



ISP Design – Using MikroTik CHR as an MPLS router

PRESENTED BY:

KEVIN MYERS,
NETWORK ARCHITECT

Profile: About Kevin Myers

Background:

- 20+ years in Networking
- Designed/Built Networks on 6 continents
- MikroTik Certified Trainer
- MikroTik, Cisco and Microsoft Certified



Community Involvement:



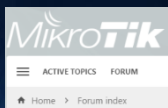
Packet Pushers (Podcast Guest / Blogger)



Group Contributor (RouterOS / WISP Talk and others)



Delegate/Roundtable contributor (NFD14)



MT Forum (Forum Veteran – Member since 2012)



Network Collective (Podcast Guest)



Profile: **About IP ArchiTechs**



Expert Networking

Whitebox | ISP | Data Center | Enterprise

- ✓ Global Consulting
- ✓ Managed Networks
- ✓ Monitoring
- ✓ Load Testing
- ✓ Development

Locations in: US | Canada | South America

Call us at: +1 855-645-7684

E-mail: consulting@iparchitech.com

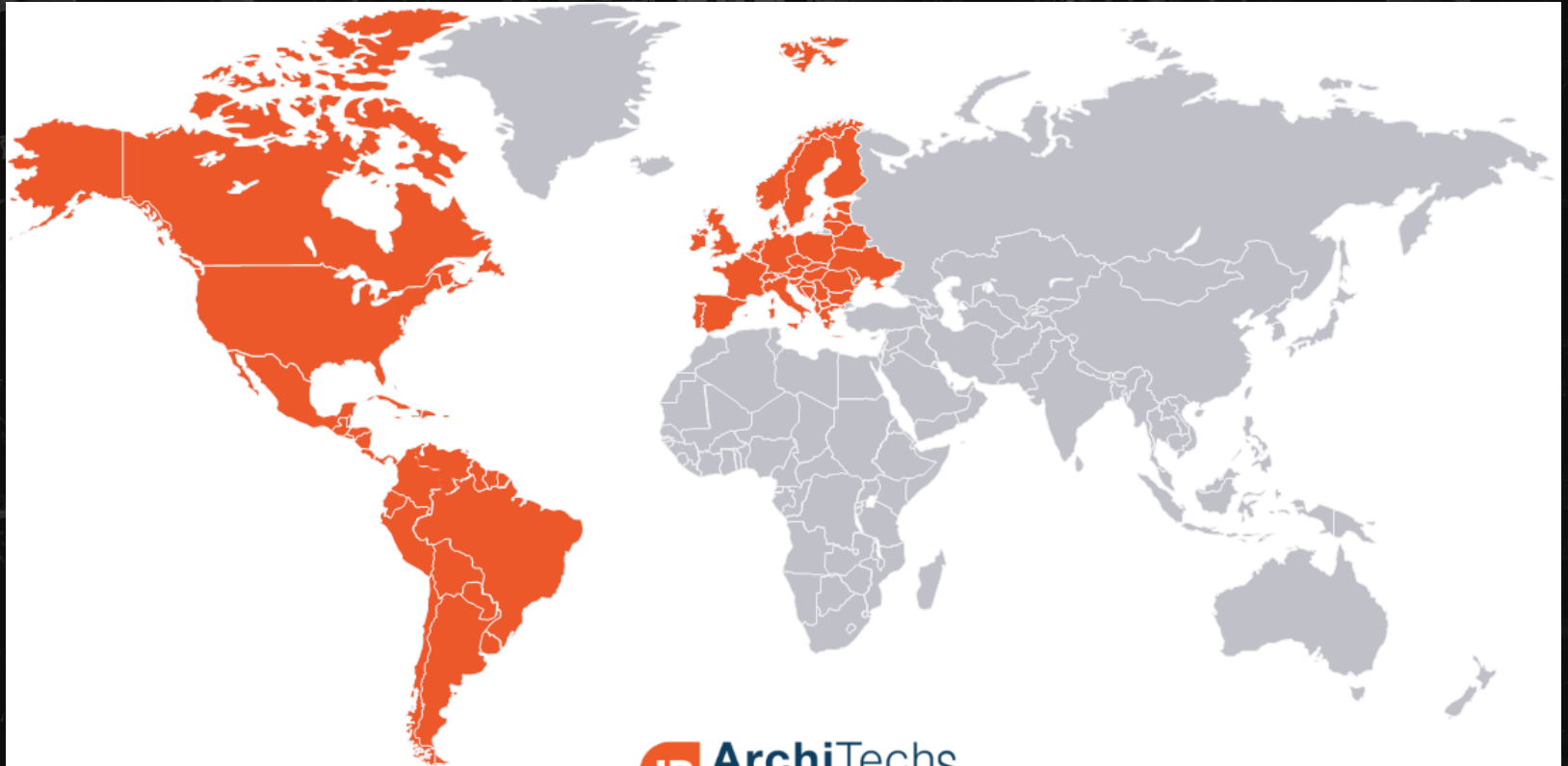
Web: www.iparchitech.com



Profile: **About IP ArchiTechs**

Now in Europe!

IPA Opened an office in Nis, Serbia in 2018



North America - South America - Europe

Goal of this presentation: When the presentation is finished, hopefully you will have walked away with a few key concepts:

- Performance characteristics of the CHR as an MPLS router
- Best practices when deploying the CHR as an MPLS router
- Benefits of using the CHR vs RouterBoard or CCR as an MPLS router

- Which Hypervisor should I use?



PROXMOX



**Microsoft
Hyper-v**

- Which Hypervisor should I use?



**Microsoft
Hyper-v**

Hyper-V is the only hypervisor currently recommended for MPLS with the MikroTik CHR.

MTU is handled differently in Hyper-V vs. ESXi and ProxMox (KVM). Packets are **not** assembled into 64k buffers in HyperV. When packets are broken down into 64k buffers, it seems to create MTU issues for the CHR.

- **Why Not ESXi or ProxMox (KVM)?**



PROXMOX

ESXi and ProxMox (KVM) both have issues when running the CHR for MPLS.

MTU is handled differently in Hyper-V vs. ESXi and ProxMox (KVM). Packets are assembled into 64k buffers which seems to create MTU issues for the CHR. This affects explicit null the most.

- Which platform is better?
- Throughput capabilities?
- x86 CPU vs. ARM/Tilera?
- MTU/Throughput concerns on different Hypervisors



VS.





Design: **CHR vs. Tilara/ARM for MPLS?**

Platform



CPU

MPLS router CPU requirements depend on load and explicit/implicit null

x86

Better for heavy computational work. Higher power draw.

Tilara

Optimized for packet transfer. Designed to be low power draw.

ARM

In between x86 and Tilara for performance.

Throughput

At 1530 bytes (L2), and 8970 bytes (L2)

x86

More CPU and power is required to move data at the same speed as a CCR

Tilara

Handles throughput at different frame sizes slightly better than x86

ARM

Handles throughput at different frame sizes similar to Tilara

MTU Handling

x86

x86 hardware and HV can typically support up to 9000 MTU.

Tilara

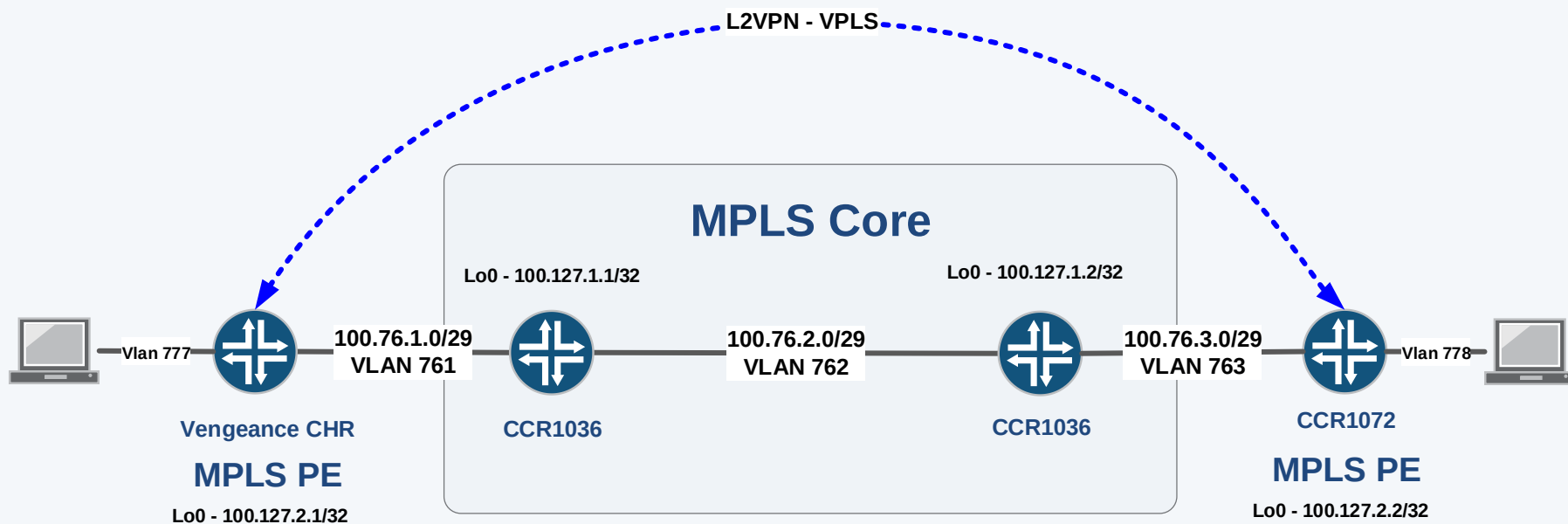
Supports up to 10222

ARM

Supports up to 9982



Design: CHR MPLS testing - logical lab setup

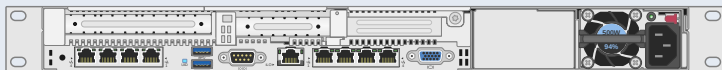




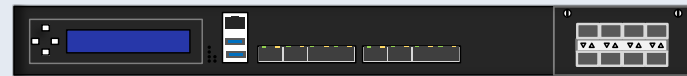
Design: CHR MPLS – switch-centric lab setup

Physical Test Network for MPLS CHR

HP 1U x86 Server



Baltic Networks Vengeance HV



MPLS PE



MPLS Core



MPLS PE

Hypervisor details – VM provisioning recommendations for Hyper V

- 2 vCPUs
- 4096 MB RAM (or more)
- Disable HyperThreading in the BIOS
- Use CPU reservation (100%)
- Disable all un-needed VM components (CD-ROM, SCSI controller, etc)
- Increase MTU to maximum on the VSWITCH/Interfaces

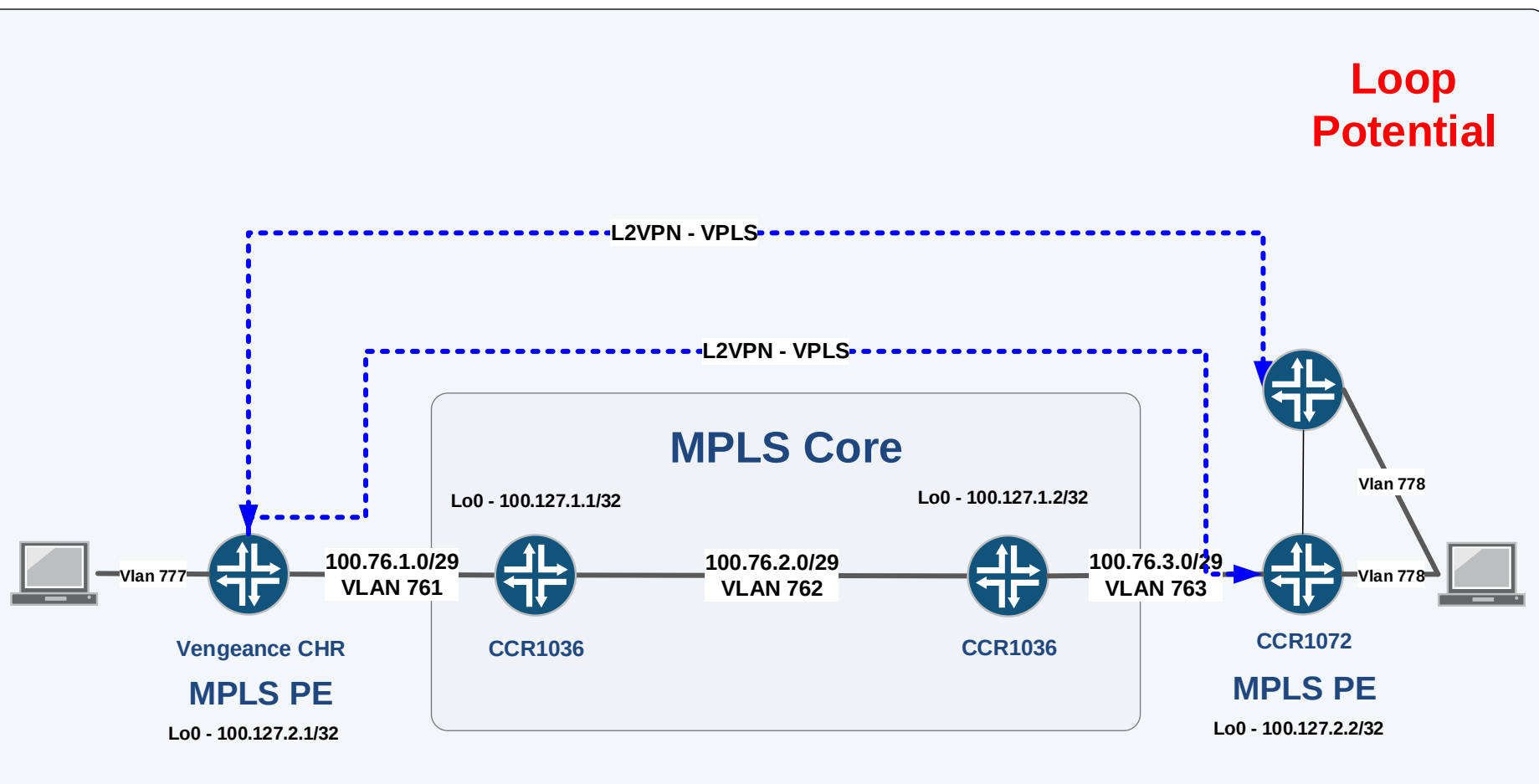
- **Concept of testing**

- Performance with VPLS
- Performance with Implicit Null vs Explicit Null
- Performance at 1530 MPLS MTU bytes, 9000 MPLS MTU bytes

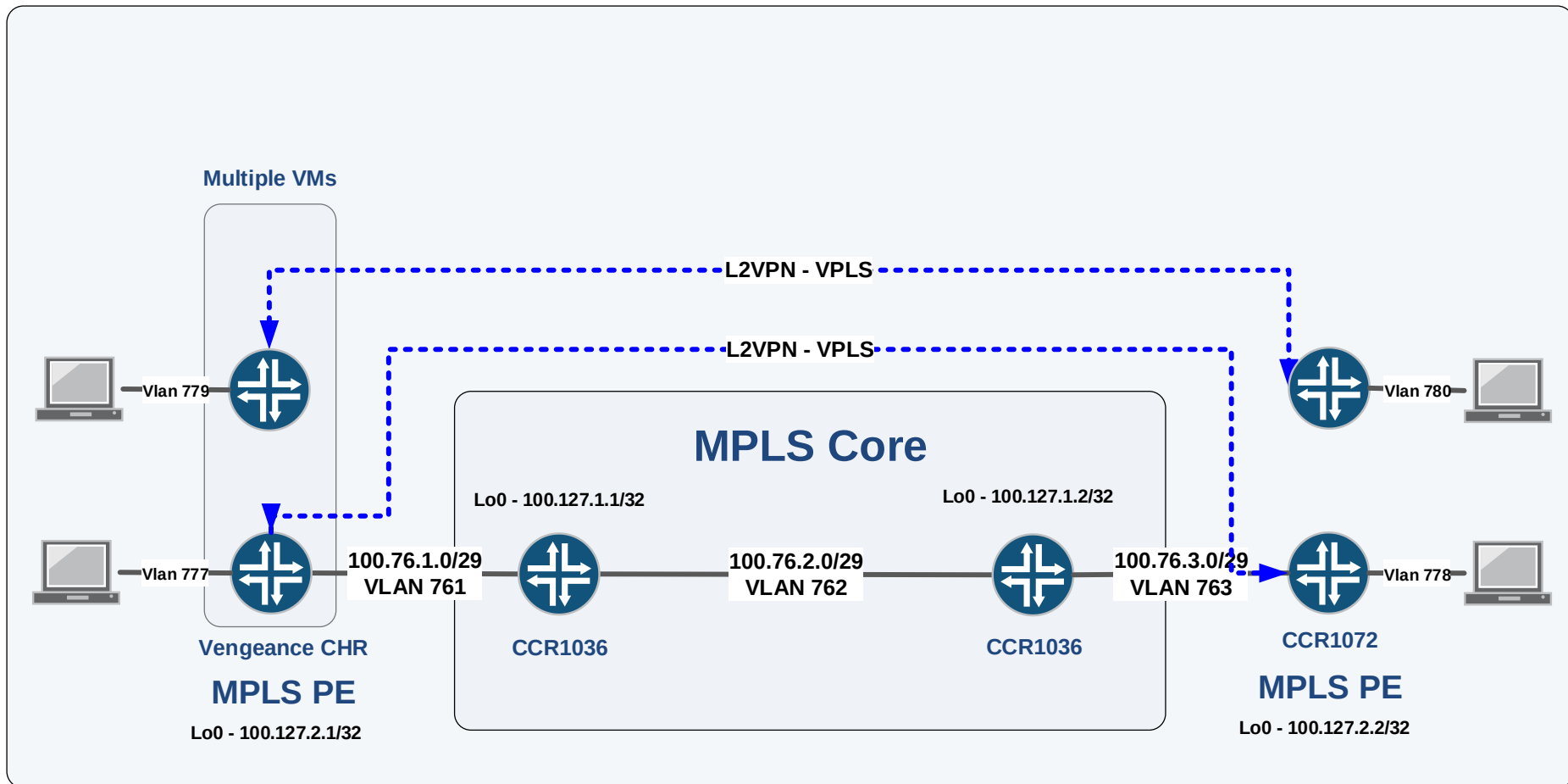
- **Performance considerations**

- Dedicate HV to CHR – don't mix applications
- TSO/LSO – Disable for best performance
- Clock speed – Highest speed possible

- VPLS is easier to make highly available than independent routers due to issues with Layer 2 looping. CHR on two HV hosts eliminates looping.



- Can deploy multiple MPLS PE routers to isolate clients when needed

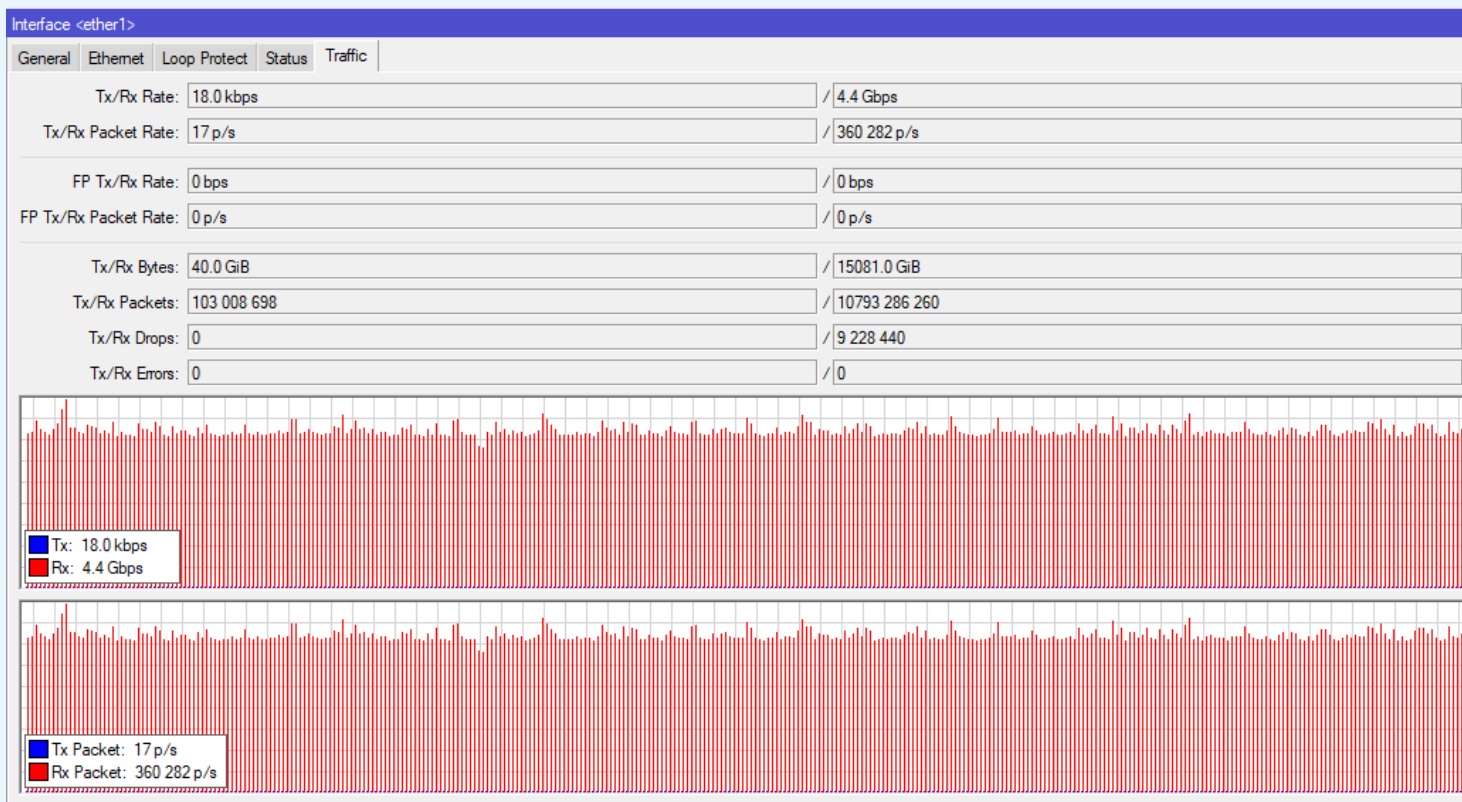


- **Implicit Null** will use Penultimate Hop Popping to deliver the packet unlabeled to the last MPLS router before the packet is forwarded into a non-MPLS network
- **Explicit Null** will keep the packet labeled until it egresses an interface that isn't MPLS capable and then the label will be stripped
- Explicit null set in MPLS → LDP will result in slightly higher CPU usage

- **Using the new multi-core bandwidth test MikroTik recently introduced for performance testing. (ROS v 6.44)**
- **It works very well!**

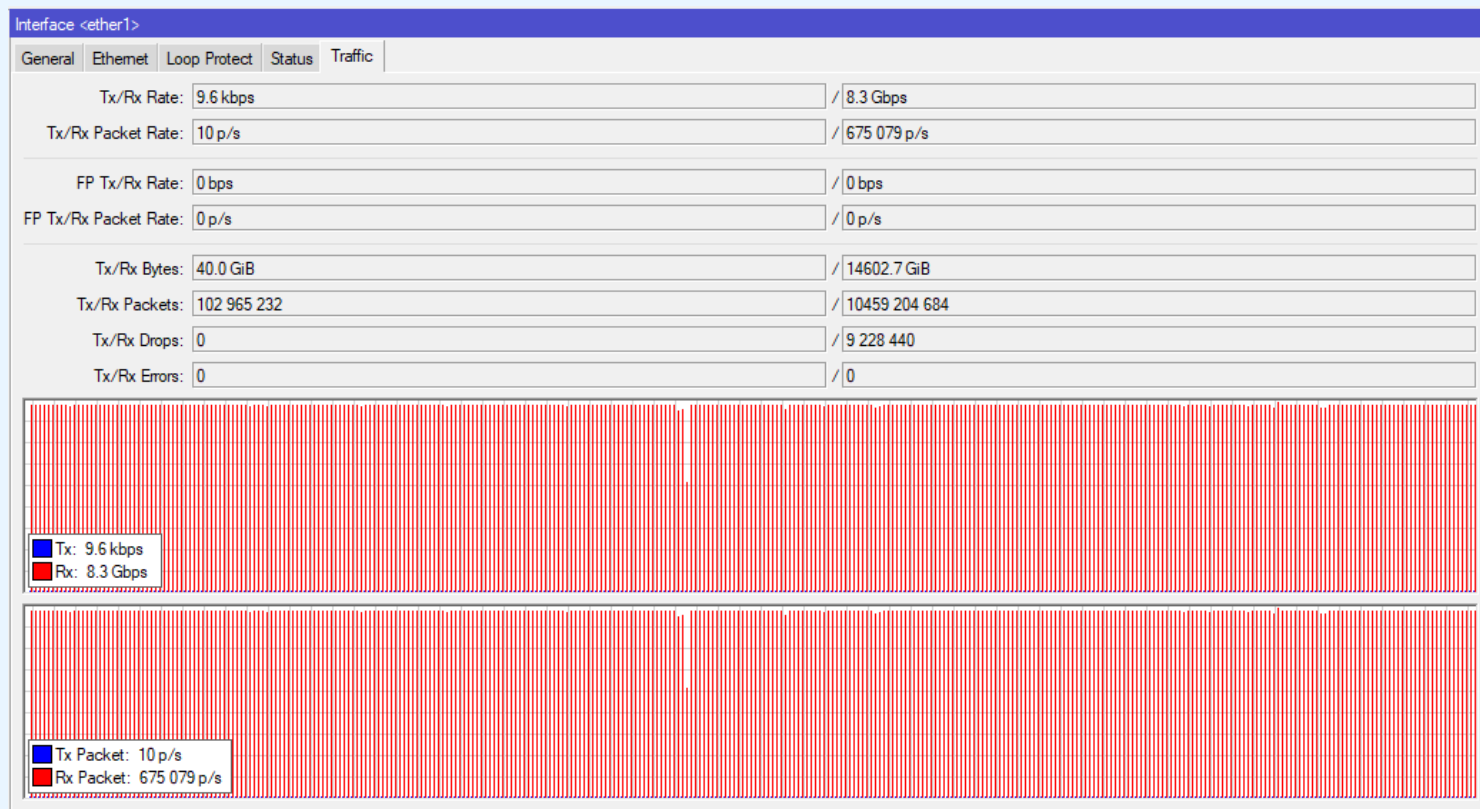
Platform	Hypervisor	CHR
Baltic Vengeance	Hyper-V 2016	6.44

Implicit Null - VPLS Throughput: **4.4 Gbps** Peak VM CPU: **15%** MPLS MTU: **1530**



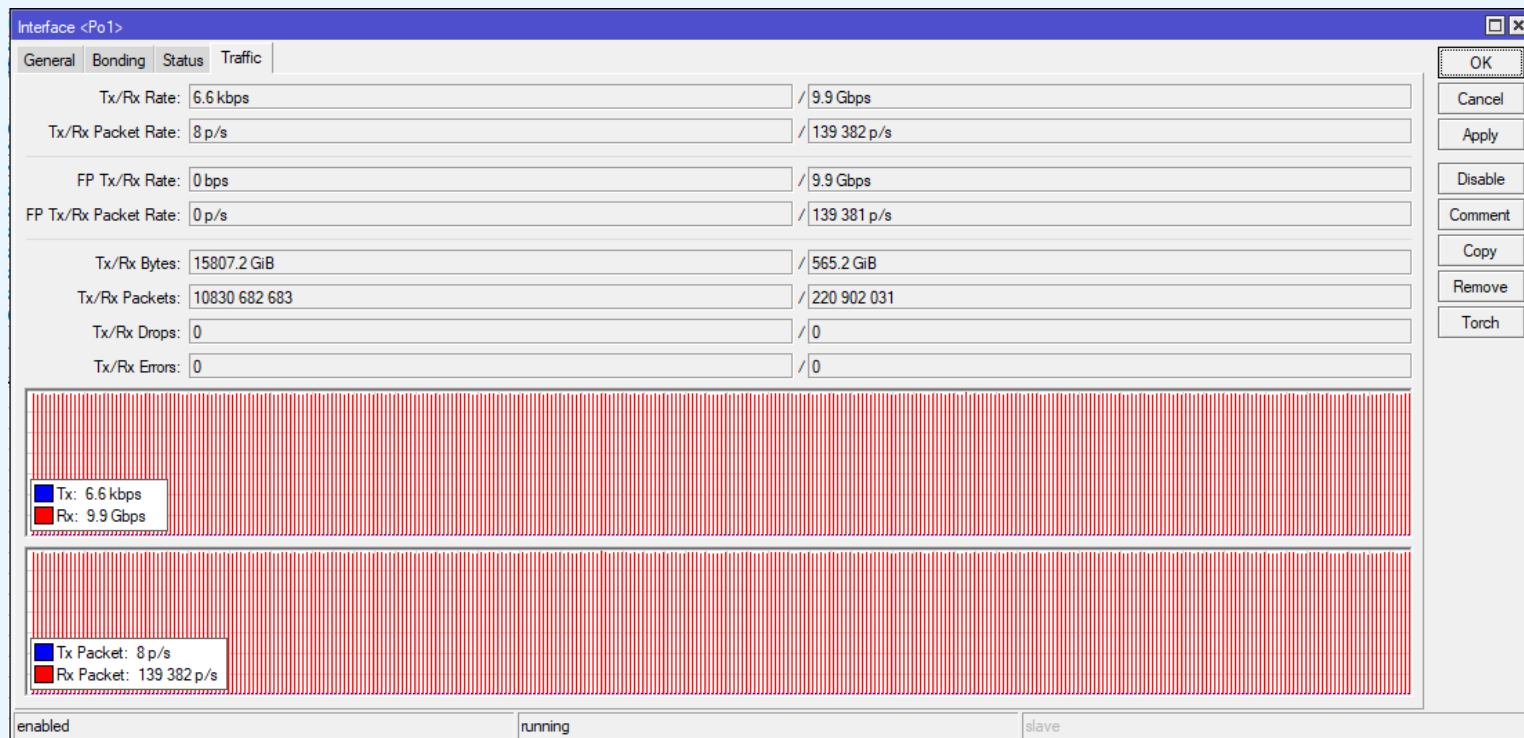
Platform	Hypervisor	CHR
Baltic Vengeance	Hyper-V 2016	6.44

Explicit Null - VPLS Throughput: **8.3 Gbps** Peak VM CPU: **20%** MPLS MTU: **1530**



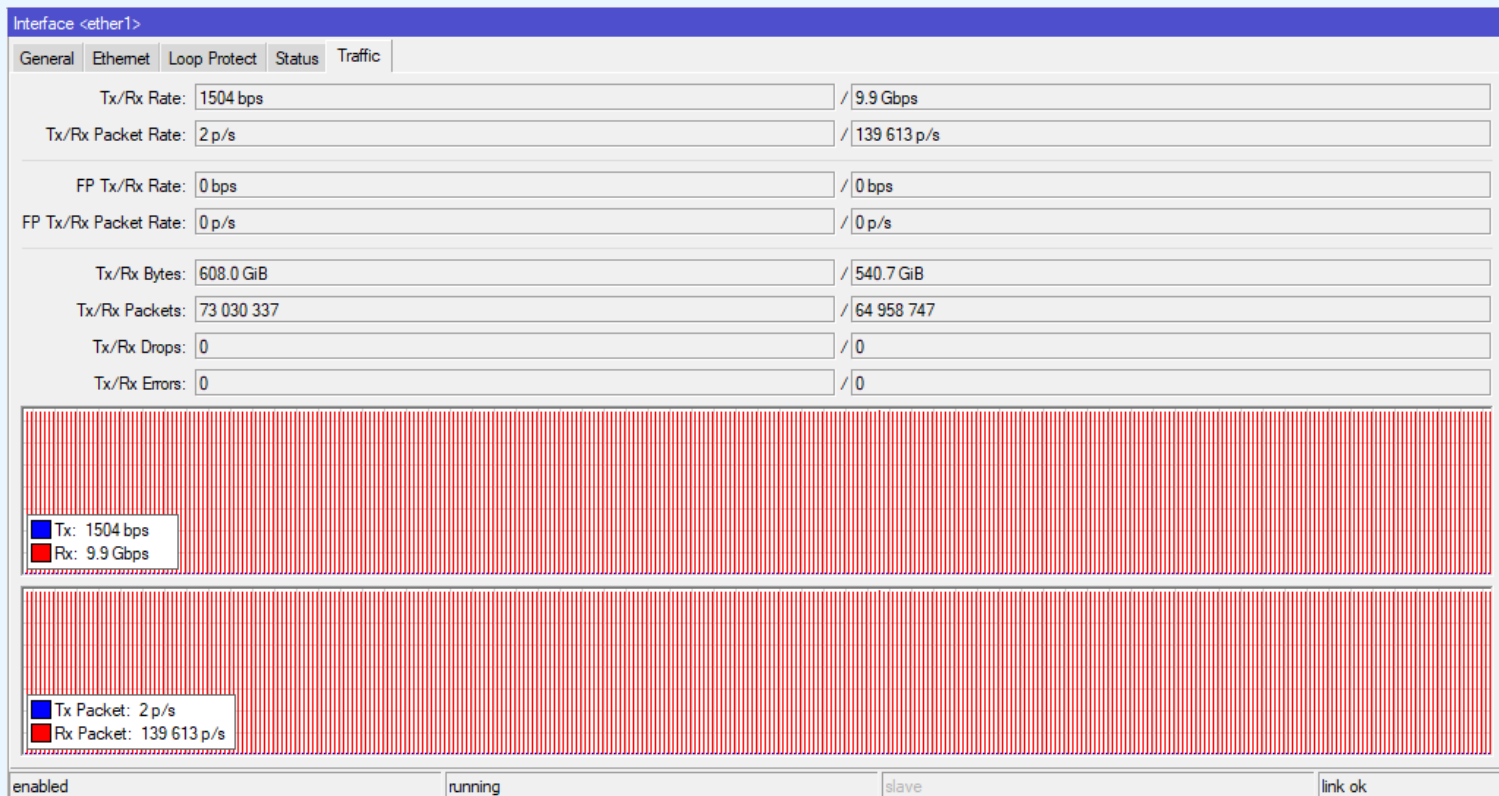
Platform	Hypervisor	CHR
Baltic Vengeance	Hyper-V 2016	6.44

Implicit Null - VPLS Throughput: **9.9 Gbps** Peak VM CPU: **7%** MPLS MTU: **9000**



Platform	Hypervisor	CHR
Baltic Vengeance	Hyper-V 2016	6.44

Explicit Null - VPLS Throughput: **9.9 Gbps** Peak VM CPU: **16%** MPLS MTU: **9000**





Design: **Questions?**

Questions??