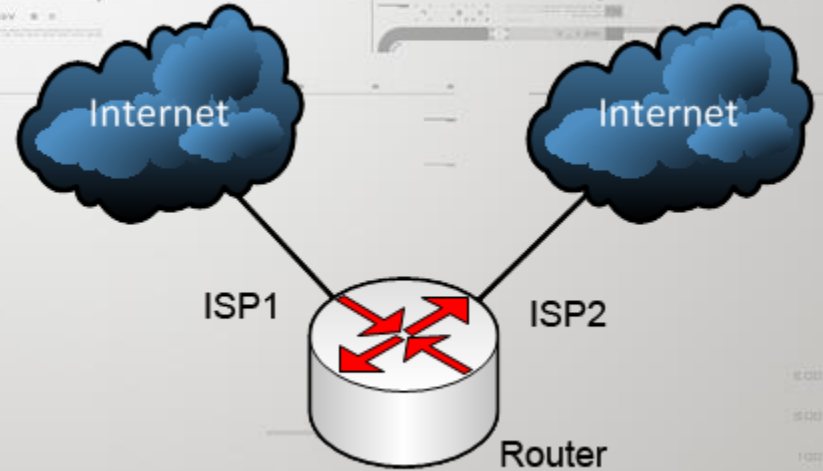
Load balancing types

For presentations on these load-balancing methods, please see

www.tiktube.com – PL 2010 and PL 2012

Load balancing types

Bandwidth based



If interface ISP1 is over 10 mbit/s; use ISP2

Why use bandwidth-based LB

- + Easily scalable
- + Takes link status into consideration
- + You have control over the connections
- + You decide when the switch to second link happens (on 10mbit link, switch after 50% util.)
- Comes with its own problems

Implementation considerations

- There are multiple ways to do bandwidth based load balancing, neither is so easy.
- MPLS TE
- Mangle + bit of scripting <-- this presentation

www.tiktube.com – PL 2010 and PL 2012

Underlying technologies

Connections and tracking them

Firewall

Filter Rules NAT Mangle Service Ports Connections Address Lists Layer7 Protocols

Tracking

	Src. Address	Dst. Address	Proto...	Connecti...	Connecti...	P2P	Timeout	TCP State
A	192.168.2.110:40244	199.167.177.38:1237	6 (tcp)				23:51:59	established
A	192.168.2.110:47716	173.194.70.188:5228	6 (tcp)				23:39:53	established
A	192.168.2.110:60474	199.167.177.59:1237	6 (tcp)				13:11:05	established
A	192.168.2.111:49823	64.4.23.141:40047	6 (tcp)				23:54:19	established
A	192.168.2.111:49833	65.55.71.73:443	6 (tcp)				23:53:59	established
A	192.168.2.111:49834	78.141.179.11:12350	6 (tcp)				23:45:59	established
A	192.168.2.111:50264	69.171.227.67:80	6 (tcp)				23:54:28	established
A	192.168.2.111:50265	69.171.227.67:80	6 (tcp)				23:54:28	established
A	192.168.2.111:50963	173.194.39.68:80	6 (tcp)				23:54:45	established
A	192.168.2.111:50980	173.194.39.65:80	6 (tcp)				23:58:36	established
A	192.168.2.111:50981	173.194.39.65:80	6 (tcp)				23:58:36	established
A	192.168.2.111:50982	173.194.39.65:80	6 (tcp)				23:58:36	established
A	192.168.2.111:50983	173.194.39.71:80	6 (tcp)				23:58:36	established
A	192.168.2.111:50984	173.194.39.71:80	6 (tcp)				23:58:36	established
A	192.168.2.111:50985	173.194.39.71:80	6 (tcp)				23:58:36	established
A	192.168.2.111:50986	173.194.39.71:80	6 (tcp)				23:58:36	established
A	192.168.2.111:50987	173.194.39.78:80	6 (tcp)				23:58:36	established
A	192.168.2.111:50988	173.194.39.78:80	6 (tcp)				23:58:36	established
A	192.168.2.111:50989	173.194.39.78:80	6 (tcp)				23:58:36	established

223 items out of 224 (1 selected) Max Entries: 0

Connection Tracking

Enabled

OK

Cancel

Apply

TCP Syn Sent Timeout: 00:00:05

TCP Syn Received Timeout: 00:00:05

TCP Established Timeout: 1d 00:00:00

TCP Fin Wait Timeout: 00:00:10

TCP Close Wait Timeout: 00:00:10

TCP Last Ack Timeout: 00:00:10

TCP Time Wait: 00:00:10

TCP Close: 00:00:10

UDP Timeout: 00:00:10

UDP Stream Timeout: 00:03:00

ICMP Timeout: 00:00:10

Generic Timeout: 00:10:00

TCP SynCookie

What is a connection

- We can define a connection as a packet flow with the same pair of source and destination IP addresses and ports.
- In case of UDP, this is would be an UDP stream.
- 192.168.2.10:49481 <-> 8.8.8.8:53

Mangle

- Mangle is a facility in ROS which allows us to “mark” packets or connections, and later use that mark for our purposes.
- Mangle marks do NOT leave the router.

Mangle – where to

The screenshot shows the Mikrotik WinBox interface. On the left is a sidebar menu with categories like Switch, Mesh, IP, IPv6, MPLS, Routing, System, Queues, Files, Log, Radius, Tools, New Terminal, MetaROUTER, Make Supout.rif, Manual, and Exit. The main window is titled 'Firewall' and has several tabs: Filter Rules, NAT, Mangle (highlighted), Service Ports, Connections, Address Lists, and Layer7 Protocols. Below the tabs are various icons and buttons, including a search bar with 'Find' and 'all' options. A table with columns for #, Action, Chain, Src. Address, Dst. Address, Proto..., Src. Port, Dst. Port, In. Inter..., and Out. Int... is visible, but it is currently empty. At the bottom left of the table area, it says '12 items'.

/ip

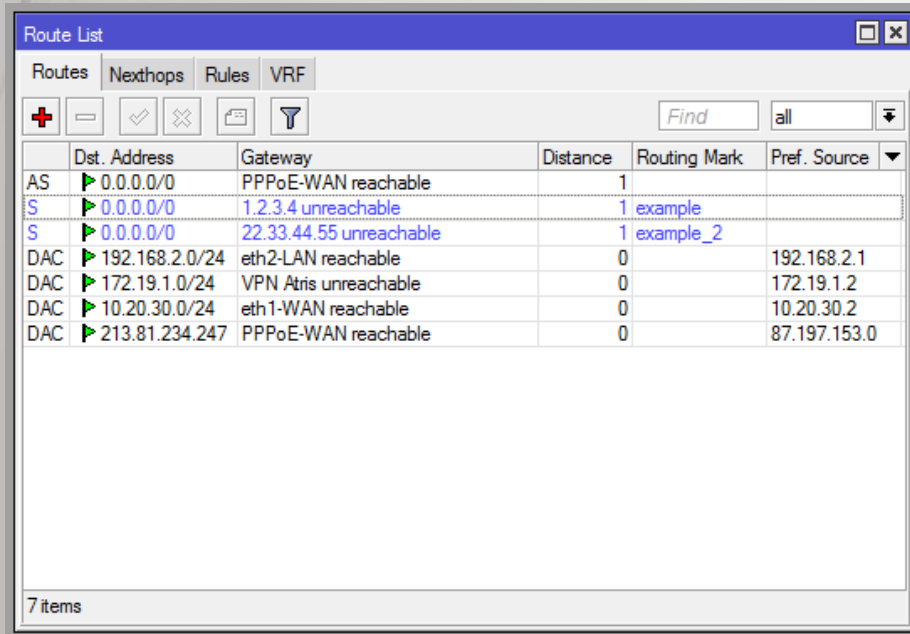
firewall

mangle

Routing tables

- A routing table tells the router which next hop to forward packets to, depending on the packets destination IP.
- `0.0.0.0/0 -> 77.21.34.12`

Routing tables – part 2

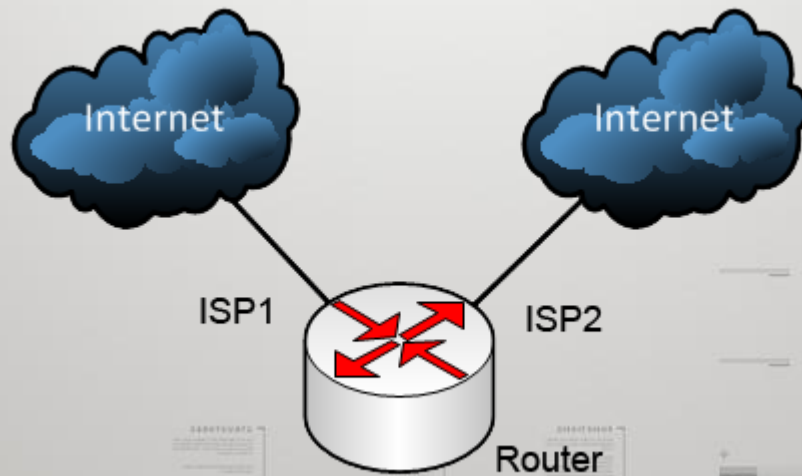


	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	▶ 0.0.0.0/0	PPPoE-WAN reachable	1		
S	▶ 0.0.0.0/0	1.2.3.4 unreachable	1	example	
S	▶ 0.0.0.0/0	22.33.44.55 unreachable	1	example_2	
DAC	▶ 192.168.2.0/24	eth2-LAN reachable	0		192.168.2.1
DAC	▶ 172.19.1.0/24	VPN Atris unreachable	0		172.19.1.2
DAC	▶ 10.20.30.0/24	eth1-WAN reachable	0		10.20.30.2
DAC	▶ 213.81.234.247	PPPoE-WAN reachable	0		87.197.153.0

7 items

- By default all packets are put into the “main” routing table
- We can create our own routing tables, and force packets to use them.

Topology



Required steps

- Create routing tables
- Setup address-lists
- Setup mangle
- Configure Traffic Monitor

Basic configuration

```
/interface ethernet
```

```
set 0 name=LAN
```

```
set 3 name=ISP_1
```

```
set 4 name=ISP_2
```

```
/ip address
```

```
add address=192.168.22.1/24 interface=LAN
```

```
add address=1.1.1.32/24 interface=ISP_1
```

```
add address=2.2.2.65/24 interface=ISP_2
```

```
/ip firewall nat
```

```
add action=masquerade chain=srcnat out-interface=ISP_1
```

```
add action=masquerade chain=srcnat out-interface=ISP_2
```


Routing tables

`/ip route`

`add gateway=1.1.1.1 distance=1`

`add gateway=2.2.2.1 distance=2`

`add gateway=1.1.1.1 routing-mark=ISP1_Route distance=1`

`add gateway=2.2.2.1 routing-mark=ISP2_Route distance=1`

Routing tables - GUI

Route List

Routes | Nexthops | Rules | VRF

Find all

	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	▶ 0.0.0.0/0	1.1.1.1 reachable ISP_1	1		
S	▶ 0.0.0.0/0	2.2.2.1 reachable ISP_2	2		
AS	▶ 0.0.0.0/0	1.1.1.1 reachable ISP_1	1	ISP1_Route	
AS	▶ 0.0.0.0/0	2.2.2.1 reachable ISP_2	1	ISP2_Route	
DAC	▶ 1.1.1.0/24	ISP_1 reachable	0		1.1.1.32
DAC	▶ 2.2.2.0/24	ISP_2 reachable	0		2.2.2.65
DAC	▶ 192.168.22.0/...	LAN reachable	0		192.168.22.1

7 items

Traffic to connected networks

	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	0.0.0.0/0	1.1.1.1 reachable ISP_1	1		
S	0.0.0.0/0	2.2.2.1 reachable ISP_2	2		
AS	0.0.0.0/0	1.1.1.1 reachable ISP_1	1	ISP1_Route	
AS	0.0.0.0/0	2.2.2.1 reachable ISP_2	1	ISP2_Route	
DAC	1.1.1.0/24	ISP_1 reachable	0		1.1.1.32
DAC	2.2.2.0/24	ISP_2 reachable	0		2.2.2.65
DAC	192.168.22.0/24	LAN reachable	0		192.168.22.1

7 items

- Connected networks are only in the “main” routing table
- We need to make sure that traffic to these networks stays in the main routing table.

Connected networks – part 2

/ip firewall address-list

```
add address=1.1.1.0/24 list=Connected
```

```
add address=2.2.2.0/24 list=Connected
```

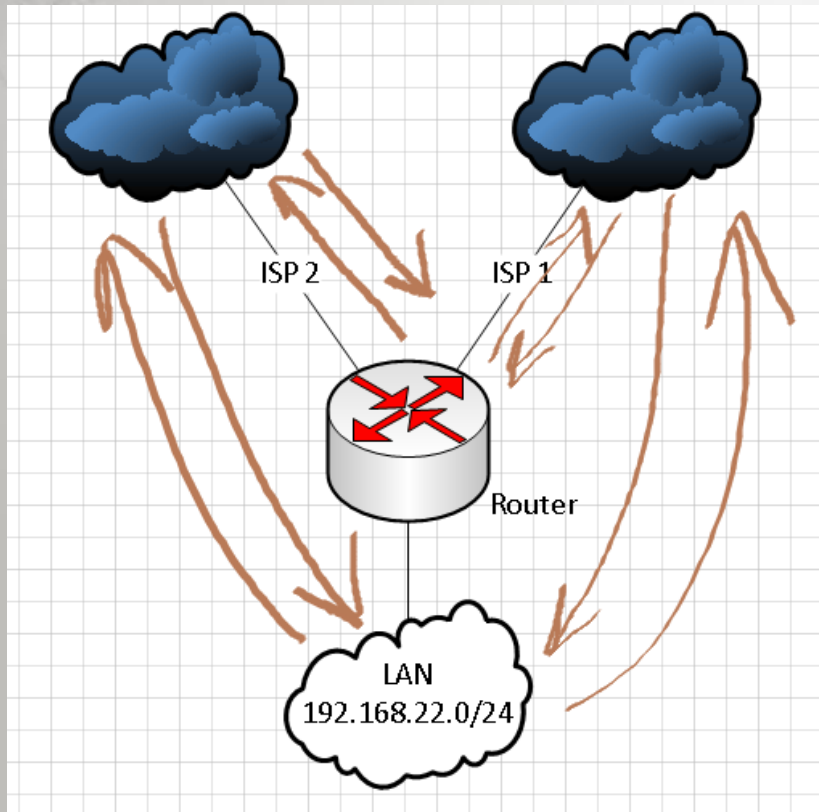
```
add address=192.168.22.0/24 list=Connected
```

```
add address=192.168.22.0/24 list=LAN
```

/ip firewall mangle

```
add chain=prerouting src-address-list=Connected  
dst-address-list=Connected action=accept
```

Topology – take 2



- In this topology, there are 4 possible traffic flows
- WAN -> Router
- Router -> WAN
- WAN -> LAN
- LAN -> WAN

Taking care of incoming connections

- When a connection is initiated from the internet through one of the ISPs we need to ensure that this connections is replied through the same ISP (from the same public IP)
- We need to mark these connections, and then put them in the proper routing table.

Router marking – WAN -> Router

- Catch the connection from internet to the router, and mark them.

`/ip firewall mangle`

```
add chain=input connection-mark=no-mark in-interface=ISP_1  
  action=mark-connection new-connection-mark=WAN1->ROS
```

```
add chain=input connection-mark=no-mark in-interface=ISP_2  
  action=mark-connection new-connection-mark=WAN2->ROS
```

Router marking – WAN -> Router

- Then put these connections into the proper routing tables.

```
add chain=output connection-mark=WAN1->ROS  
action=mark-routing new-routing-mark=ISP1_Route
```

```
add chain=output connection-mark=WAN2->ROS  
action=mark-routing new-routing-mark=ISP2_Route
```


Taking care of the LAN

- Same principle applies to the LAN.
- Connections initiated from the internet through one ISP, should be replied to through the same ISP.

LAN marking

`/ip firewall mangle`

```
add chain=forward connection-mark=no-mark in-interface=ISP_1  
  action=mark-connection new-connection-mark=WAN1->LANs
```

```
add chain=forward connection-mark=no-mark in-interface=ISP_2  
  action=mark-connection new-connection-mark=WAN2->LANs
```

```
add chain=prerouting connection-mark=WAN1->LANs src-address-list=LAN  
  action=mark-routing new-routing-mark=ISP1_Route
```

```
add chain=prerouting connection-mark=WAN2->LANs src-address-list=LAN  
  action=mark-routing new-routing-mark=ISP2_Route
```

Incoming connections - done

- We have ensured that when a connection from the internet to our router, or services inside of our network is established, it works.

LAN – partially done

- Connections from the internet to our LAN will now work through both ISPs
- So what about connections outgoing from our LAN to the internet?
- These we actually want to load-balance.

A sticky connection

- A sticky connection is a connection, that once established through one interface, will always go out that exact interface.
- This is required, because when we switch to a second link, we only need to switch new connections.
- In PCC, this is done automatically. Using our approach however, this has to be done manually.

LAN -> WAN mangle

`/ip firewall mangle`

```
add chain=prerouting connection-mark=no-mark src-address-list=LAN dst-address-list=!Connected dst-address-type=!local action=mark-connection new-connection-mark=LAN->WAN
```

```
add chain=prerouting connection-mark=LAN->WAN src-address-list=LAN action=mark-routing new-routing-mark=ISP1_Route comment="Load-Balancing here"
```

- Configuring this, we can now manually influence which routing table will our connection from LAN to the internet take.

Sticky connections

```
add chain=prerouting connection-mark=LAN->WAN routing-mark=ISP1_Route  
action=mark-connection new-connection-mark=Sticky_ISP1
```

```
add chain=prerouting connection-mark=LAN->WAN routing-mark=ISP2_Route  
action=mark-connection new-connection-mark=Sticky_ISP2
```

```
add chain=prerouting connection-mark=Sticky_ISP1 src-address-list=LAN  
action=mark-routing new-routing-mark=ISP1_Route
```

```
add chain=prerouting connection-mark=Sticky_ISP2 src-address-list=LAN  
action=mark-routing new-routing-mark=ISP2_Route
```

- This will assure that once a connection is routed through one ISP, it will stay there no matter what.

Mangle in GUI

Firewall

Filter Rules NAT Mangle Service Ports Connections Address Lists Layer7 Protocols

+ - ✓ ✗ [Icon] [Icon] Reset Counters 00 Reset All Counters Find all

#	Action	Chain	In. Interface	Connection ...	Routing Mark	Src. Addr...	Dst. Addr...	New Connection Mark	New Routing Mark
::: Connected networks - ACCEPT									
0	✓ accept	prerouting				Connected	Connected		((
::: WAN -> ROS									
1	✓ mark connection	input	ISP_1	no-mark				WAN1->ROS	((
2	✓ mark connection	input	ISP_2	no-mark				WAN2->ROS	((
3	✓ mark routing	output		WAN1->ROS					ISP1_Route ((
4	✓ mark routing	output		WAN2->ROS					ISP2_Route ((
::: WAN -> LANs									
5	✓ mark connection	forward	ISP_1	no-mark				WAN1->LANs	((
6	✓ mark connection	forward	ISP_2	no-mark				WAN2->LANs	((
7	✓ mark routing	prerouting		WAN1->LANs		LAN			ISP1_Route ((
8	✓ mark routing	prerouting		WAN2->LANs		LAN			ISP2_Route ((
::: LAN -> WAN									
9	✓ mark connection	prerouting		no-mark		LAN	!Connected	LAN->WAN	((
::: Load-Balancing here									
10	✓ mark routing	prerouting		LAN->WAN		LAN			ISP1_Route ((
::: Stick connections after this									
11	✓ mark connection	prerouting		LAN->WAN	ISP1_Route			Sticky_ISP1	((
12	✓ mark connection	prerouting		LAN->WAN	ISP2_Route			Sticky_ISP2	((
13	✓ mark routing	prerouting		Sticky_ISP1		LAN			ISP1_Route ((
14	✓ mark routing	prerouting		Sticky_ISP2		LAN			ISP2_Route ((

15 items

What's the final result?

- We can load balancing manually
- Connections go out ISP1, then we can switch the mangle rule to ISP2, but connections already using ISP1 will stay there.

Automating based on bandwidth

Traffic Monitor <LB1>

Name:

Interface:

Traffic:

Trigger:

Threshold:

On Event:

```
log warning "LB Debug: ISP1 overloaded, switching to ISP2";  
/ip firewall mangle set [find comment="Load-Balancing here"] new-routing-mark=ISP2_Route
```

enabled

OK
Cancel
Apply
Disable
Comment
Copy
Remove

Switching back

Traffic Monitor <LB2>

Name: LB2

Interface: ISP_1

Traffic: received

Trigger: below

Threshold: 5242880

On Event:

```
log warning "LB Debug: ISP1 back to normal";  
/ip firewall mangle set [find comment="Load-Balancing here"] new-routing-mark=ISP1_Route
```

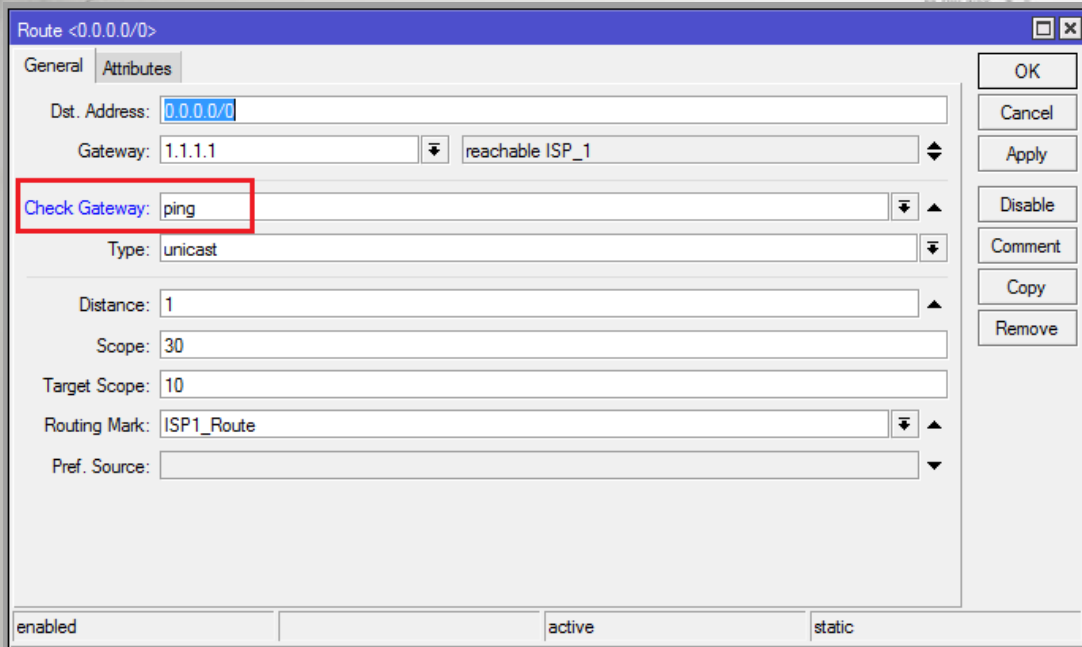
enabled

OK
Cancel
Apply
Disable
Comment
Copy
Remove

Final result

- Connections routed through ISP1, until its link is at 5mbit/s.
- After this limit all new connections will go through ISP2 until the ISP1 link is under its limit.
- Automated, bandwidth-based load balancing.

Easy Failover



The screenshot shows a network configuration window titled "Route <0.0.0.0/0>". It has two tabs: "General" and "Attributes". The "General" tab is active. The "Dst. Address" is set to "0.0.0.0/0". The "Gateway" is set to "1.1.1.1" with a dropdown menu showing "reachable ISP_1". The "Check Gateway" field is highlighted with a red box and contains the text "ping". The "Type" is set to "unicast". The "Distance" is "1", "Scope" is "30", and "Target Scope" is "10". The "Routing Mark" is set to "ISP1_Route". The "Pref. Source" is empty. At the bottom, there are three status indicators: "enabled", "active", and "static". On the right side of the window, there are buttons for "OK", "Cancel", "Apply", "Disable", "Comment", "Copy", and "Remove".

enabled	active	static
---------	--------	--------

- If the gateway can't be pinged, all routes using this gateway will become invalid.

A different approach

- This approach will not work if the link failure happens after the gateway.
- Recursive route lookup, netwatch etc.
- http://wiki.mikrotik.com/wiki/Failover_Scripting



Thanks for listening

Tomas Kirnak
t.kirnak@atris.sk

WHAT PART OF

$$i\hbar \frac{\partial}{\partial t} \Psi(\vec{r}, t) = \left(-\frac{\hbar^2}{2m} \nabla^2 + V(\vec{r}, t) \right) \Psi(\vec{r}, t)$$

DON'T YOU UNDERSTAND?

9gag.com/vinzimmermann

Find me after the presentation for any questions.