MPLS for ISPs – PPPoE over VPLS

MPLS, VPLS, PPPoE

www.atris.sk

Presenter information

Tomas Kirnak

Network design

Security, wireless

Servers

Virtualization

MikroTik Certified Trainer

Atris, Slovakia

Established 1991

Complete IT solutions

Networking, servers

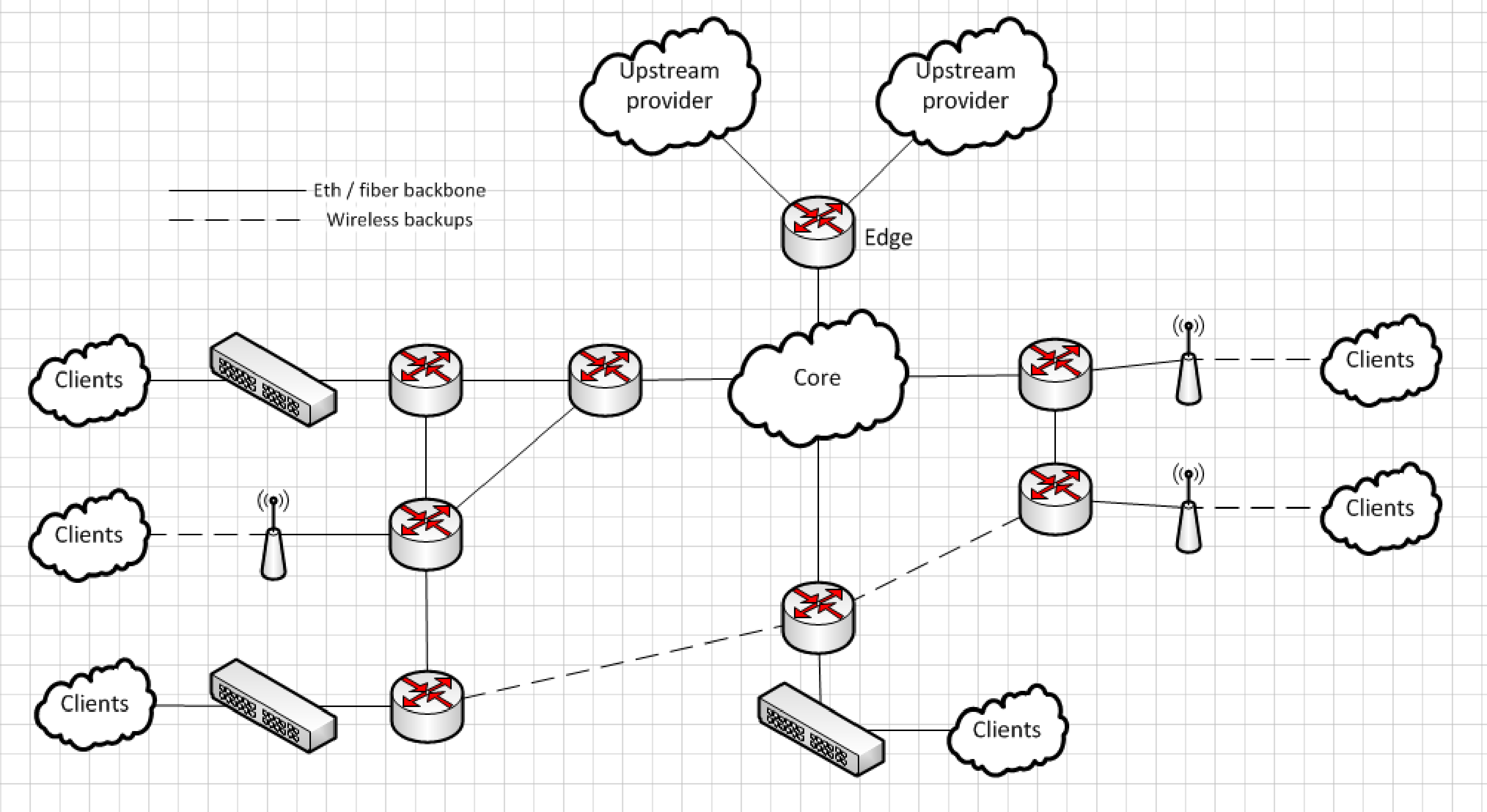
Virtualization

IP security systems

Agenda:

- PPPoE basics and advantages
- MPLS and VPLS
- MTU and MTU calculations
- MPLS PHP and ICMP in MPLS
- Configuring everything
- Tips, tricks, problems

Example provider network



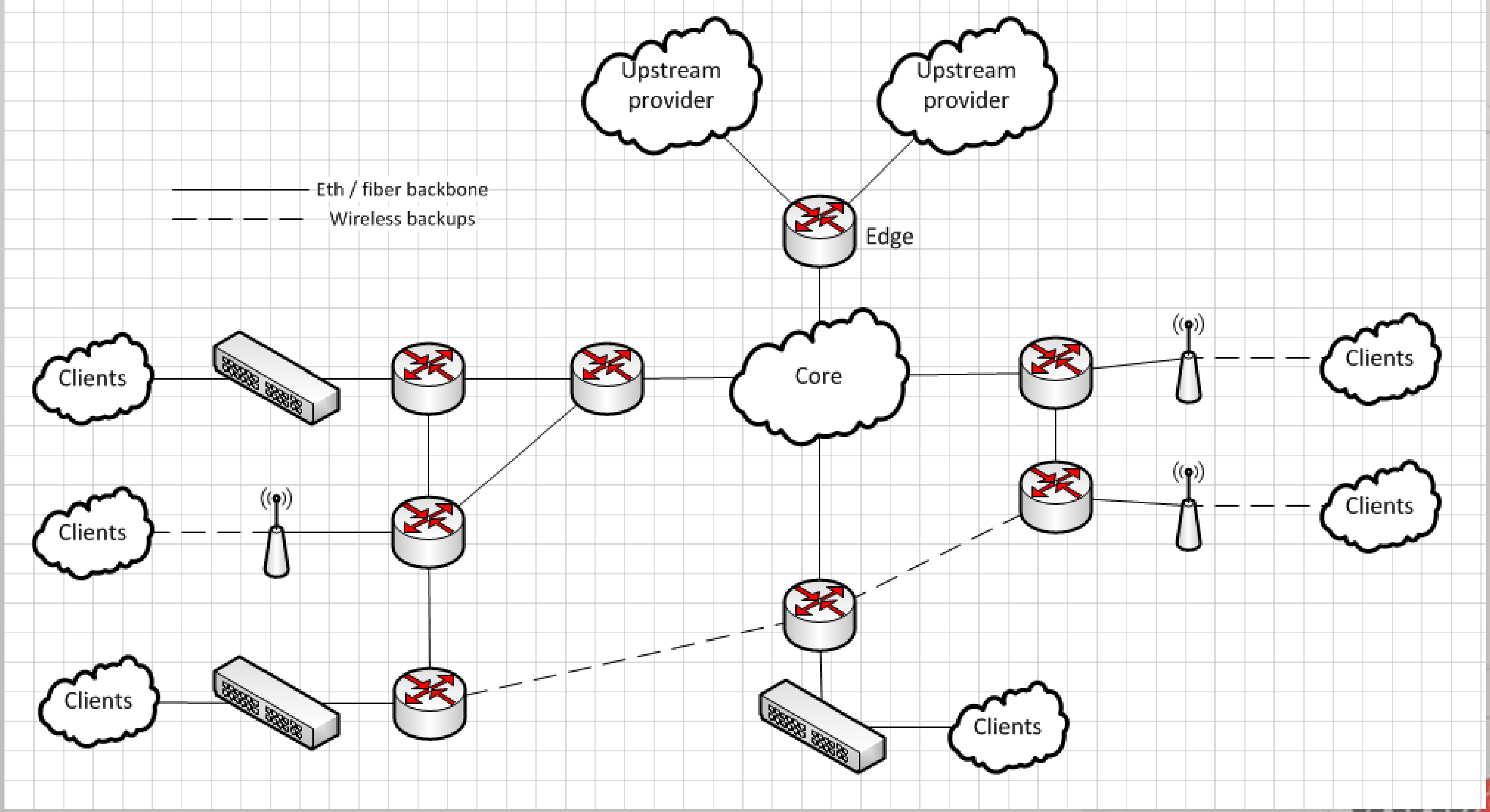
A few assumptions:

- The network is fully routed.
- OSPF is deployed and properly configured.
- Router IDs and loopbacks properly implemented.
- PPPoE is an acceptable delivery method.
- All devices support MPLS and jumbo frames.

Goals:

- Public IP assignment without the need to stretch subnets around the network.
- Conserve public IP space with use of /32s.
- Single point for authentication and accounting.
- Secure and minimize L2 segments.
- New products for customers – L2 and L3 VPNs.

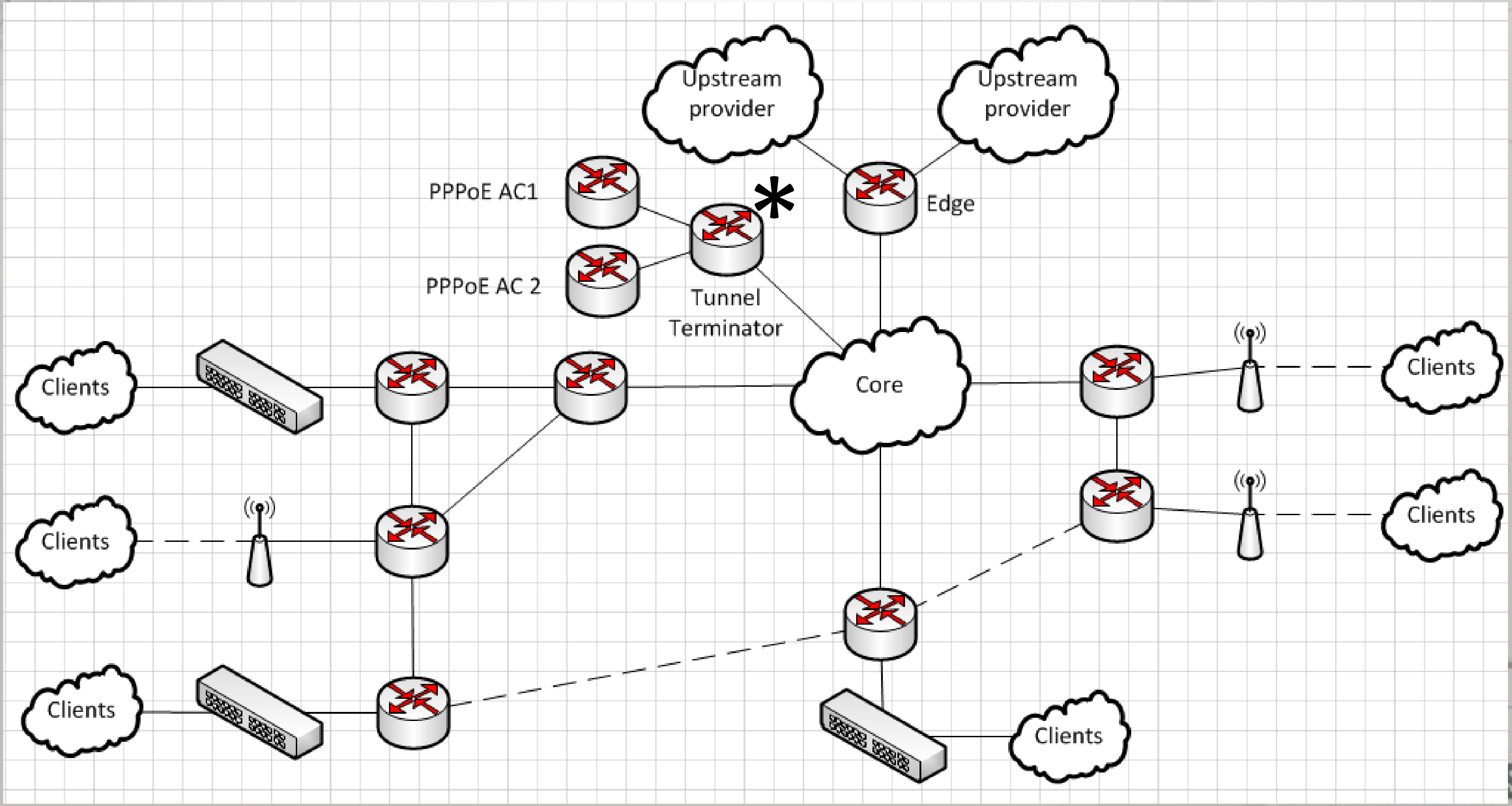
Example provider network



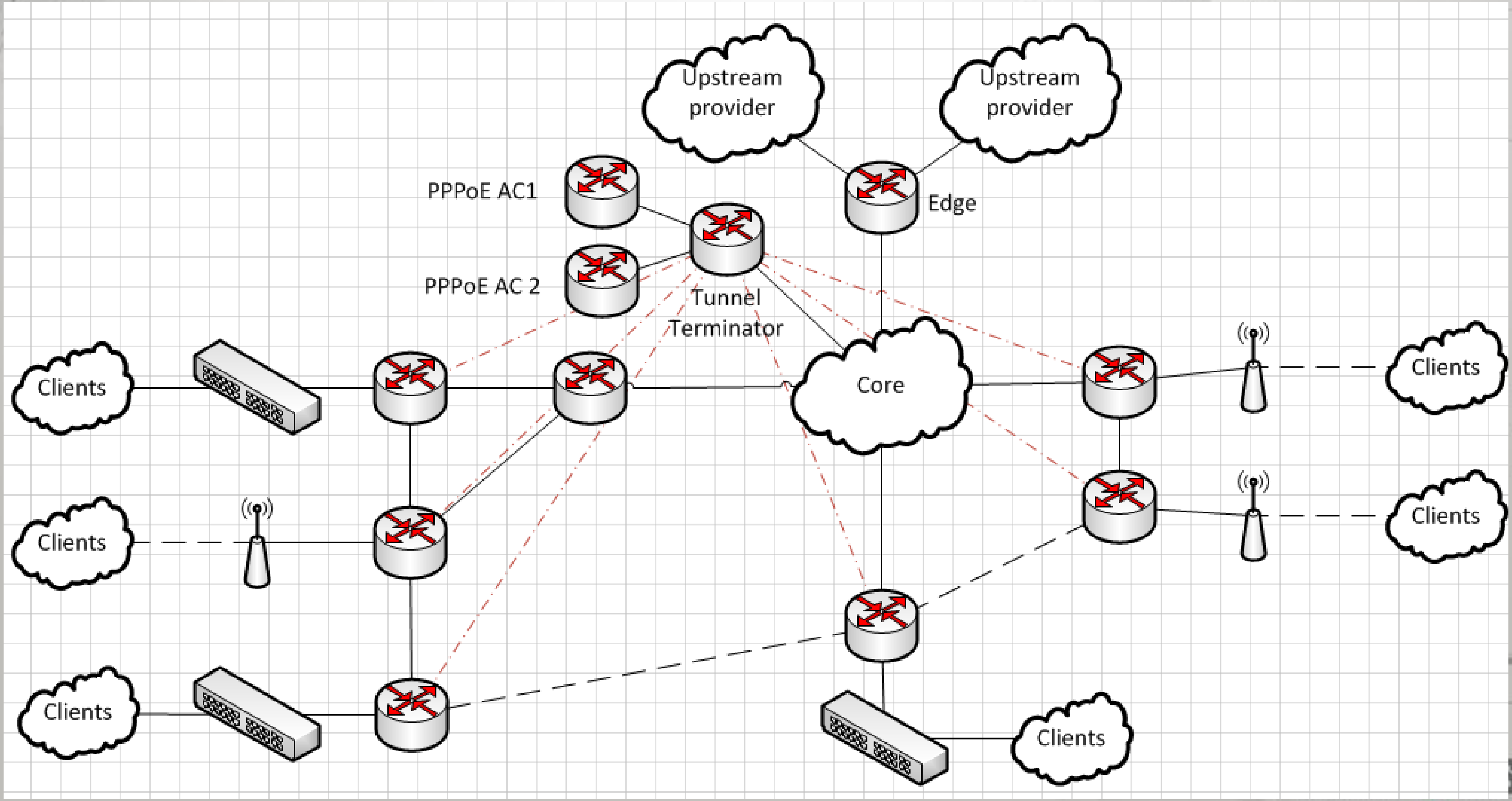
PPPoE Point-to-Point Protocol over Ethernet

- PPPoE builds a point-to-point tunnel between 2 network devices.
- Direct L2 communication between the AC and the client needed to work.
- Since the tunnel is PtP, each client can (should) be its own L2 segment.
- Username/password authentication – Radius.

Accomplishing the goals

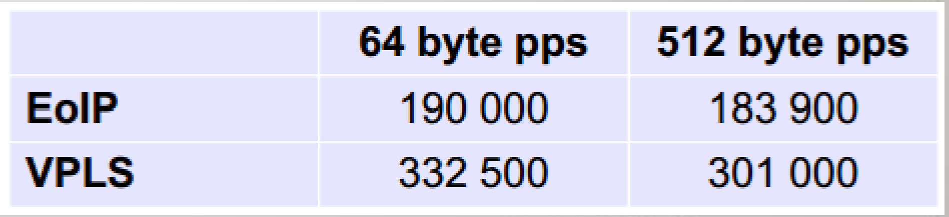


Accomplishing the goals



Stretching L2 over L3

- EoIP could be a solution for tunneling L2 over L3.
- EoIP disadvantages:
  - Fragmentation of L2 frames over multiple L3 packets
  - Performance issues
- VPLS advantages:
  - No fragmentation.
  - 60% more performance then EoIP.



	64 byte pps	512 byte pps
EoIP	190 000	183 900
VPLS	332 500	301 000

VPLS Virtual Private LAN Service

- VPLS is a method of creating transparent L2 tunnels based on MPSL signaling.

- A VPLS tunnel is presented as a separate interface to the router (same as EoIP)

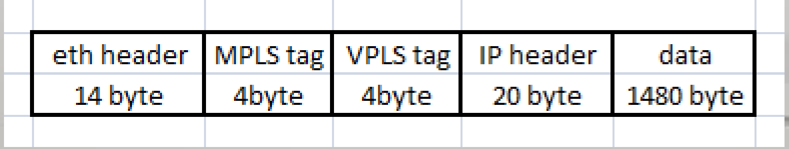
- VPLS tunnel adds one VPLS tag to the MPLS frame.

MPLS Multi-Protocol Label Switching

- In a MPLS network, each data frame is assigned a label.
- Packet-forwarding (switching) decisions are made solely on the contents of this label – no need to examine the packet itself.
- Speed benefit, since no IP routing table lookup is performed.

MPLS and label switching

- MPLS is considered a L2.5 protocol – it falls between L2 and L3.
- MPLS tags – tags are added between L2 and L3 headers
- A VPLS tag is one of multiple possible MPLS tag types



eth header	MPLS tag	VPLS tag	IP header	data
14 byte	4byte	4byte	20 byte	1480 byte

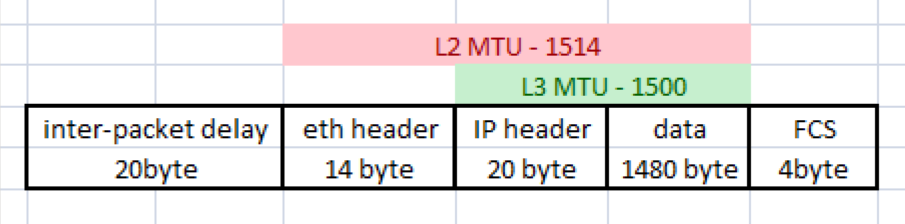
MTU Maximum Transmission Unit

- Defines the maximum byte-size of a frame that the device can handle.
- Frames larger then maximum allowed MTU are silently discarded.

- No ICMP or any other kind of error are produced, the frame is dropped without notice.

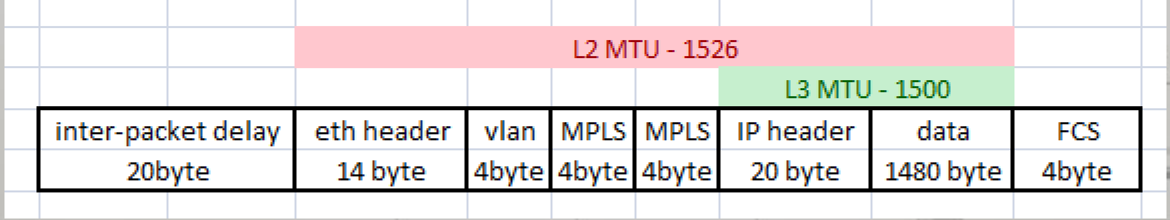
MTU Maximum Transmit Unit

- A normal frame for a switch/router



	L2 MTU - 1514			
		L3 MTU - 1500		
inter-packet delay	eth header	IP header	data	FCS
20byte	14 byte	20 byte	1480 byte	4byte

- MPLS frame inside a vlan



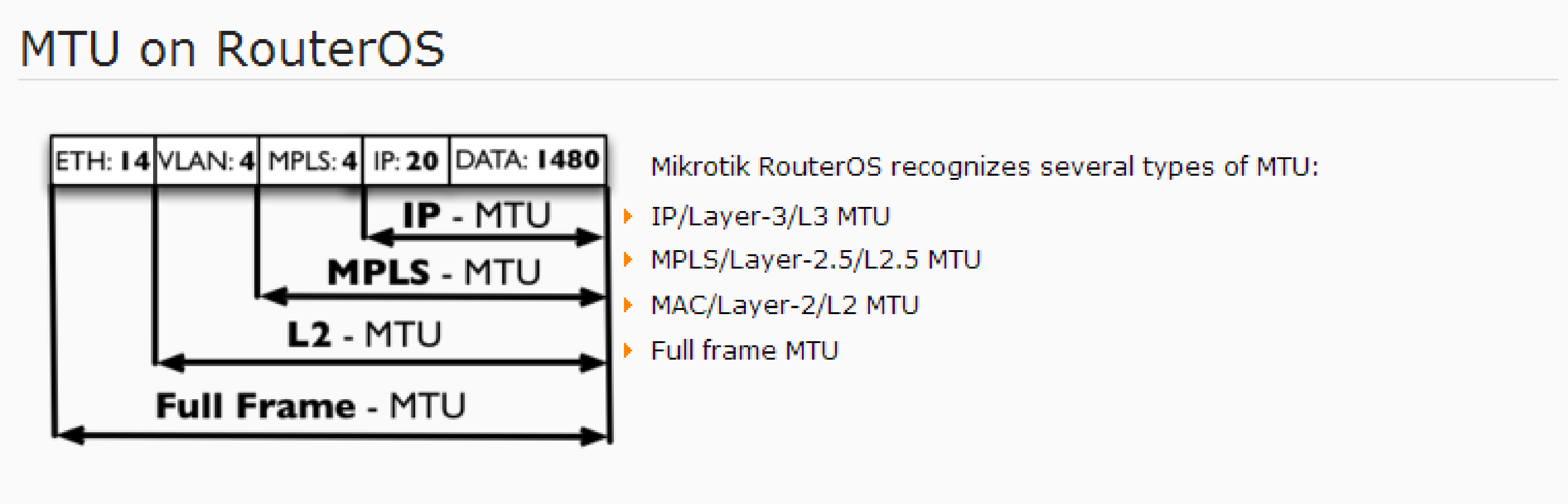
	L2 MTU - 1526						
					L3 MTU - 1500		
inter-packet delay	eth header	vlan	MPLS	MPLS	IP header	data	FCS
20byte	14 byte	4byte	4byte	4byte	20 byte	1480 byte	4byte

MTU on switches/routers

- Cheap/unmanaged switches usually only support L2MTU of 1514.
- On many switches you have to turn on “Jumbo Frames” to enable support for MTU over 1514

- Make sure your L2 infrastructure wont drop your MPLS frames, this is the biggest and most common problem with integrating MPLS.

Update for MikroTik



MTU on RouterOS

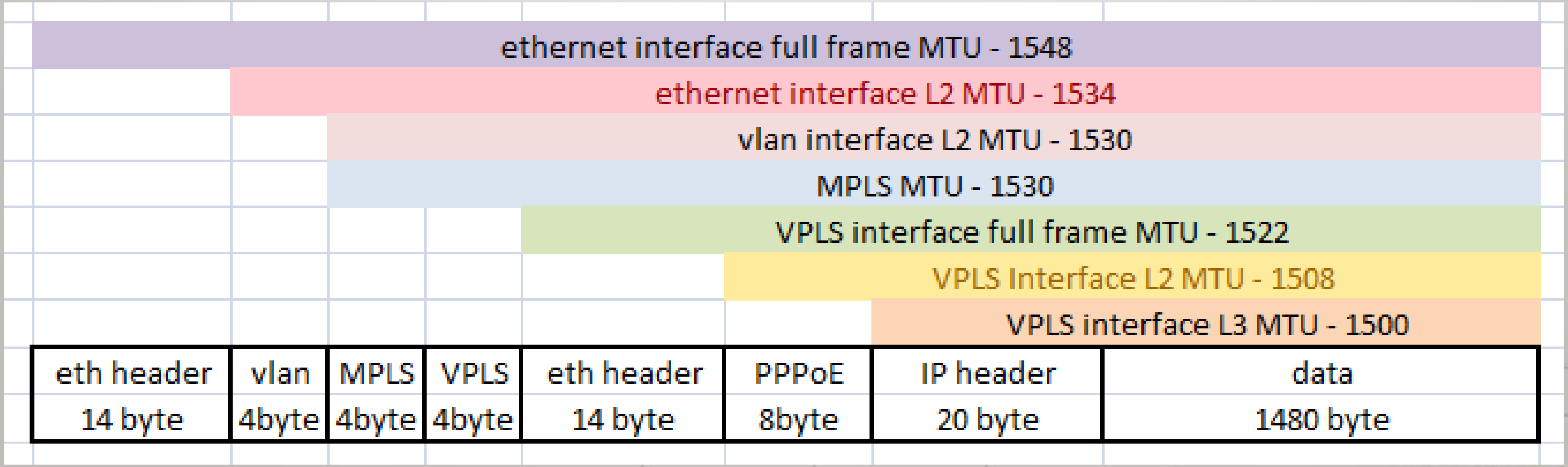
Mikrotik RouterOS recognizes several types of MTU:

- IP/Layer-3/L3 MTU
- MPLS/Layer-2.5/L2.5 MTU
- MAC/Layer-2/L2 MTU
- Full frame MTU

Check how your switch vendor defines L2MTU to avoid confusion and problems.

PPPoE over VPLS frames

- Our goal is to implement PPPoE over VPLS.
- We want full 1500 L3 MTU for our clients.



ethernet interface full frame MTU - 1548							
	ethernet interface L2 MTU - 1534						
		vlan interface L2 MTU - 1530					
		MPLS MTU - 1530					
				VPLS interface full frame MTU - 1522			
					VPLS Interface L2 MTU - 1508		
						VPLS interface L3 MTU - 1500	
eth header	vlan	MPLS	VPLS	eth header	PPPoE	IP header	data
14 byte	4byte	4byte	4byte	14 byte	8byte	20 byte	1480 byte

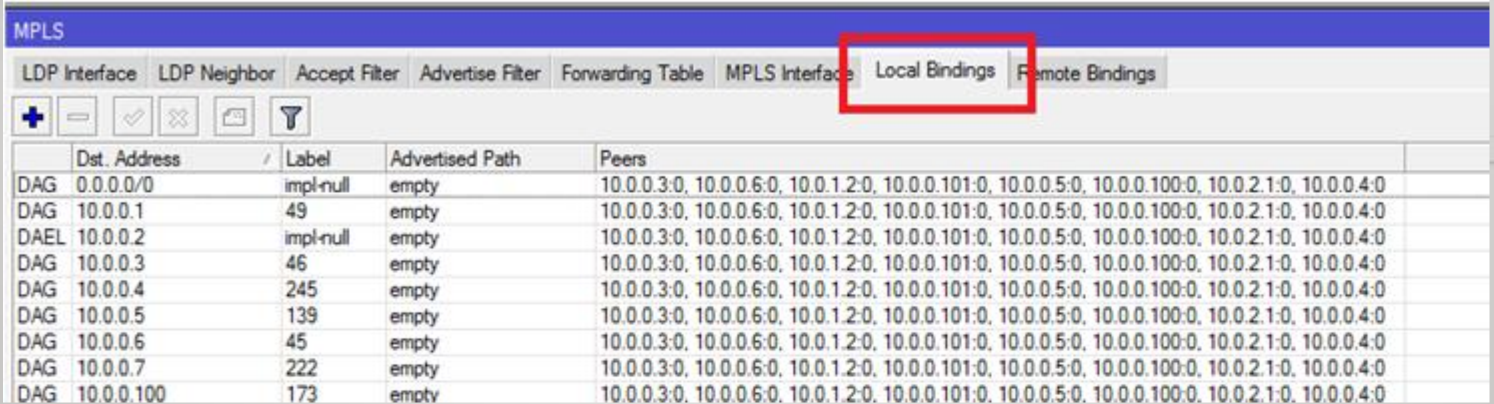
PPPoE over VPLS frames 2

Wireshark frame example.

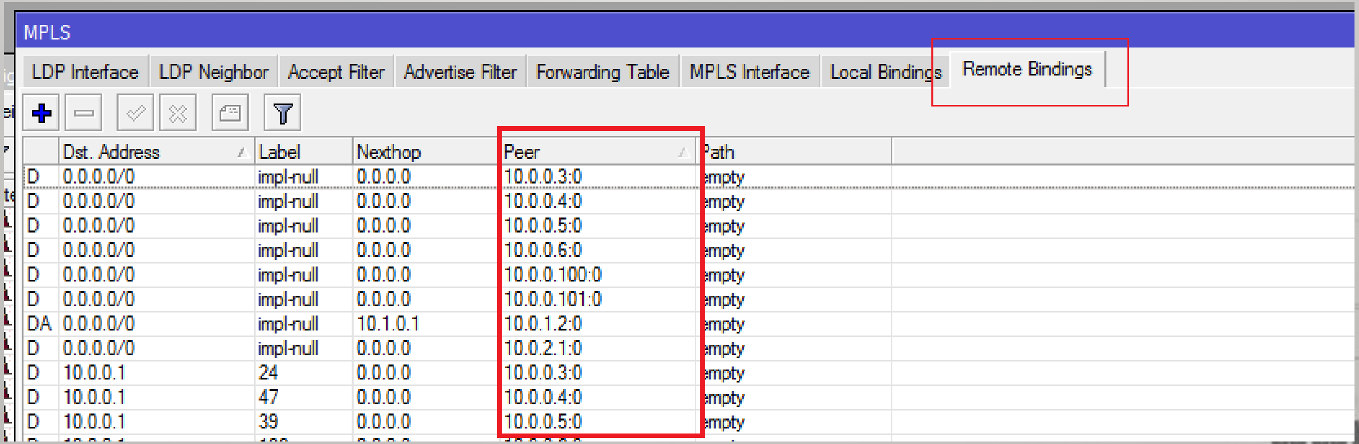
MPLS basics

- Router:
  - Assigns a separate label to each prefix in the routing table
  - Tells its peers about its label bindings
- MPLS Cloud:
  - Each router in the MPLS Cloud assigns its own label to every prefix in the routers routing table
  - Every MPLS router tells its peers about its label bindings
  - This way, all peers know about each others label bindings

MPLS tables:

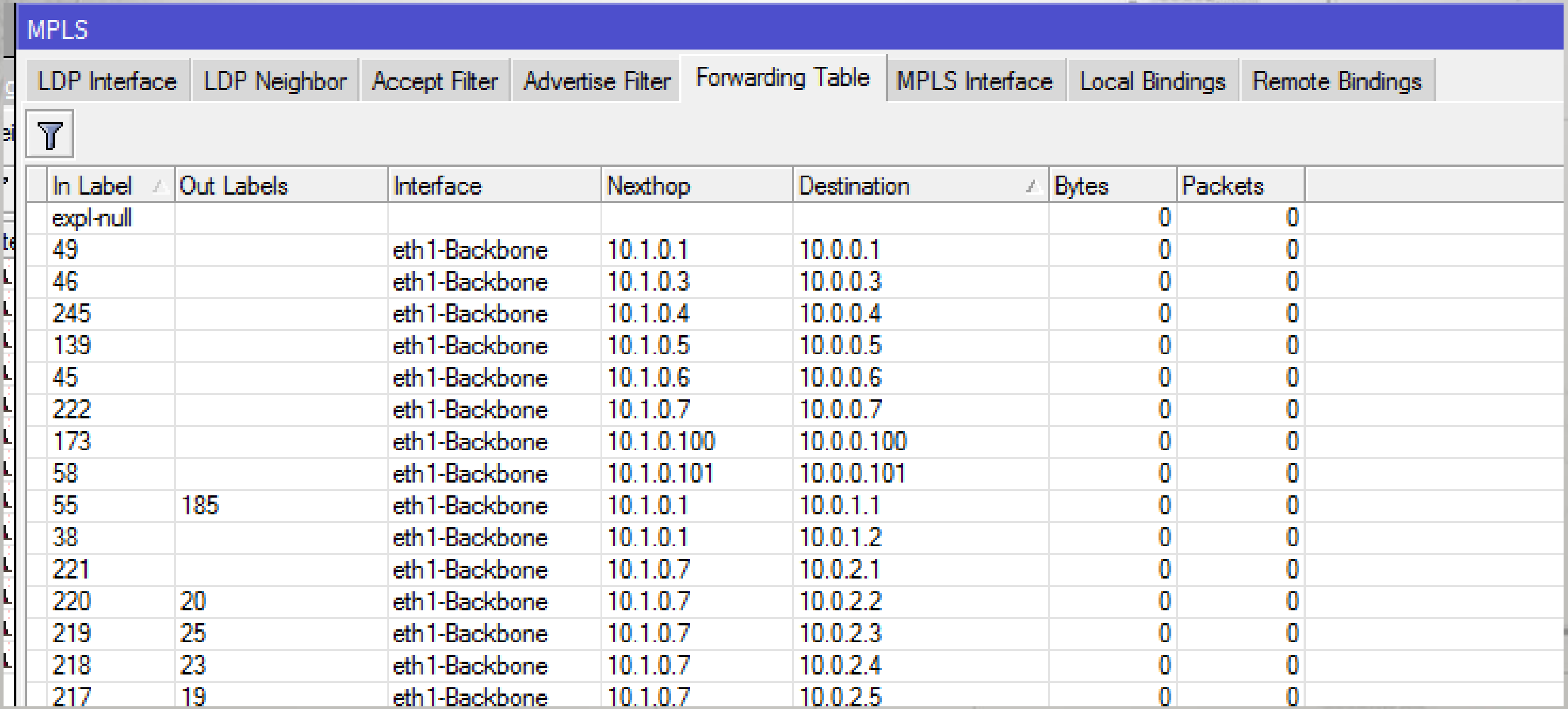


	Dst. Address	Label	Advertised Path	Peers
DAG	0.0.0.0/0	impl-null	empty	10.0.0.3:0, 10.0.0.6:0, 10.0.1.2:0, 10.0.0.101:0, 10.0.0.5:0, 10.0.0.100:0, 10.0.2.1:0, 10.0.0.4:0
DAG	10.0.0.1	49	empty	10.0.0.3:0, 10.0.0.6:0, 10.0.1.2:0, 10.0.0.101:0, 10.0.0.5:0, 10.0.0.100:0, 10.0.2.1:0, 10.0.0.4:0
DAEL	10.0.0.2	impl-null	empty	10.0.0.3:0, 10.0.0.6:0, 10.0.1.2:0, 10.0.0.101:0, 10.0.0.5:0, 10.0.0.100:0, 10.0.2.1:0, 10.0.0.4:0
DAG	10.0.0.3	46	empty	10.0.0.3:0, 10.0.0.6:0, 10.0.1.2:0, 10.0.0.101:0, 10.0.0.5:0, 10.0.0.100:0, 10.0.2.1:0, 10.0.0.4:0
DAG	10.0.0.4	245	empty	10.0.0.3:0, 10.0.0.6:0, 10.0.1.2:0, 10.0.0.101:0, 10.0.0.5:0, 10.0.0.100:0, 10.0.2.1:0, 10.0.0.4:0
DAG	10.0.0.5	139	empty	10.0.0.3:0, 10.0.0.6:0, 10.0.1.2:0, 10.0.0.101:0, 10.0.0.5:0, 10.0.0.100:0, 10.0.2.1:0, 10.0.0.4:0
DAG	10.0.0.6	45	empty	10.0.0.3:0, 10.0.0.6:0, 10.0.1.2:0, 10.0.0.101:0, 10.0.0.5:0, 10.0.0.100:0, 10.0.2.1:0, 10.0.0.4:0
DAG	10.0.0.7	222	empty	10.0.0.3:0, 10.0.0.6:0, 10.0.1.2:0, 10.0.0.101:0, 10.0.0.5:0, 10.0.0.100:0, 10.0.2.1:0, 10.0.0.4:0
DAG	10.0.0.100	173	empty	10.0.0.3:0, 10.0.0.6:0, 10.0.1.2:0, 10.0.0.101:0, 10.0.0.5:0, 10.0.0.100:0, 10.0.2.1:0, 10.0.0.4:0



	Dst. Address	Label	Nexthop	Peer	Path
D	0.0.0.0/0	impl-null	0.0.0.0	10.0.0.3:0	empty
D	0.0.0.0/0	impl-null	0.0.0.0	10.0.0.4:0	empty
D	0.0.0.0/0	impl-null	0.0.0.0	10.0.0.5:0	empty
D	0.0.0.0/0	impl-null	0.0.0.0	10.0.0.6:0	empty
D	0.0.0.0/0	impl-null	0.0.0.0	10.0.0.100:0	empty
D	0.0.0.0/0	impl-null	0.0.0.0	10.0.0.101:0	empty
DA	0.0.0.0/0	impl-null	10.1.0.1	10.0.1.2:0	empty
D	0.0.0.0/0	impl-null	0.0.0.0	10.0.2.1:0	empty
D	10.0.0.1	24	0.0.0.0	10.0.0.3:0	empty
D	10.0.0.1	47	0.0.0.0	10.0.0.4:0	empty
D	10.0.0.1	39	0.0.0.0	10.0.0.5:0	empty

After creating the MPLS forwarding table:



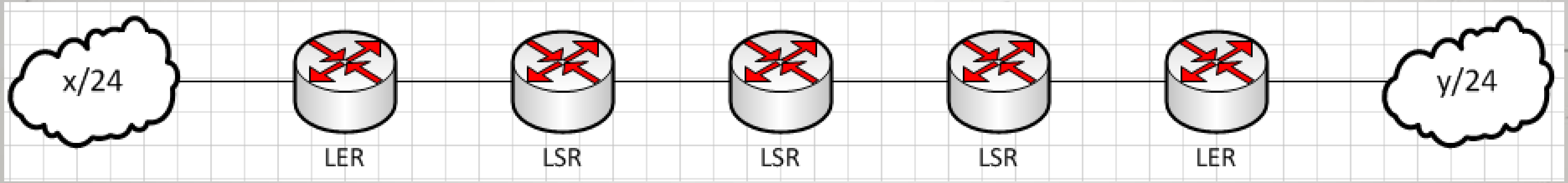
In Label	Out Labels	Interface	Nexthop	Destination	Bytes	Packets
expl-null					0	0
49		eth1-Backbone	10.1.0.1	10.0.0.1	0	0
46		eth1-Backbone	10.1.0.3	10.0.0.3	0	0
245		eth1-Backbone	10.1.0.4	10.0.0.4	0	0
139		eth1-Backbone	10.1.0.5	10.0.0.5	0	0
45		eth1-Backbone	10.1.0.6	10.0.0.6	0	0
222		eth1-Backbone	10.1.0.7	10.0.0.7	0	0
173		eth1-Backbone	10.1.0.100	10.0.0.100	0	0
58		eth1-Backbone	10.1.0.101	10.0.0.101	0	0
55	185	eth1-Backbone	10.1.0.1	10.0.1.1	0	0
38		eth1-Backbone	10.1.0.1	10.0.1.2	0	0
221		eth1-Backbone	10.1.0.7	10.0.2.1	0	0
220	20	eth1-Backbone	10.1.0.7	10.0.2.2	0	0
219	25	eth1-Backbone	10.1.0.7	10.0.2.3	0	0
218	23	eth1-Backbone	10.1.0.7	10.0.2.4	0	0
217	19	eth1-Backbone	10.1.0.7	10.0.2.5	0	0

LDP Label distribution protocol

- LDP allows the routers to learn the label bindings of their peers.

- LDP runs over IP protocol, UDP and TCP 646

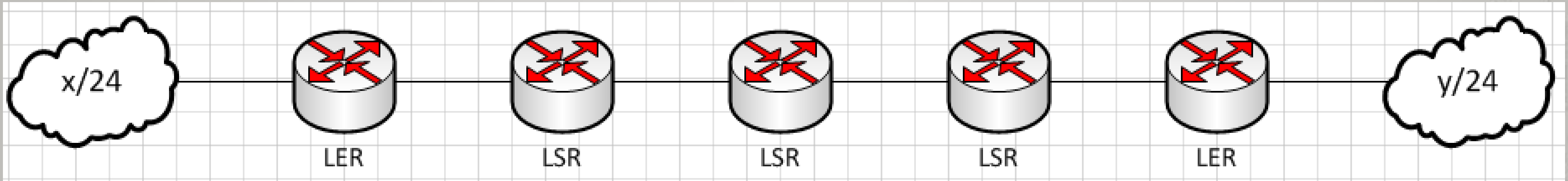
Router roles in MPLS



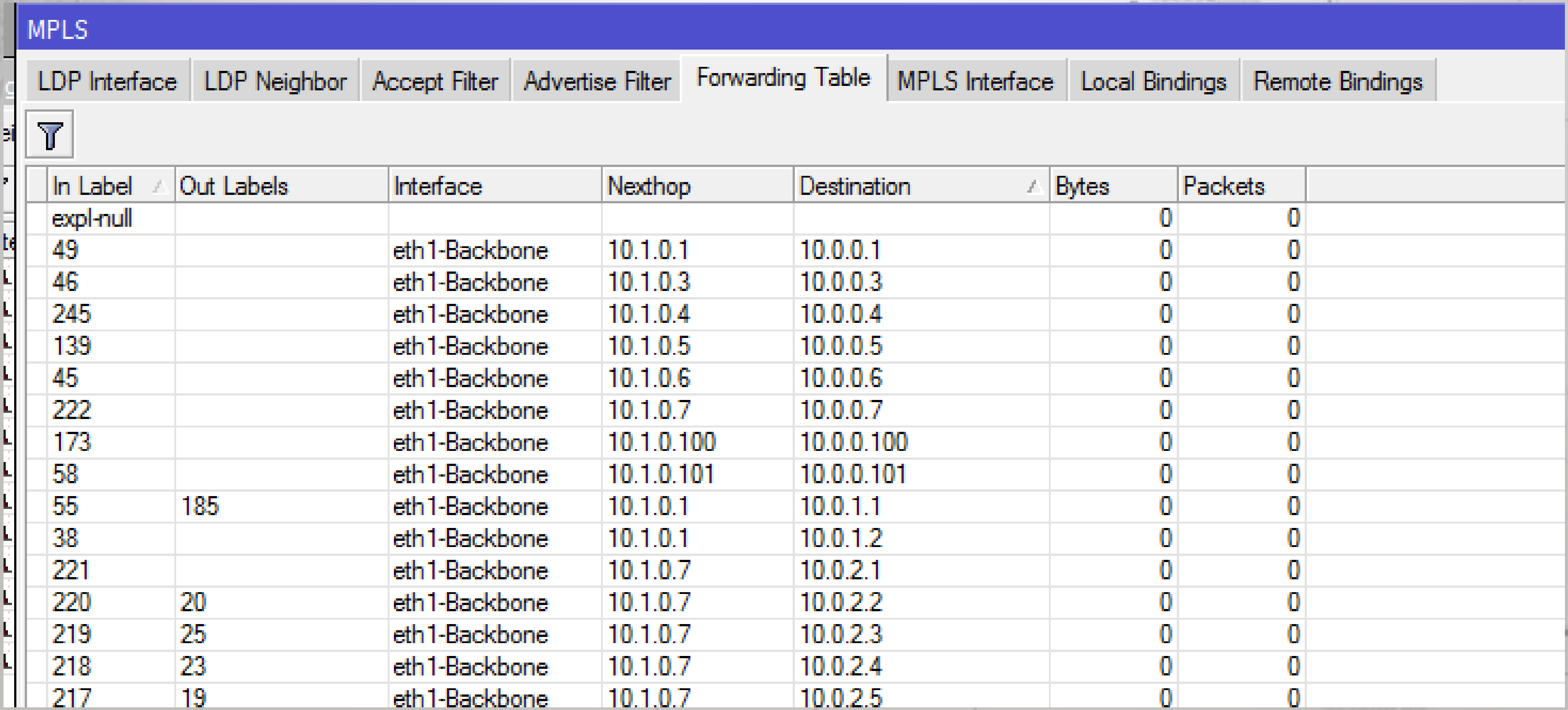
- LER – Label Edge Router
- LSR – Label Switch router
  - A single router can be a LER and LSR at the same time

Actions performed on a MPLS frame

- Push – add a label
- Pop – remove a label
- Swap – change the label



MPLS forwarding table:



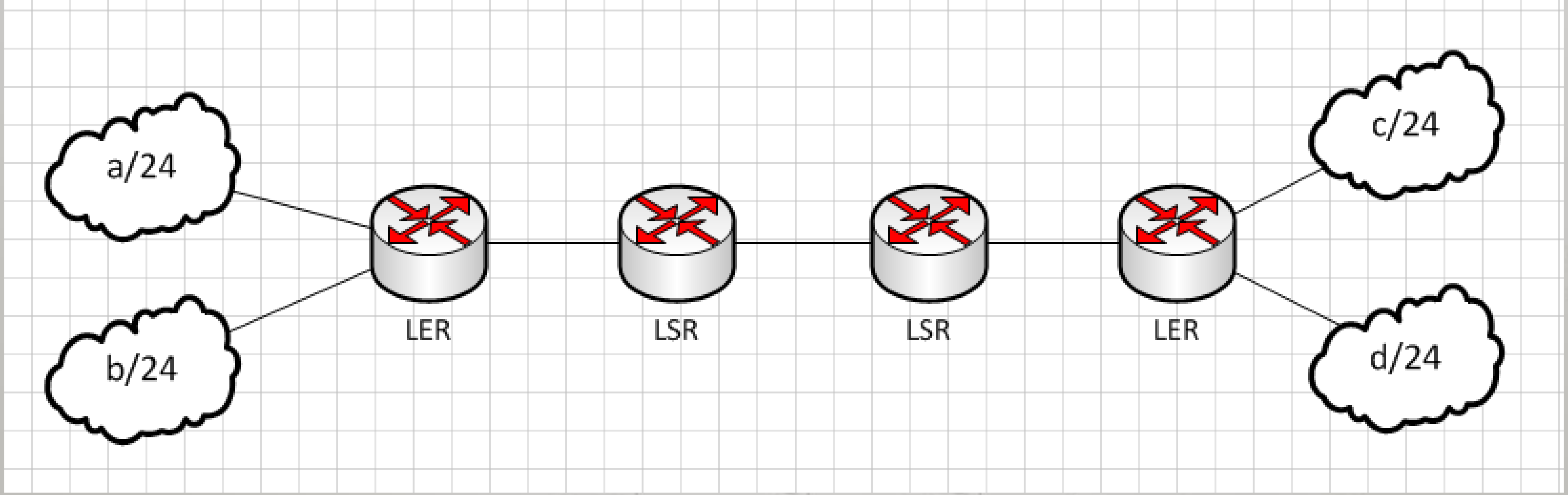
In Label	Out Labels	Interface	Nexthop	Destination	Bytes	Packets
expl-null					0	0
49		eth1-Backbone	10.1.0.1	10.0.0.1	0	0
46		eth1-Backbone	10.1.0.3	10.0.0.3	0	0
245		eth1-Backbone	10.1.0.4	10.0.0.4	0	0
139		eth1-Backbone	10.1.0.5	10.0.0.5	0	0
45		eth1-Backbone	10.1.0.6	10.0.0.6	0	0
222		eth1-Backbone	10.1.0.7	10.0.0.7	0	0
173		eth1-Backbone	10.1.0.100	10.0.0.100	0	0
58		eth1-Backbone	10.1.0.101	10.0.0.101	0	0
55	185	eth1-Backbone	10.1.0.1	10.0.1.1	0	0
38		eth1-Backbone	10.1.0.1	10.0.1.2	0	0
221		eth1-Backbone	10.1.0.7	10.0.2.1	0	0
220	20	eth1-Backbone	10.1.0.7	10.0.2.2	0	0
219	25	eth1-Backbone	10.1.0.7	10.0.2.3	0	0
218	23	eth1-Backbone	10.1.0.7	10.0.2.4	0	0
217	19	eth1-Backbone	10.1.0.7	10.0.2.5	0	0

PHP

- MPLS PHP
  - Penultimate hop popping

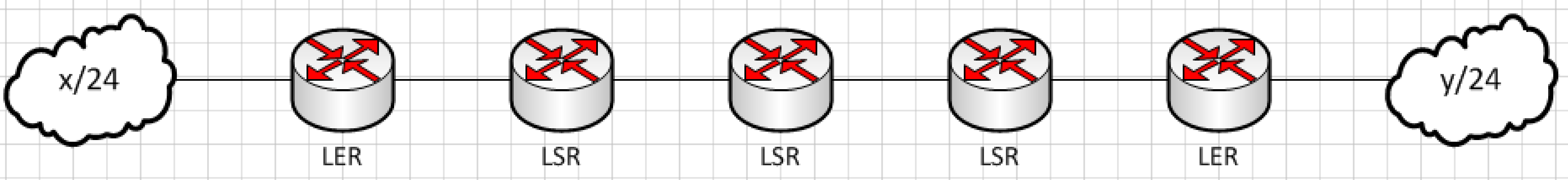
- PHP is implemented for performance reasons.
  - Without PHP, the LER would have to do 2 lookups (MPLS label forwarding table and IP routing table)

MPLS PHP



Complications - ICMP

- In a MPLS network, ICMP error packets are forwarded all the way to the original destination, not to the packets source (the source of the packet that caused the ICMP error)



MPLS ICMP Explained

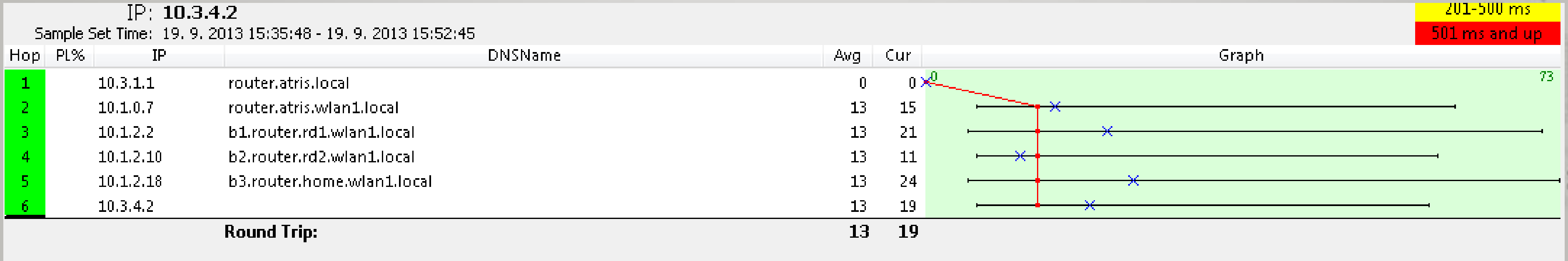
- This behavior is implemented because an MPLS switch doesn't have to be a router.
- It might not have a route to the source of the packet that caused the ICMP error. (L3 VPNs)
- The MPLS switch might not even support the IP protocol, or ICMP.

Implications

- In MPLS networks, when using trace-routes, remember the ICMP behavior.
- As long as there is a break on the MPLS path, the packet will not make it past the 1st hop, but that doesn't mean that the 2nd hop is dead.

Implications 2

- Ping times will not be reported correctly.
- Because of the MPLS ICMP behavior, the only ping you will see for all hops is the full round-trip.



IP: **10.3.4.2**
Sample Set Time: 19. 9. 2013 15:35:48 - 19. 9. 2013 15:52:45

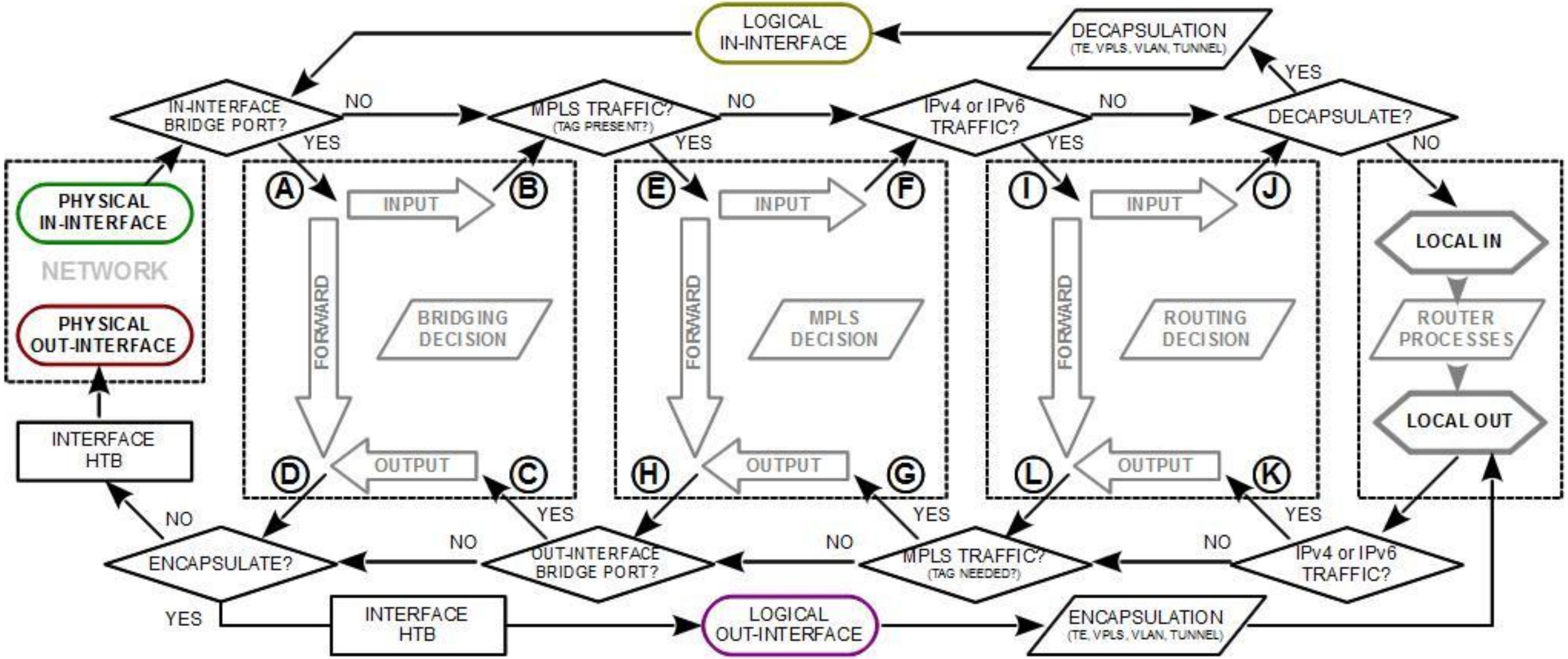
201-500 ms
501 ms and up

Hop	PL%	IP	DNSName	Avg	Cur	Graph
1		10.3.1.1	router.atris.local	0	0	
2		10.1.0.7	router.atris.wlan1.local	13	15	
3		10.1.2.2	b1.router.rd1.wlan1.local	13	21	
4		10.1.2.10	b2.router.rd2.wlan1.local	13	11	
5		10.1.2.18	b3.router.home.wlan1.local	13	24	
6		10.3.4.2		13	19	
			Round Trip:	**13**	**19**	

MPLS packet behavior:

- MPLS switched traffic:
  - Doesn’t pass through firewall
  - Doesn’t pass through NAT
  - Doesn’t pass through mangle
  - Doesn’t pass through QoS
  - Etc.
- On the LERs the traffic **will** pass the routing engine!

MikroTik RouterOS Packet Flow Diagram for version 6.x



How do I MPLS?

```
/mpls interface
   set [ find default=yes ] mpls-mtu=1550
/mpls ldp
   set enabled=yes lsr-id=RouterID transport-address=RouterID
/mpls ldp interface
   add interface=“ether1.vlan1000 - backbone.local"
```

- Remember that the RouterID from OSPF should be an actual IP on a loopback interface and be reachable.

Adding VPLS – TT

```
/interface vpls
add advertised-l2mtu=1508 name=“ether1.vlan1000.vpls1" remote-peer=10.0.2.2 vpls-id=1:0
add advertised-l2mtu=1508 name=" ether1.vlan1000.vpls2" remote-peer=10.0.2.5 vpls-id=1:0

/interface bridge
add l2mtu=1508 name="br2 - PPPoE AC“

/interface bridge port
add bridge="br2 - PPPoE AC" horizon=1 interface="ether1.vlan4000 - customers.local"
add bridge="br2 - PPPoE AC" horizon=1 interface="ether1.vlan1000.vpls1"
add bridge="br2 - PPPoE AC" horizon=1 interface="ether1.vlan1000.vpls2"
```

Securing L2 - bridges

- On RouterOS the bridge split horizon will allow us to secure the L2 segment.
- Only ports will different horizon value can communicate with each other.

Adding VPLS – Wireless AP

```
/mpls interface
set [ find default=yes ] mpls-mtu=1550
/mpls ldp
set enabled=yes lsr-id=RouterID transport-address=RouterID
/mpls ldp interface
add interface="eth1 - c1.wlan1.local“

/interface vpls
add advertised-l2mtu=1508 name="eth1.vpls1 - pppoe.ac.backbone.local“
remote-peer=10.0.0.100 vpls-id=1:0
```

PPPoE AC Config

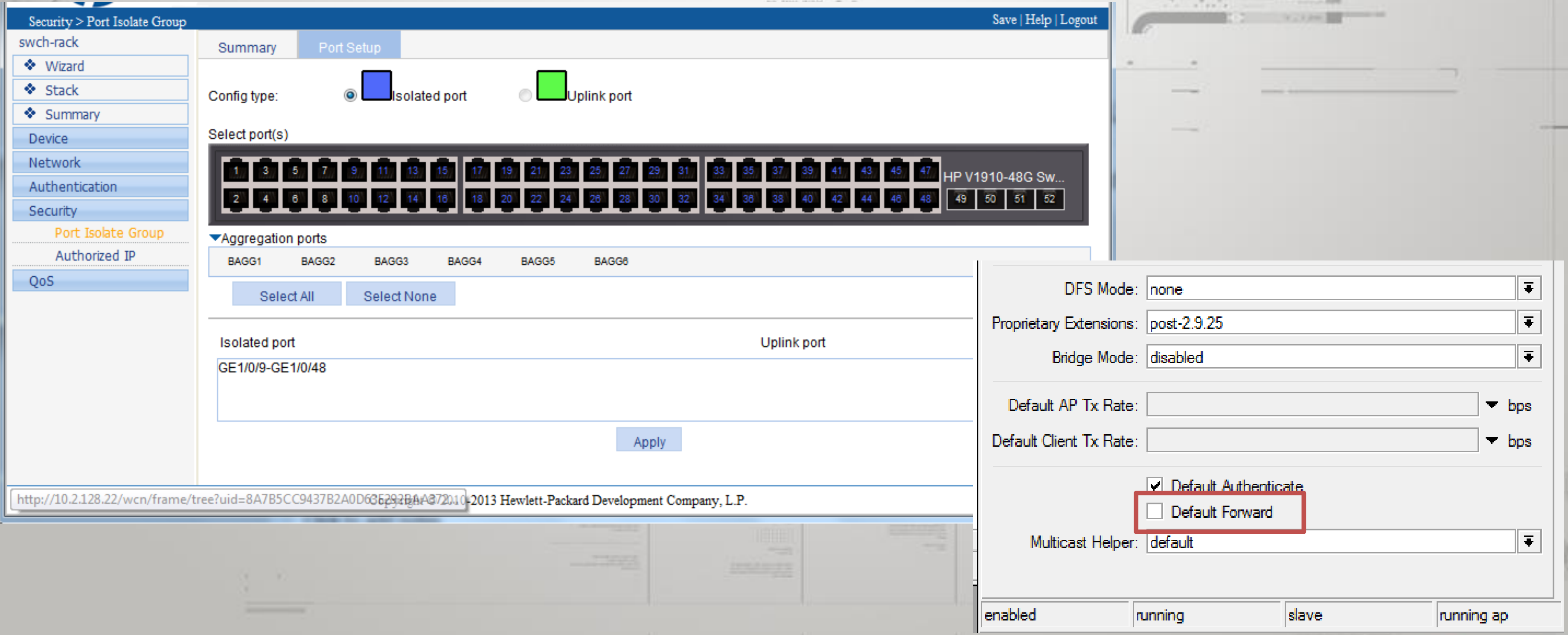
```
/ppp profile
add name="PPPoE" change-tcp-mss=no local-address=10.4.255.255 remote-address=PPPoE-pool
/ppp aaa
set use-radius=yes
/radius
add address=10.2.128.9 secret=password service=ppp

/interface pppoe-server server
add default-profile="PPPoE" disabled=no interface=ether1 max-mru=1500 max-mtu=1500
```

Securing L2 – customers

- PPPoE, being a PtP tunnel, only requires L2 connectivity between the endpoints.
- For security reasons its desired to block direct L2 communication, so your customers are protected.

- For wireless links, simply uncheck default-forward.
- For wired clients, enable port isolation on the switch.

Securing L2 - customers



Config type: ● Isolated port ○ Uplink port

Select port(s)

Aggregation ports

Isolated port: GE1/0/9-GE1/0/48

Uplink port

http://10.2.128.22/wcn/frame/tree?uid=8A7B5CC9437B2A0D6... 2010-2013 Hewlett-Packard Development Company, L.P.

DFS Mode: none

Proprietary Extensions: post-2.9.25

Bridge Mode: disabled

Default AP Tx Rate: bps

Default Client Tx Rate: bps

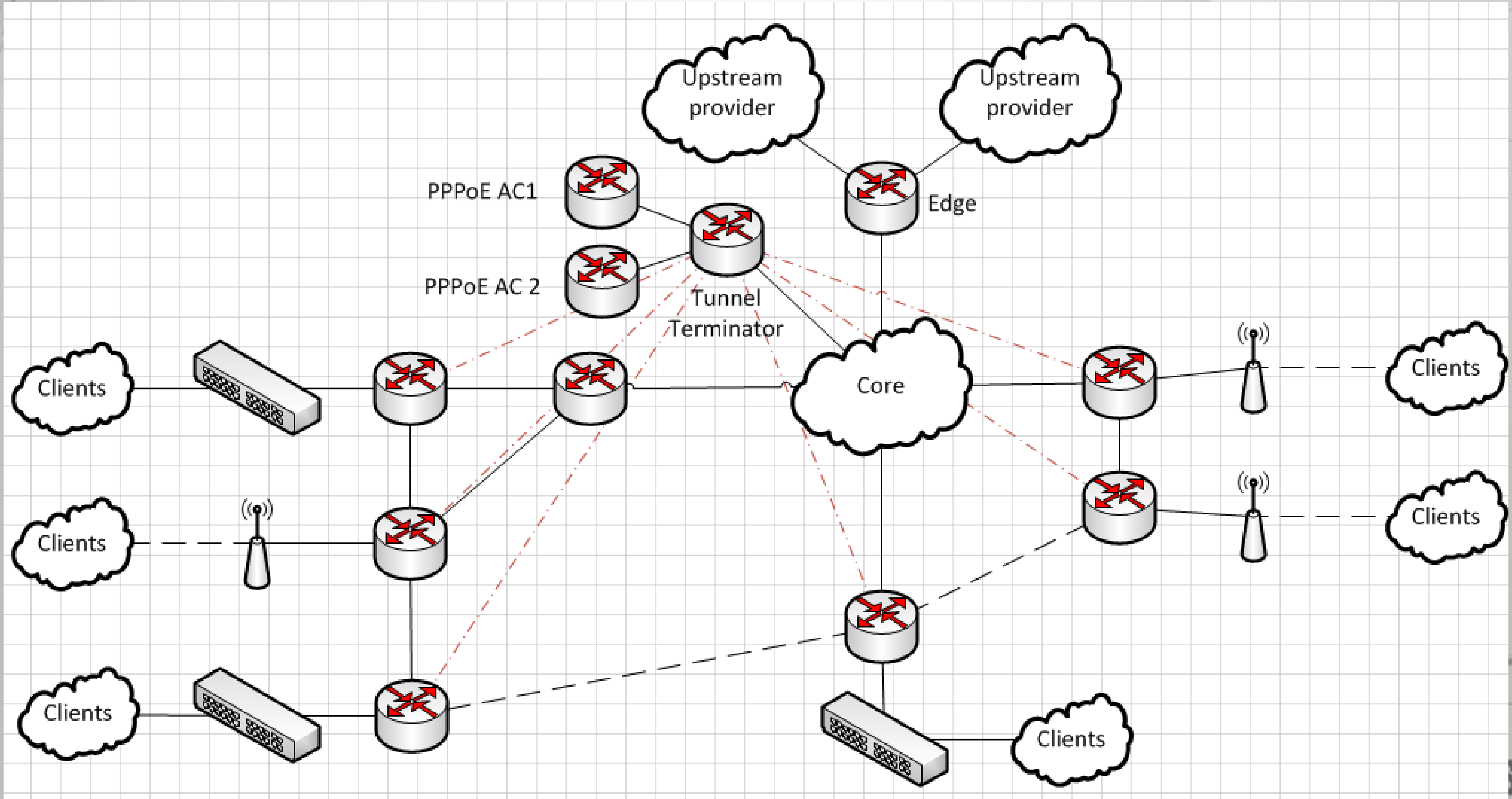
☑ Default Authenticate

□ Default Forward

Multicast Helper: default

enabled | running | slave | running ap

Accomplishing the goals



Tip: L2 VPNs

- You can offer a service for your customers, of transparent L2 VPNs just by building a VPLS tunnel, and bridging it to them.

- New service for your customers, without implementing anything. (you already have VPLS because of PPPoE)

Tips

- If something is not working, and you are sure your config is good, its probably MTU.

- Look for unmanaged switches across the MPLS path. Make sure jumbo is supported on all equipment in the MPLS path.

Watch our for cheap NICs

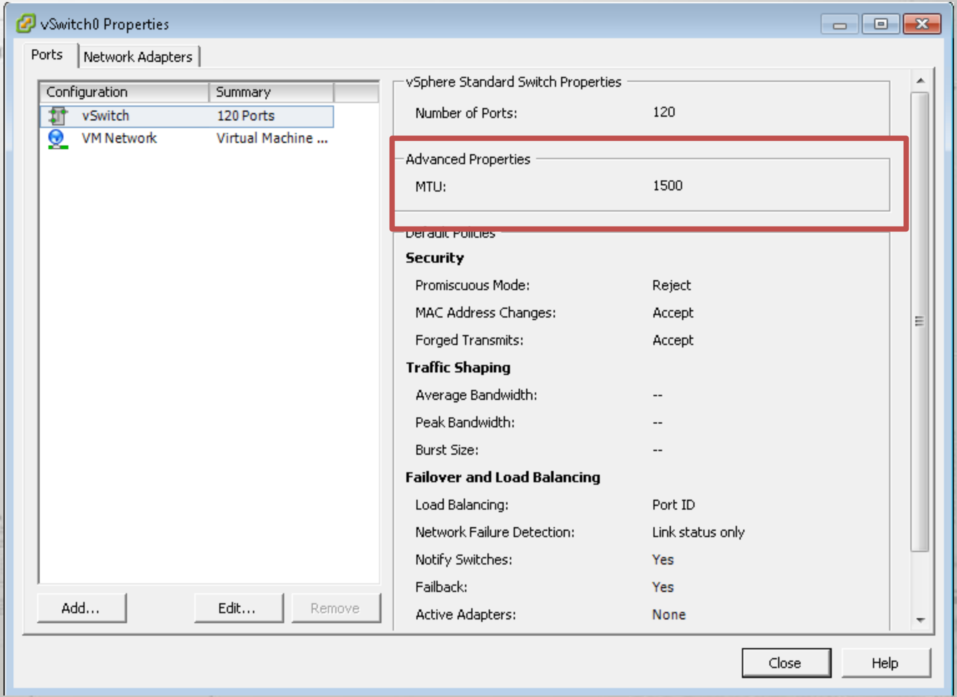
- Some NICs will not report their Max L2MTU to RouterOS.

- In this case, since RouterOS doesn’t know the NICs Max L2MTU, it ignores any frames that are >1500 (even if NIC actually supports jumbo).

- Only a problem on x86 or if ROS is a VM.

ESXi Tips

- If you are virtualizing, don’t forget to check MTU everywhere.
- Example: e1000 NIC in ESXi doesn’t support MTU >1500, even if the vSwitch does.
- Use e1000e (edit .vmx manually if needed)



Configuration	Summary
vSwitch	120 Ports
VM Network	Virtual Machine ...

vSphere Standard Switch Properties

Number of Ports: 120

Advanced Properties

MTU: 1500

Default Policies

Security

Promiscuous Mode: Reject

MAC Address Changes: Accept

Forged Transmits: Accept

Traffic Shaping

Average Bandwidth: --

Peak Bandwidth: --

Burst Size: --

Failover and Load Balancing

Load Balancing: Port ID

Network Failure Detection: Link status only

Notify Switches: Yes

Failback: Yes

Active Adapters: None

MPLS binding issue

- RouterOS creates a label binding for all prefixes in the routing table, even if the next hop is not MPLS enabled.
- Watch out for this on LERs, and create manual expl-null bindings as needed.
- Note: there is a bug in <6.3, where you cant create more than one expl-null binding.

MTU issues on ROS

- Even on RouterOS, there are MTU issues.
- Currently, a bonding interface does NOT report Max L2MTU.
- You can not use MTU >1500 if you use bonding. (no MPLS)
- Bug is reported, hopefully will be fixed. (2 months in waiting)

Issues: L3 VPNs

- L3 VPNs on v6.x are broken.
  - BGP routes not properly withdrawn
  - Redistribution inside VRF doesn’t work
  - Route leaking not working, etc.
- Use v5 if you need L3 VPNs (test test test)
- Currently L3 VPNs are not possible on CCR.
- Mikrotik support says these problems are not a priority, probably because of major changes needed to the routing engine to fix them. Lets hope for v7.

Problem with PPPoE

- One problem this delivery mechanism (PPPoE over VPLS) has is IPTV.
- Implementing IPTV with multicast requires a routed network, but you are providing PtP tunnels for each customer (therefore, multicast will not save bandwidth)
- Consider deploying multicast beside PPPoE, for example, in a separate VLAN.

Overall state of MPLS

- Overall, MPLS on Mikrotik is functional, and deployable in production (minus L3 VPNs)

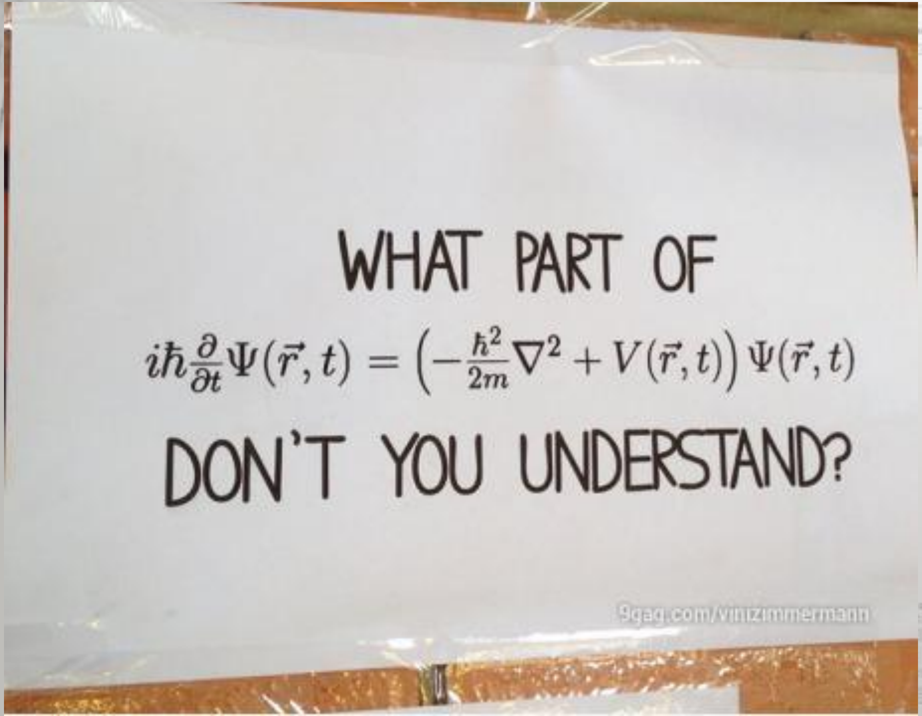
- As long as you are aware of how MPLS works (ICMP) it’s a great tool in Mikrotiks toolkit.

Final notes:

- This presentation is by no means a complete ready-to-implement solution.
- MPLS and its deployment require topology and network considerations and planning.

More material:

- If you liked this presentation look at Tiktube.com:
- US12:
  - Bandwidth-based load-balancing without MPLS TE
- EU13:
  - Building a scalable IPSec infrastructure with MikroTik



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

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

Thanks for listening

Tomas Kirnak

t.kirnak@atris.sk

www.atris.sk