

Lab

MPLS with Traffic Engineering

Last Update 2011.06.01
1.0.0

Source

- This lab was developed by ibarrere
- That is the only name listed on the site
- It is found in a blog post from May 2009 on the <http://www.networking-forum.com> web site at
 - <http://www.networking-forum.com/blog/?p=145>
- This looks like a very useful site

MPLS in a Lab Environment

- As it is difficult to create a real MPLS network in a lab environment and since simulation programs do not offer a MPLS cloud as they do for Frame Relay, this lab will use an emulation program
- In this case GNS3 running Dynagen which is controlling the Dynamips emulator
- How to use GNS3 is covered in detail in another presentation on this web site

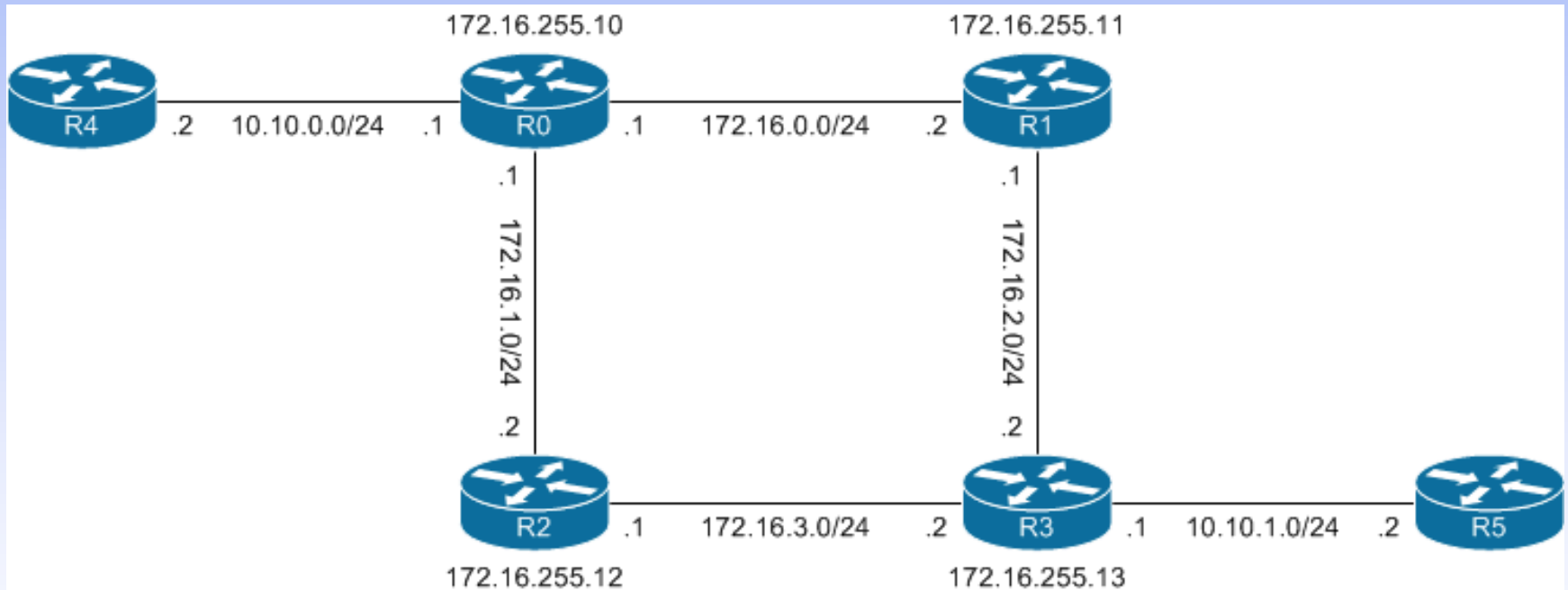
Lab Network

- Here is the lab network as setup in GNS3
- To create the topology either
 - Download and open this file in GNS3
 - MPLS With Traffic Engineering Lab.net
 - or
 - Lay the routers out as the diagram that follows shows

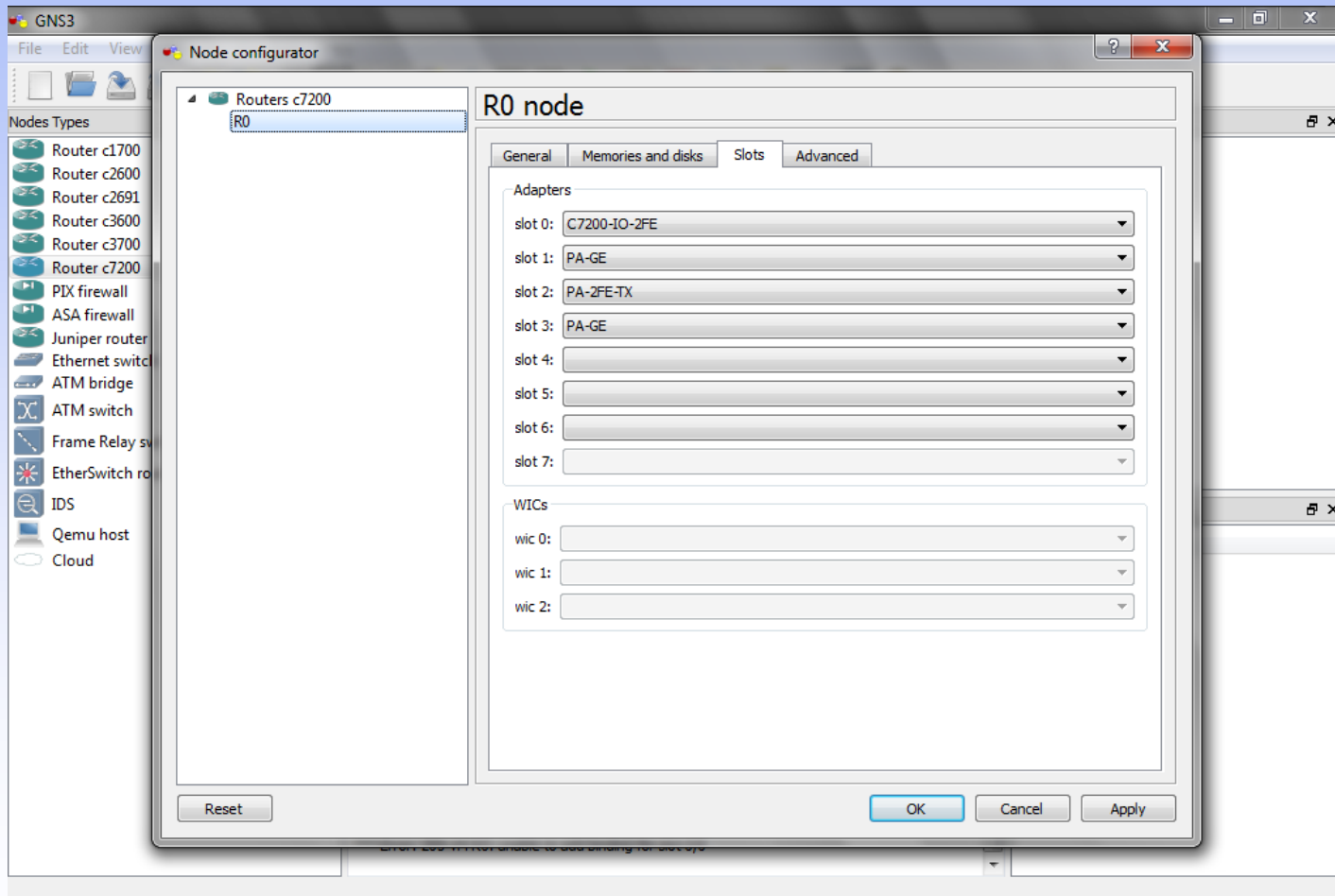
Lab Network

- When the routers are placed be sure the labels match as well
- Next the modules to be added to the routers, which are all 7200 models
- Then the connections between the routers are made
- To simplify the setup the modules are all the same even though some of them are not actually needed in all the routers

Lab Network



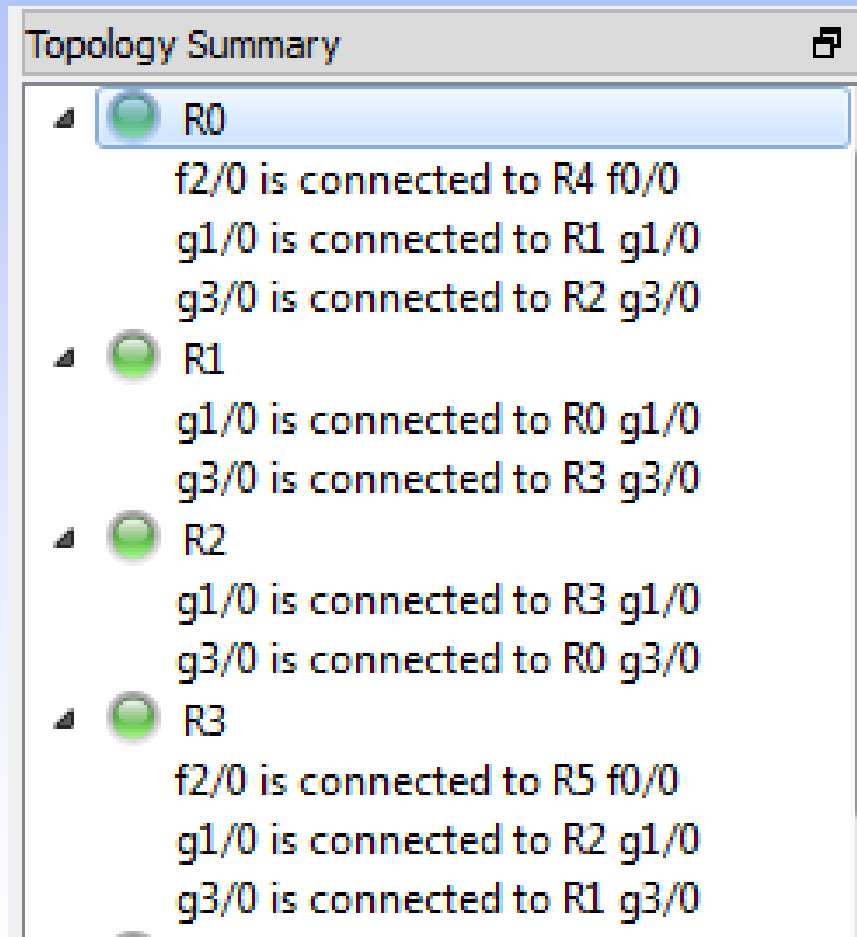
Modules Required





Modules Required

- The modules are
 - Slot 0
 - C7200-IO-FE
 - Slot 1
 - PA-GE
 - Slot 2
 - PA-2FE-TX
 - Slot 3
 - PA-GE

Connections



Connections

- ▲  R4
f0/0 is connected to R0 f2/0
- ▲  R5
f0/0 is connected to R3 f2/0

Lab Network

- The routers are all running this IOS
 - c7200-advipservicesk9-mz.150-1.M5.bin

Configuration

- As the configurations are lengthy for this many routers they are provided as separate links to copy and paste txt files listed just below this lab's link
- Download each one of these, then copy and paste the contents to each router based on the name of the file and the corresponding router

Verify Connectivity

- Let's see if it worked
- Enter
 - show mpls traffic-eng tunnels
- This shows

Verify Connectivity

```
Dynamips(14): R0, Console port - 127.0.0.1 VT
File Edit Setup Control Window Help
R0#show mpls traffic-eng tunnels
Name: R0_t2                                <Tunnel2> Destination: 172.16.255.13
Status:
  Admin: up      Oper: up      Path: valid      Signalling: connected
  path option 1, type explicit BOTTOM <Basis for Setup, path weight 2>

Config Parameters:
  Bandwidth: 158      kbps <Global> Priority: 2 2 Affinity: 0x0/0xFFFF
  Metric Type: TE <default>
  AutoRoute: enabled LockDown: disabled Loadshare: 158      bw-based
  auto-bw: disabled

Active Path Option Parameters:
  State: explicit path option 1 is active
  BandwidthOverride: disabled LockDown: disabled Verbatim: disabled

InLabel : -
OutLabel : GigabitEthernet3/0, 22
RSUP Signalling Info:
  Src 172.16.255.10, Dst 172.16.255.13, Tun_Id 2, Tun_Instance 42
  RSUP Path Info:
    My Address: 172.16.1.1
    Explicit Route: 172.16.1.2 172.16.3.1 172.16.3.2 172.16.255.13
    Record Route: NONE
    Tspec: ave rate=158 kbits, burst=1000 bytes, peak rate=158 kbits
  RSUP Resv Info:
    Record Route: NONE
    Fspec: ave rate=158 kbits, burst=1000 bytes, peak rate=158 kbits
History:
  Tunnel:
    Time since created: 52 minutes, 41 seconds
    Time since path change: 40 minutes, 33 seconds
    Number of LSP IDs <Tun_Instances> used: 42
  Current LSP:
    Uptime: 40 minutes, 33 seconds

LSP Tunnel R3_t2 is signalled, connection is up
InLabel : GigabitEthernet3/0, implicit-null
OutLabel : -
RSUP Signalling Info:
  Src 172.16.255.13, Dst 172.16.255.10, Tun_Id 2, Tun_Instance 7
  RSUP Path Info:
    My Address: 172.16.255.10
    Explicit Route: NONE
    Record Route: NONE
    Tspec: ave rate=158 kbits, burst=1000 bytes, peak rate=158 kbits
  RSUP Resv Info:
    Record Route: NONE
    Fspec: ave rate=158 kbits, burst=1000 bytes, peak rate=158 kbits
R0#
```

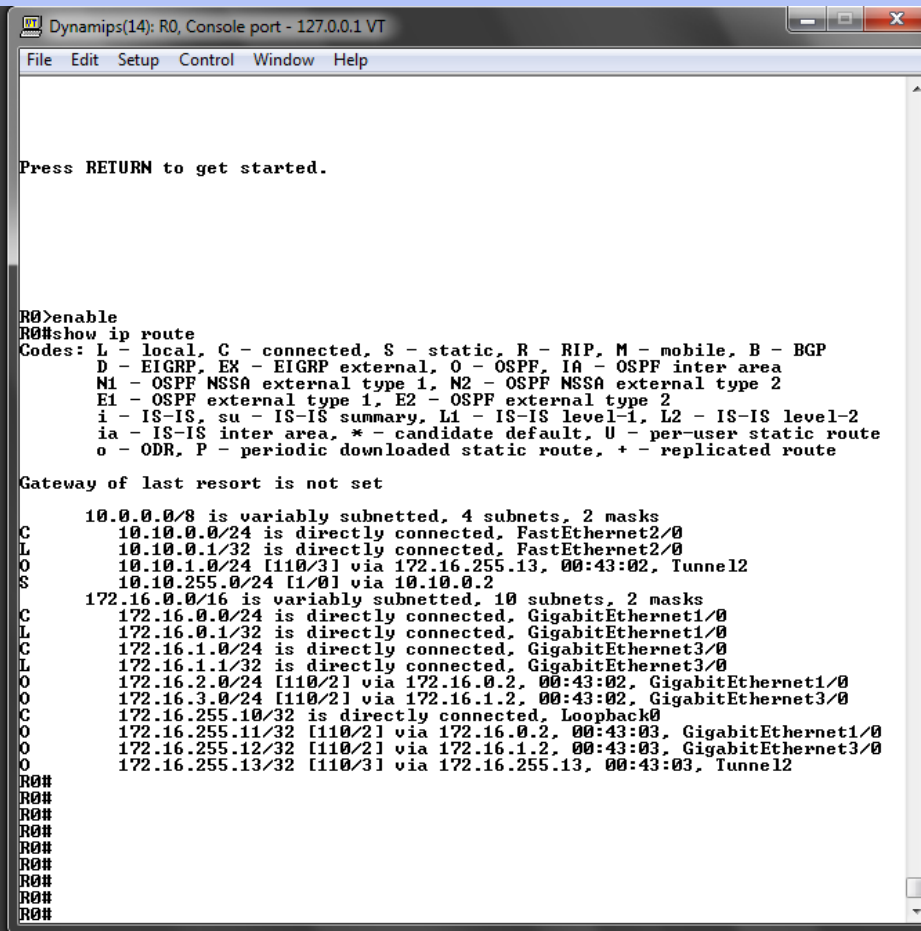
Verify Connectivity

- Ping all of the interfaces
- For example from R4 to R5
 - ping 10.10.1.2
- These should all work

Verify Connectivity

- Examine the routing table
- Here from R0

Verify Connectivity



```
Dynamips(14): R0, Console port - 127.0.0.1 VT
File Edit Setup Control Window Help

Press RETURN to get started.

R0>enable
R0#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, + - replicated route

Gateway of last resort is not set

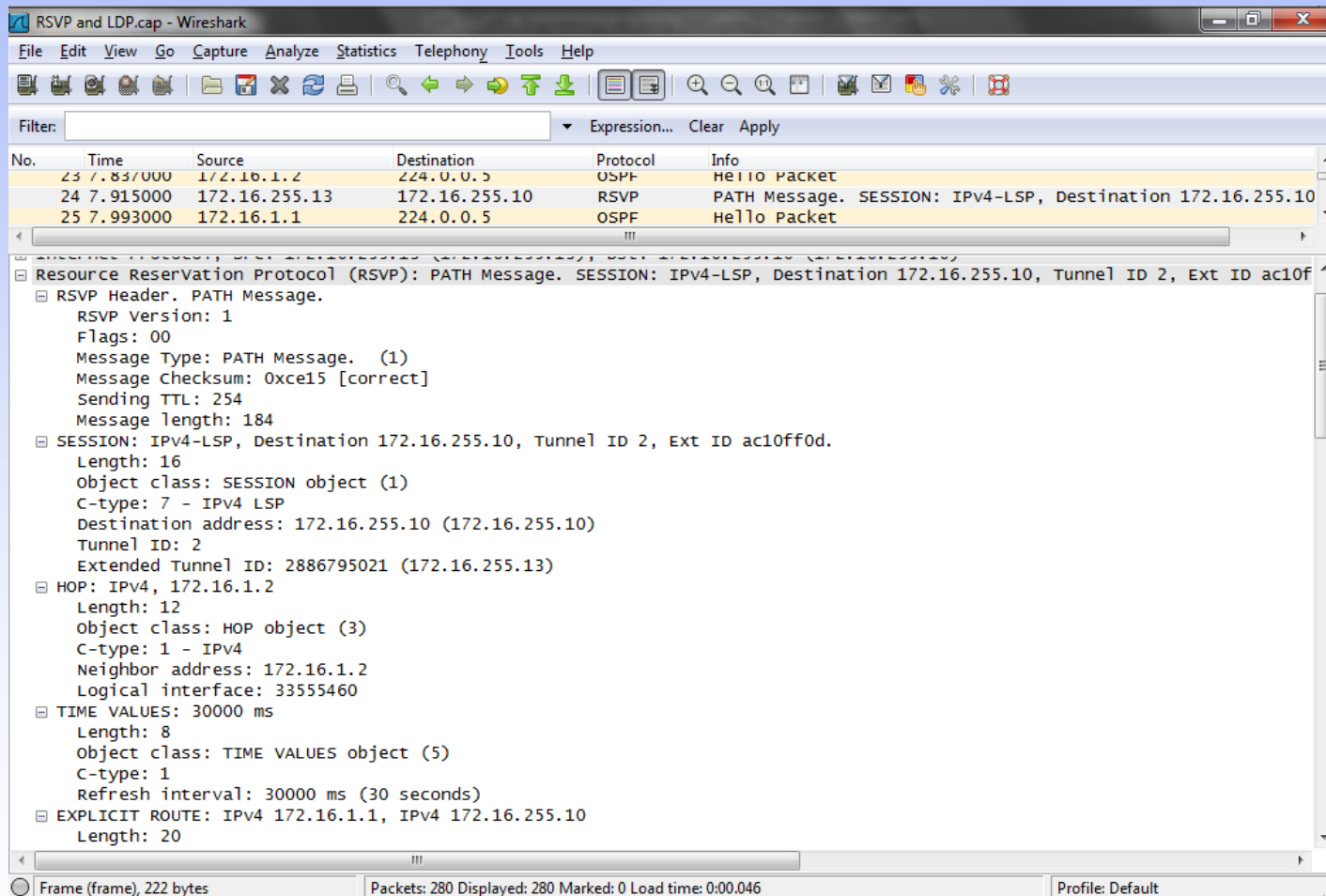
    10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.10.0.0/24 is directly connected, FastEthernet2/0
L       10.10.0.1/32 is directly connected, FastEthernet2/0
O       10.10.1.0/24 [110/3] via 172.16.255.13, 00:43:02, Tunnel2
S       10.10.255.0/24 [1/0] via 10.10.0.2

    172.16.0.0/16 is variably subnetted, 10 subnets, 2 masks
C       172.16.0.0/24 is directly connected, GigabitEthernet1/0
L       172.16.0.1/32 is directly connected, GigabitEthernet1/0
C       172.16.1.0/24 is directly connected, GigabitEthernet3/0
L       172.16.1.1/32 is directly connected, GigabitEthernet3/0
O       172.16.2.0/24 [110/2] via 172.16.0.2, 00:43:02, GigabitEthernet1/0
O       172.16.3.0/24 [110/2] via 172.16.1.2, 00:43:02, GigabitEthernet3/0
C       172.16.255.10/32 is directly connected, Loopback0
O       172.16.255.11/32 [110/2] via 172.16.0.2, 00:43:03, GigabitEthernet1/0
O       172.16.255.12/32 [110/2] via 172.16.1.2, 00:43:03, GigabitEthernet3/0
O       172.16.255.13/32 [110/3] via 172.16.255.13, 00:43:03, Tunnel2
R0#
R0#
R0#
R0#
R0#
R0#
R0#
R0#
```

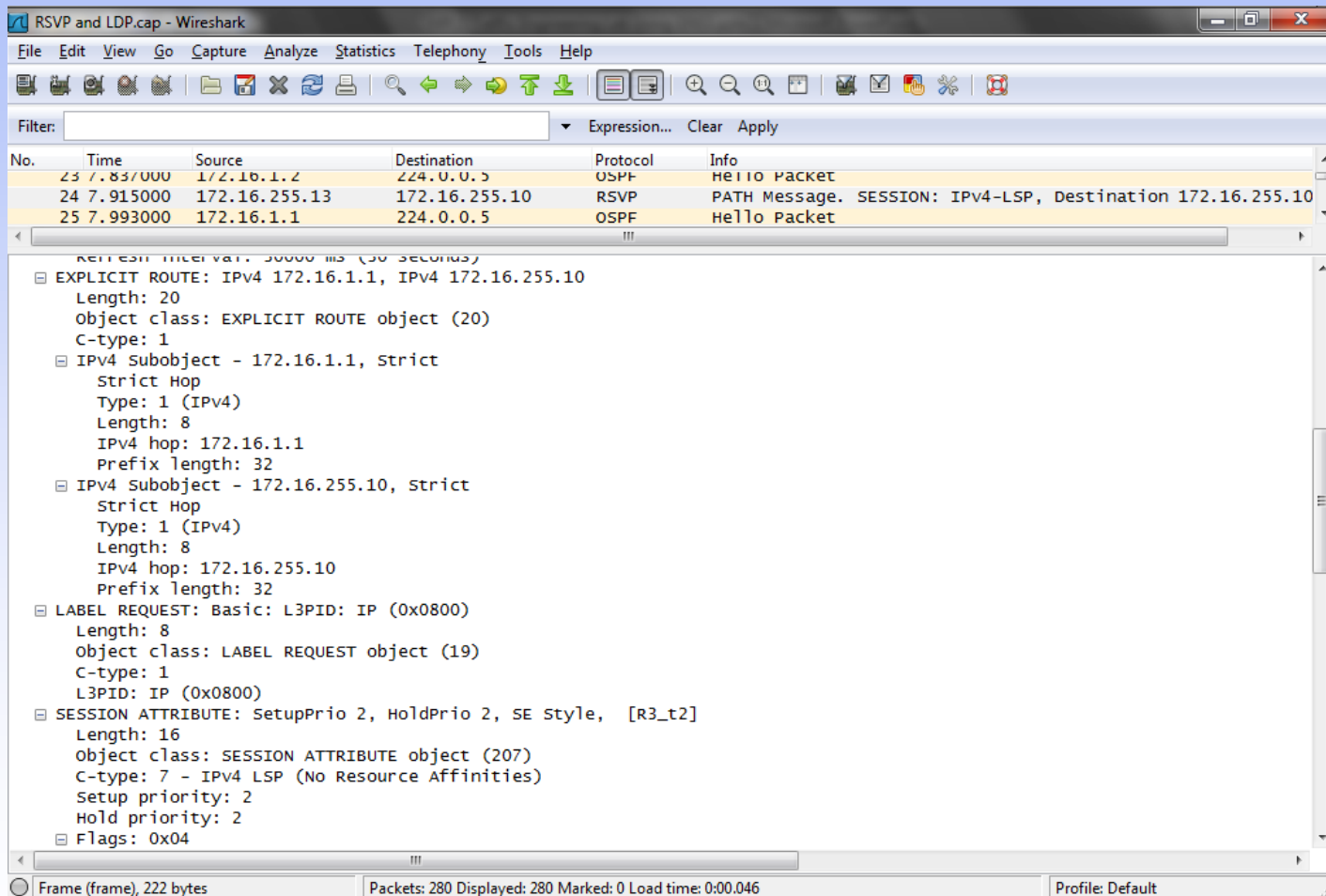
RSVP Capture File

- Let's open the same capture file
- We will see how RSVP did its work for us
- Go to frame 24
- Expand all of the RSVP level
- Here the two ends are talking to each other in a path message over tunnel 2
- As we can see there is a lot to say

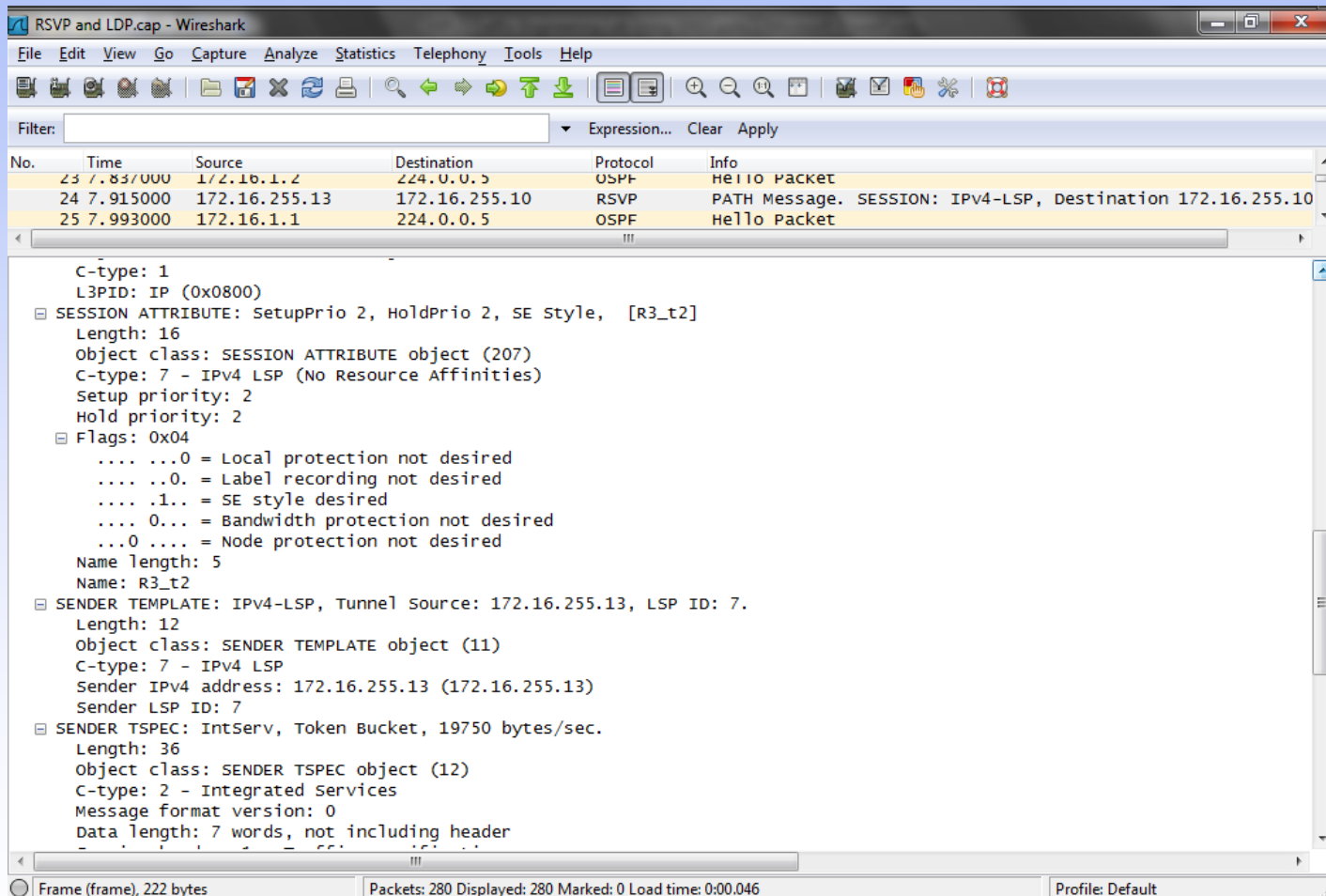
RSVP Capture File



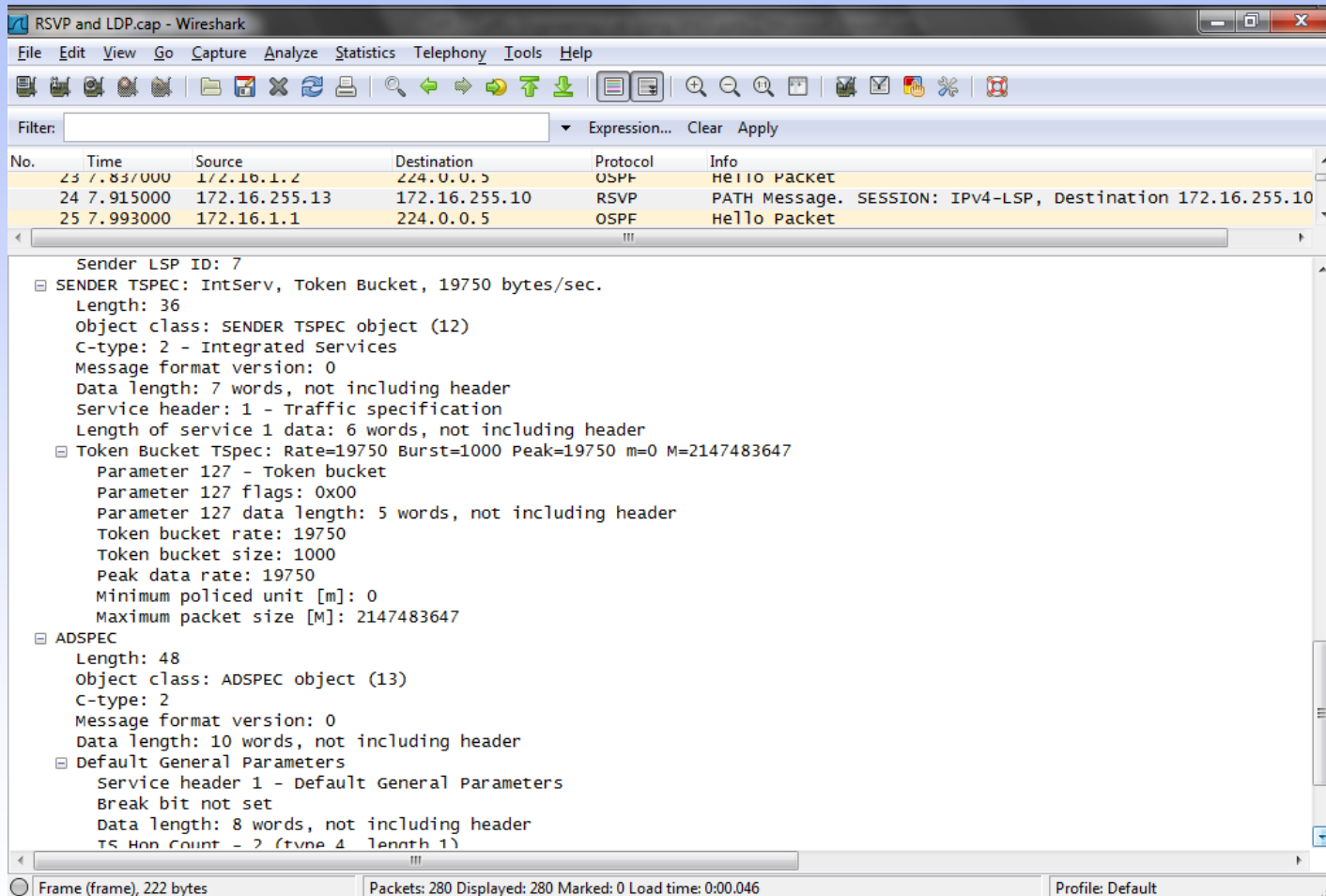
RSVP Capture File



RSVP Capture File



RSVP Capture File



The image shows a Wireshark capture window titled "RSVP and LDP.cap - Wireshark". The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Help), a toolbar, and a filter field. The packet list shows three packets:

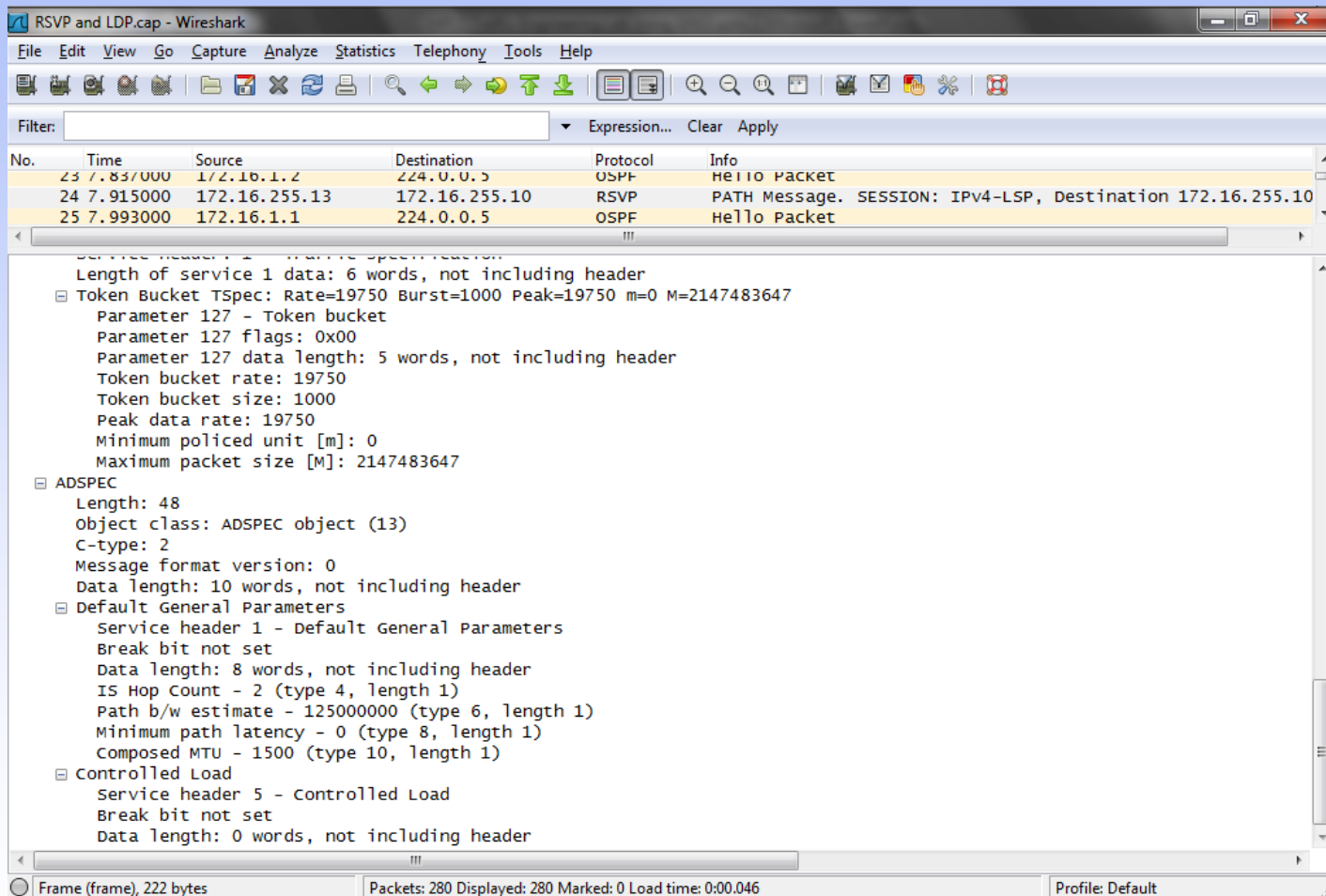
No.	Time	Source	Destination	Protocol	Info
23	7.837000	172.16.1.2	224.0.0.5	OSPF	Hello Packet
24	7.915000	172.16.255.13	172.16.255.10	RSVP	PATH Message. SESSION: IPv4-LSP, Destination 172.16.255.10
25	7.993000	172.16.1.1	224.0.0.5	OSPF	Hello Packet

The packet details pane for the selected packet (24) shows the following structure:

- Sender LSP ID: 7
- SENDER TSPEC: IntServ, Token Bucket, 19750 bytes/sec.
 - Length: 36
 - Object class: SENDER TSPEC object (12)
 - C-type: 2 - Integrated Services
 - Message format version: 0
 - Data length: 7 words, not including header
 - Service header: 1 - Traffic specification
 - Length of service 1 data: 6 words, not including header
- Token Bucket TSpec: Rate=19750 Burst=1000 Peak=19750 m=0 M=2147483647
 - Parameter 127 - Token bucket
 - Parameter 127 flags: 0x00
 - Parameter 127 data length: 5 words, not including header
 - Token bucket rate: 19750
 - Token bucket size: 1000
 - Peak data rate: 19750
 - Minimum policed unit [m]: 0
 - Maximum packet size [M]: 2147483647
- ADSPEC
 - Length: 48
 - Object class: ADSPEC object (13)
 - C-type: 2
 - Message format version: 0
 - Data length: 10 words, not including header
 - Default General Parameters
 - Service header 1 - Default General Parameters
 - Break bit not set
 - Data length: 8 words, not including header
 - TS Hop Count - 2 (type 4 length 1)

The status bar at the bottom indicates: Frame (frame), 222 bytes; Packets: 280 Displayed: 280 Marked: 0 Load time: 0:00.046; Profile: Default.

RSVP Capture File



What Do The Commands Do

- Let's look at some of the commands used to see what they do
 - ip cef
 - To turn on CEF – Cisco Express Forwarding
 - This is required by Cisco routers doing tag switching of which label switching is a form
 - As Cisco says
 - CEF switching is a proprietary form of scalable switching intended to tackle the problems associated with demand caching

What Do The Commands Do

- With CEF switching, the information which is conventionally stored in a route cache is split up over several data structures
- The CEF code is able to maintain these data structures in the Gigabit Route Processor
- The data structures that provide optimized lookup for efficient packet forwarding include
 - It is on by default in some devices
- mpls traffic-eng tunnels
 - Enables traffic engineering on the router

What Do The Commands Do

- tunnel destination 172.16.255.13
 - The IP address of the end of the tunnel
 - Where the traffic is going
- tunnel mode mpls traffic-eng
 - Specifies the encapsulation method for the tunnel
 - In this case mpls

What Do The Commands Do

- tunnel mpls traffic-eng autoroute announce
 - This tells the routing protocol to consider the existences of the tunnel in its route calculations
 - Otherwise the routing protocol will not send traffic through the tunnel
 - If not this, then a static route will have to be used

What Do The Commands Do

- tunnel mpls traffic-eng priority 2 2
 - This configures the setup and reservation priority for the tunnel
 - The first number is the priority
 - The second is the hold
 - The numbers are the same
- tunnel mpls traffic-eng bandwidth 158
 - This is the bandwidth of the tunnel in kbps
 - The range is 1 to 4294967295

What Do The Commands Do

- tunnel mpls traffic-eng path-option 1 explicit name BOTTOM
 - This sets up a path option for the tunnel
 - In this case the path is an IP explicit path related back to the tunnel destination
 - The path is also in this case given a name
- no routing dynamic
 - To prevent routing updates being sent through the tunnel
 - They still go out, just not through the tunnel

What Do The Commands Do

- mpls ip

- Enables the router to perform forwarding of MPLS labels in IPv4 packets along normally routed paths

- mpls traffic-eng tunnels

- Sets up MPLS traffic engineering for an interface

What Do The Commands Do

- ip rsvp bandwidth 750000
 - Specifies how much bandwidth can be used by traffic engineering
- ip rsvp resource-provider none
 - To enable or in this case disable any resource-provider, in other words a quality of service setting, for aggregate flow
 - Aggregate flow refers to the traffic and its corresponding QoS requirements

What Do The Commands Do

- mpls ldp autoconfig area 0
 - This command is to ease the configuration of LDP for the routing protocol
 - Without it all of the interfaces would have to be setup one by one for LDP
 - LDP is the Label Distribution Protocol
- mpls traffic-eng router-id Loopback0
 - This tells us the router identifier for the node is the IP address of that interface

What Do The Commands Do

- mpls traffic-eng area 0
 - When OSPF is the routing protocol this configures the router to flood traffic engineering for the specified OSPF area

What Do The Commands Do

- ip explicit-path name BOTTOM enable
 - Specifies an explicit path name or number
 - Enables that path
 - The commands that follow detail the path or paths
 - Such as
 - next-address 172.16.1.2
 - next-address 172.16.3.2
 - Bottom or top forces traffic though a specific path

MPLS with Traffic Engineering

- As we can see when compared to the basic MPLS configuration this one is much more complex
- But MPLS with traffic engineering is the main point to MPLS, therefore a configuration along these lines is called for