



Architech's

M A N A G E D S E R V I C E S

www.iparchitech's.com

1-855-MIKROTI(K)

Understanding VLAN Translation/Rewrites using Switches and Routers

KEVIN MYERS, NETWORK ARCHITECT /
MANAGING PARTNER

MTCINE #1409

MIKROTIK CERTIFIED TRAINER

- **Kevin Myers, Network Architect**

- 17 + years in IT, Network Architecture and Engineering

- **Areas of Design Focus:**

- MikroTik integration with large multi-vendor networks
- Design of BGP/MPLS/OSPF Service Provider Triple-Play networks
- Design of large enterprise Data Center networks

- **Certifications**

- MTCINE #1409
- Certified – CCNP, CCNA, MCP, MTCRE, MTCTCE, MTCNA



Need MikroTik help? Introducing IP SafeNET – Flat-rate, yearly comprehensive MikroTik support with hardware replacement.

Three SLA Tiers

IP SafeNET 8x5x3BD

IP SafeNET 8x5xNBD

IP SafeNET 24x7xNBD

MUM Only promotion – Try it FREE for 30 days!

• Objectives

- Learn what a VLAN Translation is and the different use cases
- Understand the different implementation types for VLAN rewrites in RouterOS
- Walk through implementation examples that uses VLAN translations in a Router and CRS switch.

- **What is VLAN Translation?** VLAN Translation (or rewrite) involves replacing one ingress tag with another and vice-versa on egress.

Example of a VLAN Translation with a ping:

source: 10.1.1.200/24 dest: 10.1.1.201/24

PING SRC: 10.1.1.200/24 –
VLAN 101

VLAN 101

```
/interface ethernet switch ingress-vlan-translation
```

```
add customer-vid=101 customer-vlan-format=untagged-or-tagged new-customer-vid=201 \  
ports=ether1 service-vlan-format=untagged-or-tagged
```



VLAN 201

PING DST: 10.1.1.201/24 –
VLAN 201

- **What is VLAN Translation?** VLAN Translation (or rewrite) involves replacing one ingress tag with another and vice-versa on egress.

Example of a VLAN Translation with a ping:

source: 10.1.1.201/24 dest: 10.1.1.200/24

PING DST: 10.1.1.200/24 –
VLAN 101

VLAN 101

```
/interface ethernet switch egress-vlan-translation
```

```
add customer-vid=201 customer-vlan-format=untagged-or-tagged new-customer-vid=101 \  
ports=ether1 service-vlan-format=untagged-or-tagged
```

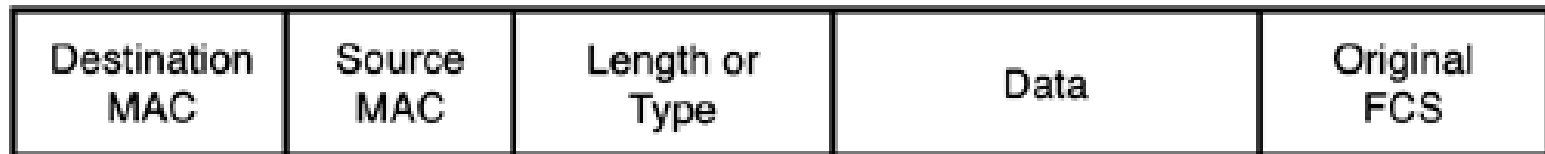


VLAN 201

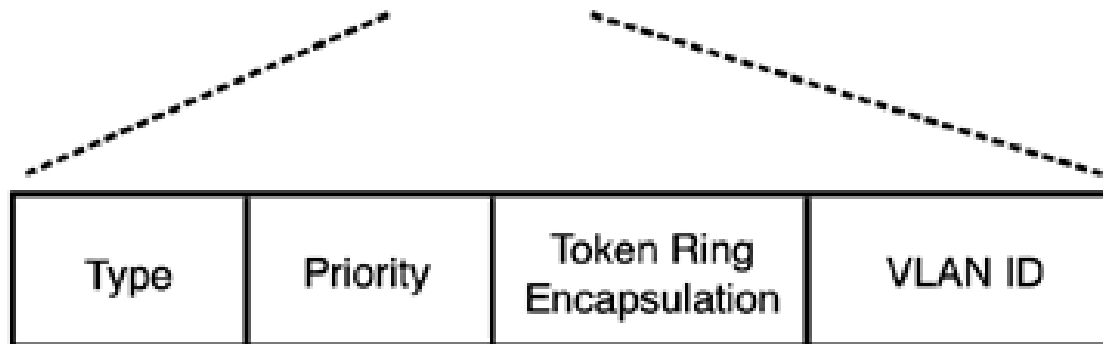
PING SRC: 10.1.1.201/24 –
VLAN 201

- **Is this QinQ? No.** VLAN Translation only involves one tag whereas QinQ adds an outer VLAN tag. In the frame below, the TAG field is rewritten from one VLAN to another while preserving all other information in the frame.

Original Ethernet Frame



802.1Q Tagged Frame

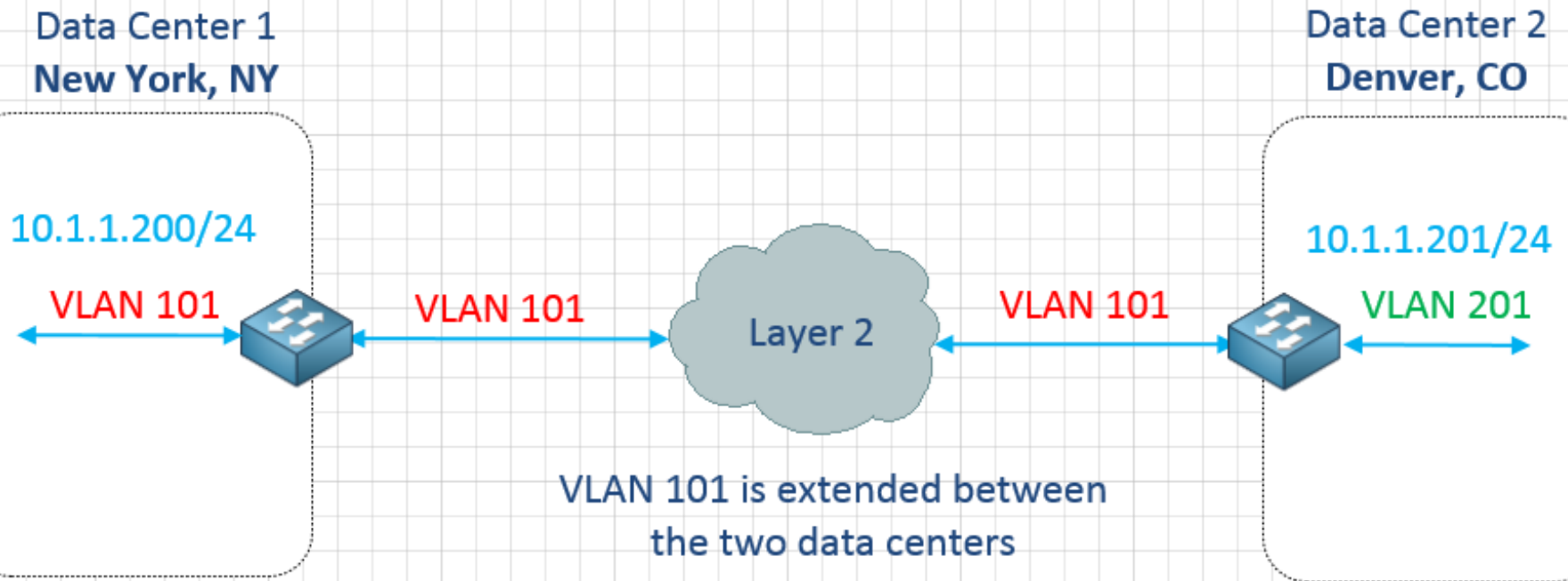


- **What problem are we trying to solve?**
- **Merging companies** – When large companies merge, there are often duplicate VLANs with different functions in campus and headquarters environments
- **Data Center Interconnect (DCI)** – When connecting large data centers at Layer 2, VLAN overlap is a major issue. A VLAN in one data center may be used for web servers, while the same VLAN number in another data center could be for database servers.
- **Service Provider** – Large carrier networks often deal with overlapping VLANs from customers and other providers. VLAN translation can be used to join segments with different tags or prevent overlap.
- **What are some other use cases for VLAN Translations?**

- **Example:**

- **Data Center 1 (NYC, NY, USA)** needs to extend Vlan 101 for web Servers on 10.1.1.0/24 to the Denver, CO DC
- **Data Center 2 (Denver, CO, USA)** uses Vlan 101 for storage replication on 192.168.222.0/24
- **VLAN 101 - 10.1.1.0/24** from NYC must be rewritten in Denver as **VLAN 201** to avoid conflict with the Storage Replication.
- Routerboards can do this via bridging and CRS via switching.

Example: VLAN 101 in NYC becomes VLAN 201 once it reaches Denver, CO. The broadcast domain (light blue) is the same even though the VLAN tag changes



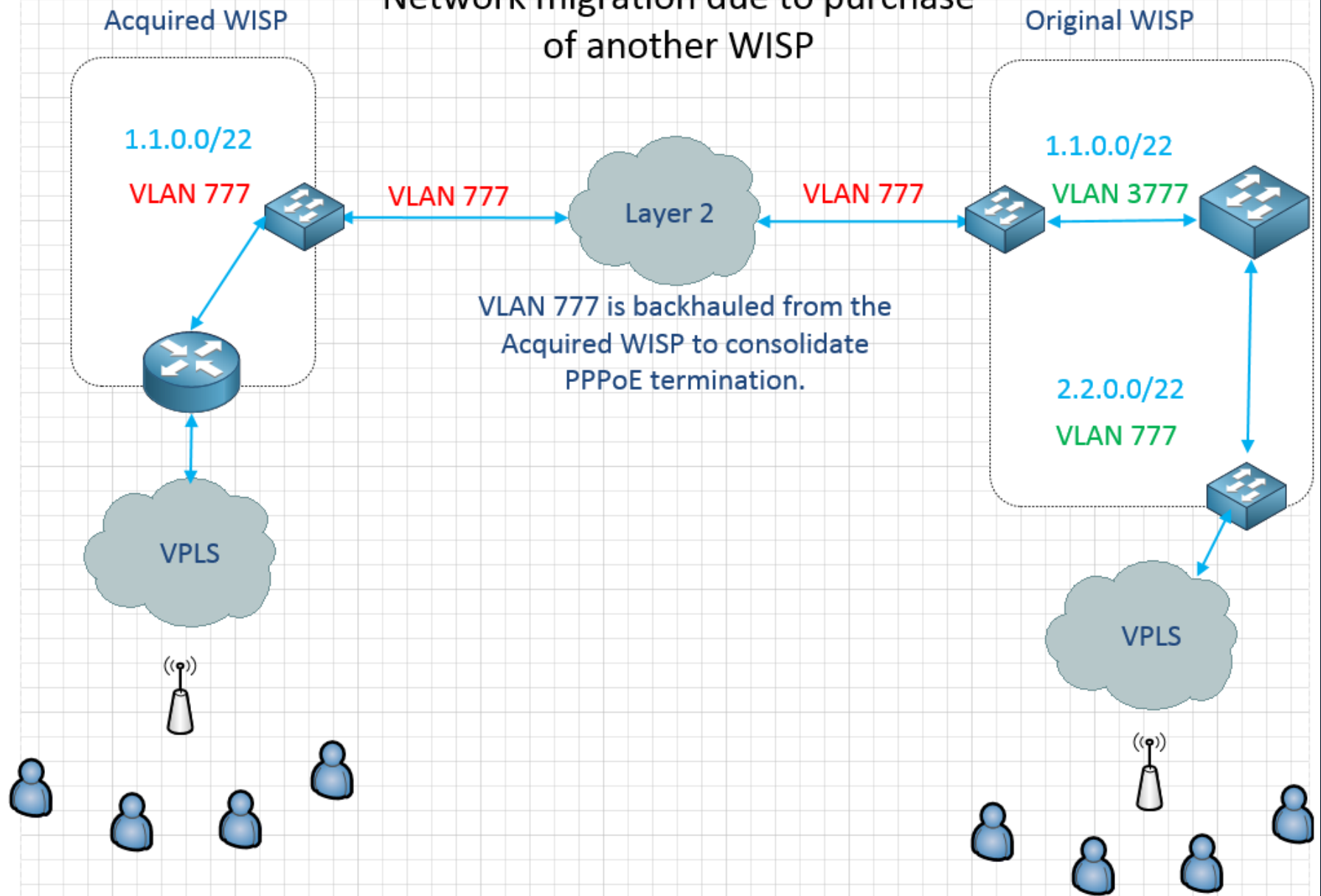
Example: VLAN 101 in NYC becomes VLAN 201 once it reaches Denver, CO. The broadcast domain (light blue) is the same even though the VLAN tag changes

Q. What is a broadcast domain?

Q. What Layer of the OSI Model does it exist at?

Q. Why would this be important when rewriting VLANs ?

Network migration due to purchase of another WISP



VLAN Translation

LAB Scenario

Configuration Objective - Translate VLAN from R1 to R2 and ping successfully between laptops

Materials Needed

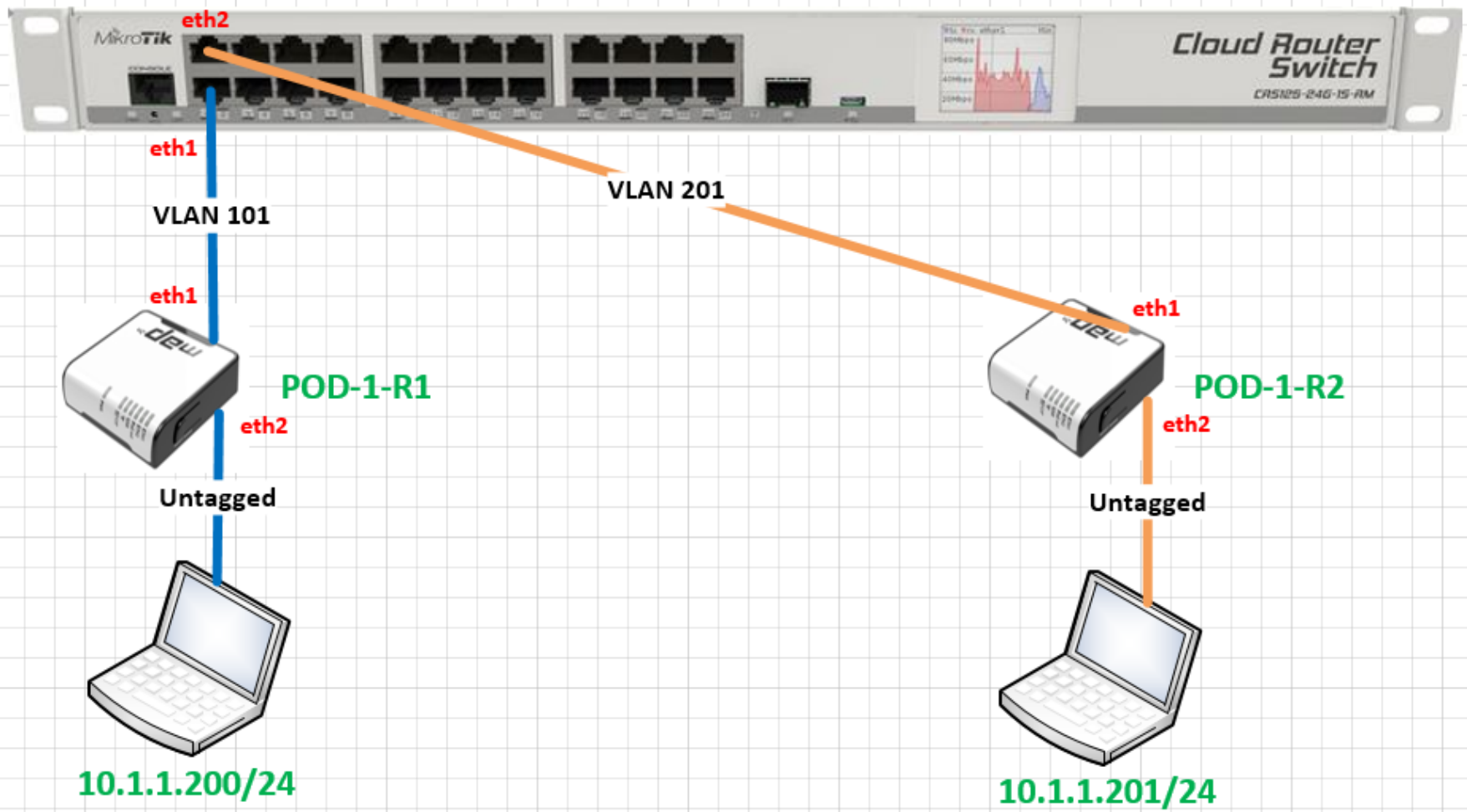
2 RouterBoards (each with at least 2 Ethernet ports)

2 Laptops with Ethernet port

4 Ethernet cables

CRS 125-24G-1S Switch

Lab Topology – Physical / Logical Connectivity for POD 1



Configuration Objective - Translate VLAN from R1 to R2 and ping successfully between laptops

VLAN Assignment

R1 VLAN – 101

R2 VLAN – 201

IP Address assignment for the Laptop

Laptop connected to R1 – 10.1.1.200/24

Laptop connected to R2 – 10.1.1.201/24

- Step 1

- Reset system config with no default on both routerboards in the pod

- Step 2

- Connect all devices according to the topology diagram.

• Step 3

- Assign the IPs for both laptops – 10.1.1.200 and .201
- Example (Windows) Using Laptop 1 (NOTE: No default gateway is needed for this lab)

Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

☐ Obtain an IP address automatically

☒ Use the following IP address:

IP address: 10 . 1 . 1 . 200

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

☐ Obtain DNS server address automatically

• Step 4

- Configure each lab Routerboard for the following

- R1 - Interface Vlan 10x on eth1 (where x is your POD)

```
/interface vlan  
add interface=ether1 l2mtu=1594 name=Vlan101 vlan-id=101
```

- R2 - Interface Vlan 20x on eth1 (where x is your POD)

```
/interface vlan  
add interface=ether1 l2mtu=1594 name=Vlan201 vlan-id=201
```

Step 5

Configure each lab Routerboard for the following

R1 – VLAN10x-Bridge (where x is your POD number)
and add ports **eth2** and **Vlan10x**

```
/interface bridge
add name=VLAN101-Bridge
/interface bridge port
add bridge=VLAN101-Bridge interface=Vlan101
add bridge=VLAN101-Bridge interface=ether2
```

R2 – VLAN20x-Bridge (where x is your POD number)
and add ports **eth2** and **Vlan20x**

```
/interface bridge
add name=VLAN201-Bridge
/interface bridge port
add bridge=VLAN201-Bridge interface=Vlan201
add bridge=VLAN201-Bridge interface=ether2
```

Step 6

Configure Ingress VLAN translation on the CRS

Ether1 – VLAN 101 to VLAN 201

Ether2 – VLAN 201 to VLAN 101

```
/interface ethernet switch ingress-vlan-translation
add customer-vid=201 customer-vlan-format=untagged-or-tagged new-customer-vid=\
  101 ports=ether2 service-vlan-format=untagged-or-tagged
add customer-vid=101 customer-vlan-format=untagged-or-tagged new-customer-vid=\
  201 ports=ether1 service-vlan-format=untagged-or-tagged
```

Step 8

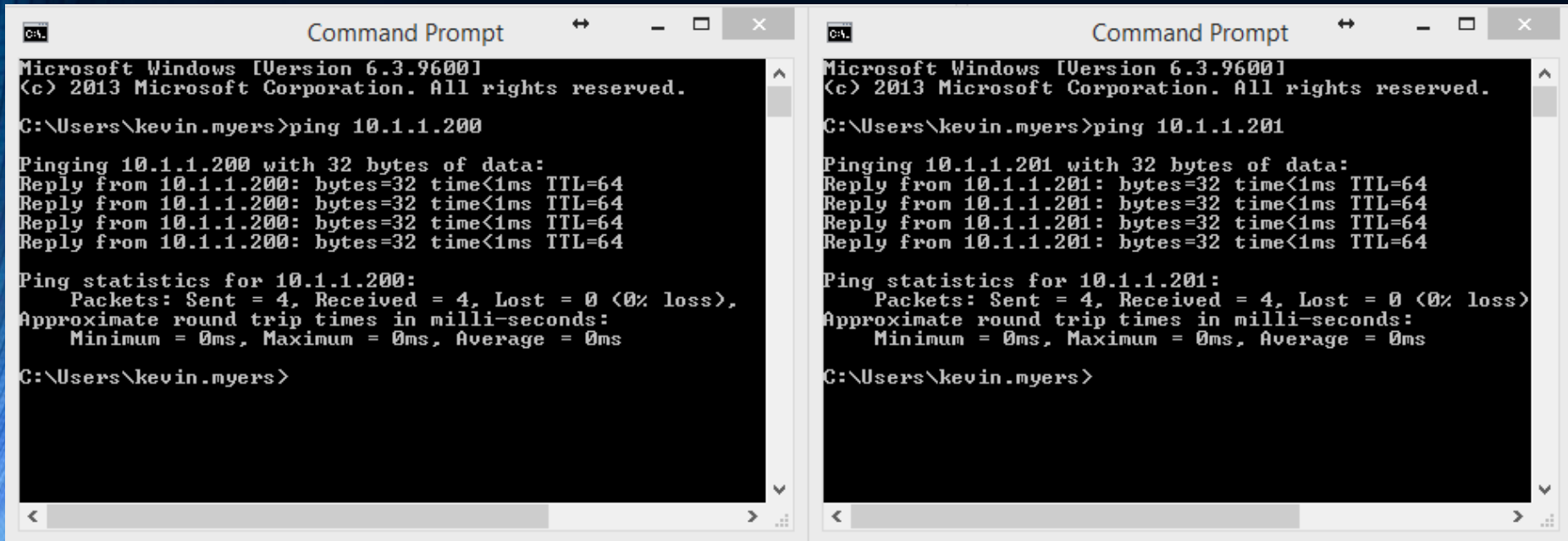
Configure Egress VLAN translation (**POD1 shown as an example**)

Ether1 – VLAN 201 to VLAN 101 (Use the correct port and VLAN for your POD)

Ether2 – VLAN 101 to VLAN 201 (Use the correct port and VLAN for your POD)

```
/interface ethernet switch egress-vlan-translation
add customer-vid=101 customer-vlan-format=untagged-or-tagged new-customer-vid=\
  201 ports=ether2 service-vlan-format=untagged-or-tagged
add customer-vid=201 customer-vlan-format=untagged-or-tagged new-customer-vid=\
  101 ports=ether1 service-vlan-format=untagged-or-tagged
```

- **Configuration Objective** - Translate VLAN from R1 to R2 in your POD and ping successfully between laptops
- **Step 9**
 - **Validation** – Ping from laptop to laptop (be sure to turn off software firewalls)



The image displays two side-by-side Windows Command Prompt windows. Both windows show the execution of a ping command from a user named kevin.myers. The left window shows a ping to 10.1.1.200, and the right window shows a ping to 10.1.1.201. Both pings are successful, with 4 packets sent and 4 received, resulting in 0% loss. The output for each ping includes the IP address, bytes, time, and TTL for each of the four replies, followed by a summary of the ping statistics.

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\kevin.myers>ping 10.1.1.200

Pinging 10.1.1.200 with 32 bytes of data:
Reply from 10.1.1.200: bytes=32 time<1ms TTL=64
Reply from 10.1.1.200: bytes=32 time<1ms TTL=64
Reply from 10.1.1.200: bytes=32 time<1ms TTL=64
Reply from 10.1.1.200: bytes=32 time<1ms TTL=64

Ping statistics for 10.1.1.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\kevin.myers>
```

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\kevin.myers>ping 10.1.1.201

Pinging 10.1.1.201 with 32 bytes of data:
Reply from 10.1.1.201: bytes=32 time<1ms TTL=64
Reply from 10.1.1.201: bytes=32 time<1ms TTL=64
Reply from 10.1.1.201: bytes=32 time<1ms TTL=64
Reply from 10.1.1.201: bytes=32 time<1ms TTL=64

Ping statistics for 10.1.1.201:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\kevin.myers>
```


How do we know that config worked?

- Packet Sniffer using port mirroring on the CRS
- Switch ports on CRS do not show up in packet sniffer unless a mirror is set up and pointed to the CPU

```
/interface ethernet switch
set bridge-type=customer-vid-used-as-lookup-vid \
  bypass-ingress-port-policing-for="" bypass-l2-security-check-filter-for="" \
  bypass-vlan-ingress-filter-for="" \
  drop-if-invalid-or-src-port-not-member-of-vlan-on-ports="" \
  drop-if-no-vlan-assignment-on-ports="" egress-mirror-ratio=1/1 \
  egress-mirror0=switch1-cpu,modified egress-mirror1=switch1-cpu,modified \
  fdb-uses=mirror0 forward-unknown-vlan=yes ingress-mirror-ratio=1/1 \
  ingress-mirror0=switch1-cpu,unmodified ingress-mirror1=\
  switch1-cpu,unmodified mac-level-isolation=yes \
  mirror-egress-if-ingress-mirrored=no mirror-tx-on-mirror-port=no \
  mirrored-packet-drop-precedence=green mirrored-packet-qos-priority=0 \
  multicast-lookup-mode=dst-ip-and-vid-for-ipv4 name=switch1 \
  override-existing-when-ufdb-full=no unicast-fdb-timeout=5m \
  unknown-vlan-lookup-mode=svl use-cvid-in-one2one-vlan-lookup=yes \
  use-svid-in-one2one-vlan-lookup=no vlan-uses=mirror0
```

Packet capture of ICMP between R1 and R2 in Wireshark

vlan-trans-8.pcap [Wireshark 1.12.3 (v1.12.3-0-gbb3e9a0 from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: icmp Expression... Clear Apply Save

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|------------|-------------|----------|--------|---------------------|
| 1 | 1970/006 | 10.1.1.200 | 10.1.1.201 | ICMP | 74 | Echo (ping) request |
| 2 | 1970/006 | 10.1.1.201 | 10.1.1.200 | ICMP | 74 | Echo (ping) reply |
| 19 | 1970/006 | 10.1.1.201 | 10.1.1.200 | ICMP | 74 | Echo (ping) request |
| 20 | 1970/006 | 10.1.1.200 | 10.1.1.201 | ICMP | 74 | Echo (ping) reply |
| 26 | 1970/006 | 10.1.1.200 | 10.1.1.201 | ICMP | 74 | Echo (ping) request |
| 27 | 1970/006 | 10.1.1.201 | 10.1.1.200 | ICMP | 74 | Echo (ping) reply |
| 49 | 1970/006 | 10.1.1.201 | 10.1.1.200 | ICMP | 74 | Echo (ping) request |
| 50 | 1970/006 | 10.1.1.200 | 10.1.1.201 | ICMP | 74 | Echo (ping) reply |
| 55 | 1970/006 | 10.1.1.200 | 10.1.1.201 | ICMP | 74 | Echo (ping) request |
| 56 | 1970/006 | 10.1.1.201 | 10.1.1.200 | ICMP | 74 | Echo (ping) reply |
| 71 | 1970/006 | 10.1.1.201 | 10.1.1.200 | ICMP | 74 | Echo (ping) request |
| 72 | 1970/006 | 10.1.1.200 | 10.1.1.201 | ICMP | 74 | Echo (ping) reply |
| 77 | 1970/006 | 10.1.1.200 | 10.1.1.201 | ICMP | 74 | Echo (ping) request |
| 78 | 1970/006 | 10.1.1.201 | 10.1.1.200 | ICMP | 74 | Echo (ping) reply |
| 95 | 1970/006 | 10.1.1.201 | 10.1.1.200 | ICMP | 74 | Echo (ping) request |
| 96 | 1970/006 | 10.1.1.200 | 10.1.1.201 | ICMP | 74 | Echo (ping) reply |
| 101 | 1970/006 | 10.1.1.200 | 10.1.1.201 | ICMP | 74 | Echo (ping) request |
| 102 | 1970/006 | 10.1.1.201 | 10.1.1.200 | ICMP | 74 | Echo (ping) reply |
| 120 | 1970/006 | 10.1.1.201 | 10.1.1.200 | ICMP | 74 | Echo (ping) request |
| 121 | 1970/006 | 10.1.1.200 | 10.1.1.201 | ICMP | 74 | Echo (ping) reply |
| 126 | 1970/006 | 10.1.1.201 | 10.1.1.200 | ICMP | 74 | Echo (ping) reply |
| 127 | 1970/006 | 10.1.1.200 | 10.1.1.201 | ICMP | 74 | Echo (ping) request |

Packet capture of ICMP from 10.1.1.200/24 (VLAN 101) to 10.1.1.201/24 (VLAN 201) in Wireshark

1 1970/006 18:59:34.090850 10.1.1.200 10.1.1.201 ICMP 74 Echo (ping) request id=0x1da7, seq=3837/64782, ttl=255 (reply in 2)

- Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
- Ethernet II, Src: Routerbo_6c:62:d8 (4c:5e:0c:6c:62:d8), Dst: Routerbo_6c:62:9c (4c:5e:0c:6c:62:9c)
- 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 101
 - 000. = Priority: Best Effort (default) (0)
 - ...0 = CFI: Canonical (0)
 - 0000 0110 0101 = ID: 101
 - Type: IP (0x0800)
- Internet Protocol Version 4, Src: 10.1.1.200 (10.1.1.200), Dst: 10.1.1.201 (10.1.1.201)
 - Version: 4
 - Header Length: 20 bytes
 - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transp
 - Total Length: 56
 - Identification: 0x7589 (30089)
 - Flags: 0x00
 - Fragment offset: 0
 - Time to live: 255
 - Protocol: ICMP (1)
 - Header checksum: 0x2ea9 [validation disabled]
 - Source: 10.1.1.200 (10.1.1.200)
 - Destination: 10.1.1.201 (10.1.1.201)
 - [Source GeoIP: Unknown]
 - [Destination GeoIP: Unknown]
- Internet Control Message Protocol

| | | |
|------|---|------------------|
| 0000 | 4c 5e 0c 6c 62 9c 4c 5e 0c 6c 62 d8 81 00 00 65 | L^..b.L^..b....e |
| 0010 | 08 00 45 00 00 38 75 89 00 00 ff 01 2e a9 0a 01 | ..E..8u. |
| 0020 | 01 c8 0a 01 01 c9 08 00 56 d3 1d a7 0e fd aa b9 | V..... |
| 0030 | b7 7d 64 71 35 d3 d5 a7 0a 06 7c b1 11 71 2d c2 | .}dq5...q- |
| 0040 | 83 b0 f0 46 34 c7 36 7a fd 40 | ...F4.6z .@ |

Packet capture of ICMP from 10.1.1.201/24 (VLAN 201) to 10.1.1.200/24 (VLAN 101) in Wireshark

2 1970/006 18:59:34.091060 10.1.1.201 10.1.1.200 ICMP 74 Echo (ping) reply id=0x1da7, seq=3837/64782, ttl=64 (request in 1)

Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)

Ethernet II, Src: Routerbo_6c:62:9c (4c:5e:0c:6c:62:9c), Dst: Routerbo_6c:62:d8 (4c:5e:0c:6c:62:d8)

802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 201

- 000. = Priority: Best Effort (default) (0)
- ...0 = CFI: Canonical (0)
- 0000 1100 1001 = ID: 201
- Type: IP (0x0800)

Internet Protocol Version 4, Src: 10.1.1.201 (10.1.1.201), Dst: 10.1.1.200 (10.1.1.200)

- Version: 4
- Header Length: 20 bytes
- Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transp
- Total Length: 56
- Identification: 0x7f3b (32571)
- Flags: 0x00
- Fragment offset: 0
- Time to live: 64
- Protocol: ICMP (1)
- Header checksum: 0xe3f7 [validation disabled]
- Source: 10.1.1.201 (10.1.1.201)
- Destination: 10.1.1.200 (10.1.1.200)
- [Source GeoIP: Unknown]
- [Destination GeoIP: Unknown]

Internet Control Message Protocol

0000 4c 5e 0c 6c 62 d8 4c 5e 0c 6c 62 9c 81 00 00 c9 L^..b.L^..b...
0010 08 00 45 00 00 38 7f 3b 00 00 40 01 e3 f7 0a 01 ..E..8.; ..@....
0020 01 c9 0a 01 01 c8 00 00 5e d3 1d a7 0e fd aa b9^.....
0030 b7 7d 64 71 35 d3 d5 a7 0a 06 7c b1 11 71 2d c2 .}dq5... ..|..q-
0040 83 b0 f0 46 34 c7 36 7a fd 40 ...F4.6z ..@

Configuration example - Bridging

This configuration can be used in almost any RouterBoard to translate between VLANs using a bridge.

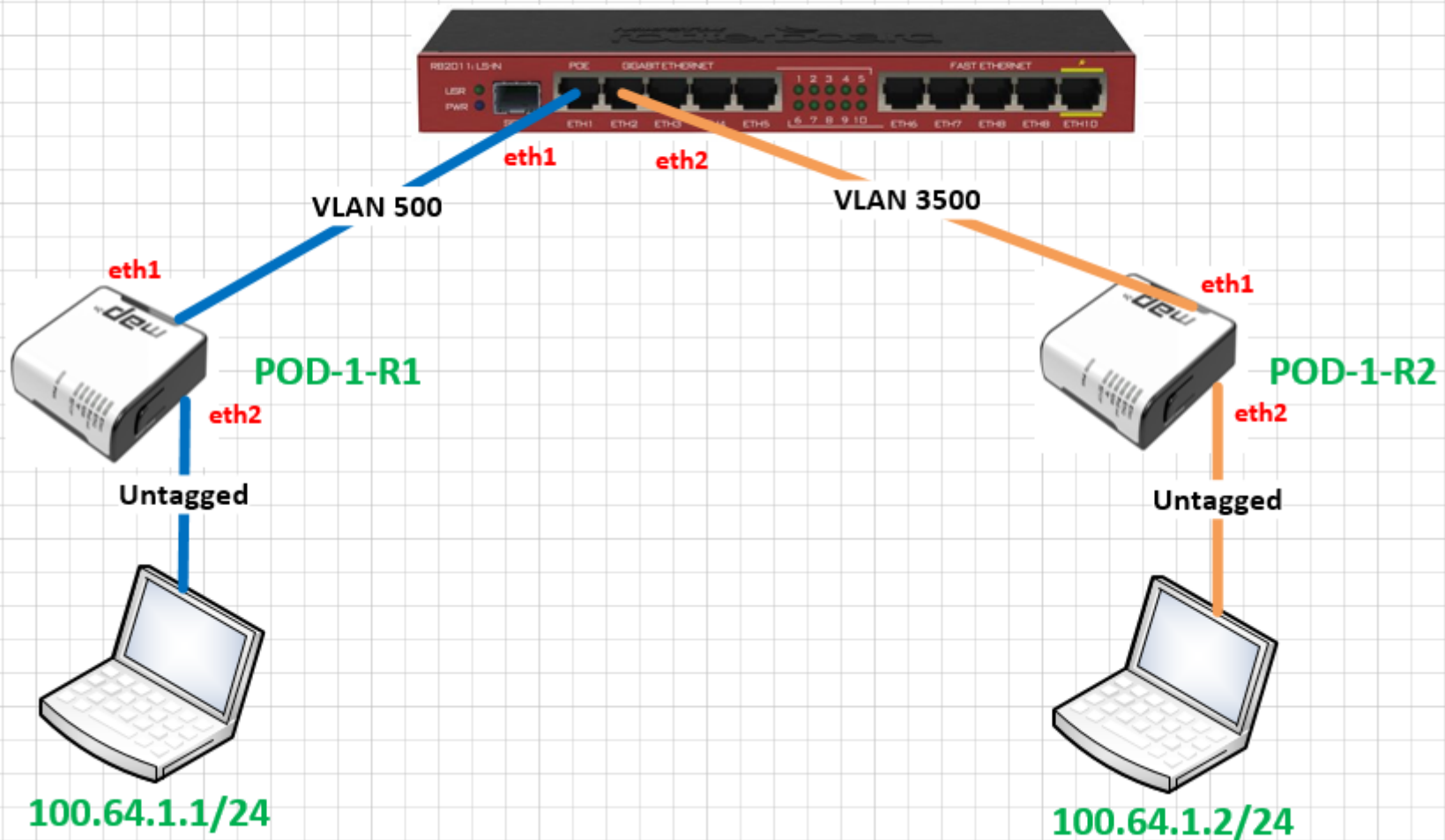
Create VLAN 500 and 3100 interface VLANs

```
/interface vlan
add interface=ether1 name=Vlan500 vlan-id=500
add interface=ether2 name=Vlan3500 vlan-id=3500
```

Create Bridge and add VLAN interface ports

```
/interface bridge
add name=Loopback0
add name=Vlan500-to-3500-Bridge
/interface bridge port
add bridge=Vlan500-to-3500-Bridge interface=Vlan3500
add bridge=Vlan500-to-3500-Bridge interface=Vlan500
```


Lab Topology – Physical / Logical Connectivity for VLAN rewrite using bridging



When to use switching vs. bridging – what is the difference?

- **Speed and packet latency is the answer** – switching is done in hardware and so the traffic going through the VLAN translation can happen at wire speed without CPU load or latency of processing. 1 Gigabit in the case of the CRS

When to use switching vs. bridging – what is the difference?

- Use CRS switches for applications that require higher throughput and port density.
 - Data Center Interconnect (DCI)
 - Service Provider core or aggregation layers
 - WISP Core layer to support acquisition and migration of a new WISP.
- Use routers for lower speed scenarios like at the edge.

MUM Giveaway #1

(4) Google Virtual Reality Headsets



MUM Giveaway #2

(4) R/C Quadcopters



Questions?

The content of this presentation will be available at
iparchitech.com

Please come see us at the IP ArchiTechs booth in the Exhibitor Hall

Email: kevin.myers@iparchitech.com

Office: (303) 590-9943

Web: www.iparchitech.com

Thank you for your time and enjoy the MUM!!