ELEMENTS®

Using DSCP for traffic separation, routing and load balancing

Presenter information

Tomas Kirnak

Network design and security
Ethernet and wireless
Servers & Virtualization
Network management & monitoring

MikroTik Certified Trainer
MikroTik Certified Consultant

RF elements s.r.o.

Wireless equipment manufacturer
Enclosures
Antennas
Shields

RF ELEMENTS®

Agenda

- What is DSCP
- Scenario 1:
  Using DSCP for routing and load-balancing
- Scenario 2:
  Using DSCP for traffic-separation and routing

What is DSCP

What DSCP actually is

- DSCP - **D**ifferentiated **S**ervices **C**ode **P**oint
- Uses DS field (Differentiated services field) in each IP packet
- Replace the outdated TOS field

How to DSCP

- DSCP can be injected into the IPv4 packet by any device processing the packet:
  - Packet's origin can choose what DSCP to send packet with
  - A router processing the packet can also inject/modify/remove the DSCP of the packet

- In RouterOS – mangle facility

DSCP sticks

- DSCP is not like a packet / connection mark
- DSCP is an actual field in the IP packet

- It sticks to the packet for the whole duration of that packet's life

- We can therefore use it over the whole path the packet takes

What is it used for

- DSCP is for QoS – not entirely true

- “Differentiated services or DiffServ is a […] mechanism for classifying and managing network traffic […].”

  - www.wikipedia.com

- Once traffic is classified using a DSCP tag, we can do QoS
- But this classification can also be used for different purposes

Scenario 1

BGP is wonderful, but…

- BGP is wonderful and the internet would not work without it
- But there are many problems
  - very little equipment (that doesn't cost over 9000 $) supports it
  - network admins don't know how to use it
  - very hard to get a normal ISP to BGP peer with you
  - administrative paperwork required (get an AS, get a PA or PI block)
  - money…
  - BGP security … non existent (anyone can announce anything)

Why are we talking about BGP?

- If an organization (SMB, a small enterprise, a school) needs to get multiple upstream connections…
- They are not gonna be BGP

- A different (and easier on all fronts) mechanism is needed for load-balancing and failover

Just to prove a point…

- How many ASs* are in your country?
- http://bgp.he.net/report/prefixes#_countriesv4

* - Not DDoS asses – thanks for that yesterday Tom Smyth

Lets get back on topic

- DSCP can be used as a classifier to spread traffic over multiple ISPs
- We divide our traffic into a certain number of “streams” and assign different DSCP to each stream
- Later we can route based on DSCP tag to different ISPs

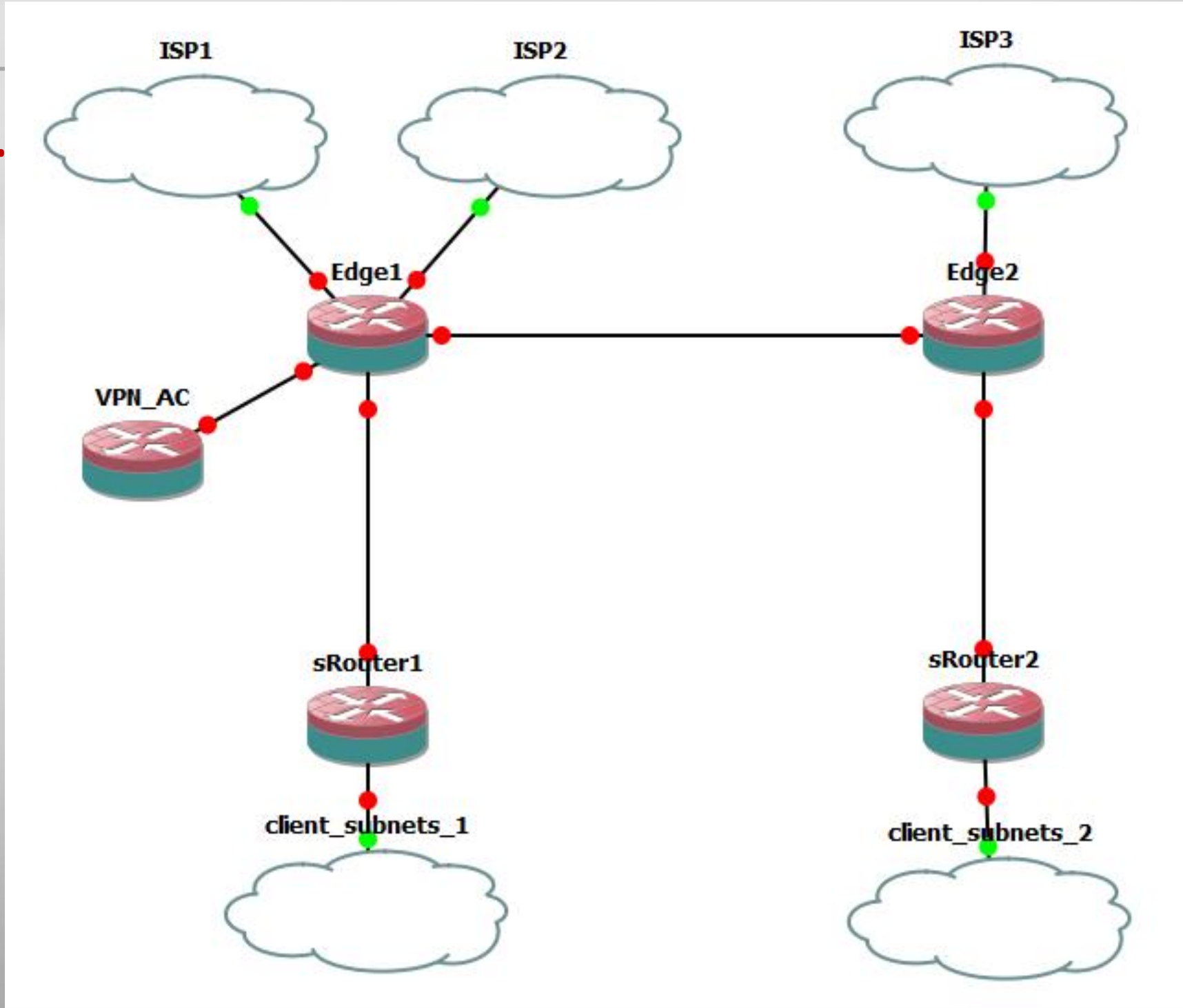
Does this sound familiar?

- This is basically PCC
- So why do we need DSCP?
- Because we are talking about medium size networks, that can have ISPs spread over multiple locations (therefore multiple routers)

RF ELEMENTS®

Let me explain…

- Assume this is an SMB
- Assume Edge1 and Edge2 are in separate buildings
- We need DSCP



How to accomplish this?

- Load balancing with MikroTik 101:

  http://tiktube.com/video/DofH3iFnjDJomGEoIDFqnrquKlEoLqHq=

- If you don’t understand what the next few slides mean, watch that presentation first

- It will explain the basics of Load-Balancing on MikroTik

How to accomplish this

- First fix outgoing traffic to directly attached subnets:

  /ip firewall mangle

  add chain=prerouting dst-address-list=Upstream-Connected src-address-list=Private-Networks action=mark-connection new-connection-mark=Upstream-Connected comment="Traffic to Upstream-Connected networks - Should remain in main routing table"

  add chain=prerouting connection-mark=Upstream-Connected

- Now fix traffic to private subnets:

  /ip firewall mangle

  add chain=prerouting dst-address-list=Private-Networks src-address-list=Private-Networks action=mark-connection new-connection-mark=Private-Networks comment="Traffic to Private-Networks - Should remain in main routing table"

  add chain=prerouting connection-mark=Private-Networks

How to accomplish this 2

- Populate the address-lists

```
/ip firewall address-list
add address=x.x.x.x/27 list=Upstream-Connected
add address=x.x.x.x/29 list=Upstream-Connected

/ip firewall address-list
add address=10.0.0.0/8 list=Private-Networks
add address=172.16.0.0/12 list=Private-Networks
add address=192.168.0.0/16 list=Private-Networks
```

Where to configure this

- Do this on both Edge1 and Edge2
- Make sure to configure correct subnets in the Upstream-Connected address-list

Traffic from WAN to ROS

- Ensure traffic to ROS arriving over certain ISP goes back out that ISP interface

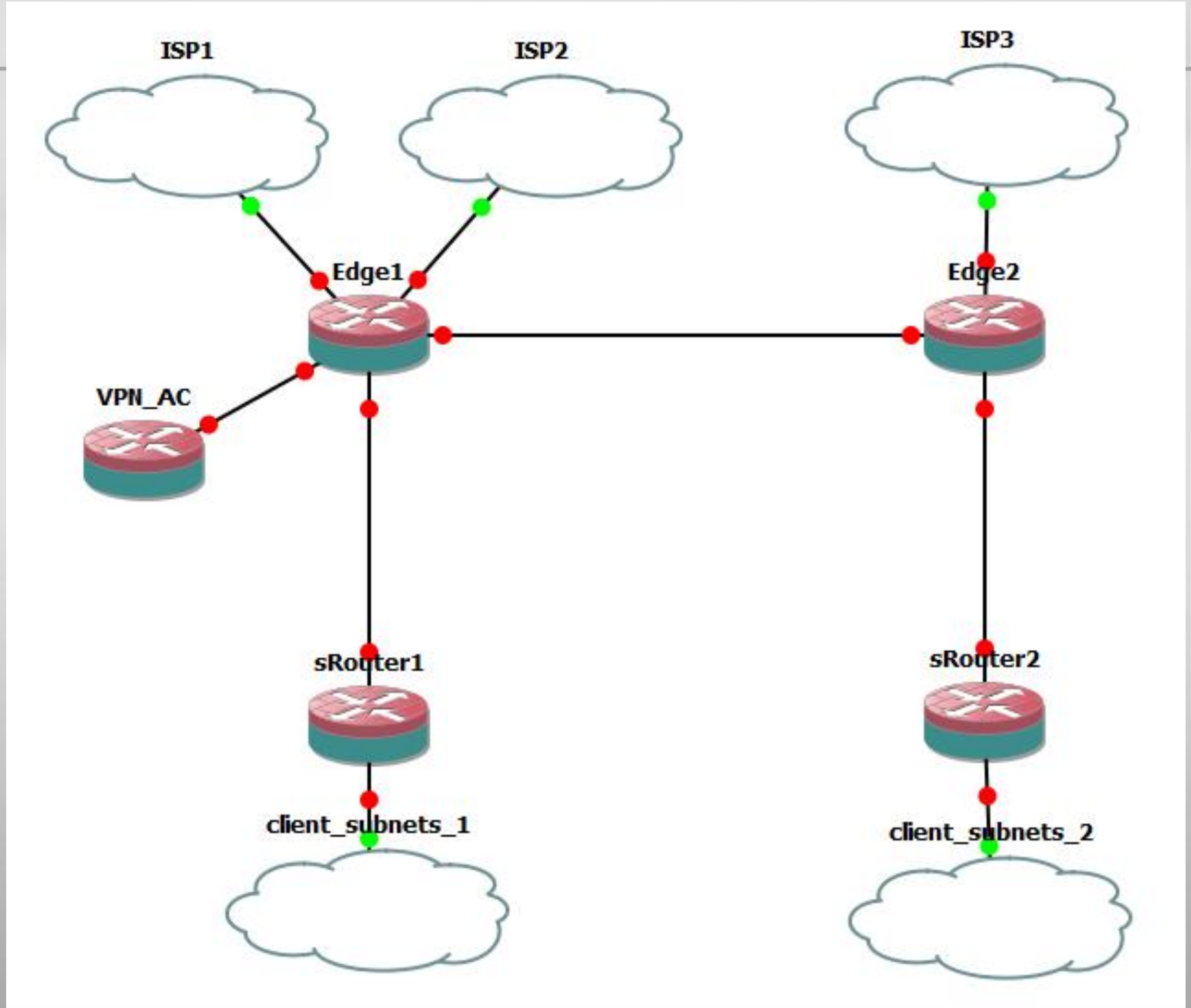
- This is necessary because of NAT
- Your ISP’s security should block you
- And makes your QoS easier

Traffic from WAN to ROS

- One Edge1:

/ip firewall mangle

```
add action=mark-connection chain=input comment="WAN -> ROS" connection-mark=no-mark in-interface="ether1 - ISP1" new-connection-mark=ISP1->ROS passthrough=no
add action=mark-routing chain=output connection-mark=ISP1->ROS new-routing-mark=ISP1 passthrough=no
add action=mark-connection chain=input connection-mark=no-mark in-interface="ether2.pppoe1 - ISP2" new-connection-mark=ISP2->ROS passthrough=no
add action=mark-routing chain=output connection-mark=ISP2->ROS new-routing-mark=ISP2 passthrough=no
add action=mark-connection chain=input comment="Edge2 WAN -> ROS" connection-mark=no-mark in-interface="ether5.vlan1003 - wifi.backbone.local" new-connection-mark=Edge2->ROS passthrough=no
add action=mark-routing chain=output connection-mark=Edge2->ROS new-routing-mark=Edge2 passthrough=no
```


ISP1
ISP2
ISP3
Edge1
Edge2
VPN_AC
sRouter1
sRouter2
client_subnets_1
client_subnets_2

Traffic from WAN to ROS

- One Edge2:

/ip firewall mangle

```
add action=mark-connection chain=input comment="WAN -> ROS" connection-mark=no-mark in-interface=ether5 new-connection-mark=ISP3->ROS passthrough=no
add action=mark-routing chain=output connection-mark=ISP3->ROS new-routing-mark=ISP3 passthrough=no
```

```
add action=mark-connection chain=input connection-mark=no-mark in-interface="br1.vlan1003 - wifi.backbone.local" comment="Edge1 WAN->ROS" new-connection-mark=Edge1->ROS passthrough=no
add action=mark-routing chain=output connection-mark=Edge1->ROS new-routing-mark=Edge1 passthrough=no
```

Traffic from WAN to LANs

- Traffic from WAN to our routers is not good
- Same principle also applies to traffic from WAN to our LANs tho
- Lets fix that too

Traffic from WAN to LANs

- One Edge1:

/ip firewall mangle

add action=mark-connection chain=forward comment="WAN -> LANs" connection-mark=no-mark in-interface="ether1 - ISP1" new-connection-mark=ISP1->LANs passthrough=no

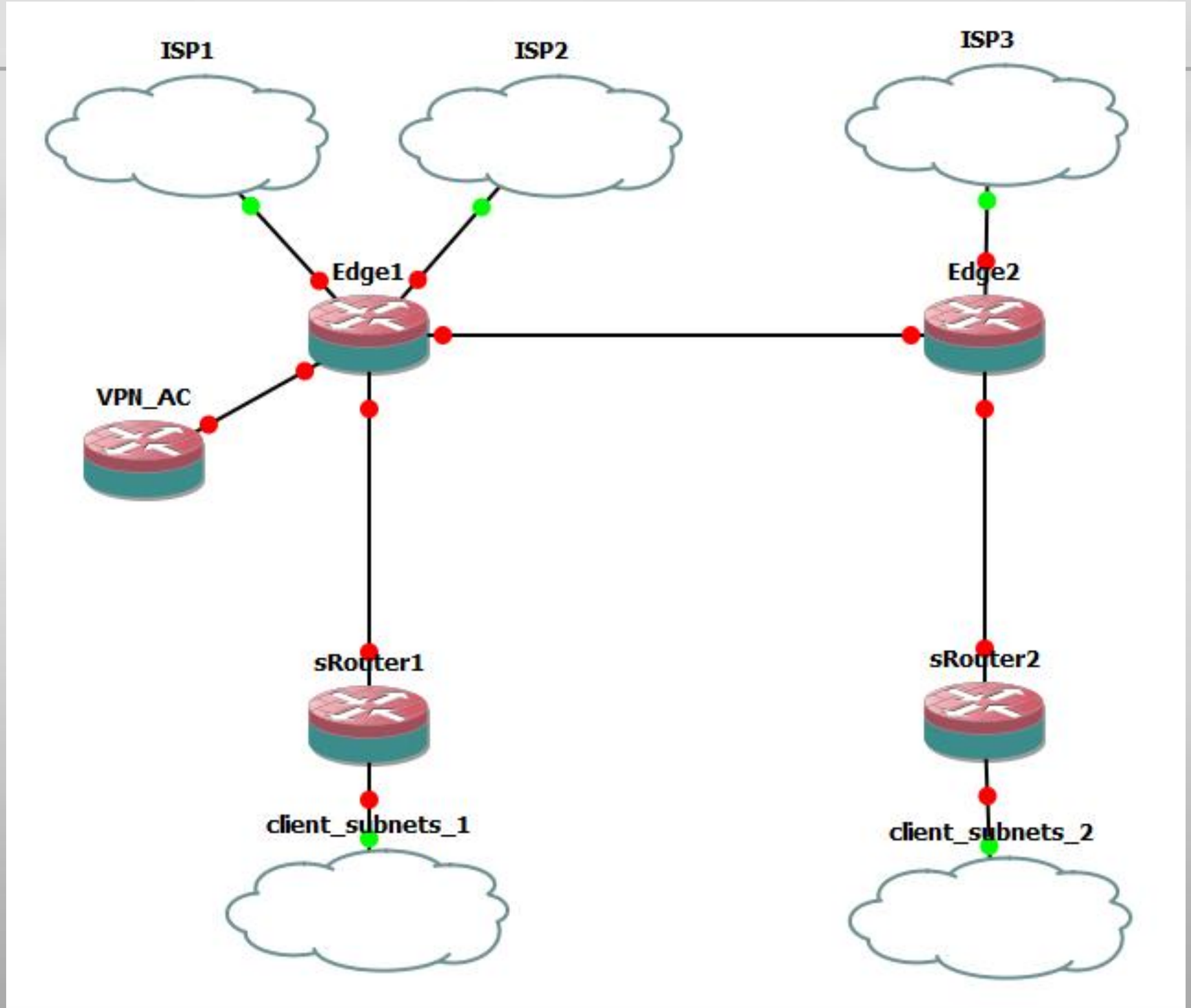
add action=mark-routing chain=prerouting connection-mark=ISP1->LANs new-routing-mark=ISP1 passthrough=no src-address-list=Private-Networks

add action=mark-connection chain=forward connection-mark=no-mark in-interface="ether2.pppoe1 - ISP2" new-connection-mark=ISP2->LANs passthrough=no

add action=mark-routing chain=prerouting connection-mark=ISP2->LANs new-routing-mark=ISP2 passthrough=no src-address-list=Private-Networks

add action=mark-connection chain=forward comment="Edge2 -> LAN" connection-mark=no-mark in-interface="ether5.vlan1003 - wifi.backbone.local" new-connection-mark=Edge2->LANs passthrough=no

add action=mark-routing chain=prerouting connection-mark=Edge2->LANs new-routing-mark=Edge2 passthrough=no src-address-list=Private-Network


ISP1
ISP2
ISP3
Edge1
Edge2
VPN_AC
sRouter1
sRouter2
client_subnets_1
client_subnets_2

Traffic from WAN to ROS

- One Edge2:

/ip firewall mangle

```
add action=mark-connection chain=forward comment="WAN -> LANs" connection-mark=no-mark in-interface=ether5 new-connection-mark=ISP3->LANs passthrough=no
add action=mark-routing chain=prerouting connection-mark=ISP3->LANs new-routing-mark=ISP3 passthrough=no src-address-list=Private-Networks
```

```
add action=mark-connection chain=forward connection-mark=no-mark in-interface="br1.vlan1003 - wifi.backbone.local" new-connection-mark=Edge1->LANs passthrough=no
add action=mark-routing chain=prerouting connection-mark=Edge1->LANs new-routing-mark=Edge1 passthrough=no src-address-list=Private-Networks
```

Done with the basics

- So at this point, we are done with the basics
- If you are not asleep right now, good job
- Also at this point, you are probably thinking:

"FFFFFUUUUUUUU how am I ever gonna get through all that code"

- Confused RouterOS user

Finishing up

- Lets finish with actual client traffic routing based on DSCP tag
- Edge1:

```
/ip firewall mangle
add action=mark-connection chain=prerouting comment="LAN -> WAN" connection-mark=no-mark dst-address-list=!Private-Networks dst-address-type=!local new-connection-mark=LAN->WAN src-address-list=Private-Networks
add action=mark-routing chain=prerouting comment="DSCP 51 to ISP1" connection-mark=LAN->WAN dscp=51 new-routing-mark=ISP1 src-address-list=Private-Networks
add action=mark-routing chain=prerouting comment="DSCP 52 to ISP2" connection-mark=LAN->WAN dscp=52 new-routing-mark=ISP2 src-address-list=Private-Networks
add action=mark-routing chain=prerouting comment="DSCP 53 to ISPS3" connection-mark=LAN->WAN dscp=53 new-routing-mark=Edge2 src-address-list=Private-Networks
```

Finishing up

- Lets finish with actual client traffic routing based on DSCP tag
- Edge2:

```
/ip firewall mangle
add action=mark-connection chain=prerouting comment="LAN -> WAN" connection-mark=no-mark dst-address-list=!Private-Networks dst-address-type=!local new-connection-mark=LAN->WAN src-address-list=Private-Networks
add action=mark-routing chain=prerouting comment="DSCP 51 to ISP1" connection-mark=LAN->WAN dscp=51 new-routing-mark=Edge1 src-address-list=Private-Networks
add action=mark-routing chain=prerouting comment="DSCP 52 to ISP2" connection-mark=LAN->WAN dscp=52 new-routing-mark=Edge1 src-address-list=Private-Networks
add action=mark-routing chain=prerouting comment="DSCP 53 to ISPS3" connection-mark=LAN->WAN dscp=53 new-routing-mark=ISP3 src-address-list=Private-Networks
```

Now for routing tables

- Now lets fix the routing tables on both routers:
- Edge1:

/ip route

add distance=1 gateway=x.x.x.x routing-mark=ISP1

add distance=1 gateway=y.y.y.y routing-mark=ISP2

add distance=1 gateway=z.z.z.z routing-mark=Edge2

- Edge2:

/ip route

add distance=1 gateway=t.t.t.t routing-mark=Edge1

add distance=1 gateway=v.v.v.v routing-mark=ISP3

Now for routing tables

- Now edges are completely configured and ready to load-balance based on a DSCP mark
- So now lets do PCC load balancing on subnet terminating routers
- We will be assigning DSCPs with PCC

Now for balancing on subnet routers

/ip firewall mangle

```
add action=change-dscp chain=forward comment="PCC load balancing" dst-address-list=!LANs new-dscp=51 passthrough=no per-connection-classifier=both-addresses:5/0 src-address-list=LANs
add action=change-dscp chain=forward dst-address-list=!LANs new-dscp=52 passthrough=no per-connection-classifier=both-addresses:5/1 src-address-list=LANs
add action=change-dscp chain=forward dst-address-list=!LANs new-dscp=52 passthrough=no per-connection-classifier=both-addresses:5/2 src-address-list=LANs
add action=change-dscp chain=forward dst-address-list=!LANs new-dscp=53 passthrough=no per-connection-classifier=both-addresses:5/3 src-address-list=LANs
add action=change-dscp chain=forward dst-address-list=!LANs new-dscp=53 passthrough=no per-connection-classifier=both-addresses:5/4 src-address-list=LANs
```

- This balances our ISPs with a 1:2:2 ratio

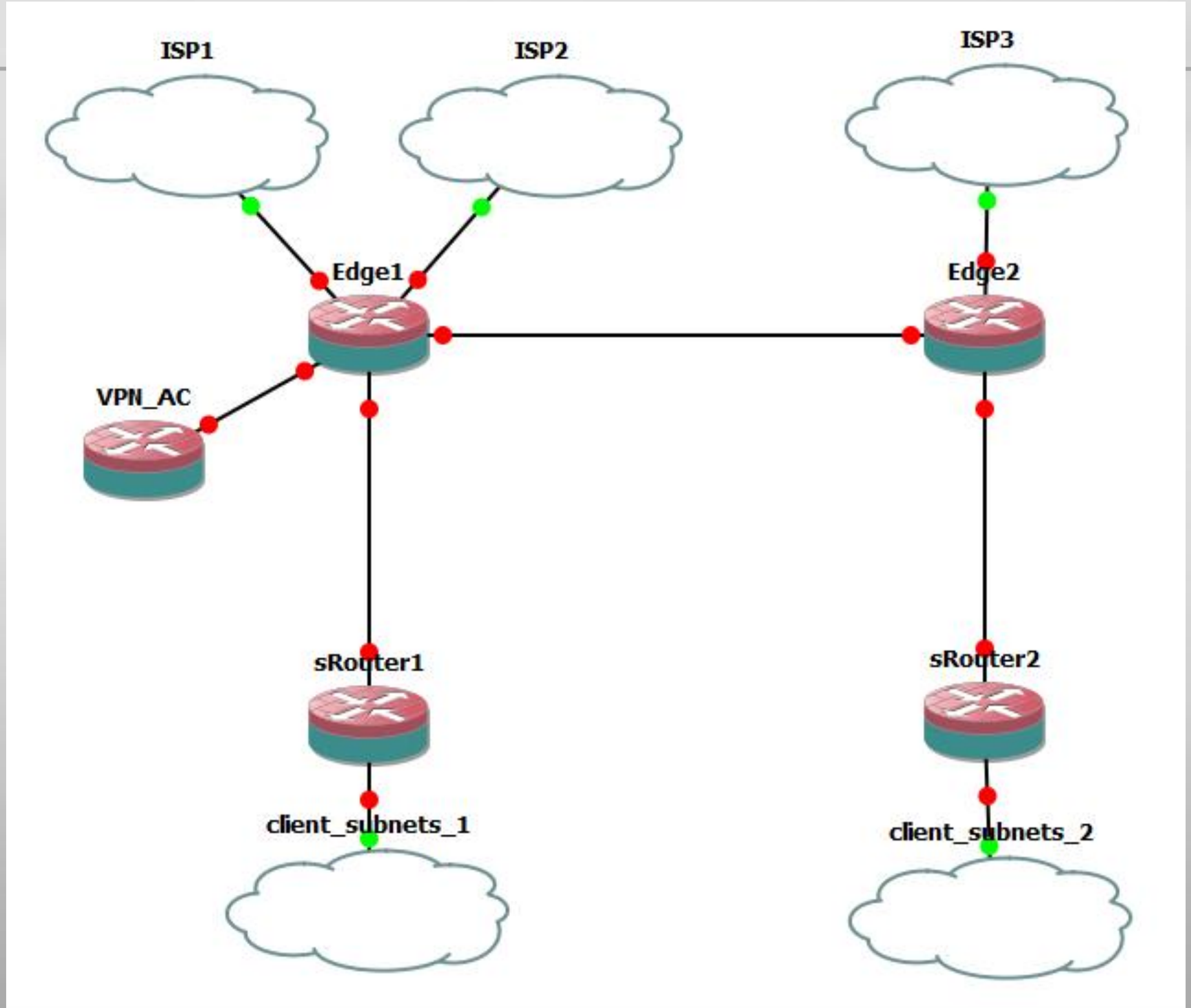
Little note on last config

- Watch out for the address list in the last commands as well
- Fill it with your local subnets behind the subnet terminating routers

Final result

- Connection from WAN to our routers and services inside of our LAN will work and respond back correctly

- Any connection from inside of our LANs will be balanced across all ISPs with a 1:2:2 ratio


ISP1
ISP2
ISP3
Edge1
Edge2
VPN_AC
sRouter1
sRouter2
client_subnets_1
client_subnets_2

Scenario 2

Where is this useful?

- You can provide our-of-country traffic termination to your customers
- This can be used to get over geographic content blocks

- Your customers can decide what traffic to send out-of-country (because of latency etc.)

Let customers decide where their traffic flows

- With routing choice being based on DSCP, you can let customers choose which routing path to take

- Customers can source their traffic with a certain DSCP, and therefore influence where your infrastructure will route that traffic

Disclaimer

- I am not a lawyer, and I cant tell you if this is legal for you or not.

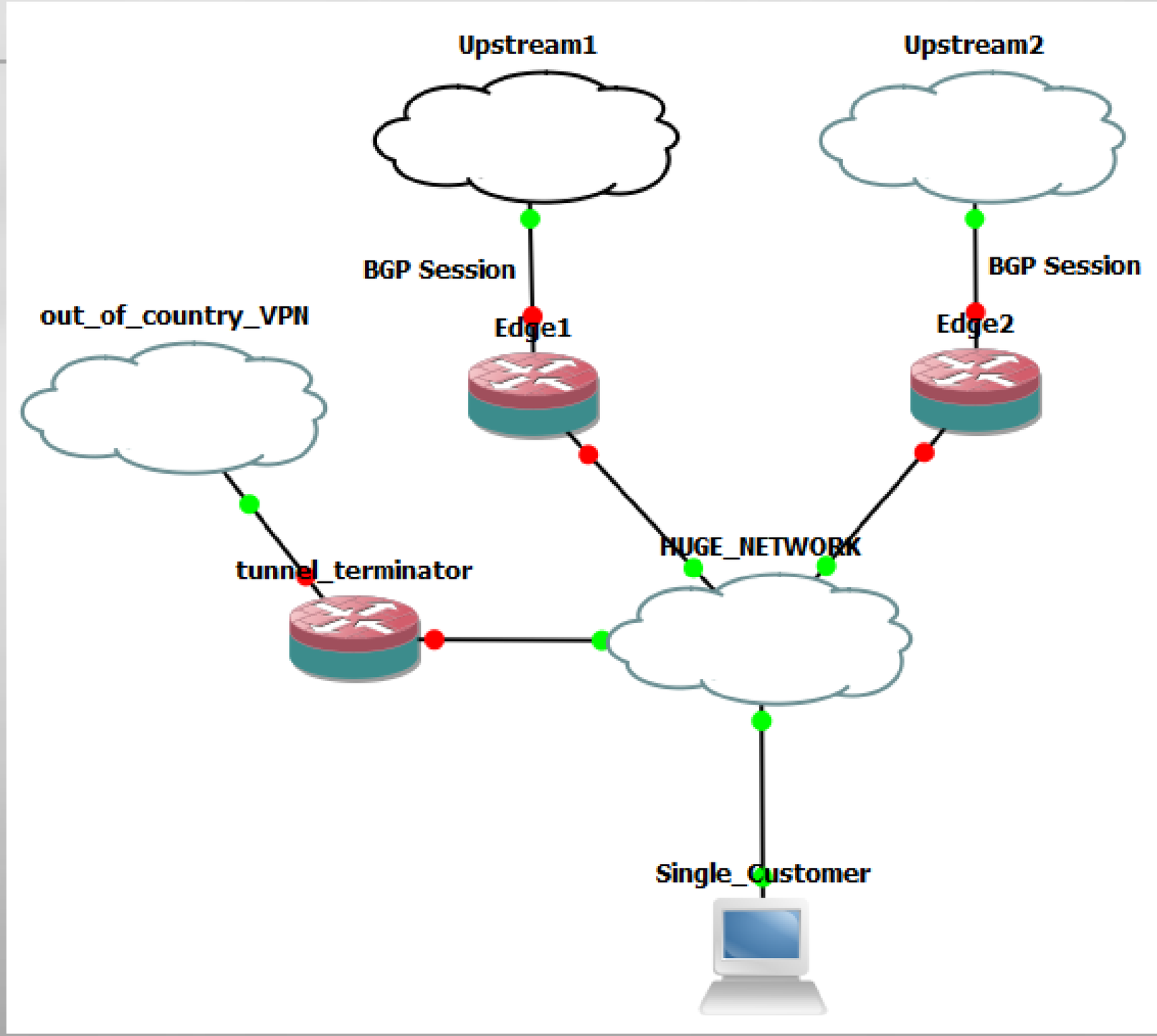
- Please find out if this is legal or not for you.
- Please find out if you are not breaking any local laws or EULA or ToS'es etc…

- Am I not in any way responsive for how you use this

ELEMENTS

Topology

- Customer can send traffic with certain DSCP
- That traffic will leave through out-of-country VPN



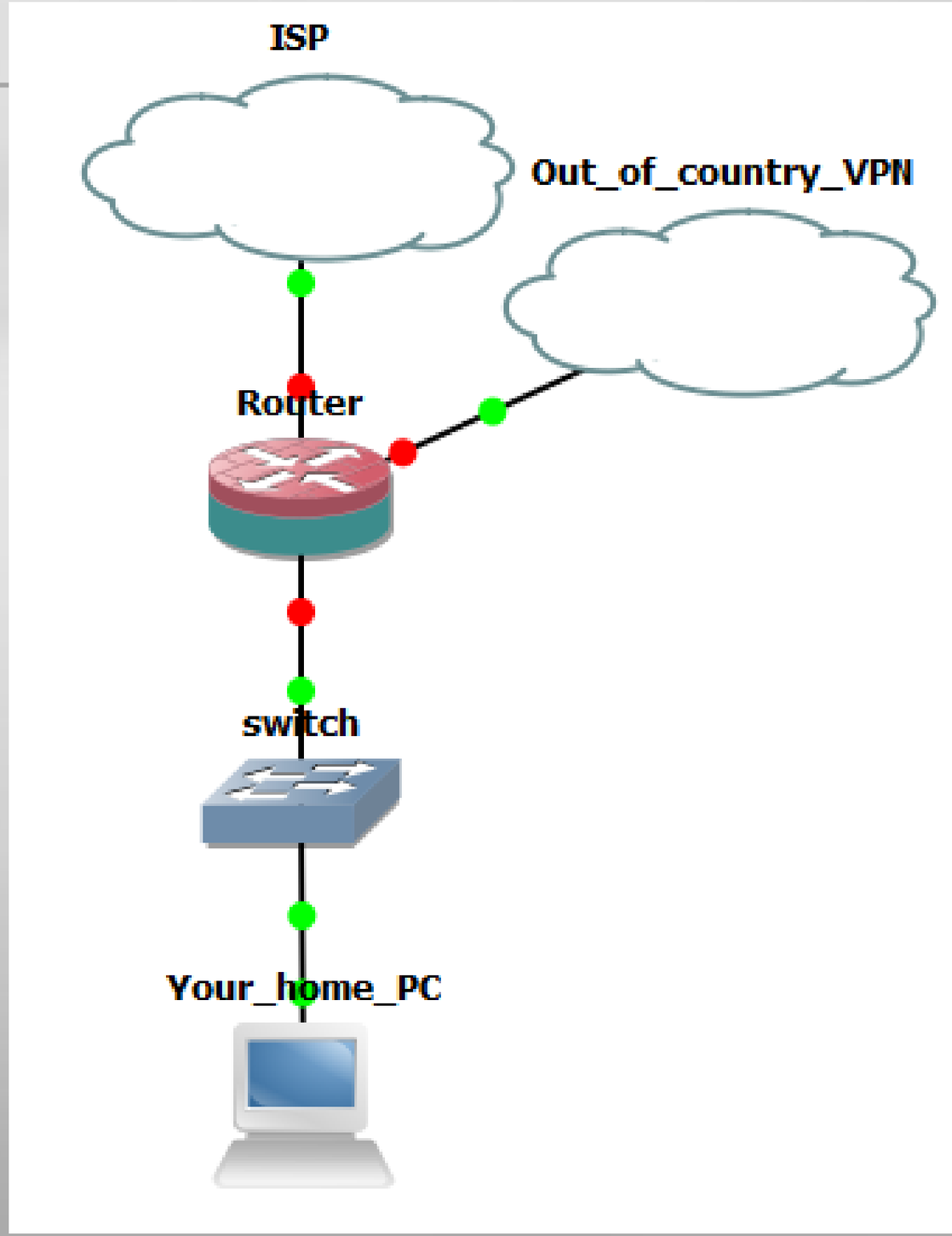
Example usage

- Certain web service are available to US-based IP addresses only
- You can buy a US VPN (there are many providers)
- You can then send traffic with a certain DSCP through the US VPN tunnel, so to those web service, you will appear as a US customer, therefore avoiding the country-based block
- !! See the disclaimer !!

Example usage

- Same situation for your home usage

- The config is the same here, send traffic with certain DSCP to out-of-country VPN tunnel



So why do this?

- You can set a single web-browser to be a “US web browser”
- Windows (and other OSs) allow you to send traffic out with difference DSCP based on which application originates the traffic
- Also useful for gamers – get on different country servers

So how to do this?

- Same concept as before

/ip firewall mangle

add action=mark-connection chain=prerouting comment="LAN -> WAN" connection-mark=no-mark dst-address-list=!Private-Networks dst-address-type=!local new-connection-mark=LAN->WAN src-address-list=Private-Networks

add action=mark-routing chain=prerouting comment="VPN traffic - based on dscp mark" connection-mark=LAN->WAN dscp=22 new-routing-mark=ooc_VPN src-address-list=Private-Networks

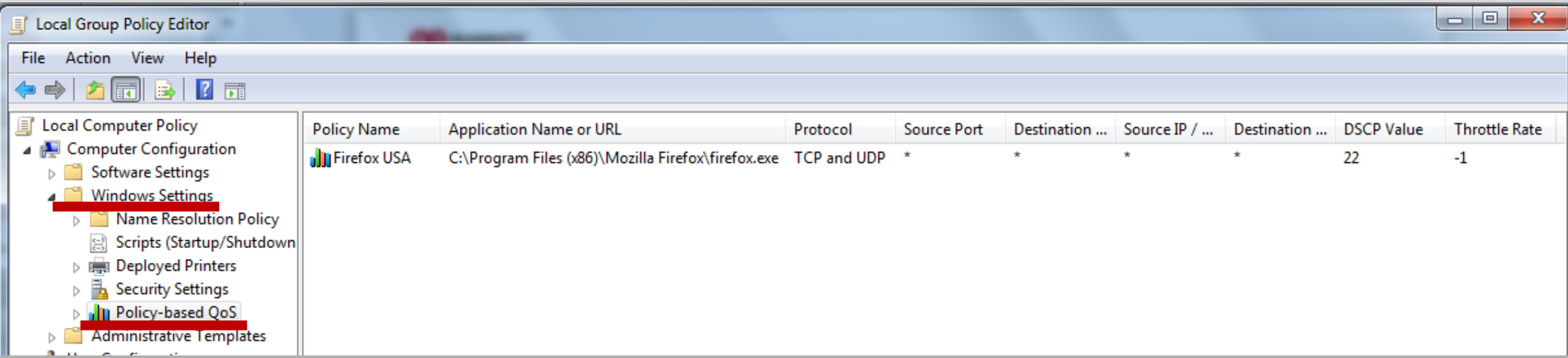
Little note on last config

- Again watch out for the address list in the last commands
- Fill it with your local subnets behind the subnet terminating routers
- This is just to filer outgoing only packets, incoming packets cant be put into the VPN routing table

Client config

- How to configure windows to send only one browser out through the out-of-country VPN (using DSCP)?

- Start > gpedit.msc



Client config

- If using Windows, of course there is a catch (Thanks Microsoft!)

- If your PC is not in AD, or out of an AD network, the QoS setting actually does nothing

- Fix with a registry change:

  http://support.microsoft.com/kb/2733528

  See the “Let me fix it myself” section

Demo

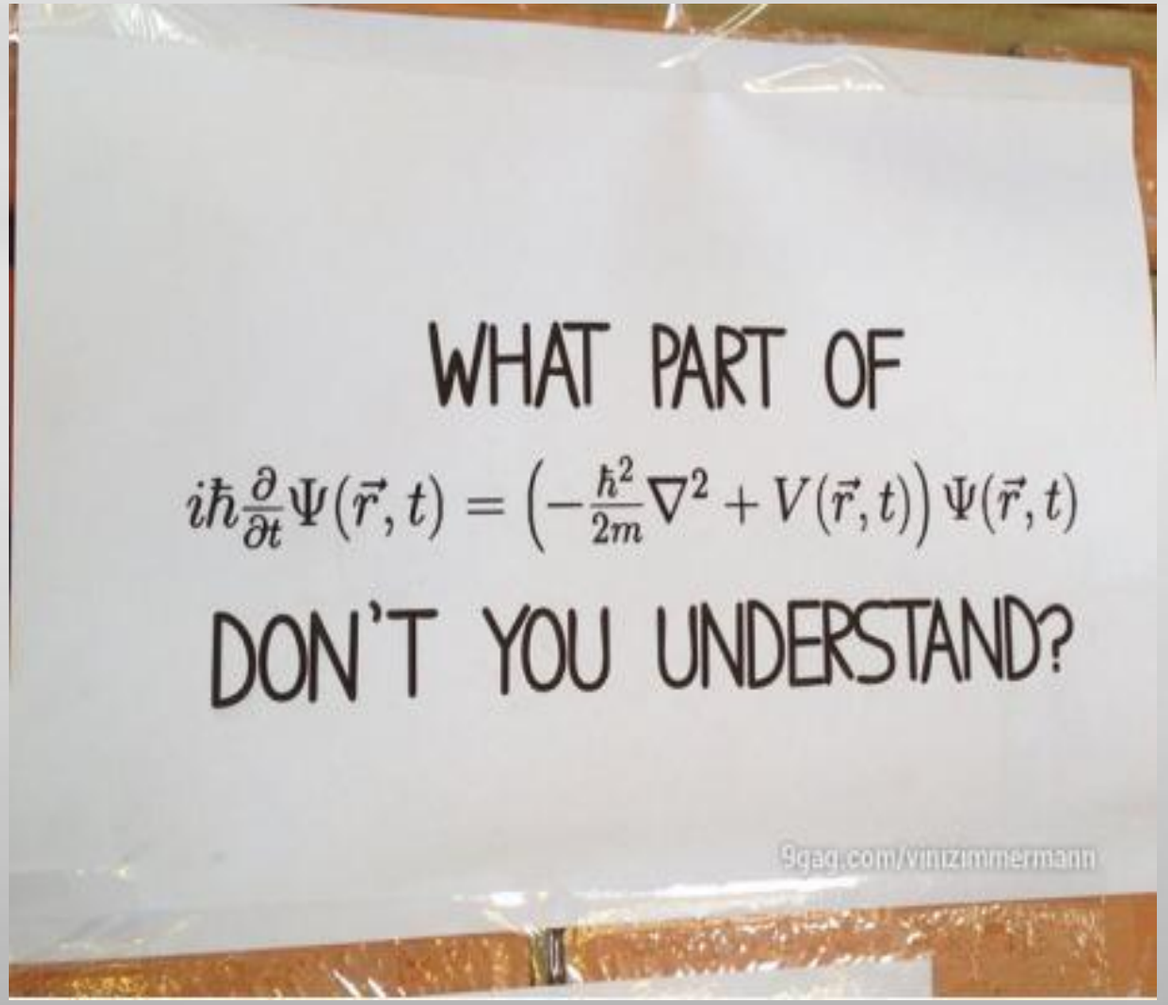
- How to configure?
- How it looks when working correctly.
- Live demo!

That's it!

- If you liked this presentation, more material from me:
  - **Bandwidth-based load-balancing without MPLS TE**
  - **Scalable IPSec (IPSec basics + L2TP /IPSec)**
  - **MPLS for ISPs (MPLS basics + PPPoE over VPLS)**
  - **OSPF to the customer**

- Find the presentations on www.tiktube.com

  Search for "Kirnak"



If you have any questions, please ask now, or find me after the presentation.

Thanks for listening

Tomas Kirnak

kirnak@rfelements.com