Basic Setup

Let's suppose that we have two uplinks: **GW1**, **GW2**. It can be addresses of ADSL modems (like *192.168.1.1* and*192.168.2.1*), or addresses of PPP interfaces (like *pppoe-out1* and *pptp-out1*). Then, we have some policy routing rules, so all outgoing traffic is marked with **ISP1** (which goes to **GW1**) and **ISP2** (which goes to **GW2**) marks. And we want to monitor **Host1** via **GW1**, and **Host2** via **GW2** - those may be some popular Internet websites, like Google, Yahoo, etc.

First, create routes to those hosts via corresponding gateways:

```
/ip route

add dst-address=Host1 gateway=GW1 scope=10

add dst-address=Host2 gateway=GW2 scope=10
```

Now we create rules for **ISP1** routing mark (one for main gateway, and another one for failover):

```
/ip route

add distance=1 gateway=Host1 routing-mark=ISP1 check-gateway=ping

add distance=2 gateway=Host2 routing-mark=ISP1 check-gateway=ping
```

Those routes will be resolved recursively (see Manual:IP/Route#Nexthop_lookup), and will be active only if **HostN** is pingable.

Then the same rules for **ISP2** mark:

```
/ip route

add distance=1 gateway=Host2 routing-mark=ISP2 check-gateway=ping

add distance=2 gateway=Host1 routing-mark=ISP2 check-gateway=ping
```

Multiple host checking per Uplink

If **Host1** or **Host2** in #Basic Setup fails, corresponding link is considered failed too. For redundancy, we may use several hosts per uplink: let's monitor **Host1A** and **Host1B** via **GW1**, and **Host2A** and **Host2B** via **GW2**. Also, we'll use double recursive lookup, so that there were fewer places where **HostN** is mentioned.

As earlier, first we need routes to our checking hosts:

```
/ip route

add dst-address=Host1A gateway=GW1 scope=10

add dst-address=Host1B gateway=GW1 scope=10

add dst-address=Host2A gateway=GW2 scope=10

add dst-address=Host2B gateway=GW2 scope=10
```

Then, let's create destinations to "virtual" hops to use in further routes. I'm using *10.1.1.1* and *10.2.2.2* as an example:

```
/ip route

add dst-address=10.1.1.1 gateway=Host1A scope=10 target-scope=10 check-gateway=ping

add dst-address=10.1.1.1 gateway=Host1B scope=10 target-scope=10 check-gateway=ping

add dst-address=10.2.2.2 gateway=Host2A scope=10 target-scope=10 check-gateway=ping

add dst-address=10.2.2.2 gateway=Host2B scope=10 target-scope=10 check-gateway=ping
```

And now we may add default routes for clients:

```
/ip route

add distance=1 gateway=10.1.1.1 routing-mark=ISP1

add distance=2 gateway=10.2.2.2 routing-mark=ISP1

add distance=1 gateway=10.2.2.2 routing-mark=ISP2

add distance=2 gateway=10.1.1.1 routing-mark=ISP2
```

GW1= 192.168.1.1

GW2=187.16.47.73

GW3=192.168.25.1

Host A: www.uol.com.br

HostB: www.google.com.br

```
/ip route

add dst-address=200.221.2.45 gateway=192.168.1.1 scope=10

add dst-address=216.58.216.67 gateway=192.168.1.1 scope=10

add dst-address=200.221.2.45 gateway=187.16.47.73 scope=10

add dst-address=216.58.216.67 gateway=187.16.47.73 scope=10

add dst-address=200.221.2.45 gateway=192.168.25.1 scope=10

add dst-address=216.58.216.67 gateway=192.168.25.1 scope=10
```

```
/ip route

add dst-address=10.1.1.1 gateway=200.221.2.45 scope=10 target-scope=10
check-gateway=ping

add dst-address=10.1.1.1 gateway=216.58.216.67 scope=10 target-scope=10
check-gateway=ping
```

```
/ip route

add distance=1 gateway=10.1.1.1 routing-mark=ISP1

add distance=2 gateway=10.2.2.2 routing-mark=ISP1

add distance=1 gateway=10.2.2.2 routing-mark=ISP2

add distance=2 gateway=10.1.1.1 routing-mark=ISP2

add distance=1 gateway=10.1.1.1 routing-mark=ISP3

add distance=2 gateway=10.2.2.2 routing-mark=ISP3
```