IMPLEMENTATION OF SEAMLESS LAYER 2 VPN CONNECTIVITY:

A MikroTik-Based MPLS and VPLS Workshop"

SUBASH SUBEDI

Abstract

This project demonstrates the design, configuration, and verification of a Multiprotocol Label Switching (MPLS) core to provide Virtual Private LAN Service (VPLS) using MikroTik RouterOS. The primary objective is to establish a transparent Layer 2 bridge between geographically separated Customer Edge (CE) sites, allowing them to operate within the same broadcast domain over a Provider (P) network.

The implementation utilizes OSPF as the Interior Gateway Protocol (IGP) for core reachability and Label Distribution Protocol (LDP) for signaling and label exchange. Key technical configurations include the optimization of MTU settings (1508 bytes) to prevent fragmentation of encapsulated frames and the binding of VPLS instances to local bridge interfaces on Provider Edge (PE) routers.

The success of the deployment is validated through OSPF adjacency checks, LDP session monitoring, and end-to-end ICMP connectivity between customer sites. Furthermore, a deep-packet analysis via Wireshark confirms the encapsulation of Ethernet frames within MPLS labels (the "inner" and "outer" label stack), proving the core's ability to remain "IP-unaware" while switching customer traffic. This setup serves as a scalable model for Service Providers to offer high-performance site-to-site connectivity.

Table of Contents

List of Figures

1. Introduction

This document provides a comprehensive guide to setting up MPLS (Multiprotocol Label Switching) and VPLS (Virtual Private LAN Service) on MikroTik routers using GNS3. The setup consists of a Provider (P) router (Main), two Provider Edge (PE) routers, and two Customer Edge (CE) routers. VPLS creates a transparent Layer 2 Ethernet bridge, allowing CE1 and CE2 to communicate on the same subnet across an MPLS core.

2. Prerequisites

Emulation: GNS3

Images: MikroTik CHR v7.x (Tested on 7.21rc4).

Protocol Knowledge: OSPF (IGP), LDP (Label Distribution), and VPLS (Tunneling).

3. Network Topology Overview

Main (P): Loopback 1.1.1.1/32

PE1: Loopback 2.2.2.2/32 | PE2: Loopback 3.3.3.3/32

CE1: 192.168.10.1/24 | CE2: 192.168.10.2/24

Core Links: * Main (ether1) to PE1 (ether1): 10.0.1.0/30

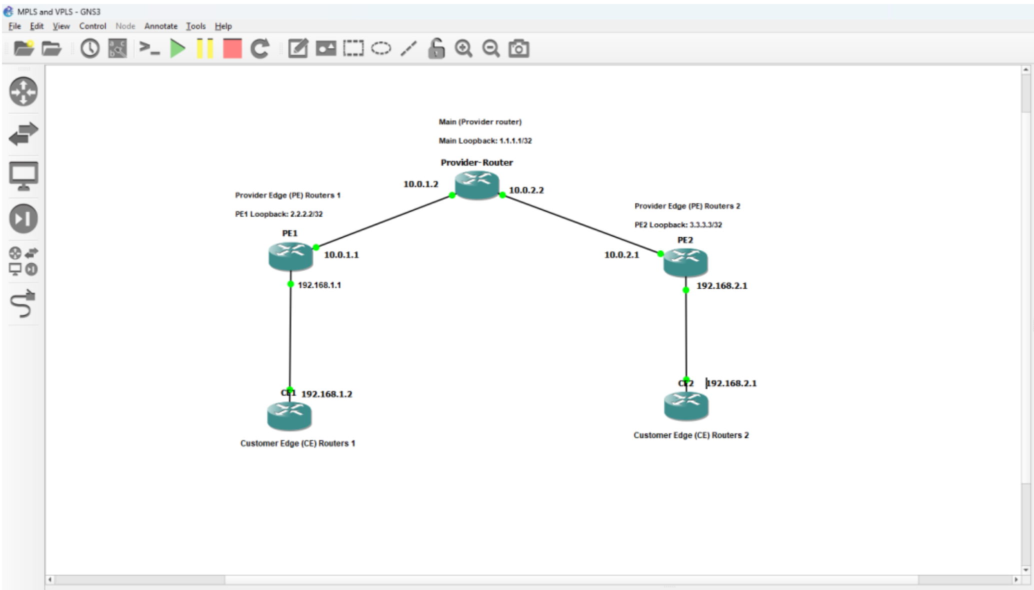


Figure 1

4. Configuring the Routers

Configurations are provided in RouterOS CLI format. Apply them via terminal or export/import scripts.

4.1 Main Router Configuration

The Main router handles label switching between PE1 and PE2..

❖ **Set Identity and Loopback:**

```
/system identity set name=Main
/interface bridge add name=loopback
/ip address add address=1.1.1.1/32 interface=loopback
```

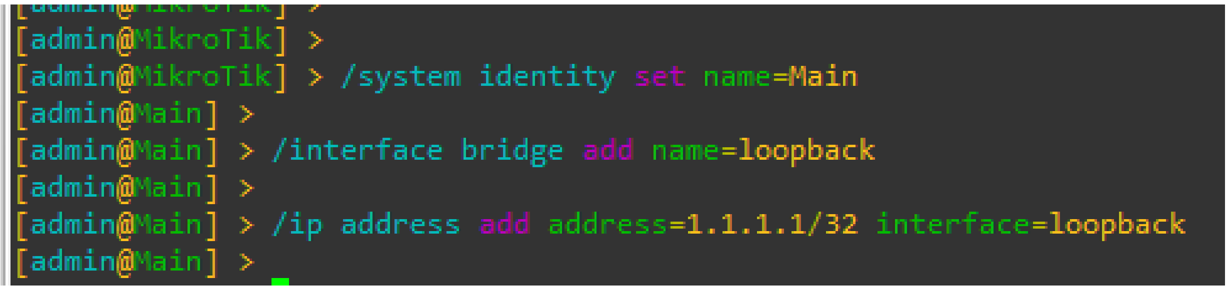


Figure 2 : CMD to set Identity and Loopback (Main Router)

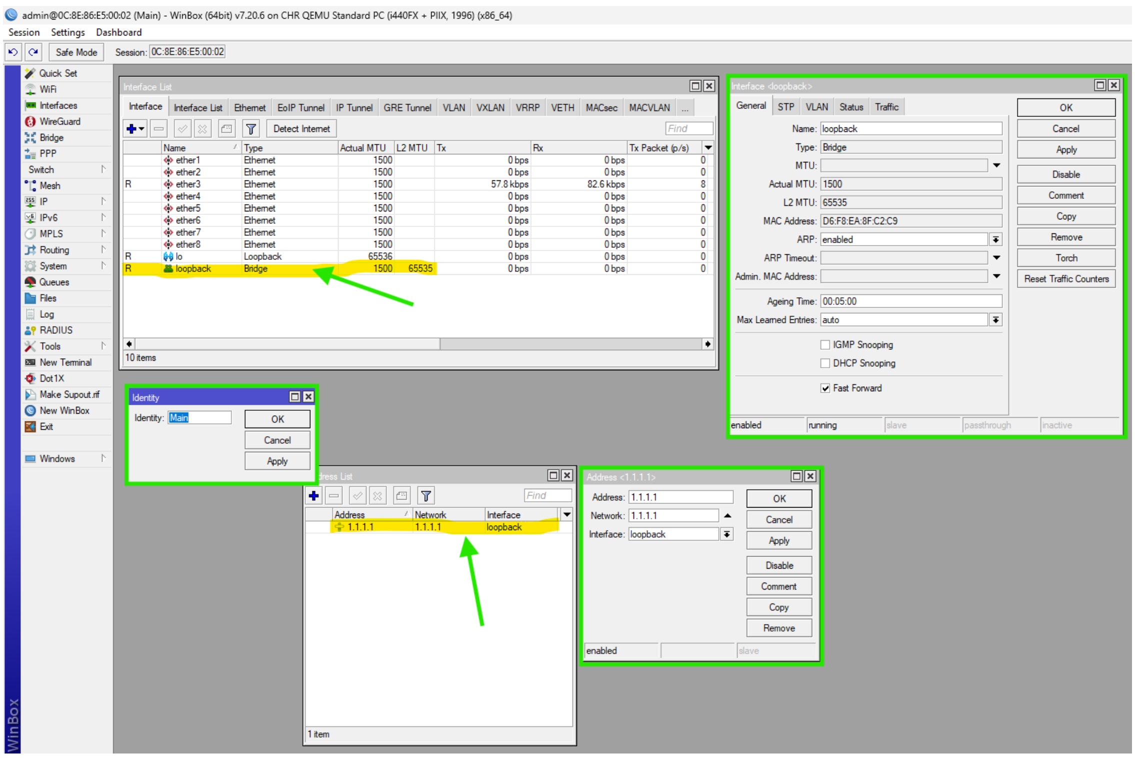


Figure 3: WINBOX to set Identity and Loopback (Main Router)

❖ Core Interface IPs (Connecting to PE1 and PE2):

```
/ip address add address=10.0.1.2/30 interface=ether1 comment="To PE1"

/ip address add address=10.0.2.2/30 interface=ether2 comment="To PE2"
```

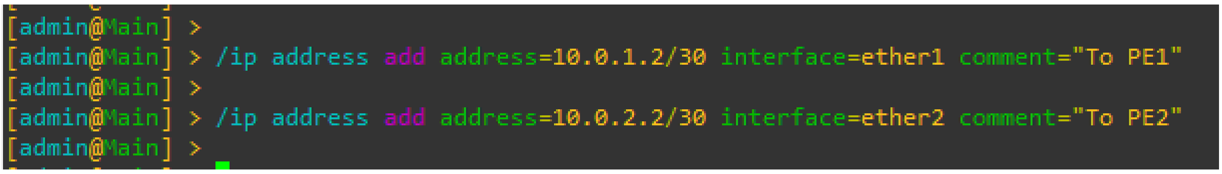


Figure 4: CMD to set Core Interface IPs (Connecting to PE1 and PE2) (Main Router)

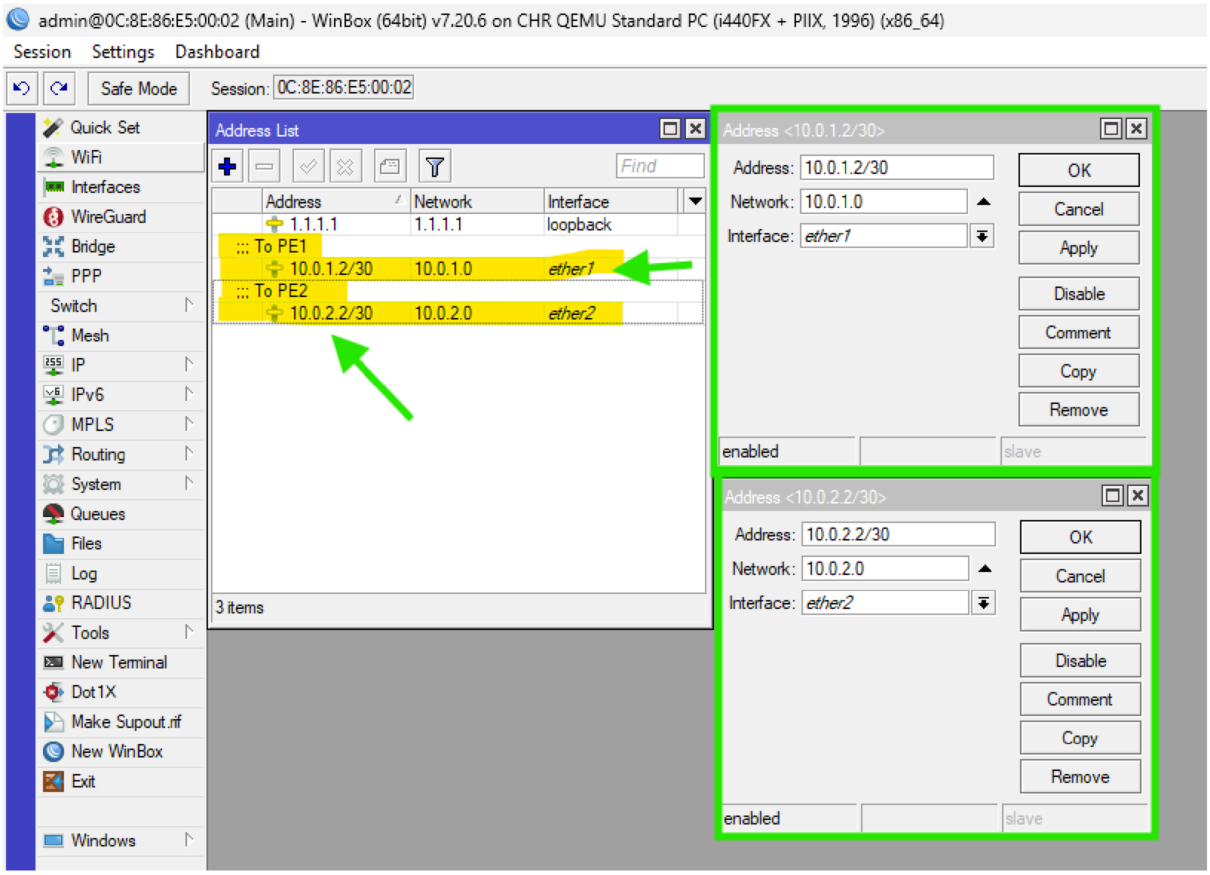
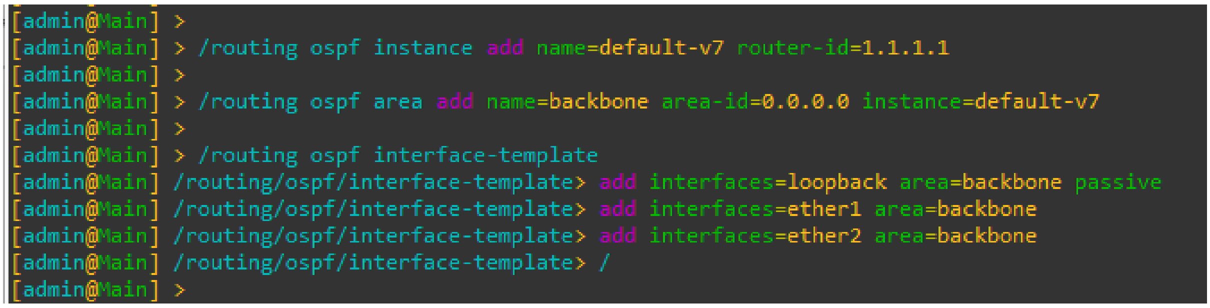


Figure 5: WINBOX Process to set Core Interface IPs (Connecting to PE1 and PE2) (Main Router)

❖ OSPF Setup:

```
/routing ospf instance add name=default-v7 router-id=1.1.1.1
/routing ospf area add name=backbone area-id=0.0.0.0 instance=default-v7

/routing ospf interface-template
add interfaces=loopback area=backbone passive
add interfaces=ether1 area=backbone
add interfaces=ether2 area=backbone
```



```
[admin@Main] >
[admin@Main] > /routing ospf instance add name=default-v7 router-id=1.1.1.1
[admin@Main] >
[admin@Main] > /routing ospf area add name=backbone area-id=0.0.0.0 instance=default-v7
[admin@Main] >
[admin@Main] > /routing ospf interface-template
[admin@Main] /routing/ospf/interface-template> add interfaces=loopback area=backbone passive
[admin@Main] /routing/ospf/interface-template> add interfaces=ether1 area=backbone
[admin@Main] /routing/ospf/interface-template> add interfaces=ether2 area=backbone
[admin@Main] /routing/ospf/interface-template> /
[admin@Main] >
```

Figure 6:CMD for OSPF setup (Main Router)

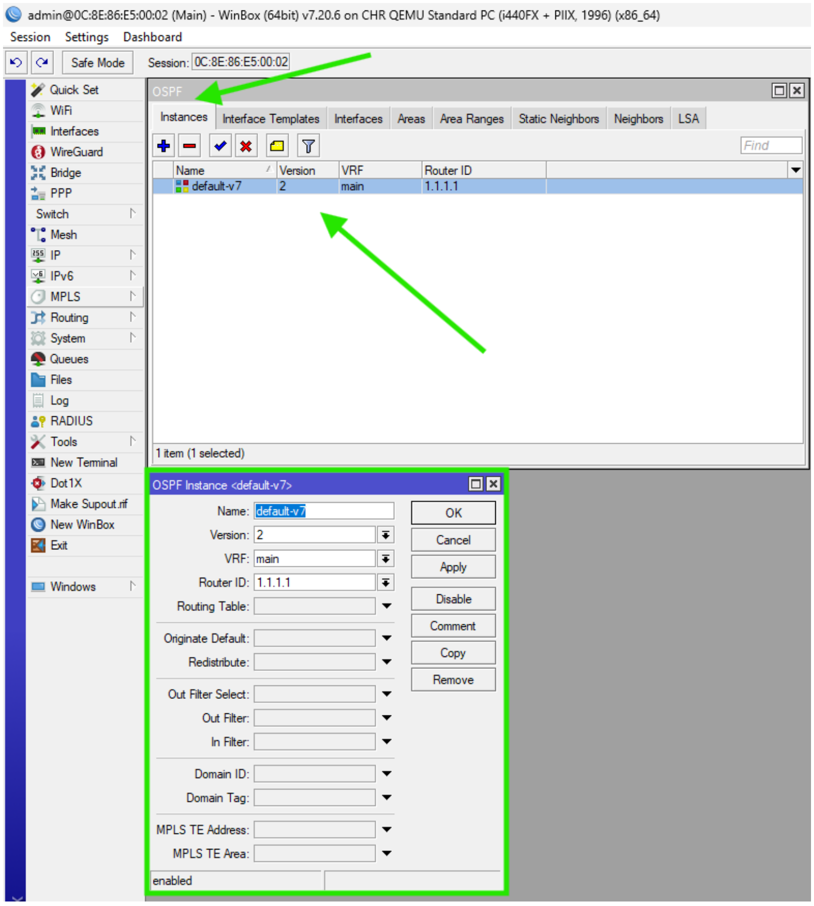


Figure 7: WINBOX Process OSPF SETUP (Main Router)

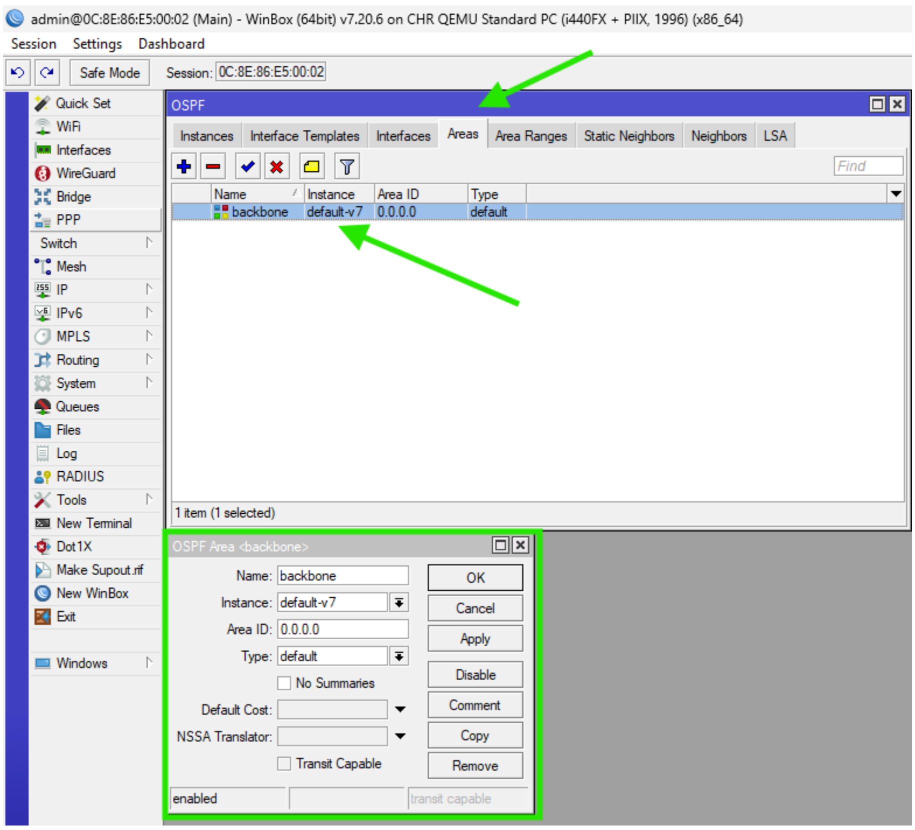


Figure 8: WINBOX Process for OSPF SETUP (Main Router)

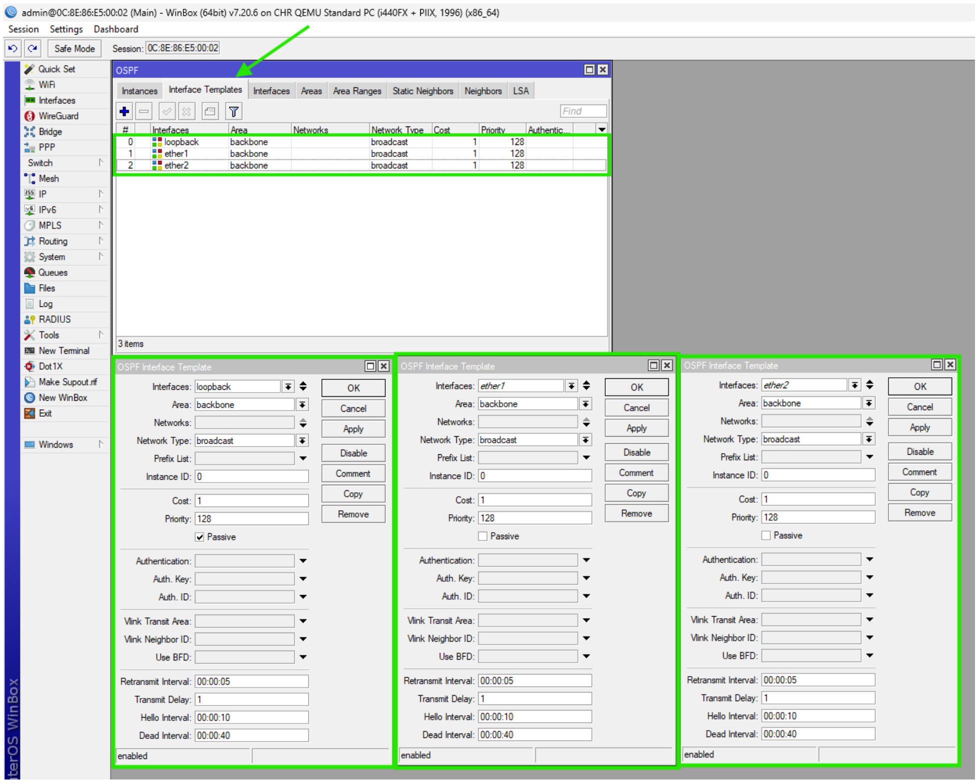


Figure 9: WINBOX Process for OSPF SETUP (Main Router)

❖ MPLS and LDP:

```
/mpls ldp add lsr-id=1.1.1.1 transport-addresses=1.1.1.1

/mpls ldp set [find lsr-id=1.1.1.1] disabled=no
```

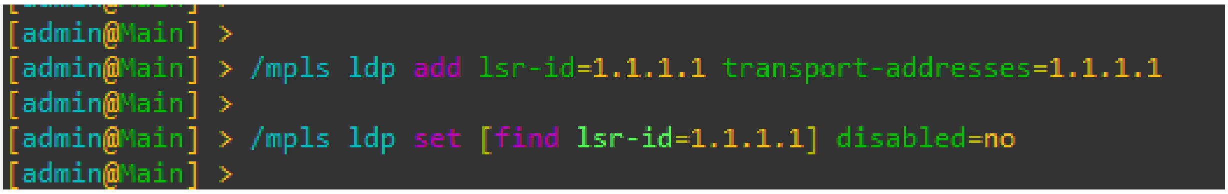


Figure 10: CMD to setup MPLS and LDP (Main Router)

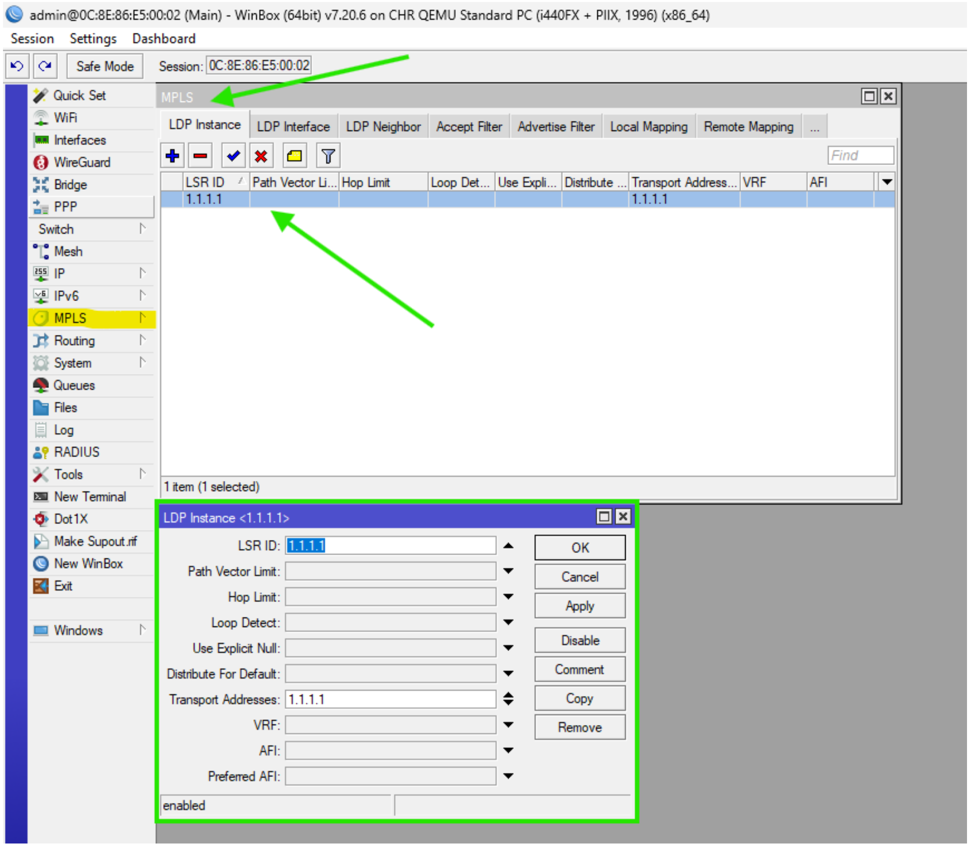


Figure 11: WINBOX Process for Setup MPLS and LDP (Main Router)

❖ Enable LDP on Interfaces

```
/mpls ldp interface add interface=ether1
/mpls ldp interface add interface=ether2
```

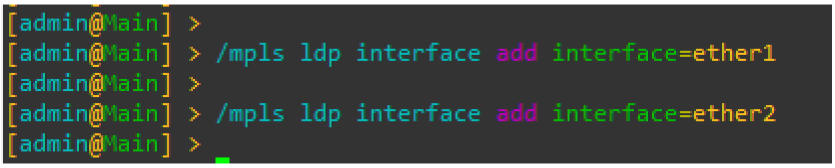


Figure 12: CMD to Enable LDP on Interfaces (Main Router)

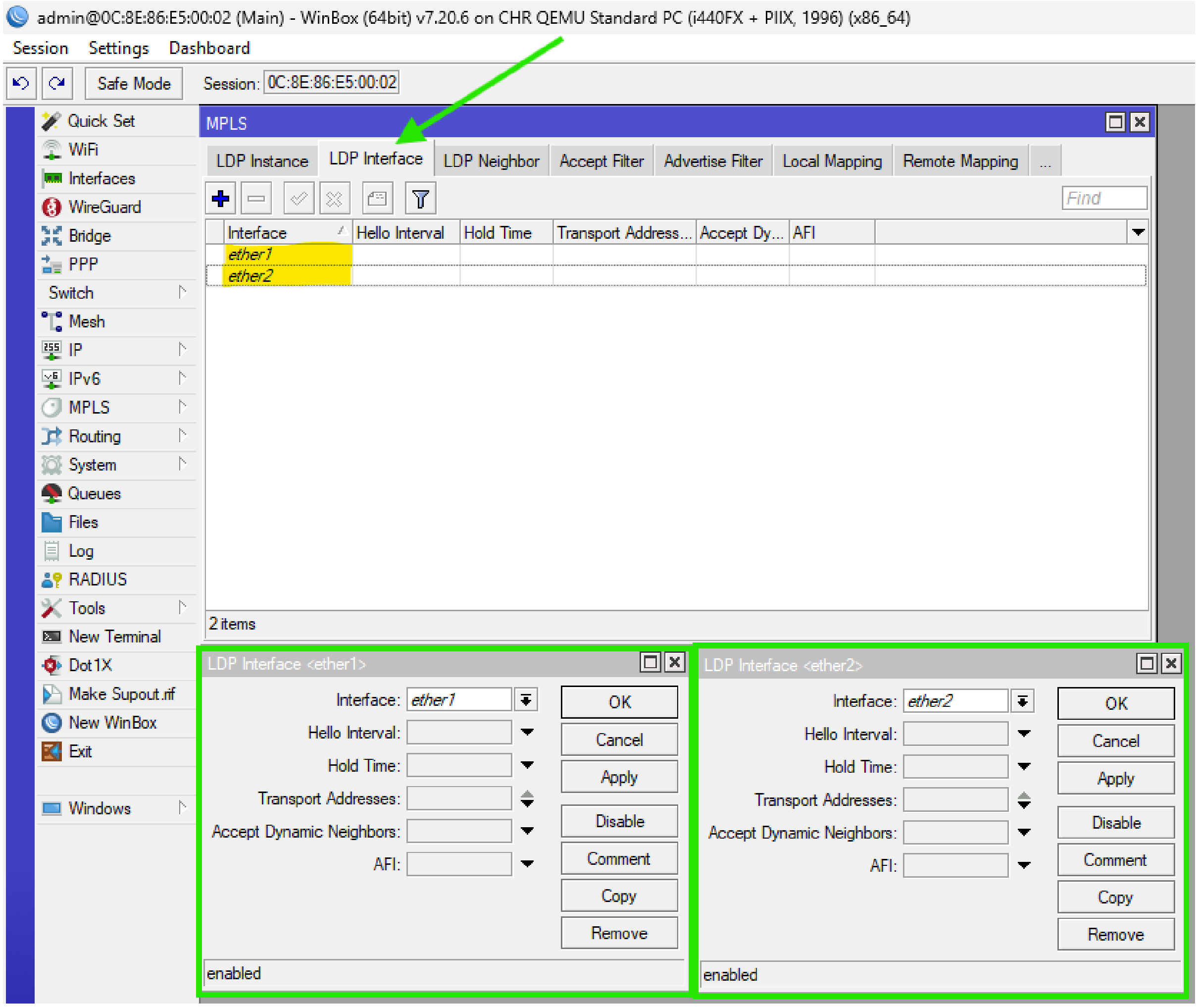


Figure 13: WINBOX Process to Enable LDP on Interfaces (Main Router)

❖ Set MPLS MTU (Crucial for VPLS to prevent fragmentation)

```
/mpls interface set [find where interface=ether1] mpls-mtu=1508
/mpls interface set [find where interface=ether2] mpls-mtu=1508

Or

/mpls interface add interface=ether1 mpls-mtu=1508
/mpls interface add interface=ether2 mpls-mtu=1508
```

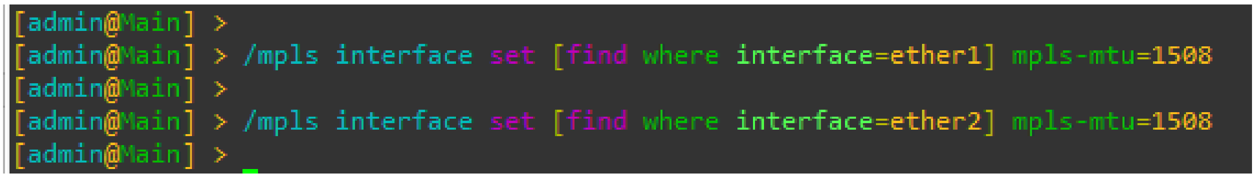


Figure 14: CMD to Setup MPLS MTU (Crucial for VPLS to prevent fragmentation) i (Main Router)

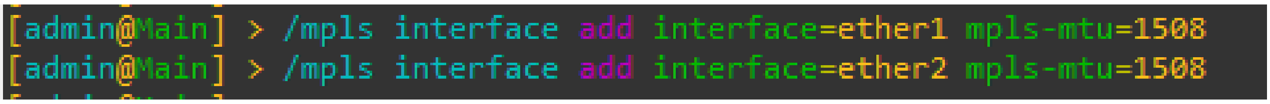


Figure 15: CMD to Setup MPLS MTU (Crucial for VPLS to prevent fragmentation) ii (Main Router)

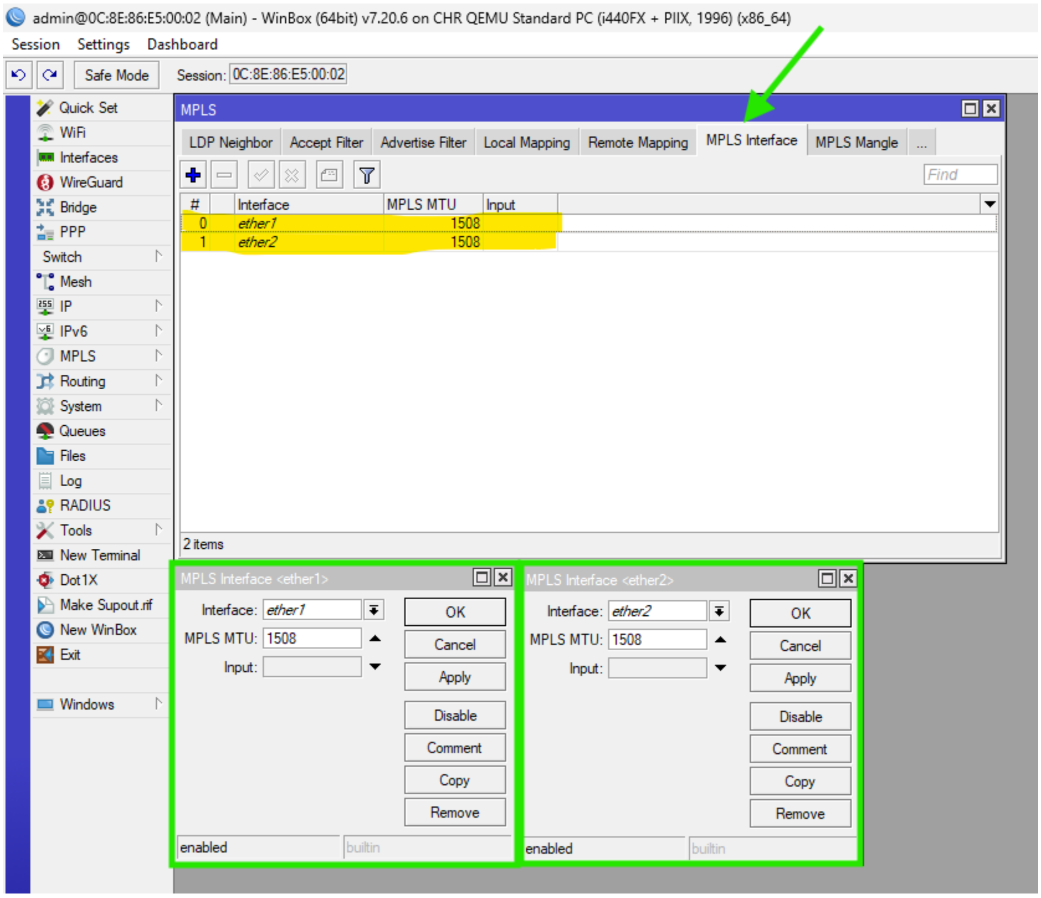


Figure 16: WINBOX to Setup MPLS MTU (Crucial for VPLS to prevent fragmentation) ii (Main Router)

4.2 PE1 Router Configuration

PE1 bridges the local customer (CE1) into the MPLS VPLS tunnel.

❖ Set Identity and Loopback:

```
/system identity set name=PE1
/interface bridge add name=loopback
/ip address add address=2.2.2.2/32 interface=loopback
```

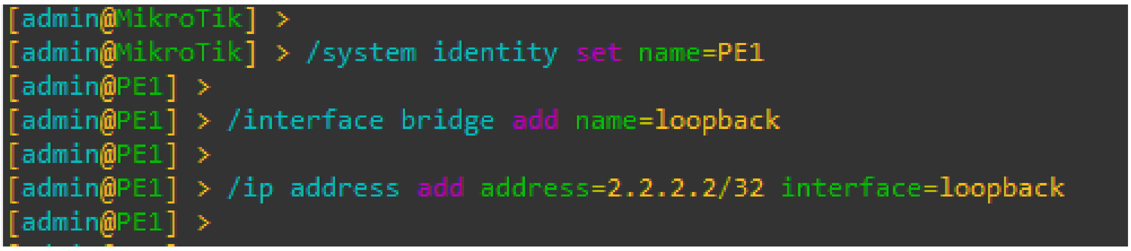


Figure 17: CMD to Set Identity and Loopback (PE1 Router)

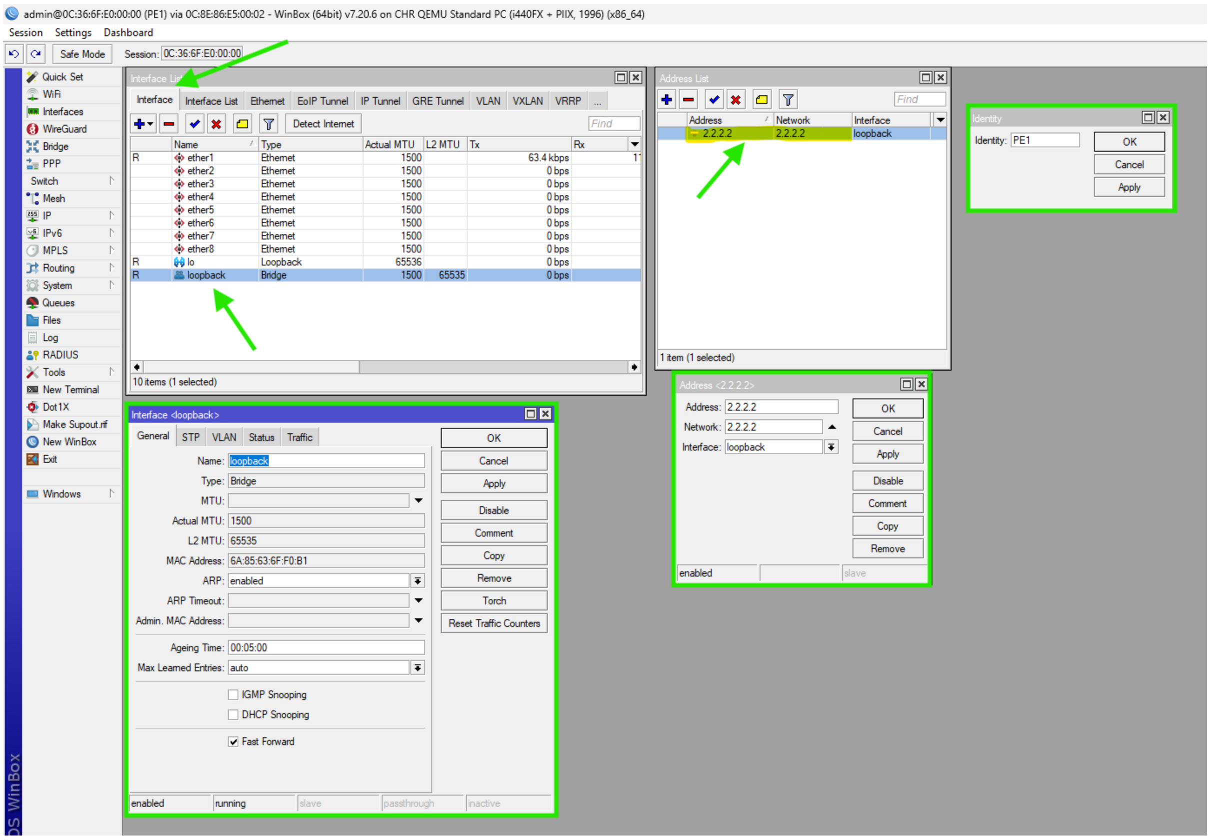


Figure 18: WINBOX Process to Set Identity and Loopback (PE1 Router)

❖ Interface IPs:

```
/ip address add address=10.0.1.1/30 interface=ether1 comment="Connect to Main"
```

Figure 19: CMD to set Interface Ips (PE1 Router)

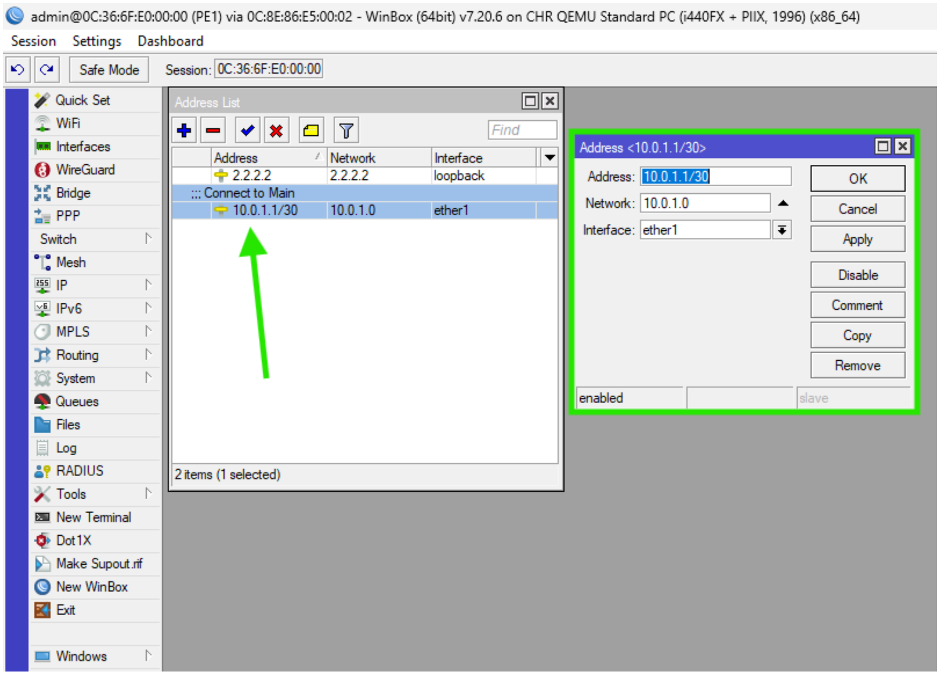


Figure 20: WINBOX Process to set Interface IPs (PE1 Router)

❖ OSPF Setup (To reach Main and PE2):

```
/routing ospf instance add name=default-v7 router-id=2.2.2.2
/routing ospf area add name=backbone area-id=0.0.0.0 instance=default-v7
/routing ospf interface-template
add interfaces=loopback area=backbone passive
add interfaces=ether1 area=backbone
```

```
[admin@PE1] >
[admin@PE1] > /routing ospf instance add name=default-v7 router-id=2.2.2.2
[admin@PE1] >
[admin@PE1] > /routing ospf area add name=backbone area-id=0.0.0.0 instance=default-v7
[admin@PE1] >
[admin@PE1] > /routing ospf interface-template
[admin@PE1] /routing/ospf/interface-template> add interfaces=loopback area=backbone passive
[admin@PE1] /routing/ospf/interface-template> add interfaces=ether1 area=backbone
[admin@PE1] /routing/ospf/interface-template> /
[admin@PE1] >
```

Figure 21: CMD for OSPF Setup (PE1 Router)

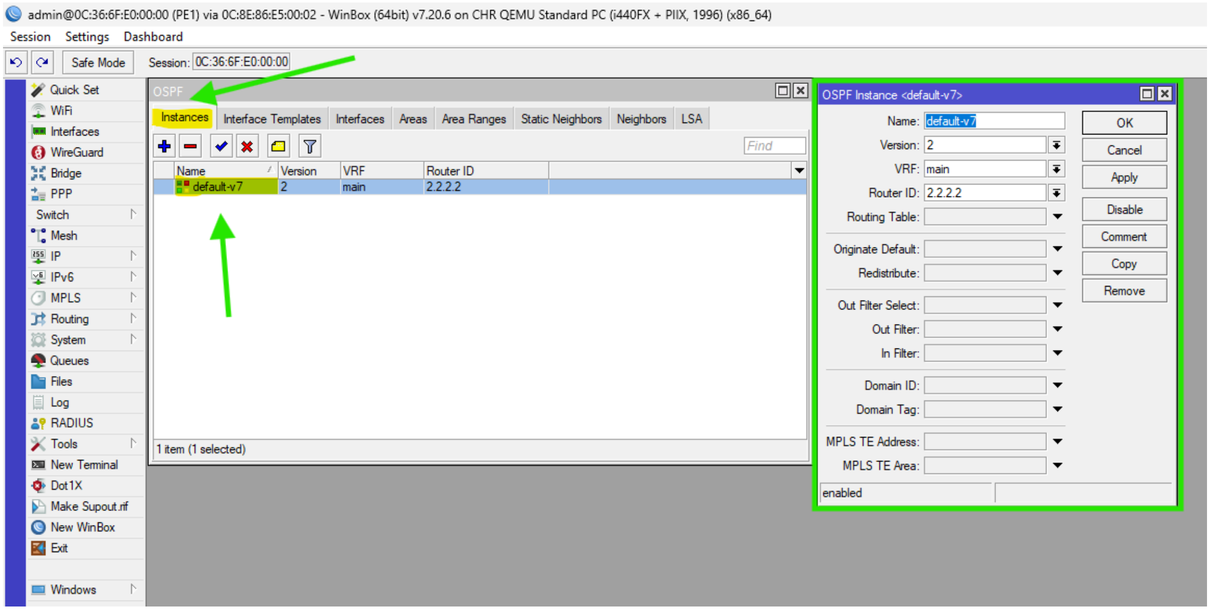


Figure 22: WINBOX Process for OSPF Setup i (PE1 Router)

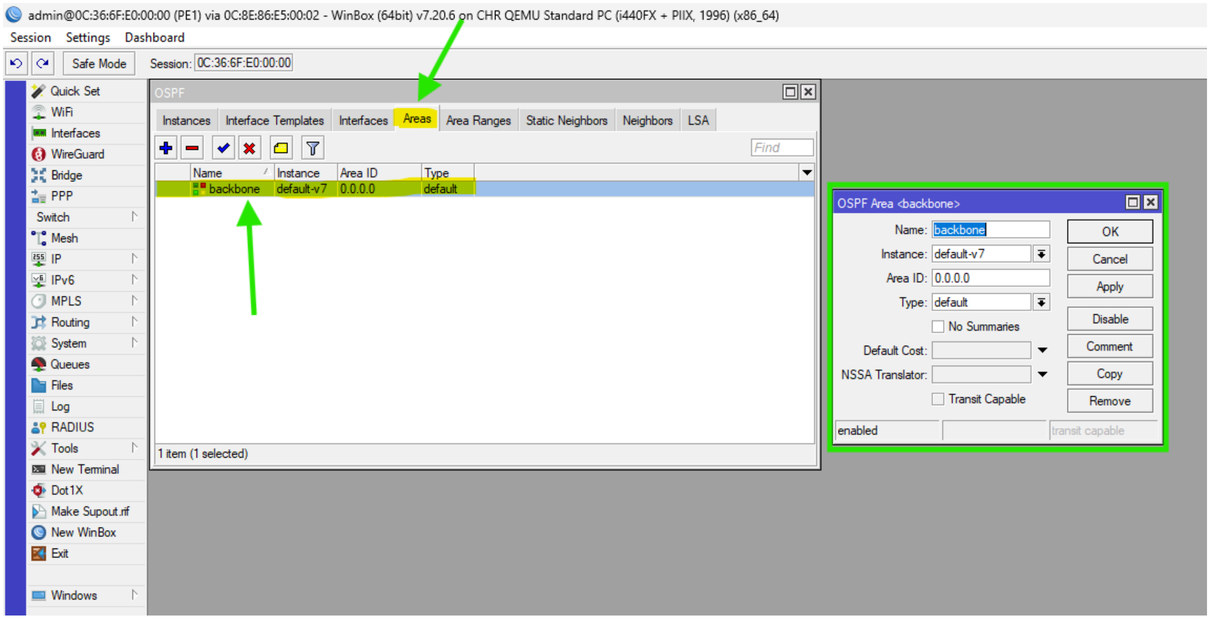


Figure 23: WINBOX Process for OSPF Setup ii (PE1 Router)

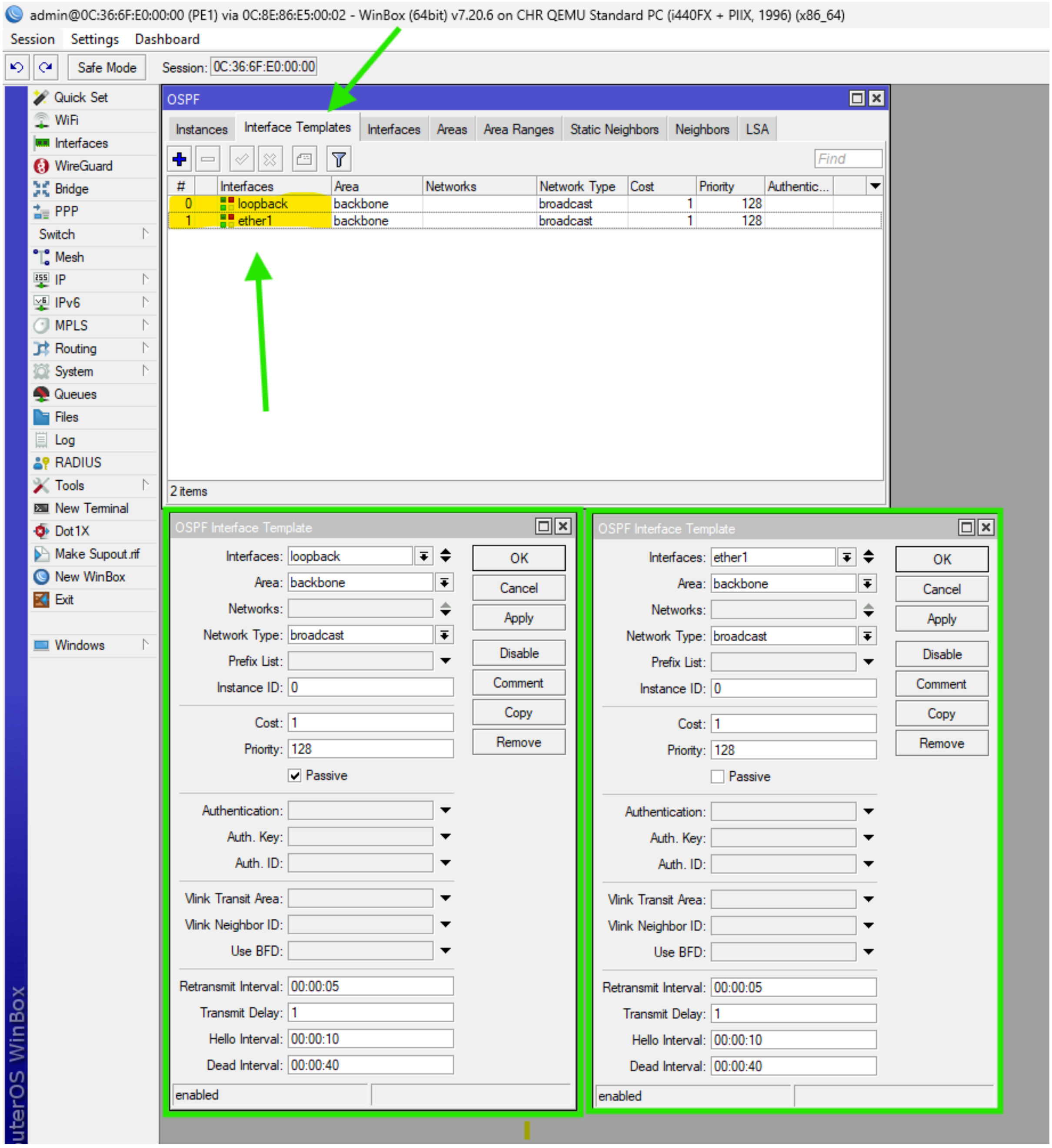


Figure 24: WINBOX Process for OSPF Setup iii (PE1 Router)

❖ MPLS and LDP:

```
/mpls ldp add lsr-id=2.2.2.2 transport-addresses=2.2.2.2
/mpls ldp set [find lsr-id=2.2.2.2] disabled=no
/mpls ldp interface add interface=ether1
```

```
[admin@PE1] >
[admin@PE1] > /mpls ldp add lsr-id=2.2.2.2 transport-addresses=2.2.2.2
[admin@PE1] >
[admin@PE1] > /mpls ldp set [find lsr-id=2.2.2.2] disabled=no
[admin@PE1] >
[admin@PE1] > /mpls ldp interface add interface=ether1
[admin@PE1] >
```

Figure 25: CMD to Setup MPLS & LDP (PE1 Router)

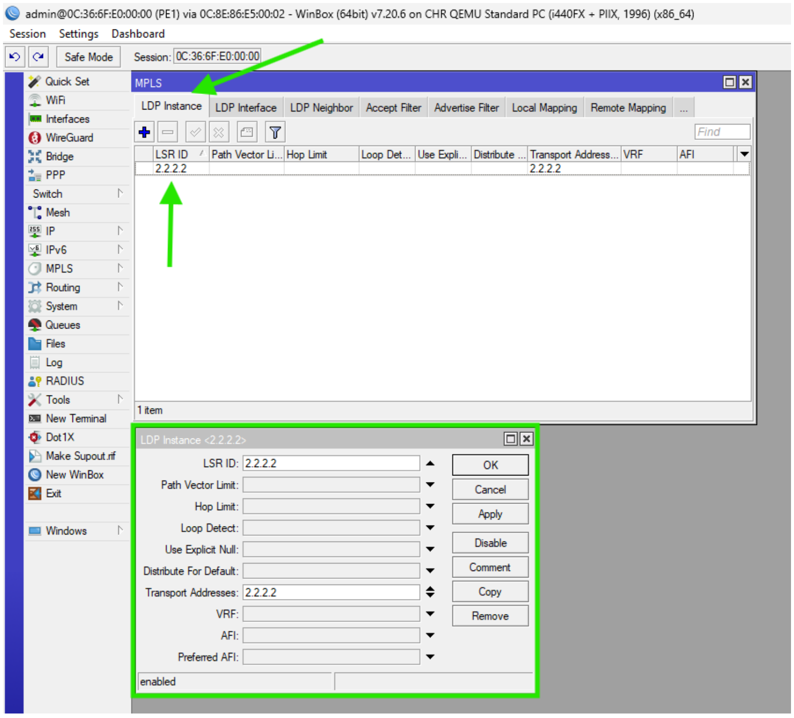


Figure 26: WINBOX Process for Setup MPLS & LDP i (PE1 Router)

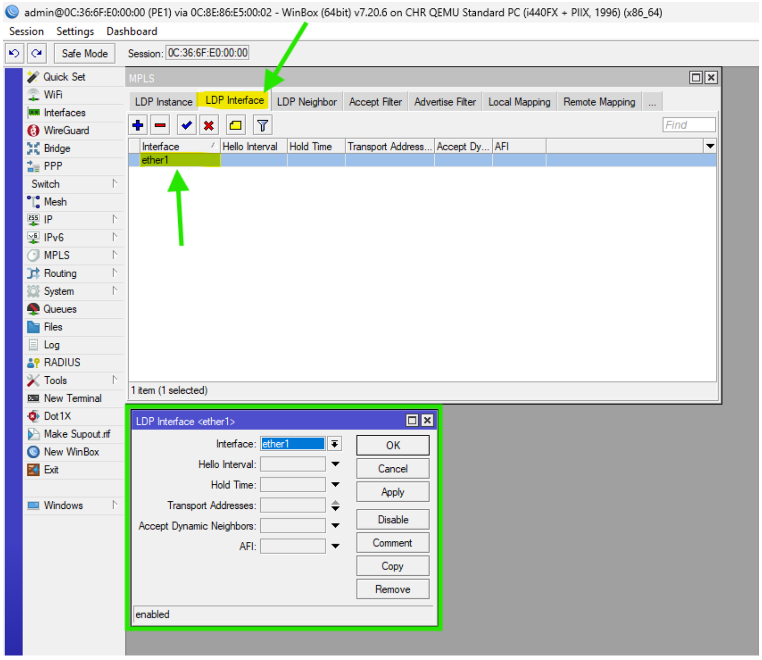


Figure 27: WINBOX Process for Setup MPLS & LDP ii (PE1 Router)

❖ VPLS Setup (Note: 'peer' instead of 'remote-peer'):

```
/interface vpls add name=vpls1 peer=3.3.3.3 vpls-id=10:1 disabled=no
```

Figure 28: CMD for VPLS Setup (Note: 'peer' instead of 'remote-peer') (PE1 Router)

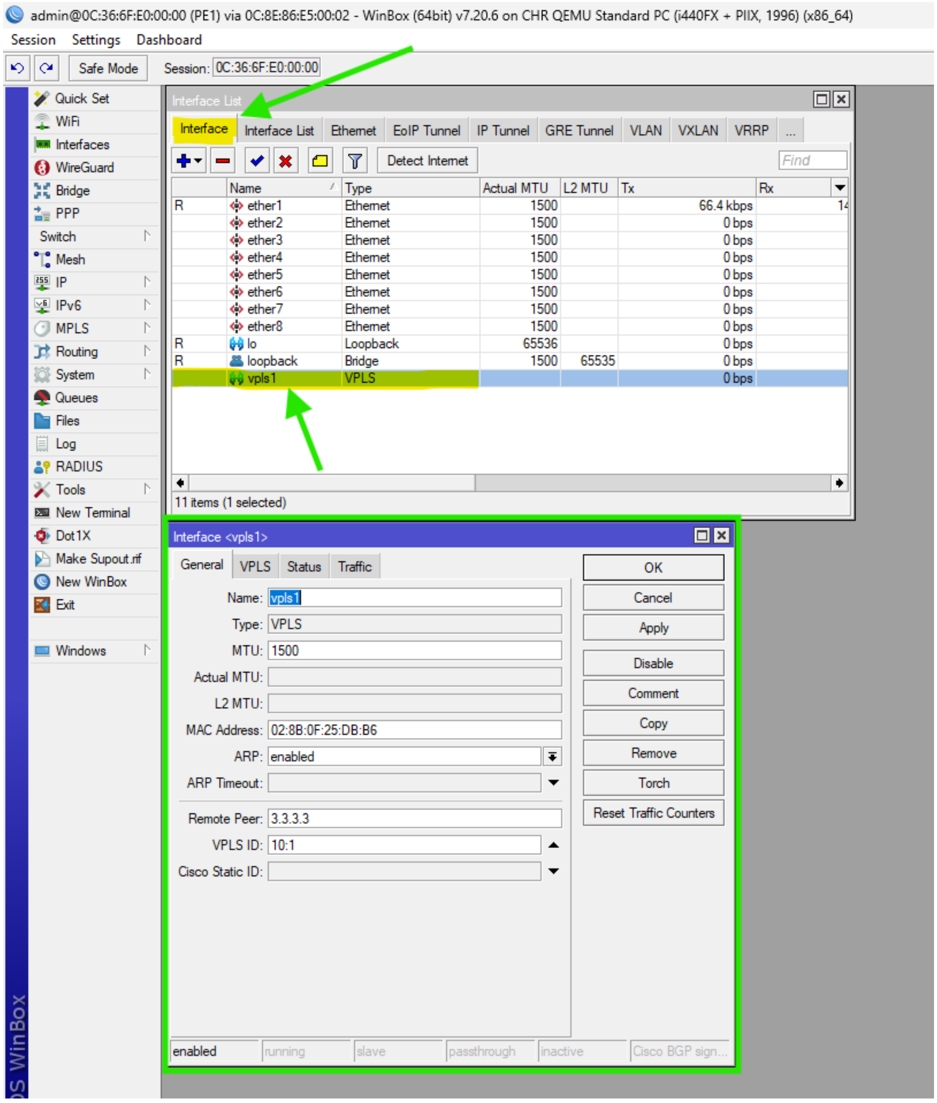


Figure 29: WINBOX Process for VPLS Setup (Note: 'peer' instead of 'remote-peer') (PE1 Router)

❖ Customer Bridge (Binds the VPLS tunnel to the Customer Port):

```
/interface bridge add name=br-customer
/interface bridge port add bridge=br-customer interface=ether2
/interface bridge port add bridge=br-customer interface=vpls1
```

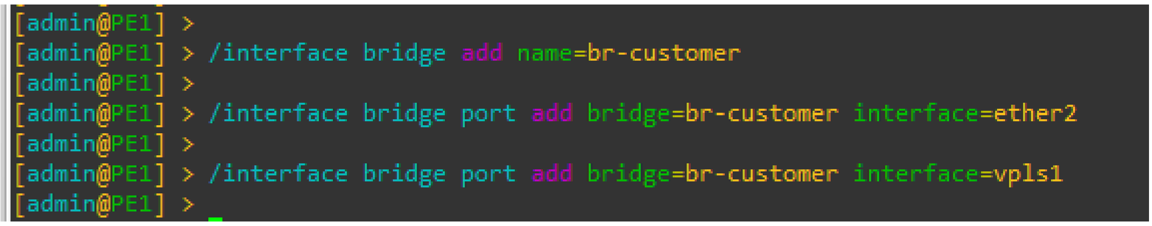


Figure 30:CMD to Setup Customer Bridge (Binds the VPLS tunnel to the Customer Port) (PE1 Router)

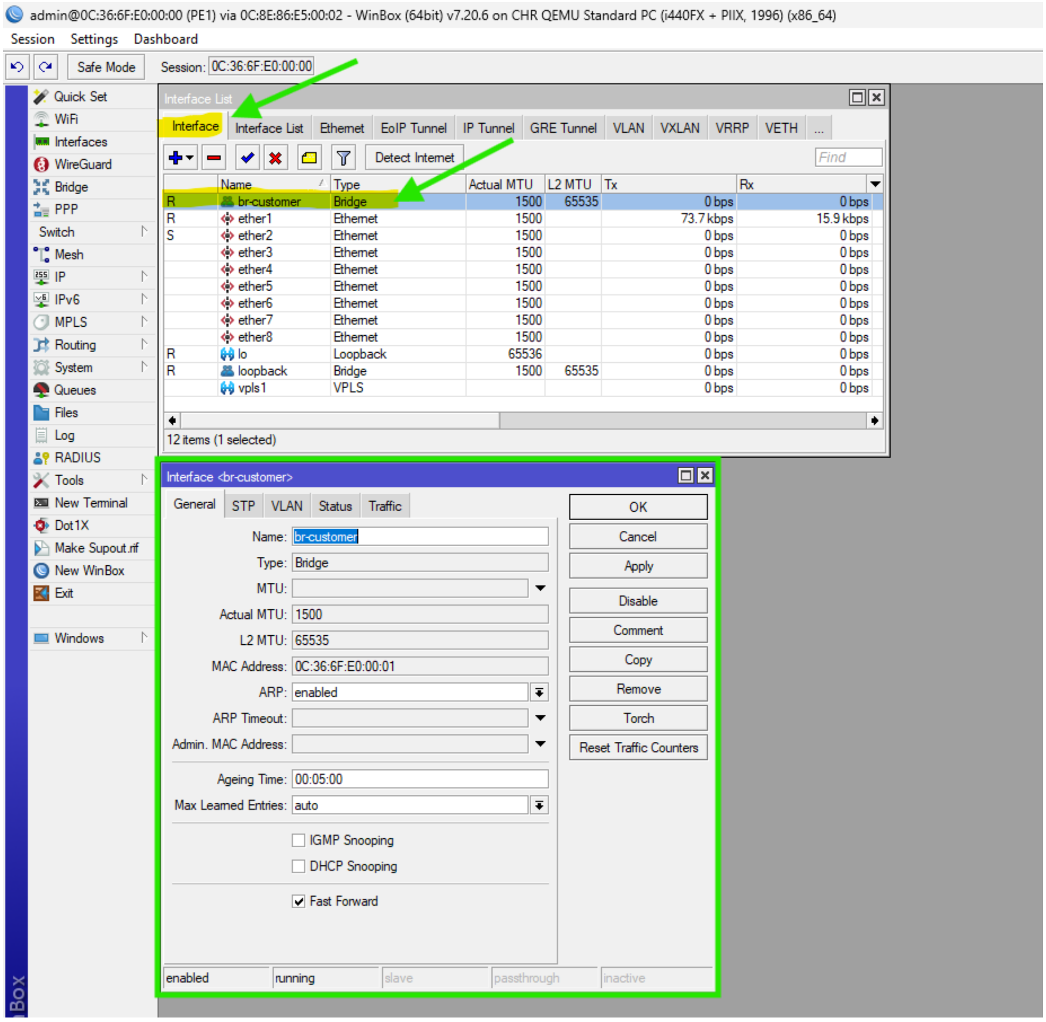


Figure 31: WINBOX for Setup Customer Bridge (Binds the VPLS tunnel to the Customer Port) (PE1 Router)

❖ MPLS MTU

/mpls interface set [find where interface=ether1] mpls-mtu=1508
Or
/mpls interface add interface=ether1 mpls-mtu=1508

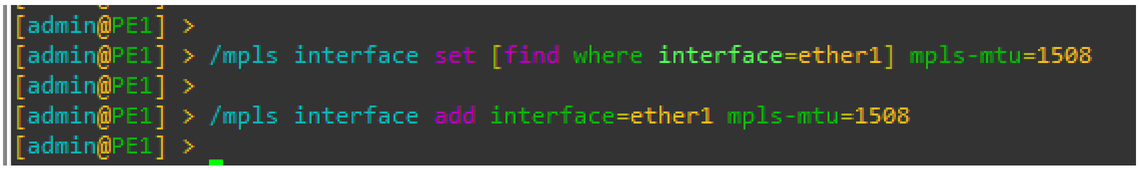


Figure 32: CMD to Setup MPLS MTU (PE1 Router)

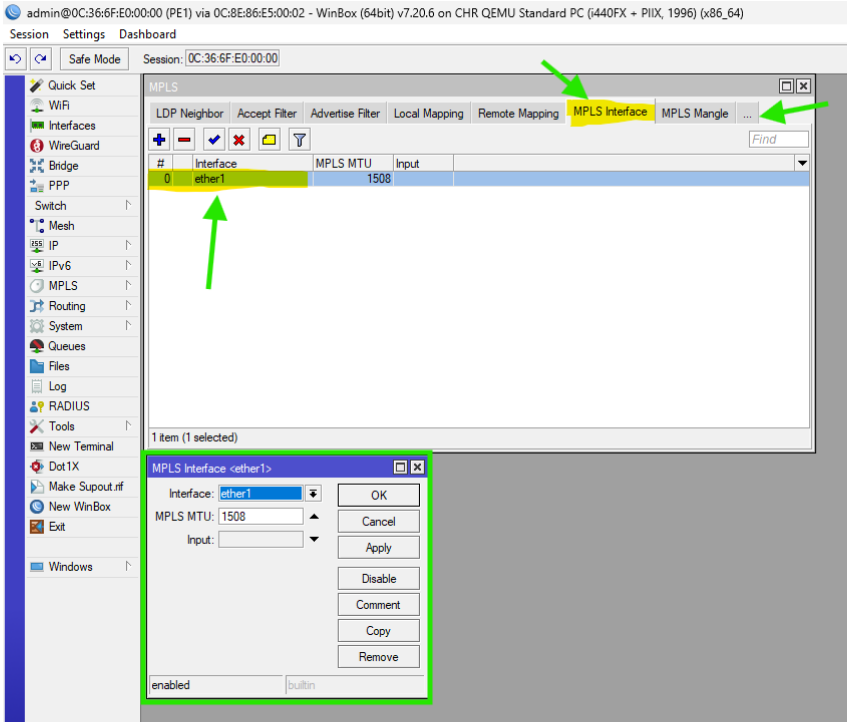


Figure 33: WINBOX for Setup MPLS MTU (PE1 Router)

4.3 PE2 Router Configuration

Like PE1, but adjusted IPs and remote-peer.

❖ Set Identity and Loopback:

```
/system identity set name=PE2
/interface bridge add name=loopback
/ip address add address=3.3.3.3/32 interface=loopback
```

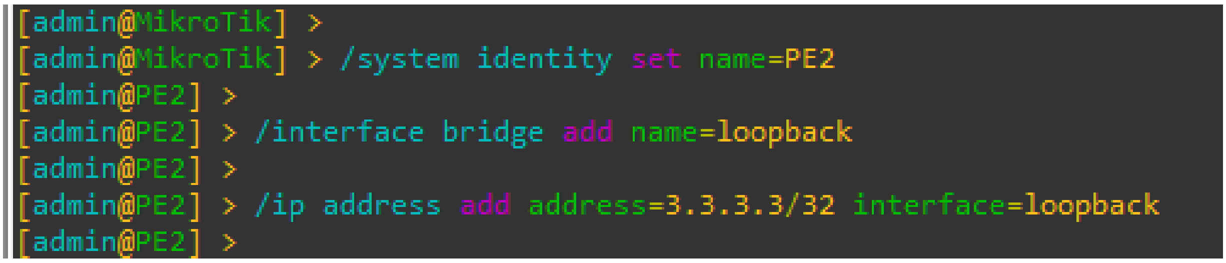


Figure 34: CMD to Set Identity and Loopback (PE2 Router)

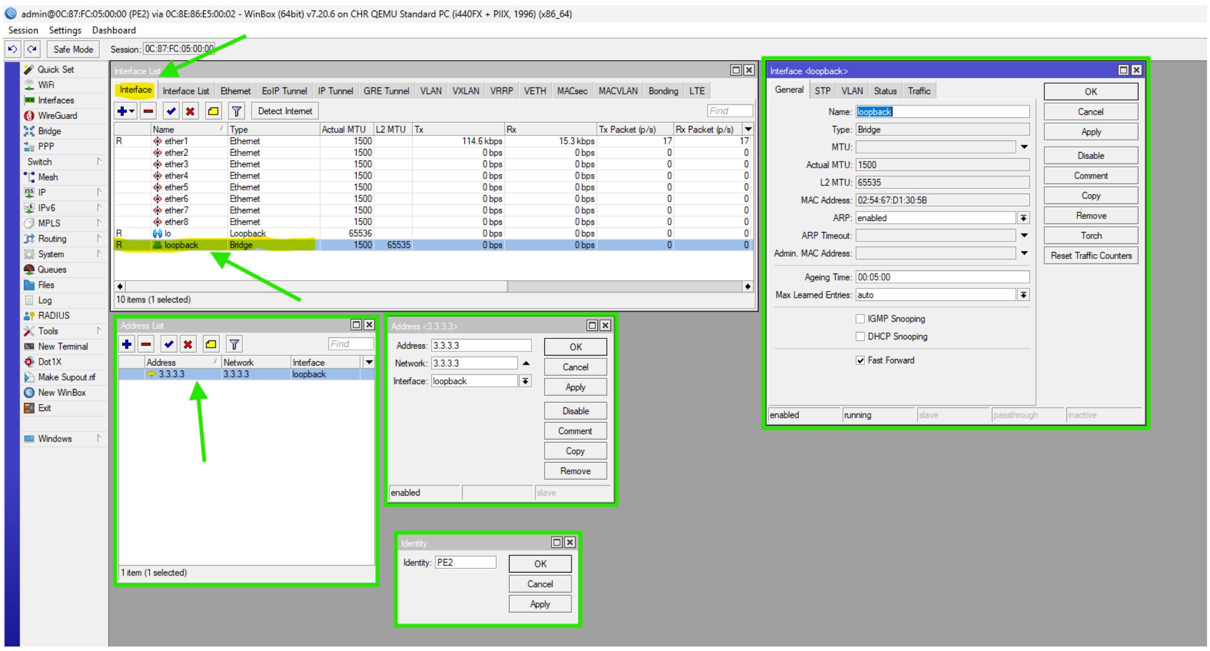


Figure 35: WINBOX Process to Set Identity and Loopback (PE2 Router)

❖ IP Addressing (Core side):

```
/ip address add address=10.0.2.1/30 interface=ether1 comment="Connect to Main"
```

Figure 36: CMD to set Interface Ips (PE1 Router)

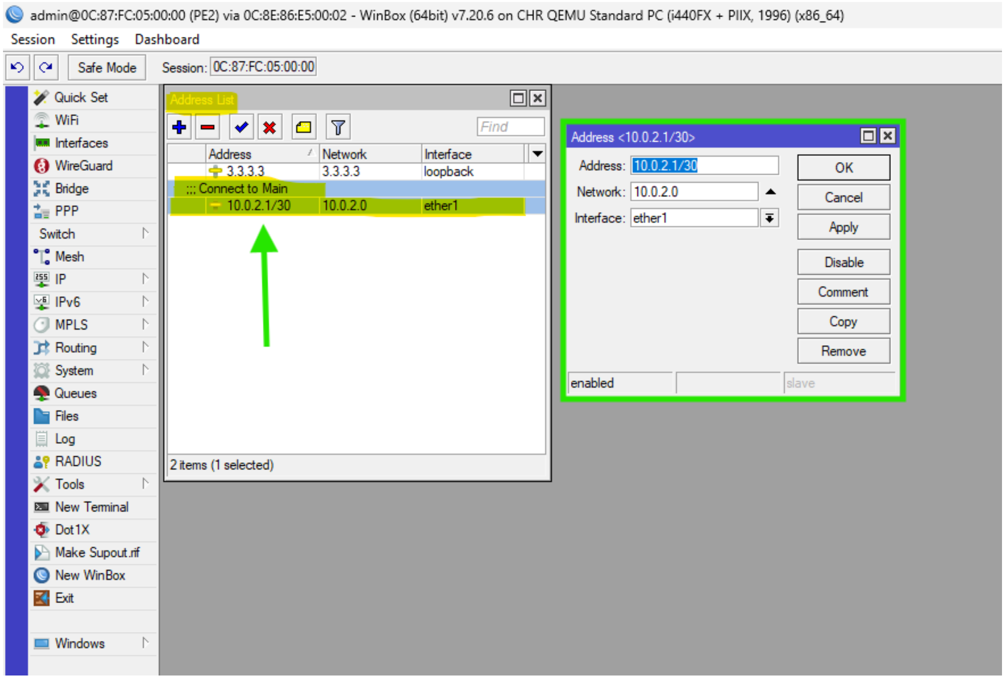


Figure 37: WINBOX Process to set Interface IPs (PE1 Router)

❖ OSPF Setup:

```
/routing ospf instance add name=default-v7 router-id=3.3.3.3
/routing ospf area add name=backbone area-id=0.0.0.0 instance=default-v7
/routing ospf interface-template
add interfaces=loopback area=backbone passive
add interfaces=ether1 area=backbone
```

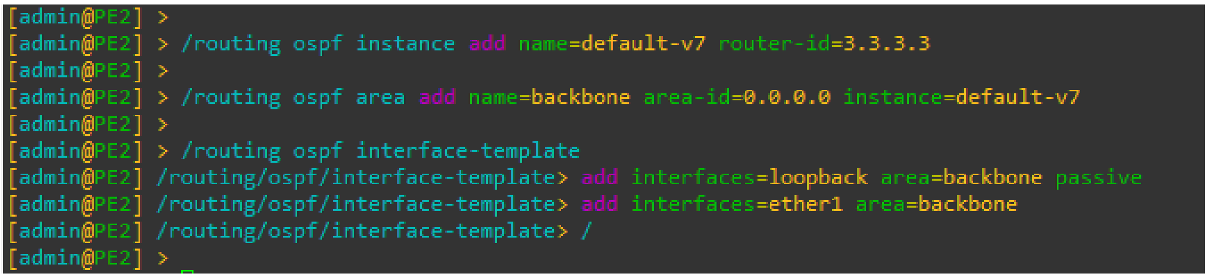


Figure 38: CMD for OSPF Setup (PE2 Router)

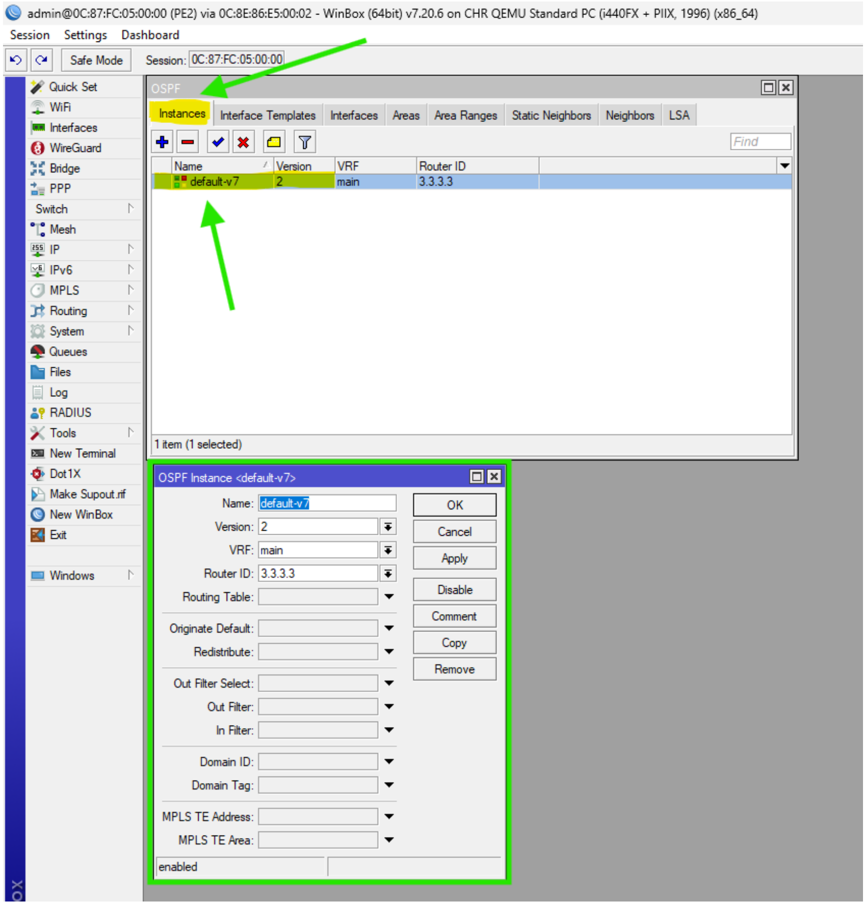


Figure 39: WINBOX Process for OSPF Setup i (PE2 Router)

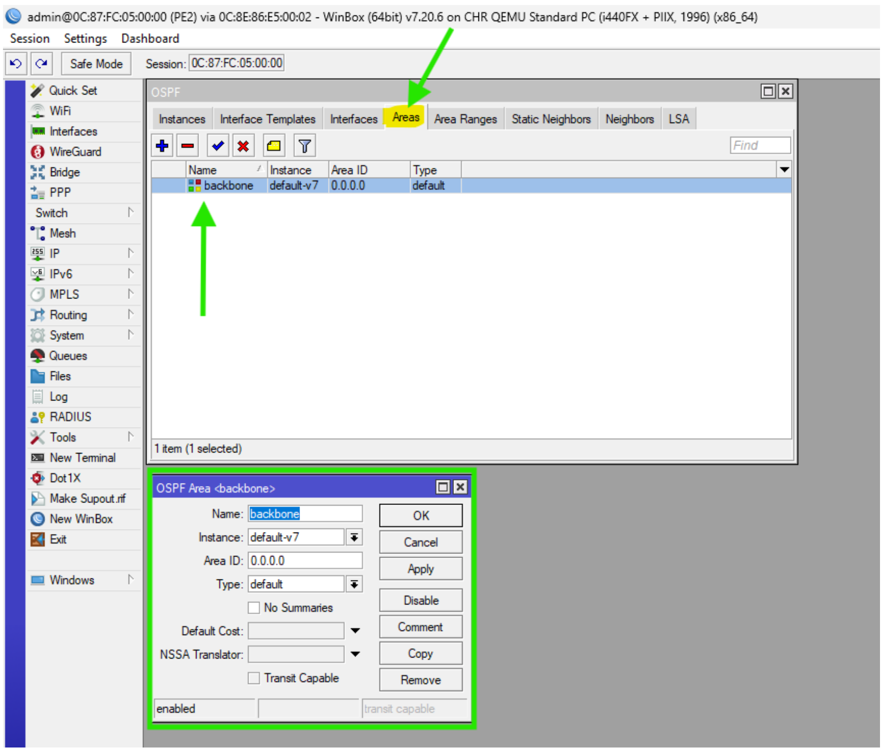


Figure 40: WINBOX Process for OSPF Setup ii (PE2 Router)

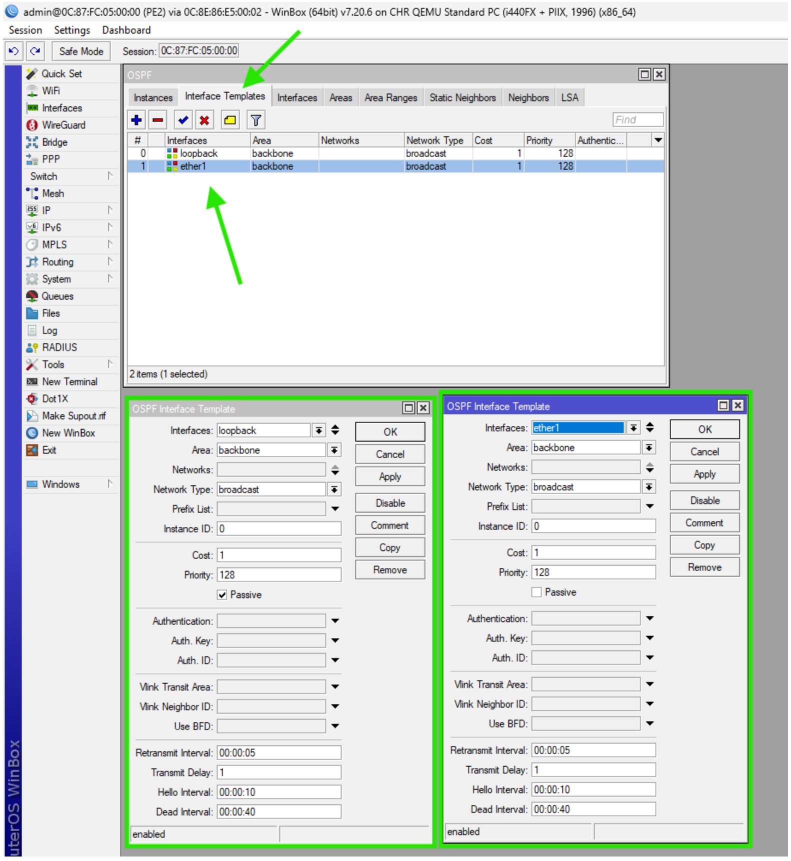


Figure 41: WINBOX Process for OSPF Setup iii (PE2 Router)

❖ MPLS and LDP:

```
/mpls ldp add lsr-id=3.3.3.3 transport-addresses=3.3.3.3
/mpls ldp set [find lsr-id=3.3.3.3] disabled=no
/mpls ldp interface add interface=ether1
```

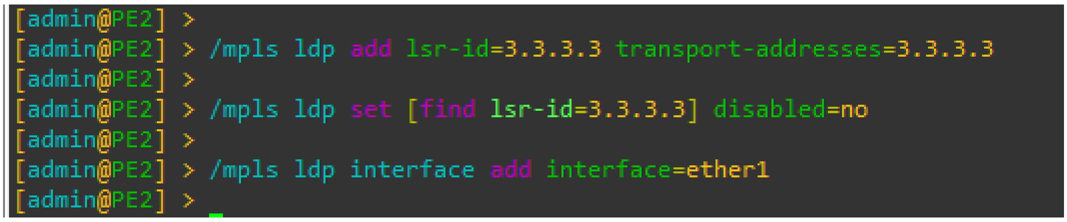


Figure 42: CMD to Setup MPLS & LDP (PE2 Router)

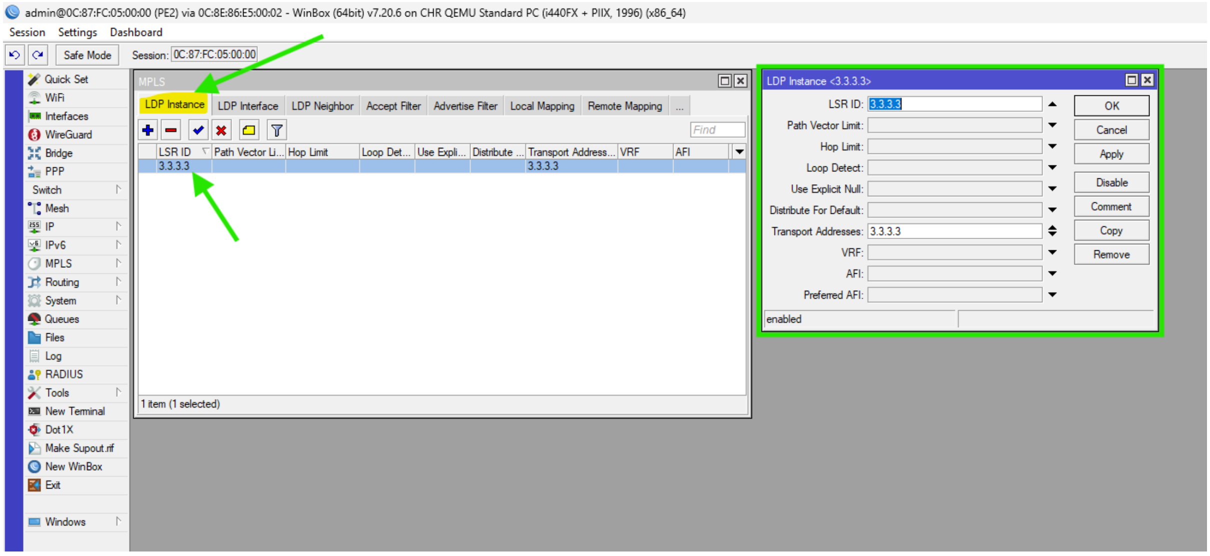


Figure 43: WINBOX Process for Setup MPLS & LDP i (PE2 Router)

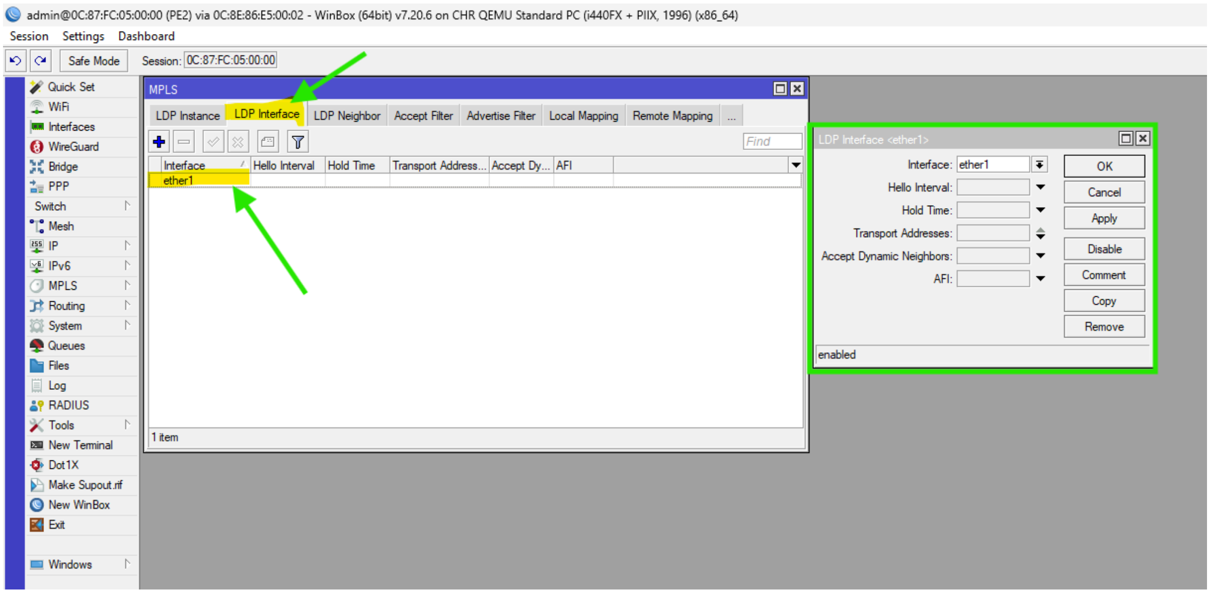


Figure 44: WINBOX Process for Setup MPLS & LDP ii (PE2 Router)

❖ VPLS Setup (Points to PE1: 2.2.2.2):

```
/interface vpls add name=vpls1 peer=2.2.2.2 vpls-id=10:1 disabled=no
```

```
[admin@PE2] >
[admin@PE2] > /interface vpls add name=vpls1 peer=2.2.2.2 vpls-id=10:1 disabled=no
[admin@PE2] >
```

Figure 45: CMD for VPLS Setup (Note: 'peer' instead of 'remote-peer') (PE2 Router)

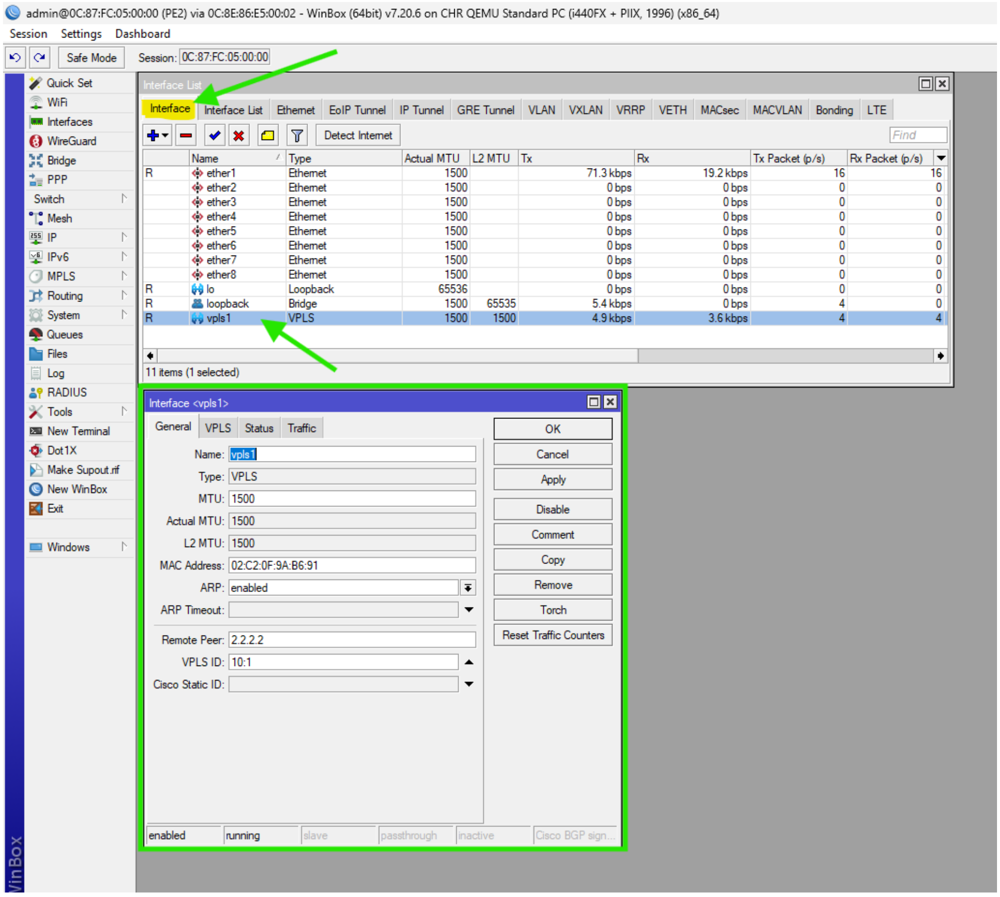


Figure 46: WINBOX Process for VPLS Setup (Note: 'peer' instead of 'remote-peer') (PE2 Router)

❖ Customer Bridge:

```
/interface bridge add name=br-customer
/interface bridge port add bridge=br-customer interface=ether2
/interface bridge port add bridge=br-customer interface=vpls1
```

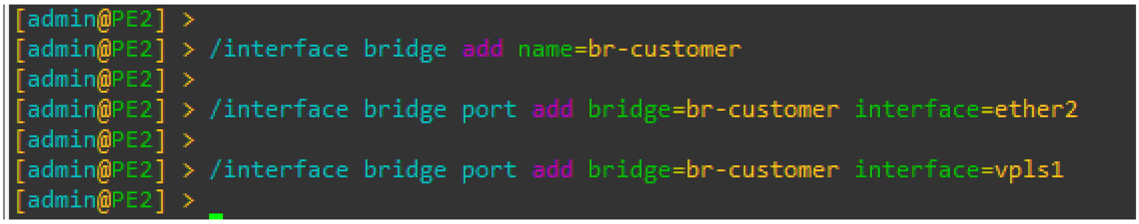


Figure 47: CMD to Setup Customer Bridge (Binds the VPLS tunnel to the Customer Port) (PE2 Router)

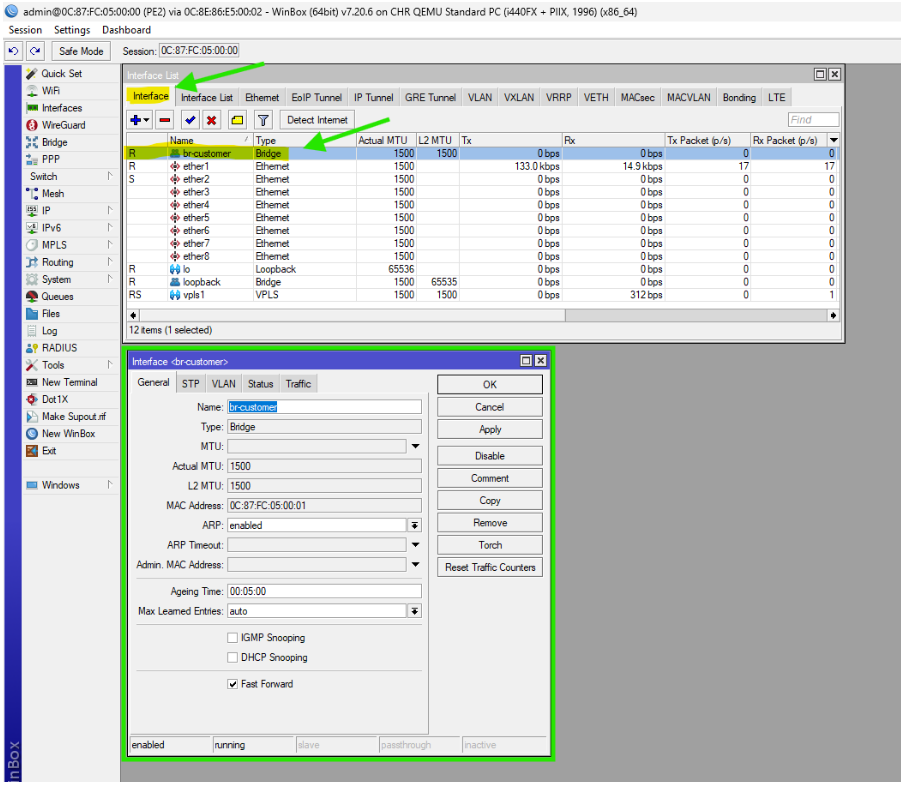


Figure 48: WINBOX for Setup Customer Bridge (Binds the VPLS tunnel to the Customer Port) (PE2 Router)

❖ MPLS MTU

```
/mpls interface set [find where interface=ether1] mpls-mtu=1508
                                Or
/mpls interface add interface=ether1 mpls-mtu=1508
```

Figure 49: CMD to Setup MPLS MTU (PE2 Router)

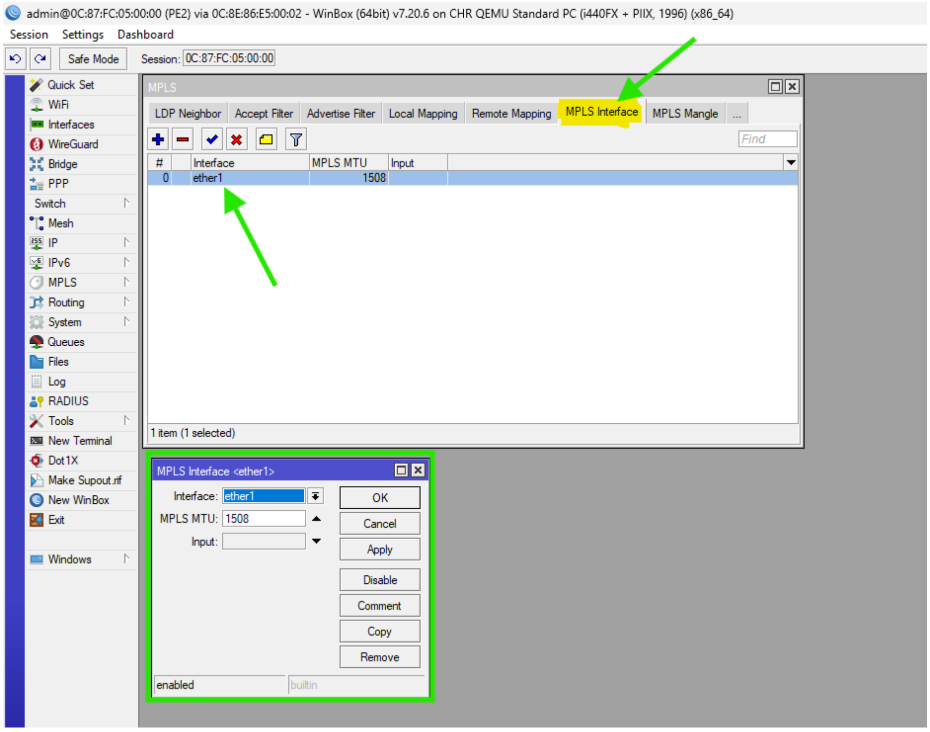


Figure 50: WINBOX for Setup MPLS MTU (PE2 Router)

4.4 CE1 Router Configuration

Assuming CE1 is also a MikroTik for simplicity (use similar for other routers).

❖ **Set Identity:**

```
/system identity set name=CE1
```

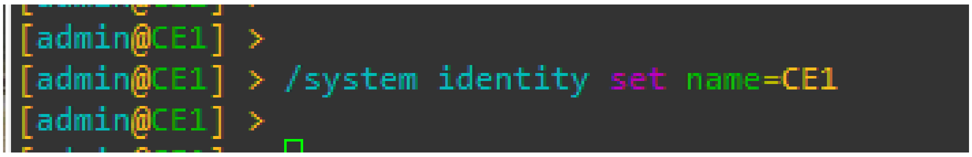


Figure 51: CMD to Steup Identity

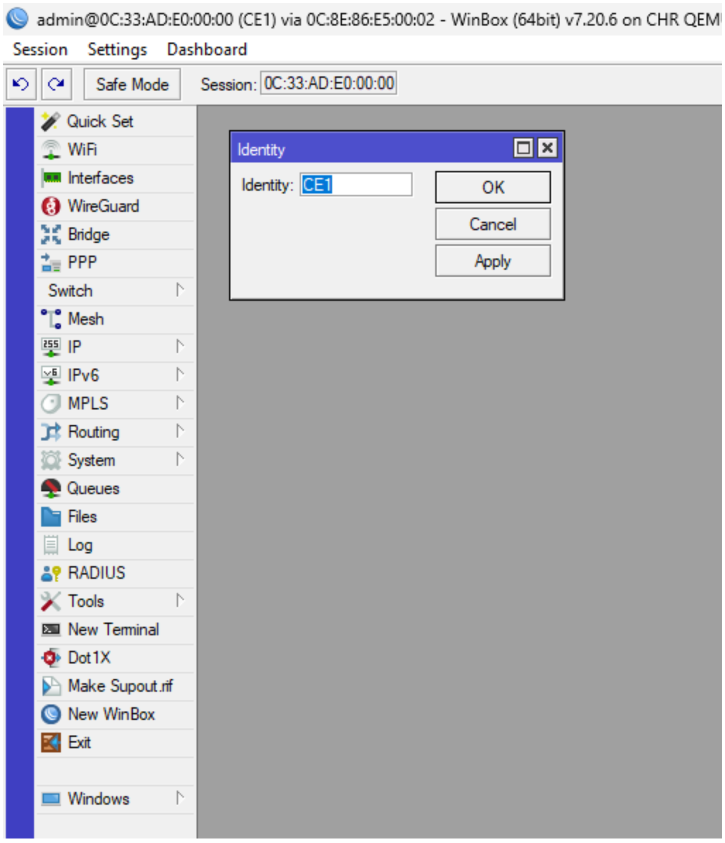


Figure 52: WINBOX for set Identity

❖ **Interface IP and Route:**

```
/ip address add address=192.168.10.1/24 interface=ether1
```

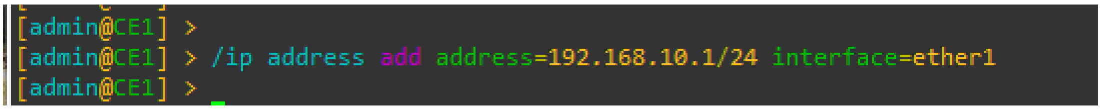


Figure 53: CMD to Setup Interface IP and Route

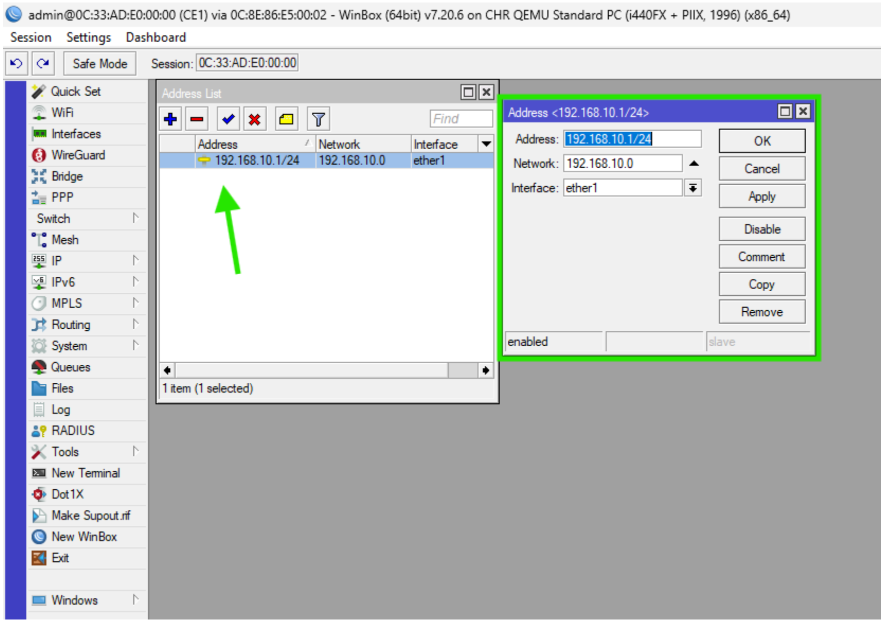


Figure 54: WINBOX to Steup Interface IP and Route

4.5 CE2 Router Configuration

Assuming CE2 is also a MikroTik for simplicity (use similar for other routers).

❖ Set Identity:

```
/system identity set name=CE2
```

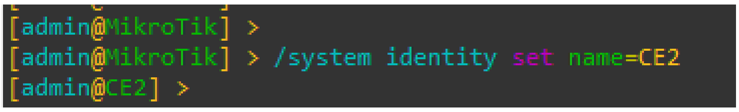


Figure 55: CMD to Steup Identity

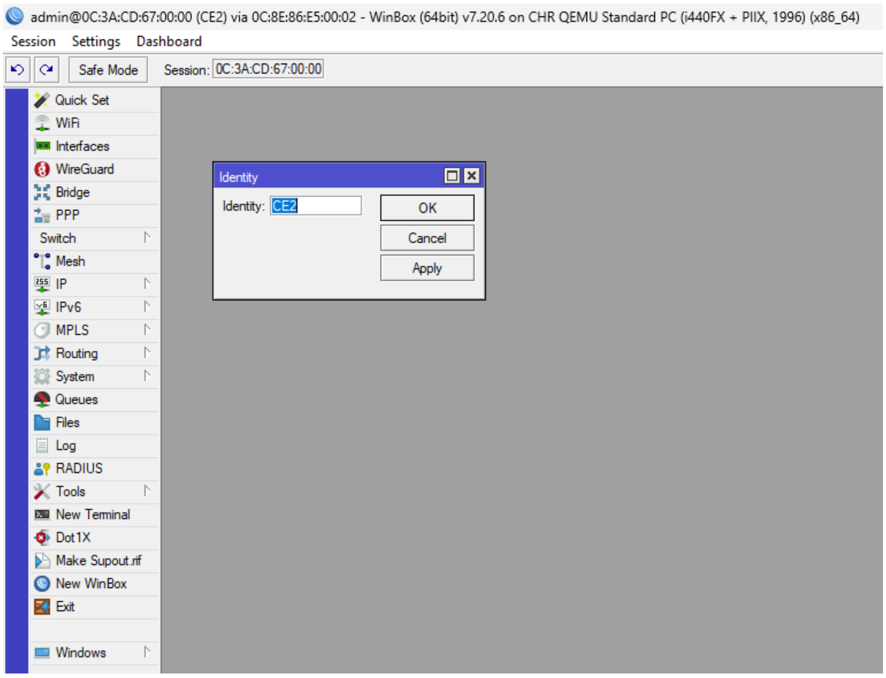


Figure 56: WINBOX for set Identity

❖ Interface IP (Shared Subnet):

```
/ip address add address=192.168.10.2/24 interface=ether1
```

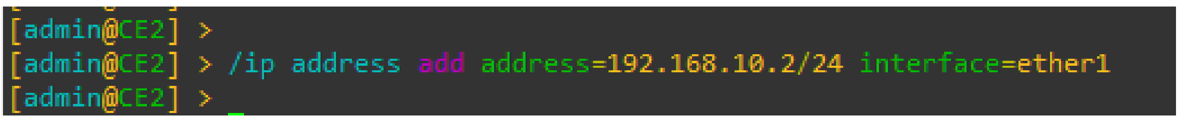


Figure 57: CMD to Setup Interface IP and Route

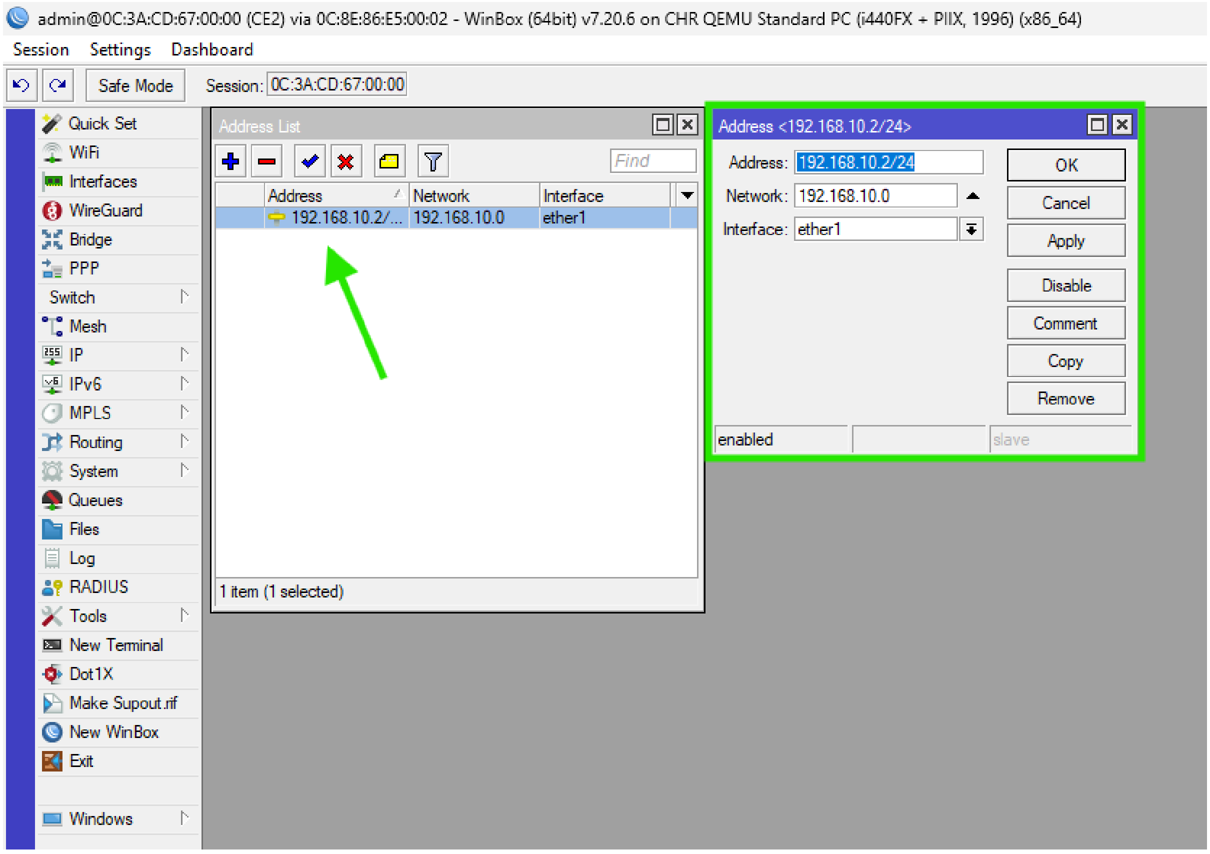


Figure 58: WINBOX to Steup Interface IP and Route

5. Verifying the MPLS and VPLS Setup

OSPF Adjacency: Run **/routing ospf neighbor print**. All core links must show state="Full".

```
[admin@Main] >
[admin@Main] >  routing/ospf/neighbor/print
Flags: V - virtual; D - dynamic
 0  D instance=default-v7 area=backbone address=10.0.2.1 priority=128 router-id=3.3.3.3 dr=10.0.2.1 bdr=10.0.2.2
      state="Full" state-changes=6 adjacency=1h19m38s timeout=38s

 1  D instance=default-v7 area=backbone address=10.0.1.1 priority=128 router-id=2.2.2.2 dr=10.0.1.1 bdr=10.0.1.2
      state="Full" state-changes=6 adjacency=1h19m38s timeout=38s
[admin@Main] >
```

Figure 59: Output of ospf neighbor Main Router

```
[admin@PE1] >
[admin@PE1] >  routing/ospf/neighbor/print
Flags: V - virtual; D - dynamic
 0  D instance=default-v7 area=backbone address=10.0.1.2 priority=128 router-id=1.1.1.1 dr=10.0.1.1 bdr=10.0.1.2
      state="Full" state-changes=6 adjacency=1h19m34s timeout=32s
[admin@PE1] >
```

Figure 60: Output of ospf neighbor PE1 Router

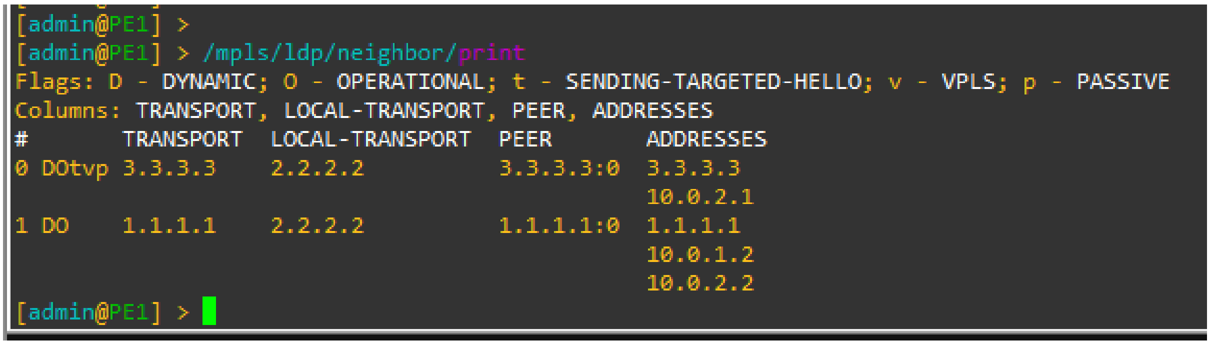
```
[admin@PE2] > routing/ospf/neighbor/print
Flags: V - virtual; D - dynamic
 0  D instance=default-v7 area=backbone address=10.0.2.2 priority=128 router-id=1.1.1.1 dr=10.0.2.1 bdr=10.0.2.2
      state="Full" state-changes=6 adjacency=1h18m46s timeout=30s
[admin@PE2] >
```

Figure 61: Output of ospf neighbor PE2 Router

LDP Session: Run **/mpls ldp neighbor print**. Should show DO (Dynamic, Operational) flags.

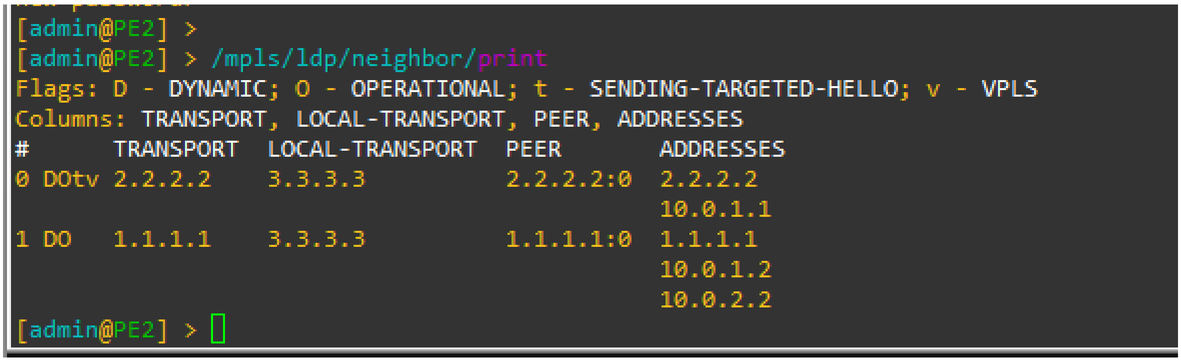
```
[admin@Main] > /mpls/ldp/neighbor/print
Flags: D - DYNAMIC; O - OPERATIONAL; p - PASSIVE
Columns: TRANSPORT, LOCAL-TRANSPORT, PEER, ADDRESSES
#     TRANSPORT  LOCAL-TRANSPORT  PEER       ADDRESSES
0 DOp 2.2.2.2    1.1.1.1          2.2.2.2:0  2.2.2.2
                                             10.0.1.1
1 DOp 3.3.3.3    1.1.1.1          3.3.3.3:0  3.3.3.3
                                             10.0.2.1
[admin@Main] >
```

Figure 62: Output of ldp neighbor Main Router



```
[admin@PE1] >
[admin@PE1] > /mpls/ldp/neighbor/print
Flags: D - DYNAMIC; O - OPERATIONAL; t - SENDING-TARGETED-HELLO; v - VPLS; p - PASSIVE
Columns: TRANSPORT, LOCAL-TRANSPORT, PEER, ADDRESSES
#       TRANSPORT  LOCAL-TRANSPORT  PEER       ADDRESSES
0 DOtvp 3.3.3.3    2.2.2.2          3.3.3.3:0  3.3.3.3
                                               10.0.2.1
1 DO    1.1.1.1    2.2.2.2          1.1.1.1:0  1.1.1.1
                                               10.0.1.2
                                               10.0.2.2
[admin@PE1] >
```

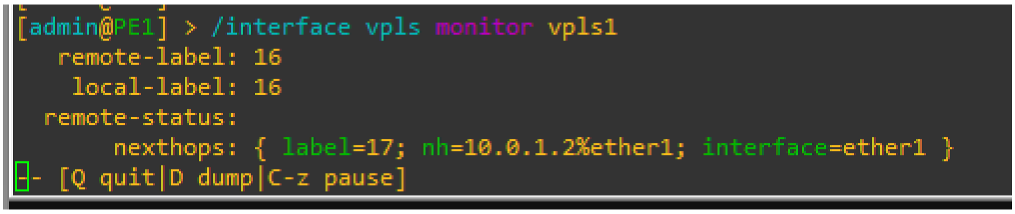
Figure 63: Output of ldp neighbor PE1 Router



```
[admin@PE2] >
[admin@PE2] > /mpls/ldp/neighbor/print
Flags: D - DYNAMIC; O - OPERATIONAL; t - SENDING-TARGETED-HELLO; v - VPLS
Columns: TRANSPORT, LOCAL-TRANSPORT, PEER, ADDRESSES
#      TRANSPORT  LOCAL-TRANSPORT  PEER       ADDRESSES
0 DOtv 2.2.2.2    3.3.3.3          2.2.2.2:0  2.2.2.2
                                              10.0.1.1
1 DO   1.1.1.1    3.3.3.3          1.1.1.1:0  1.1.1.1
                                              10.0.1.2
                                              10.0.2.2
[admin@PE2] >
```

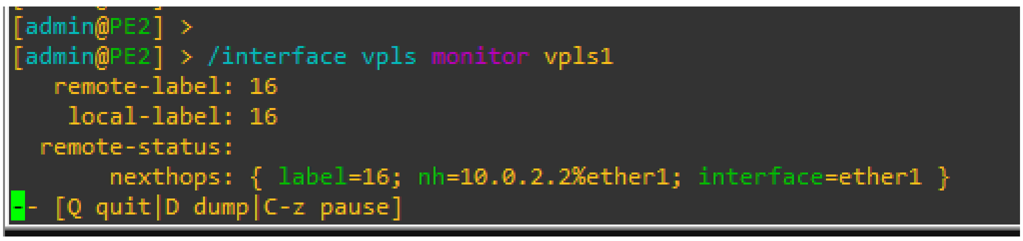
Figure 64: Output of ldp neighbor PE2 Router

VPLS Tunnel: Run **/interface vpls monitor vpls1**. Should show remote-label and status: running.



```
[admin@PE1] > /interface vpls monitor vpls1
   remote-label: 16
    local-label: 16
  remote-status:
       nexthops: { label=17; nh=10.0.1.2%ether1; interface=ether1 }
-- [Q quit|D dump|C-z pause]
```

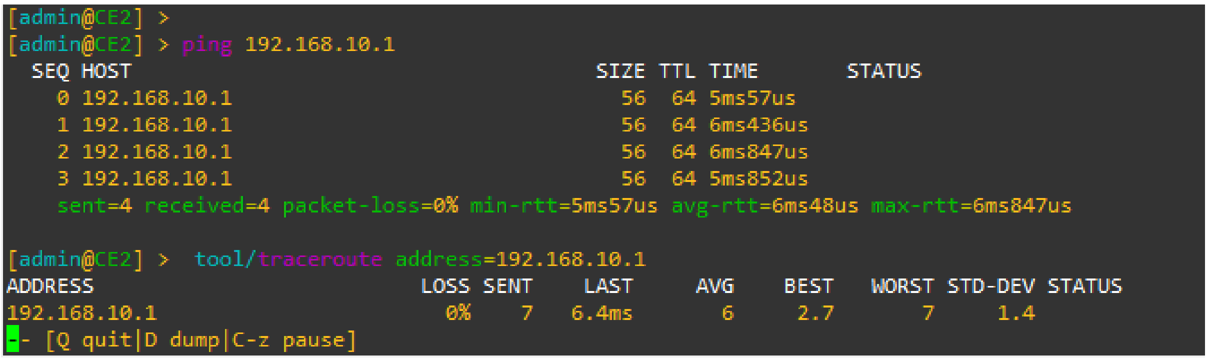
Figure 65: Output of vpls monitor PE1 Router



```
[admin@PE2] >
[admin@PE2] > /interface vpls monitor vpls1
   remote-label: 16
    local-label: 16
  remote-status:
       nexthops: { label=16; nh=10.0.2.2%ether1; interface=ether1 }
-- [Q quit|D dump|C-z pause]
```

Figure 66: Output of vpls monitor PE2 Router

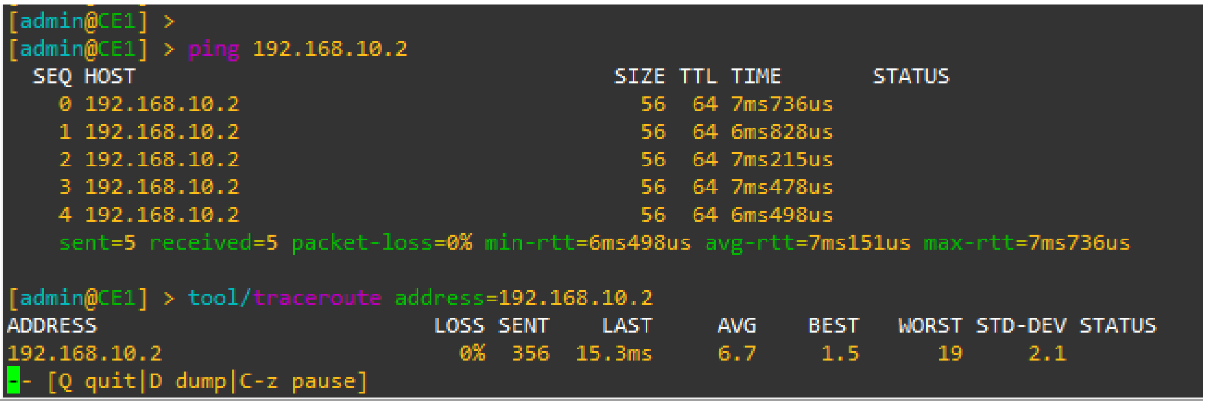
End-to-End Test: From CE1, run **ping 192.168.10.2 OR** From CE2, run **ping 192.168.10.1**. Success confirms the Layer 2 bridge is transparently passing traffic across the MPLS core.



```
[admin@CE2] >
[admin@CE2] > ping 192.168.10.1
  SEQ HOST                                     SIZE TTL TIME       STATUS
    0 192.168.10.1                               56  64 5ms57us
    1 192.168.10.1                               56  64 6ms436us
    2 192.168.10.1                               56  64 6ms847us
    3 192.168.10.1                               56  64 5ms852us
    sent=4 received=4 packet-loss=0% min-rtt=5ms57us avg-rtt=6ms48us max-rtt=6ms847us

[admin@CE2] >  tool/traceroute address=192.168.10.1
ADDRESS                          LOSS SENT    LAST     AVG    BEST   WORST STD-DEV STATUS
192.168.10.1                       0%    7   6.4ms       6     2.7       7     1.4
-- [Q quit|D dump|C-z pause]
```

Figure 67: CE1 PING and TRACEROUTE Report



```
[admin@CE1] >
[admin@CE1] > ping 192.168.10.2
  SEQ HOST                                     SIZE TTL TIME       STATUS
    0 192.168.10.2                               56  64 7ms736us
    1 192.168.10.2                               56  64 6ms828us
    2 192.168.10.2                               56  64 7ms215us
    3 192.168.10.2                               56  64 7ms478us
    4 192.168.10.2                               56  64 6ms498us
    sent=5 received=5 packet-loss=0% min-rtt=6ms498us avg-rtt=7ms151us max-rtt=7ms736us

[admin@CE1] > tool/traceroute address=192.168.10.2
ADDRESS                          LOSS SENT    LAST     AVG    BEST   WORST STD-DEV STATUS
192.168.10.2                       0%  356  15.3ms     6.7     1.5      19     2.1
-- [Q quit|D dump|C-z pause]
```

Figure 68: CE2 PING and TRACEROUTE Report

6. Troubleshooting Common Issues

- **OSPF Not Converging:** Check **interface IPs**, MTU (set MPLS MTU to 1508 if fragmentation issues).
- **LDP Not Establishing:** Ensure **loopbacks** are **advertised in OSPF**, transport-address set correctly.
- **VPLS Down:** Verify remote-peer is reachable, same vpls-id, LDP up.
- **No Connectivity:** Check bridge ports, ARP tables, and use /tool sniffer for packet captures.
- **GNS3 Performance:** Reduce RAM if crashes; your PC specs handle it well.
- **MTU Issues:** Set **/mpls set dynamic-interface-mtu=1500** or **adjust manually**.
- Common Commands:

    /log print for errors.

    /mpls ldp bindings print for label info.

7. Wireshark Traffic

7.1 Identify basic protocols first

Before jumping into MPLS/VPLS, confirm that the traffic is capture correct.

ip

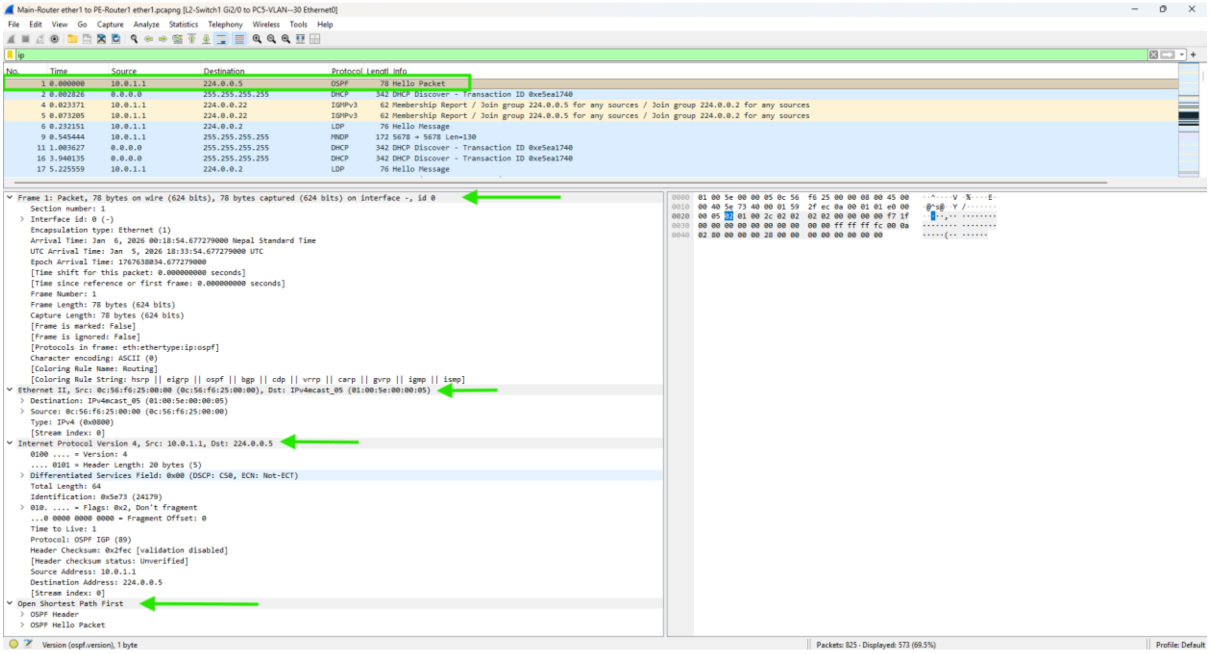


Figure 69

We will see normal ip traffic as show in upper figure

If you see nothing capture might be wrong interface or encapsulated only

7.2 Check OSPF (control plane)

OSPF is very important - it builds routes for MPLS.

ospf

We will see:

Hello packets

DBD (Database Description)

LSR / LSU / LSAck

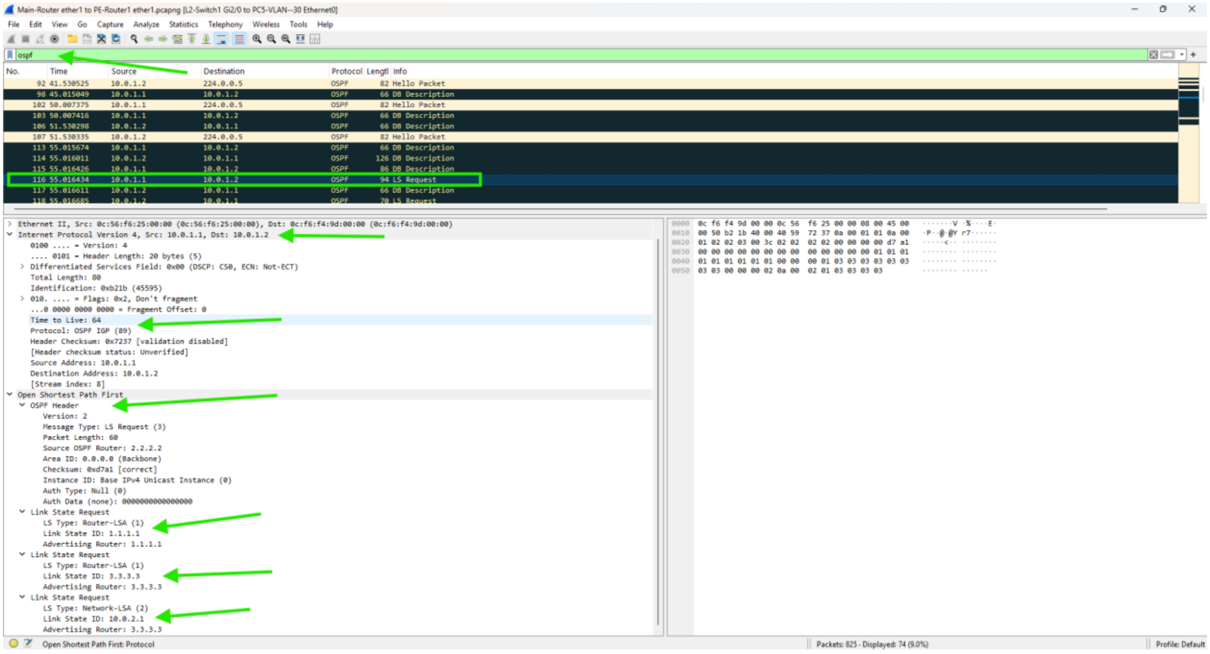


Figure 70

Check:

- Router ID
- Area ID
- Hello Interval
- Dead Interval
- Link State Request

7.3 Verify OSPF Neighbor formation

Look specifically for Hello packets.

ospf && ospf.msg == 1

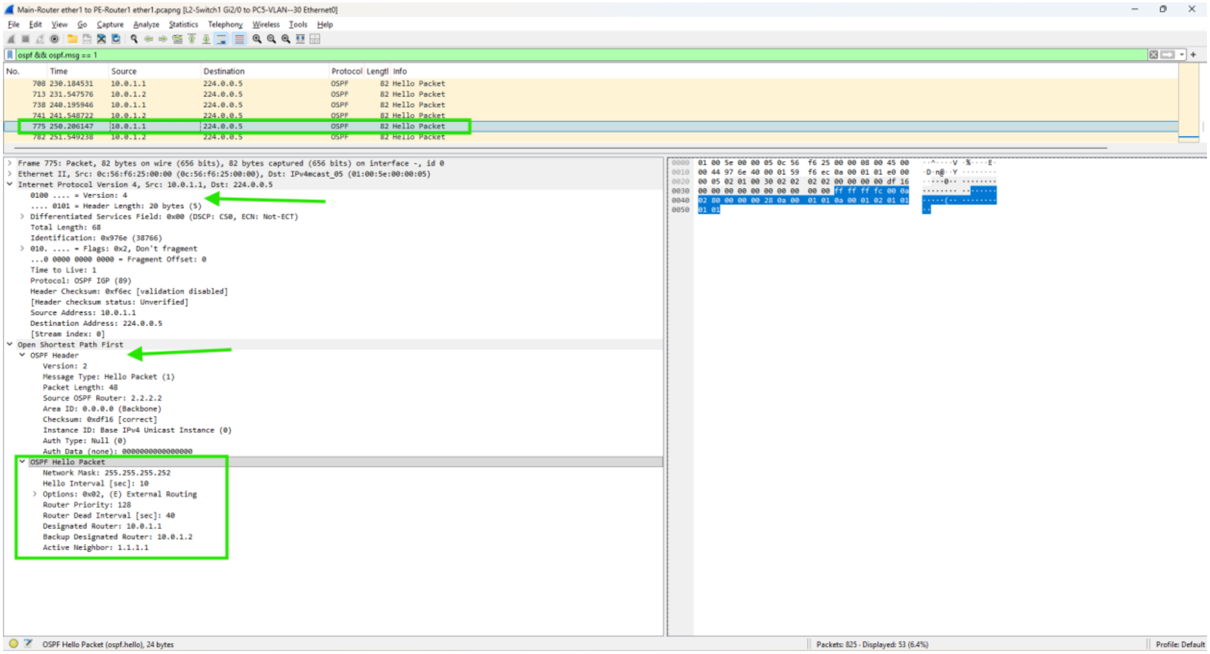


Figure 71

Check:

Source IP

Destination 224.0.0.5

Neighbor Router IDs

If neighbors don't reach FULL, MPLS/VPLS won't establish.

7.4 Check MPLS packets

mpls

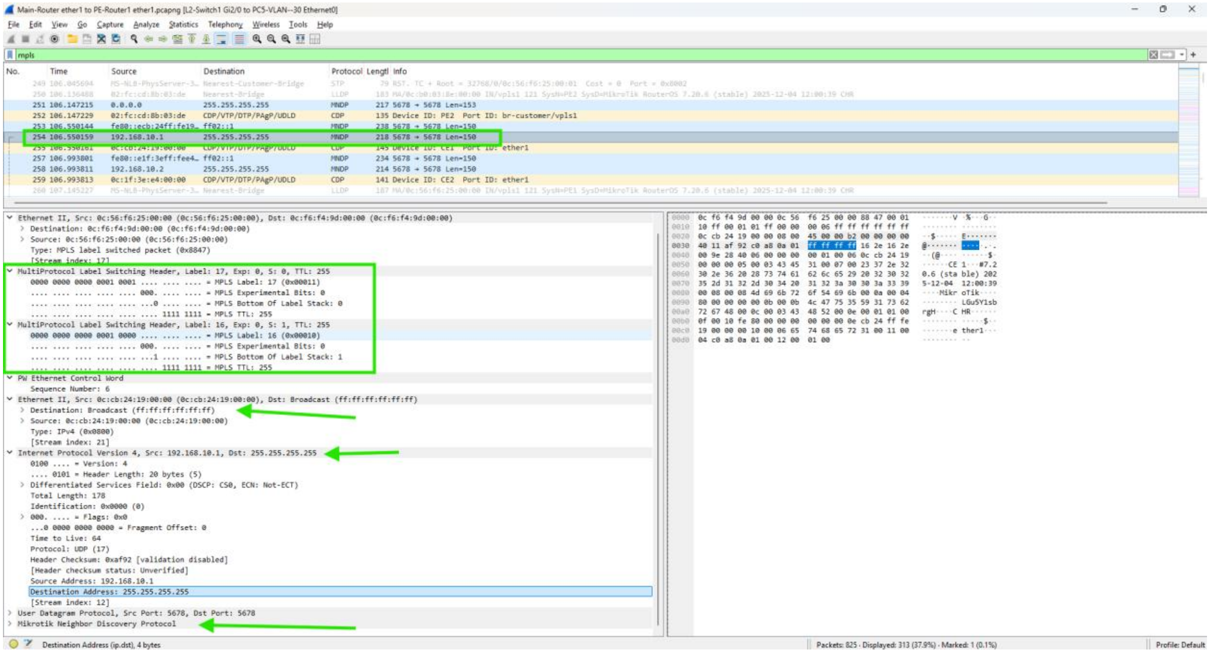


Figure 72

7.5 Check LDP (Label Distribution Protocol)

MPLS labels come from LDP (or RSVP).

ldp

You should see:

LDP Hello

LDP Initialization

Label Mapping messages

Destination ports:

TCP 646

UDP 646

No LDP = MPLS labels won’t exist

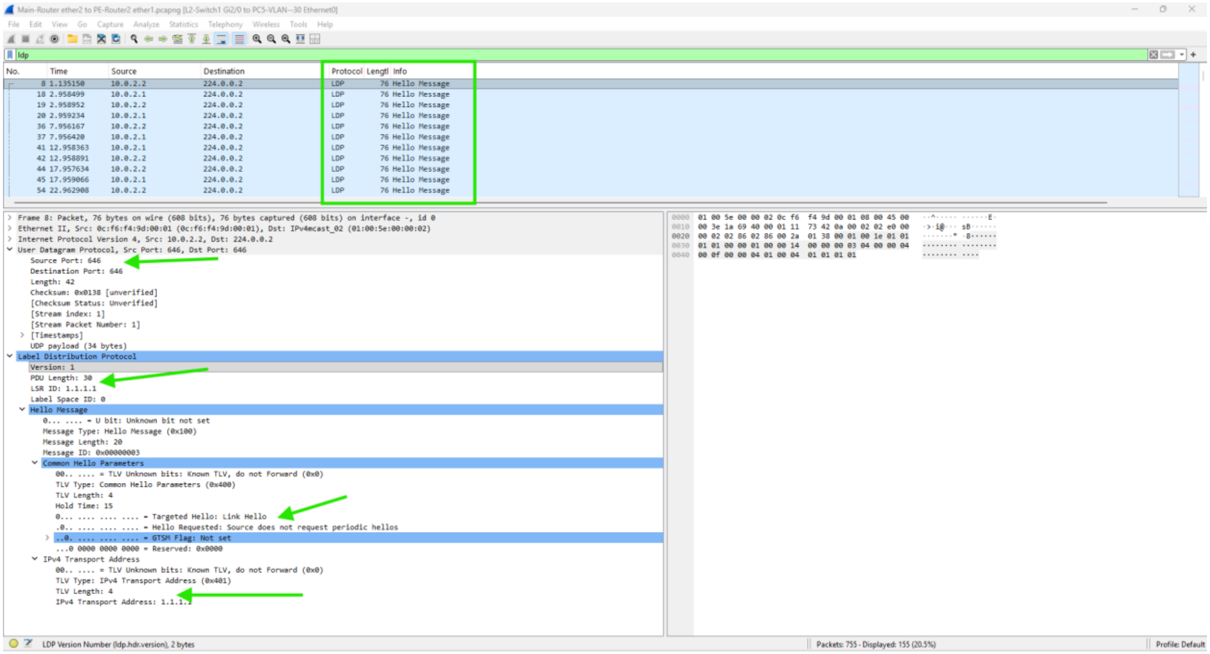


Figure 73

7.6 Verify VPLS traffic

VPLS usually runs over MPLS with Ethernet frames.

eth && mpls OR mpls && eth.type == 0x8847

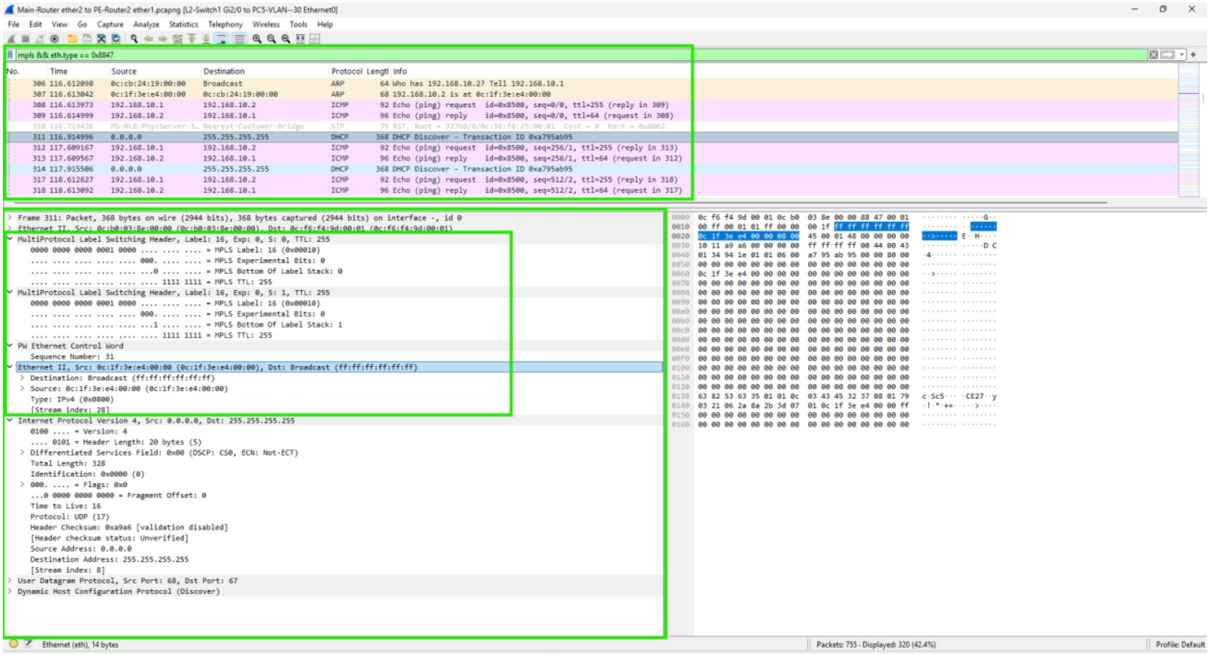
Click packet on expand:

Ethernet

MPLS

Inner Ethernet Frame

If you see **Ethernet inside MPLS**, that's VPLS



7.7 Check MAC learning (important for VPLS)

VPLS behaves like a switch.

arp or eth.dst == ff:ff:ff:ff:ff:ff

You should see:

ARP requests flooding

MAC addresses being learned

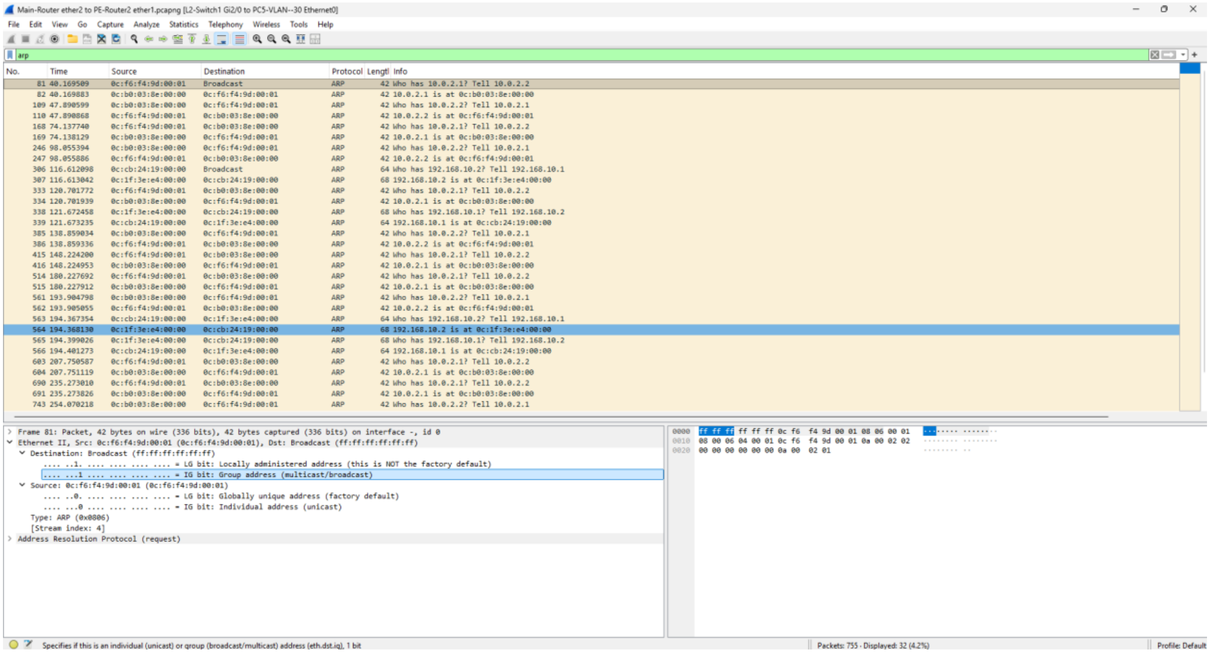


Figure 74

8. Conclusion

The successful implementation of this workshop demonstrates that MPLS and VPLS provide a robust, scalable solution for extending Layer 2 networks across a routed IP core. By decoupling the customer's Ethernet traffic from the provider's routing infrastructure, the network achieves a high degree of transparency and efficiency.

Key takeaways from this project include:

Infrastructure Synergy: The integration of OSPF for path determination and LDP for label switching creates a high-performance "label-switched" environment that minimizes CPU overhead on core routers.

Layer 2 Transparency: VPLS successfully creates a "virtual wire," allowing customer sites to communicate as if they were physically connected to the same local switch, supporting protocols that standard Layer 3 VPNs cannot easily transport.

The MTU Factor: The workshop highlights the critical importance of MTU overhead; by increasing the core MTU to 1508, the network avoids the performance degradation caused by packet fragmentation during encapsulation.

Ultimately, this MikroTik-based architecture provides a cost-effective and enterprise-ready blueprint for Service Providers looking to deliver seamless site-to-site connectivity while maintaining a simplified, IP-unaware core.