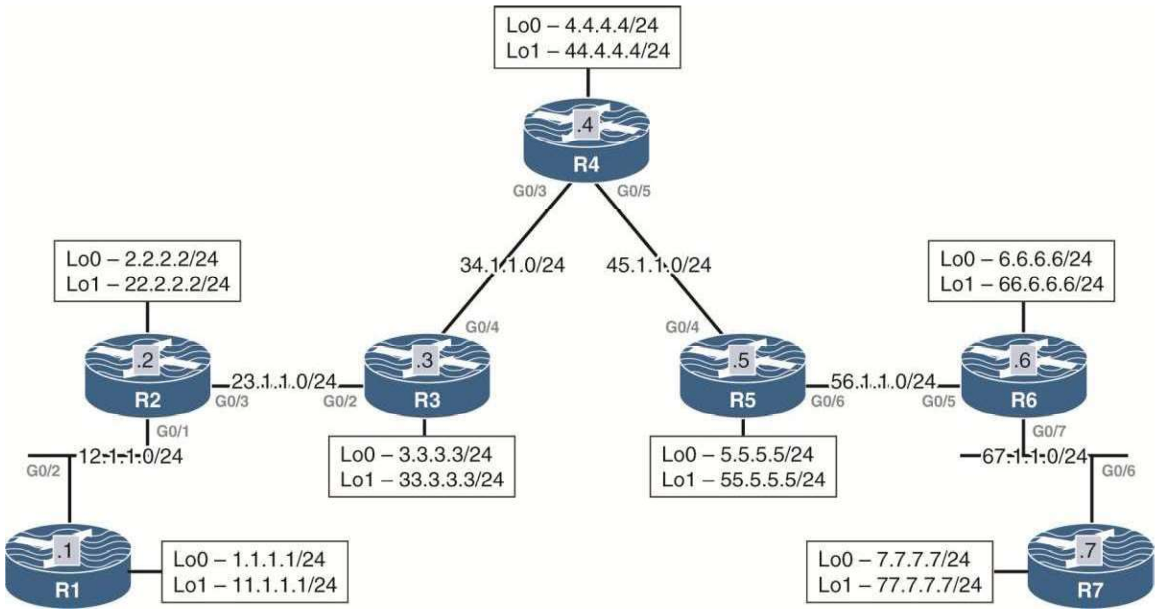
Lab 1
Configuring Label Distribution Protocol



Lab Setup:

To copy and paste the initial configurations, go to the Initial-config folder → MPLS folder → Lab-1.

Task 1

Configure OSPF Area 0 on all links in the previous topology except the Loopback1 interfaces. Configure the OSPF router IDs of these routers to be 0.0.0.*x*, where *x* is the router number.

The following configures OSPF on all the routers in the topology. The OSPF router ID has been manually assigned to each router using the 0.0.0.x format, where x represents the router number. OSPF for Area 0 is enabled on all the gigabit and loopback 0 interfaces:

On R1:

```
R1(config)#router ospf 1
R1(config-router)#router-id 0.0.0.1
R1(config-router)#network 12.1.1.1 0.0.0.0 area 0
R1(config-router)#network 1.1.1.1 0.0.0.0 area 0
```

On R2:

```
R2(config)#router ospf 1
R2(config-router)#router-id 0.0.0.2
R2(config-router)#network 12.1.1.2 0.0.0.0 area 0
R2(config-router)#network 2.2.2.2 0.0.0.0 area 0
R2(config-router)#network 23.1.1.2 0.0.0.0 area 0
```

You should see the following console message:

```
%OSPF-5-ADJCHG: Process 1, Nbr 0.0.0.1 on GigabitEthernet0/1 from
LOADING to FULL, Loading Done
```

On R3:

```
R3(config)#router ospf 1
R3(config-router)#router-id 0.0.0.3
R3(config-router)#network 3.3.3.3 0.0.0.0 area 0
R3(config-router)#network 34.1.1.3 0.0.0.0 area 0
R3(config-router)#network 23.1.1.3 0.0.0.0 area 0
```

You should see the following console message:

```
%OSPF-5-ADJCHG: Process 1, Nbr 0.0.0.2 on GigabitEthernet0/2 from
LOADING to FULL, Loading Done
```

On R4:

```
R4(config)#router ospf 1
R4(config-router)#router-id 0.0.0.4
R4(config-router)#network 45.1.1.4 0.0.0.0 area 0
R4(config-router)#network 34.1.1.4 0.0.0.0 area 0
R4(config-router)#network 4.4.4.4 0.0.0.0 area 0
```

You should see the following console message:

```
%OSPF-5-ADJCHG: Process 1, Nbr 0.0.0.3 on GigabitEthernet0/3 from
LOADING to FULL, Loading Done
```

On R5:

```
R5(config)#router ospf 1
R5(config-router)#router-id 0.0.0.5
R5(config-router)#network 5.5.5.5 0.0.0.0 area 0
R5(config-router)#network 56.1.1.5 0.0.0.0 area 0
R5(config-router)#network 45.1.1.5 0.0.0.0 area 0
```

You should see the following console message:

```
%OSPF-5-ADJCHG: Process 1, Nbr 0.0.0.4 on GigabitEthernet0/4 from
LOADING to FULL, Loading Done
```

On R6:

```
R6(config)#router ospf 1
R6(config-router)#router-id 0.0.0.6
R6(config-router)#network 6.6.6.6 0.0.0.0 area 0
R6(config-router)#network 67.1.1.6 0.0.0.0 area 0
R6(config-router)#network 56.1.1.6 0.0.0.0 area 0
```

You should see the following console message:

```
%OSPF-5-ADJCHG: Process 1, Nbr 0.0.0.5 on GigabitEthernet0/5 from
LOADING to FULL, Loading Done
```

On R7:

```
R7(config)#router ospf 1
R7(config-router)#router-id 0.0.0.7
R7(config-router)#network 67.1.1.7 0.0.0.0 area 0
R7(config-router)#network 7.7.7.7 0.0.0.0 area 0
```

You should see the following console message:

```
%OSPF-5-ADJCHG: Process 1, Nbr 0.0.0.6 on GigabitEthernet0/6 from
LOADING to FULL, Loading Done
```

The console log messages in the above verify the OSPF adjacencies between the devices. The routing table on R1 and R7 below show the routes they learn via OSPF. Each loopback 0 interface from each device in the topology is present in their routing tables, thereby confirming successful OSPF route advertisements:

On R1:

```
R1#show ip route ospf | begin Gate
Gateway of last resort is not set

      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/2] via 12.1.1.2, 00:09:15, GigabitEthernet0/2
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/3] via 12.1.1.2, 00:09:05, GigabitEthernet0/2
      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/4] via 12.1.1.2, 00:08:03, GigabitEthernet0/2
      5.0.0.0/32 is subnetted, 1 subnets
O        5.5.5.5 [110/5] via 12.1.1.2, 00:07:07, GigabitEthernet0/2
      6.0.0.0/32 is subnetted, 1 subnets
O        6.6.6.6 [110/6] via 12.1.1.2, 00:05:57, GigabitEthernet0/2
      7.0.0.0/32 is subnetted, 1 subnets
O        7.7.7.7 [110/7] via 12.1.1.2, 00:05:03, GigabitEthernet0/2
      23.0.0.0/24 is subnetted, 1 subnets
O        23.1.1.0 [110/2] via 12.1.1.2, 00:09:15, GigabitEthernet0/2
      34.0.0.0/24 is subnetted, 1 subnets
O        34.1.1.0 [110/3] via 12.1.1.2, 00:09:05, GigabitEthernet0/2
      45.0.0.0/24 is subnetted, 1 subnets
O        45.1.1.0 [110/4] via 12.1.1.2, 00:08:03, GigabitEthernet0/2
      56.0.0.0/24 is subnetted, 1 subnets
O        56.1.1.0 [110/5] via 12.1.1.2, 00:06:07, GigabitEthernet0/2
      67.0.0.0/24 is subnetted, 1 subnets
O        67.1.1.0 [110/6] via 12.1.1.2, 00:05:57, GigabitEthernet0/2
```

On R7:

```
R7#show ip route ospf | begin Gate
Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
O        1.1.1.1 [110/7] via 67.1.1.6, 00:05:21, GigabitEthernet0/6
      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/6] via 67.1.1.6, 00:05:21, GigabitEthernet0/6
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/5] via 67.1.1.6, 00:05:21, GigabitEthernet0/6
      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/4] via 67.1.1.6, 00:05:21, GigabitEthernet0/6
      5.0.0.0/32 is subnetted, 1 subnets
```

```
O        5.5.5.5 [110/3] via 67.1.1.6, 00:05:21, GigabitEthernet0/6
      6.0.0.0/32 is subnetted, 1 subnets
O        6.6.6.6 [110/2] via 67.1.1.6, 00:05:21, GigabitEthernet0/6
      12.0.0.0/24 is subnetted, 1 subnets
O        12.1.1.0 [110/6] via 67.1.1.6, 00:05:21, GigabitEthernet0/6
      23.0.0.0/24 is subnetted, 1 subnets
O        23.1.1.0 [110/5] via 67.1.1.6, 00:05:21, GigabitEthernet0/6
      34.0.0.0/24 is subnetted, 1 subnets
O        34.1.1.0 [110/4] via 67.1.1.6, 00:05:21, GigabitEthernet0/6
      45.0.0.0/24 is subnetted, 1 subnets
O        45.1.1.0 [110/3] via 67.1.1.6, 00:05:21, GigabitEthernet0/6
      56.0.0.0/24 is subnetted, 1 subnets
O        56.1.1.0 [110/2] via 67.1.1.6, 00:05:21, GigabitEthernet0/6
```

Task 2

Configure Label Distribution Protocol on the links interconnecting the routers in this topology. Ensure that the LDP ID is based on the IP address assigned to the Loopback0 interfaces of these routers. You may override a command from the previous task to accomplish this task.

In MPLS, routers assign to certain prefixes, identifiers known as a label. These labels are used to forward data traffic between routers that are configured for MPLS label switching. Such routers are known as Label-Switch Routers or LSRs. LSRs must advertise their label-to-prefix mapping to neighboring LSRs in order to establish a label-switched path(LSP). LSRs use a label distribution protocol to facilitate this exchange.

Tag Distribution Protocol (TDP) and Label Distribution Protocol (LDP) are examples of these label distribution protocols used within the MPLS architecture. TDP is Cisco proprietary, is obsoleted, and replaced with the standardized LDP. LDP will be the focus of this lab.

The MPLS label distribution protocol that should be used to exchange labels is configured using the mpls label protocol command in Cisco IOS. Prior to IOS version 12.4, the default was TDP. After TDP was obsoleted, LDP became the default setting on Cisco IOS routers. The show mpls label protocol ? command below confirms this:

On R1:

```
R1(config)#mpls label protocol ?
```

```
ldp Use LDP (default)
tdp  Use TDP
```

In the above, the “(default)” next to the “ldp” option in the command signifies that LDP is the default for modern IOS routers.

Once LDP is enabled on two adjacent LSRs, a bidirectional LDP session is formed and the two LSRs become LDP peers. As LDP peers, they exchange label mapping information between each other over the bidirectional LDP session. The session is considered bidirectional because routers can both send and receive label information over the same session.

LDP session establishment is a two-step process:

1. Discover potential peers
2. Establish sessions with those peers

The first step involves setting up an LDP hello adjacency. This method utilizes one of two methods to discover potential peers: Basic and Extended. Basic discovery relies on link-level multicast to find directly-connected peers while extended discovery uses targeted destination addresses to discover peers reachable at the network level. After establishing a hello adjacency, LDP moves on to form an actual LDP session. This session is based on TCP and is the session over which LDP data is actually transferred. This task deals primarily with the basic discovery method and how LDP uses it to form the LDP TCP session.

Using the basic discovery method, LSRs configured for LDP will multicast LDP hello messages to the 224.0.0.2, all-routers, link-level multicast group. These hellos are sent as UDP messages destined to UDP port 646, the well-known LDP UDP port. Once two routers exchange LDP link-level UDP hello messages, they form a hello adjacency. Once potential neighbors are discovered, LDP proceeds to establish a TCP transport session between the peers. It is over this TCP session that label mappings are reliably exchanged between peers.

This task requires the routers to form LDP sessions with their directly-connected routers. Since the routers are directly-connected, the basic discovery mechanism described above is sufficient to complete the task. To accomplish this, MPLS label switching needs to be enabled on the router. Typically, MPLS label switching is enabled by default on most Cisco IOS platforms. If MPLS is not enabled it can be enabled using the mpls ip command in global configuration mode.

After enabling globally, similar to IP routing, each interface that participates in MPLS label switching needs to be explicitly configured to do so. MPLS label switching and LDP are enabled on the interconnecting links between all the devices with the interface-level command mpls ip. For explanation purposes, for now, the mpls ip command is only issued on R1 and R2 connecting gigabit interfaces:

On R1:

```
R1(config)#interface g0/2
R1(config-if)#mpls ip
```

On R2:

```
R2(config)#interface g0/1
R2(config-if)#mpls ip
```

As covered above, LDP's basic discovery method periodically sends UDP hello's out the interfaces enabled for MPLS. These hello's are source and destined to the UDP port 646.The debug ip udp command displays the header details of these hello packets as shown below on R1 and R2:

On R1:

```
UDP: rcvd src=12.1.1.2(646), dst=224.0.0.2(646), length=42
```

On R2:

```
UDP: rcvd src=12.1.1.2(646), dst=224.0.0.2(646), length=42

UDP: rcvd src=12.1.1.1(646), dst=224.0.0.2(646), length=42

UDP: rcvd src=12.1.1.1(646), dst=224.0.0.2(646), length=42
```

R1 and R2 are sending hello messages onto the link, but there has been no indication of an LDP adjacency forming between the two of them. In these situations the show mpls ldp discovery command can be used to see what LDP hellos the router has received and the interfaces on which they were received. The following shows the output of the same command on R1:

On R1:

```
R1#show mpls ldp discovery

 Local LDP Identifier:
    11.1.1.1:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet0/2 (ldp): xmit/recv
```

```
            LDP Id: 22.2.2.2:0; no route
```

The output above reports that R1 has received a hello from a router with LDP ID 22.2.2.2. Following the LDP ID of the remote peer is a line that reads "no route". Earlier, using the debug ip udp command, it was shown that R1 and R2 are using a source IP address of 12.1.1.1 and 12.1.1.2 respectively when sending their LDP hellos. If this is the case, why does R1 believe the LDP hello was received from an LDP ID of 22.2.2.2?

The reason for this can be understood by examining the UDP hello message sent by R2 to the 224.0.0.2 address below.

```
Internet Protocol Version 4, Src: 12.1.1.2, Dst: 224.0.0.2
User Datagram Protocol, Src Port: 646, Dst Port: 646
Label Distribution Protocol
    Version: 1
    PDU Length: 30
    LSR ID: 22.2.2.2
    Label Space ID: 0
    Hello Message
        0... .... = U bit: Unknown bit not set
        Message Type: Hello Message (0x100)
        Message Length: 20
        Message ID: 0x00000000
        Common Hello Parameters
        IPv4 Transport Address
            00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
            TLV Type: IPv4 Transport Address (0x401)
            TLV Length: 4
            IPv4 Transport Address: 22.2.2.2
```

The UDP hello contains a transport address field. The IPv4 transport address field designates the IP address the sending LDP router is instructing receiving LDP routers to use, when forming the TCP LDP session. R1 will also include its own chosen address as the transport address in the hello towards R2.

By default, LDP uses the LDP router-ID as the transport address. Similar to IGPs and BGP, if not statically configured, the router uses the highest loopback IP address as the LDP router ID. Since the LDP router ID is also used as the transport address, the highest loopback IP address on the router becomes both the LDP router ID and the transport address for the LDP session. The highest IP address on R1 on an operational interface is 11.1.1.1, and on R2 it is 22.2.2.2. Both routers use these as the transport addresses.

In order to allow a TCP session to be established using these addresses, R1 and R2 should have reachability to these addresses. This means they must have routes in their routing tables to reach them. The **show mpls ldp discovery** output from R1 above indicated that it did not have a route to this network using the words "no route". The **show ip route**

22.2.2.2 command on R1 confirms this situation:

On R1:

```
R1#show ip route 22.2.2.2

% Network not in table
```

Similarly, R2's show mpls ldp discovery command indicates it has no route to reach the 11.1.1.1 address on R1. its show ip route 11.1.1.1 command also confirms:

On R2:

```
R2#show mpls ldp discovery

 Local LDP Identifier:
    22.2.2.2:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet0/1 (ldp): xmit/recv
            LDP Id: 11.1.1.1:0; no route

R2#show ip route 11.1.1.1

% Network not in table
```

Without routes to the transport addresses, the LDP session will not form. Typically, reachability to the LDP transport is provided by an IGP. The lab topology uses OSPF as the underlying IGP in the network. The reason neither R1 or R2 have routes to reach these addresses is because they are the addresses assigned to the loopback 1 interface on R1 and R2. These interfaces have not been added to the OSPF process running on the routers as shown below:

On R1:

```
R1#show ip ospf interface brief

Interface     PID   Area             IP Address/Mask    Cost  State Nbrs F/C
Lo0           1     0                1.1.1.1/32         1     LOOP  0/0
Gi0/2         1     0                12.1.1.1/24        1     BDR   1/1
```

On R2:

```
R2#show ip ospf interface brief
```

```
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Lo0          1     0               2.2.2.2/32         1     LOOP  0/0
Gi0/3        1     0               23.1.1.2/24        1     BDR   1/1
Gi0/1        1     0               12.1.1.2/24        1     DR    1/1
```

This is why the task specifies using loopback 0 as the LDP router-ID. The LDP router-ID can be manually set with the mpls ldp router-id command. Appending the force keyword to this command will cause the LDP router ID changes to happen immediately. This command is issued on R1 and R2:

On R1 and R2:

```
Rx(config)#mpls ldp router-id lo0 force
```

Notice the packet capture of the UDP hello from R2 once again. The transport address now reflects the IPv4 address of the loopback 0 interface on R2. The same happens on R1:

```
Internet Protocol Version 4, Src: 12.1.1.2, Dst: 224.0.0.2
User Datagram Protocol, Src Port: 646, Dst Port: 646
Label Distribution Protocol
    Version: 1
    PDU Length: 30
    LSR ID: 2.2.2.2
    Label Space ID: 0
    Hello Message
        0... .... = U bit: Unknown bit not set
        Message Type: Hello Message (0x100)
        Message Length: 20
        Message ID: 0x00000000
        Common Hello Parameters
        IPv4 Transport Address
            00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
            TLV Type: IPv4 Transport Address (0x401)
            TLV Length: 4
            IPv4 Transport Address: 2.2.2.2
```

The show mpls ldp discovery command on R1 reflects the change in the LDP ID as well:

On R1:

```
R1#show mpls ldp discovery

 Local LDP Identifier:
    1.1.1.1:0
```

```
    Discovery Sources:
    Interfaces:
        GigabitEthernet0/2 (ldp): xmit/recv
            LDP Id: 2.2.2.2:0
```

In the above, the “no route” text is no longer present because R1 has a route to reach 2.2.2.2 in its routing table. The same applies to R2. Since R1 and R2 have reachability to each other's loopback addresses, they form LDP neighbor adjacencies and a TCP session with each other. This is seen with the show mpls ldp neighbor command:

```
R1#show mpls ldp neighbor

    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0
     TCP connection: 2.2.2.2.32572 - 1.1.1.1.646
     State: Oper; Msgs sent/rcvd: 24/24; Downstream
     Up time: 00:03:55
     LDP discovery sources:
       GigabitEthernet0/2, Src IP addr: 12.1.1.2
        Addresses bound to peer LDP Ident:
          12.1.1.2        23.1.1.2        2.2.2.2         22.2.2.2
```

On R2:

```
R2#show mpls ldp neighbor

    Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 2.2.2.2:0
     TCP connection: 1.1.1.1.646 - 2.2.2.2.32572
     State: Oper; Msgs sent/rcvd: 24/24; Downstream
     Up time: 00:04:09
     LDP discovery sources:
       GigabitEthernet0/1, Src IP addr: 12.1.1.1
        Addresses bound to peer LDP Ident:
          12.1.1.1        1.1.1.1         11.1.1.1
```

Notice the LDP identifier value in the output above. It is a 6 byte value that is composed of two parts :

1. 4 byte Router ID
2. 2 byte local space ID

The first four bytes identify the LSR with a router ID. The determination of this value was explained earlier. The last two indicate the label space. LDP supports two types of label spaces: Platform-wide label space or per-interface label space. The label space identifies at what level the assigned labels are significant on the LSR. For per-interface label spaces, the labels assigned have an interface-level significance. This simply means that, when performing label operations, the interface and MPLS label are used as matching criteria.

Platform-wide label space uses the same label space for all interfaces on the LSR. This simply means that, when performing label operations, the MPLS label is the only matching criteria. Platform-wide label space is represented as ":0" in the last two bytes of the LDP identifier. This is what is seen in the LDP identifier above on R1 and R2.

The output above also includes the details about the TCP connection. The TCP connection from R2 is sourced from it's 2.2.2.2 address to R1's 1.1.1.1 address. Note the port number used is TCP port 646 (assigned to LDP) on R1 and from R2, it is 32572. This makes R2 the TCP client and R1 the TCP server. LSRs determine the TCP active or passive role by comparing their transport addresses. The device with a higher transport address (2.2.2.2) plays the active role.

To complete this task, the above configurations are replicated on all the routers. Note, for the remaining devices, the router ID is first set with the mpls ldp router-id lo0 command. Following this, LDP is enabled per interface with the mpls ip command. For this reason, the force keyword is not required while setting the LDP router ID:

On R3, R4, R5, R6 and R7

```
Rx(config)#mpls ldp router-id lo0
```

On R2:

```
R2(config)#interface g0/3
R2(config-if)#mpls ip
```

On R3:

```
R3(config)#interface g0/2
R3(config-if)#mpls ip

R3(config)#interface g0/4
R3(config-if)#mpls ip
```

You should see the following console message:

```
%LDP-5-NBRCHG: LDP Neighbor 2.2.2.2:0 (2) is UP
```

On R4:

```
R4(config)#interface g0/3
R4(config-if)#mpls ip

R4(config)#interface g0/5
R4(config-if)#mpls ip
```

You should see the following console message:

```
%LDP-5-NBRCHG: LDP Neighbor 3.3.3.3:0 (1) is UP
```

On R5:

```
R5(config)#mpls ldp router-id lo0

R5(config)#interface g0/4
R5(config-if)#mpls ip

R5(config)#interface g0/6
R5(config-if)#mpls ip
```

You should see the following console message:

```
%LDP-5-NBRCHG: LDP Neighbor 4.4.4.4:0 (1) is UP
```

On R6:

```
R6(config)#interface g0/5
R6(config-if)#mpls ip

R6(config-if)#interface g0/7
R6(config)#mpls ip
```

You should see the following console message:

```
%LDP-5-NBRCHG: LDP Neighbor 5.5.5.5:0 (1) is UP
```

On R7:

```
R7(config)#interface g0/6
R7(config-if)#mpls ip
```

You should see the following console message:

```
%LDP-5-NBRCHG: LDP Neighbor 6.6.6.6:0 (1) is UP
```

After completing the above, LDP sessions successfully form between the devices. The show mpls ldp neighbor command for all routers can be used to verify this:

On R1:

```
R1#show mpls ldp neighbor

    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0
        TCP connection: 2.2.2.2.32572 - 1.1.1.1.646
        State: Oper; Msgs sent/rcvd: 33/33; Downstream
        Up time: 00:14:30
        LDP discovery sources:
          GigabitEthernet0/2, Src IP addr: 12.1.1.2
        Addresses bound to peer LDP Ident:
          12.1.1.2        23.1.1.2        2.2.2.2         22.2.2.2
```

On R2:

```
R2#show mpls ldp neighbor

    Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 1.1.1.1.646 - 2.2.2.2.32572
        State: Oper; Msgs sent/rcvd: 33/33; Downstream
        Up time: 00:14:30
        LDP discovery sources:
          GigabitEthernet0/1, Src IP addr: 12.1.1.1
        Addresses bound to peer LDP Ident:
          12.1.1.1        1.1.1.1         11.1.1.1
    Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 3.3.3.3.40659 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 20/20; Downstream
        Up time: 00:03:28
        LDP discovery sources:
          GigabitEthernet0/3, Src IP addr: 23.1.1.3
        Addresses bound to peer LDP Ident:
          23.1.1.3        34.1.1.3        3.3.3.3         33.3.3.3
```

On R3:

```
R3#show mpls ldp neighbor
```

```
    Peer LDP Ident: 4.4.4.4:0; Local LDP Ident 3.3.3.3:0
        TCP connection: 4.4.4.4.28926 - 3.3.3.3.646
        State: Oper; Msgs sent/rcvd: 22/21; Downstream
        Up time: 00:04:35
        LDP discovery sources:
          GigabitEthernet0/4, Src IP addr: 34.1.1.4
        Addresses bound to peer LDP Ident:
          34.1.1.4        45.1.1.4        4.4.4.4         44.4.4.4
    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 3.3.3.3:0
        TCP connection: 2.2.2.2.646 - 3.3.3.3.40659
        State: Oper; Msgs sent/rcvd: 20/20; Downstream
        Up time: 00:03:28
        LDP discovery sources:
          GigabitEthernet0/2, Src IP addr: 23.1.1.2
        Addresses bound to peer LDP Ident:
          12.1.1.2        23.1.1.2        2.2.2.2         22.2.2.2
```

On R4:

R4#**show mpls ldp neighbor**

```
    Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 4.4.4.4:0
        TCP connection: 3.3.3.3.646 - 4.4.4.4.28926
        State: Oper; Msgs sent/rcvd: 21/22; Downstream
        Up time: 00:04:35
        LDP discovery sources:
          GigabitEthernet0/3, Src IP addr: 34.1.1.3
        Addresses bound to peer LDP Ident:
          23.1.1.3        34.1.1.3        3.3.3.3         33.3.3.3
    Peer LDP Ident: 5.5.5.5:0; Local LDP Ident 4.4.4.4:0
        TCP connection: 5.5.5.5.38712 - 4.4.4.4.646
        State: Oper; Msgs sent/rcvd: 21/22; Downstream
        Up time: 00:04:20
        LDP discovery sources:
          GigabitEthernet0/5, Src IP addr: 45.1.1.5
        Addresses bound to peer LDP Ident:
          45.1.1.5        56.1.1.5        5.5.5.5         55.5.5.5
```

On R5:

R5#**show mpls ldp neighbor**

```
    Peer LDP Ident: 4.4.4.4:0; Local LDP Ident 5.5.5.5:0
        TCP connection: 4.4.4.4.646 - 5.5.5.5.38712
```

```
        State: Oper; Msgs sent/rcvd: 22/21; Downstream
        Up time: 00:04:20
        LDP discovery sources:
          GigabitEthernet0/4, Src IP addr: 45.1.1.4
        Addresses bound to peer LDP Ident:
          34.1.1.4        45.1.1.4        4.4.4.4         44.4.4.4
    Peer LDP Ident: 6.6.6.6:0; Local LDP Ident 5.5.5.5:0
        TCP connection: 6.6.6.6.57365 - 5.5.5.5.646
        State: Oper; Msgs sent/rcvd: 21/21; Downstream
        Up time: 00:04:03
        LDP discovery sources:
          GigabitEthernet0/6, Src IP addr: 56.1.1.6
        Addresses bound to peer LDP Ident:
          56.1.1.6        67.1.1.6        6.6.6.6         66.6.6.6
```

On R6:

```
R6#show mpls ldp neighbor

    Peer LDP Ident: 5.5.5.5:0; Local LDP Ident 6.6.6.6:0
        TCP connection: 5.5.5.5.646 - 6.6.6.6.57365
        State: Oper; Msgs sent/rcvd: 21/21; Downstream
        Up time: 00:04:03
        LDP discovery sources:
          GigabitEthernet0/5, Src IP addr: 56.1.1.5
        Addresses bound to peer LDP Ident:
          45.1.1.5        56.1.1.5        5.5.5.5         55.5.5.5
    Peer LDP Ident: 7.7.7.7:0; Local LDP Ident 6.6.6.6:0
        TCP connection: 7.7.7.7.36337 - 6.6.6.6.646
        State: Oper; Msgs sent/rcvd: 21/21; Downstream
        Up time: 00:03:43
        LDP discovery sources:
          GigabitEthernet0/7, Src IP addr: 67.1.1.7
        Addresses bound to peer LDP Ident:
          67.1.1.7        7.7.7.7         77.7.7.7
```

On R7:

```
R7#show mpls ldp neighbor

    Peer LDP Ident: 6.6.6.6:0; Local LDP Ident 7.7.7.7:0
        TCP connection: 6.6.6.6.646 - 7.7.7.7.36337
        State: Oper; Msgs sent/rcvd: 21/21; Downstream
        Up time: 00:03:43
        LDP discovery sources:
```

```
         GigabitEthernet0/6, Src IP addr: 67.1.1.6
       Addresses bound to peer LDP Ident:
         56.1.1.6        67.1.1.6        6.6.6.6         66.6.6.6
```

Task 3

Configure the interval for LDP neighbor discovery to be 15 seconds, with a hold timer of 45 seconds on all LSRs.

As mentioned in the earlier task, LSRs use the discovery mechanism to discover each other by periodically sending hello messages. These messages are UDP based over the UDP port 646. Similar to the IGPs, LDP also sends and expects to receive hellos at a regular interval. This is based on the hello interval timer. Failure to receive hellos within the time specified as hold time will result in tearing down the LDP session.

The show mpls ldp discovery details command can be used to verify the default hello/hold timers. The example output from R1 is used to demonstrate this. Notice, the hello timer of 5000 is expressed in milliseconds. This equates to 5 seconds. The hold timer is 15 seconds:

On R1:

```
R1#show mpls ldp discovery detail

 Local LDP Identifier:
    1.1.1.1:0
    Discovery Sources:
    Interfaces:
     GigabitEthernet0/2 (ldp): xmit/recv
         Enabled: Interface config
         Hello interval: 5000 ms; Transport IP addr: 1.1.1.1
         LDP Id: 2.2.2.2:0
           Src IP addr: 12.1.1.2; Transport IP addr: 2.2.2.2
           Hold time: 15 sec; Proposed local/peer: 15/15 sec
           Reachable via 2.2.2.2/32
           Password: not required, none, in use
            Clients: IPv4
```

This task requires the hello timer to be increased to 15 seconds and the hold timer, to 45 seconds. This can be modified with the mpls ldp discovery hello interval 15 and mpls ldp discovery hello holdtime 45 command. For both cases, the desired time is entered in seconds:

```
R1(config)#mpls ldp discovery hello interval ?
  <1-65535>  Hello interval in seconds

R1(config)#mpls ldp discovery hello holdtime ?
  <1-65535>  Holdtime in seconds
```

The commands are entered on all the routers in the topology:

On all routers:

```
Rx(config)#mpls ldp discovery hello holdtime 45
Rx(config)#mpls ldp discovery hello interval 15
```

Example output from R1 verifies the timer modifications:

On R1:

```
R1#show mpls ldp discovery detail

 Local LDP Identifier:
    1.1.1.1:0
    Discovery Sources:
    Interfaces:
     GigabitEthernet0/2 (ldp): xmit/recv
         Enabled: Interface config
         Hello interval: 15000 ms; Transport IP addr: 1.1.1.1
         LDP Id: 2.2.2.2:0
           Src IP addr: 12.1.1.2; Transport IP addr: 2.2.2.2
           Hold time: 45 sec; Proposed local/peer: 45/45 sec
           Reachable via 2.2.2.2/32
           Password: not required, none, in use
            Clients: IPv4
```

Task 4

Configure the session keepalives and hold timers of all routers to 30 and 90 seconds, respectively.

LDP also includes a mechanism to monitor the state of the TCP transport session between LDP peers. As with BGP, the session is maintained with the keepalive timers for every peer session. The capture below is an example of the LDP keepalive message:

```
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 2.2.2.2
Transmission Control Protocol, Src Port: 646, Dst Port: 39248, Seq: 1,
Ack: 1, Len: 18
Label Distribution Protocol
    Version: 1
    PDU Length: 14
    LSR ID: 1.1.1.1
    Label Space ID: 0
    Keep Alive Message
        0... .... = U bit: Unknown bit not set
        Message Type: Keep Alive Message (0x201)
        Message Length: 4
        Message ID: 0x0000076c
```

The default keepalive interval is 60 seconds as seen in the output below. If an LSR does not hear a keepalive from the neighbor for the period of the session hold time value, default 180 seconds, the LSR concludes that the peer is down and can terminate the TCP transport session.

On R1:

```
R1#show mpls ldp parameters

LDP Feature Set Manager: State Initialized
  LDP features:
    Basic
    IP-over-MPLS
    TDP
    IGP-Sync
    Auto-Configuration
    TCP-MD5-Rollover
Protocol version: 1
Session hold time: 180 sec; keep alive interval: 60 sec
Discovery hello: holdtime: 45 sec; interval: 15 sec
Discovery targeted hello: holdtime: 90 sec; interval: 10 sec
Downstream on Demand max hop count: 255
LDP for targeted sessions
LDP initial/maximum backoff: 15/120 sec
LDP loop detection: off
```

This task requires the keepalive timer and session hold time to be reduced to 30 and 90 seconds. This can be achieved by using the mpls ldp holdtime 90 command. Changing the holdtime implicitly modifies the keepalive time to 1/3rd of the hold time value. This command is issued on all routers in the topology:

On all routers:

```
Rx(config)#mpls ldp holdtime 90

%Previously established sessions may not use the new holdtime.
```

Notice the message *%Previously established sessions may not use the new holdtime* in the output above. If the timers are not affected, the clear mpls ldp neighbor * command can be used to reset the session between the device so the new timers can take effect. The example output from R1 now shows the modified timers:

On R1:

```
R1#show mpls ldp parameters

LDP Feature Set Manager: State Initialized
  LDP features:
    Basic
    IP-over-MPLS
    TDP
    IGP-Sync
    Auto-Configuration
    TCP-MD5-Rollover
Protocol version: 1
Session hold time: 90 sec; keep alive interval: 30 sec
Discovery hello: holdtime: 45 sec; interval: 15 sec
Discovery targeted hello: holdtime: 90 sec; interval: 10 sec
Downstream on Demand max hop count: 255
LDP for targeted sessions
LDP initial/maximum backoff: 15/120 sec
LDP loop detection: off
```

Task 5

Configure the LDP router ID of R1 to be its Loopback1 interface. You should not reload the router to accomplish this task.

Recall from Task 2, by default the LDP RID chosen by R1 was the highest loopback interface address, 11.1.1.1. This address was also used by LDP as the transport address in the LDP hellos it sent to R2. Since this address wasn't advertised into OSPF, due to missing reachability information, R1 and R2 did not form LDP adjacencies with each other. However, by using the mpls ldp router-id command, the LDP router ID was manually set to the loopback 0 address. This address is advertised into OSPF, resulting in R1 and R2 forming successful LDP sessions with each other.

This task requires resetting the LDP router ID to the 11.1.1.1 address on R1. The task also requires the change to take immediate effect without a restart on the routers. For this purpose, the command mpls ldp router-id lo1 force will be used on R1. The force keyword forces LDP to use the new router ID without requiring a router reload. Additionally, the 11.1.1.1 prefix is advertised into OSPF area 0 using the network command:

On R1:

```
R1(config)#router ospf 1
R1(config-router)#network 11.1.1.1 0.0.0.0 area 0

R1(config)#mpls ldp router-id loopback 1 force
```

You should see the following console message:

```
%LDP-5-NBRCHG: LDP Neighbor 2.2.2.2:0 (1) is DOWN (LDP Router ID
changed)
```

The show mpls ldp neighbor command on R1 verifies the above configuration:

```
R1#show mpls ldp neighbor

    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 11.1.1.1:0
     TCP connection: 2.2.2.2.646 - 11.1.1.1.22898
     State: Oper; Msgs sent/rcvd: 18/20; Downstream
     Up time: 00:00:57
     LDP discovery sources:
       GigabitEthernet0/2, Src IP addr: 12.1.1.2
        Addresses bound to peer LDP Ident:
          12.1.1.2        2.2.2.2         22.2.2.2        23.1.1.2
```

Task 6

The MPLS label space on a router is platform dependent. By default, the routers begin numbering the labels from 16 up to 100,000. Change the MPLS label space such that the routers use the following label ranges:

Router	Label range:
R1	100 - 199
R2	200 – 299
R3	300 – 399
R4	400 – 499
R5	500 – 599
R6	600 – 699
R7	700 – 799

MPLS LDP label values are in total 20 bits long. Cisco IOS reserves labels 0 through 15 for special purposes. This means the available label values to assign to prefixes range from 16 through 1,048,575. IOS however defaults to using label range 16 through 100,000. This can be seen with the show mpls label range command below from R1:

On R1:

```
R1#show mpls label range

Downstream Generic label region: Min/Max label: 16/100000
```

This task requires modifying the label range assigned by LDP on each router to that specified in the task.

This is done with the mpls label range *min max* command on all routers:

On R1:

```
R1(config)#mpls label range 100 199
```

On R2:

```
R2(config)#mpls label range 200 299
```

On R3:

```
R3(config)#mpls label range 300 399
```

On R4:

```
R4(config)#mpls label range 400 499
```

On R5:

```
R5(config)#mpls label range 500 599
```

On R6:

```
R6(config)#mpls label range 600 699
```

On R7:

```
R7(config)#mpls label range 700 799
```

The show mpls label range command is issued to verify the new range. However, notice the example output taken from R1 below. It states the the configured range will take effect after the next reload:

On R1:

```
R1#show mpls label range

Downstream Generic label region: Min/Max label: 16/199
[Configured range for next reload: Min/Max label: 100/199]
```

After issuing a write mem command on the routers to save the configuration, a reload is performed on all routers using the reload command.

```
Rx#write memory
Building configuration...
[OK]

Rx#reload
Proceed with reload? [confirm]
```

Once the routers reload, they start using the new range specified. The example output from R1 shows the new label range:

On R1:

```
R1#show mpls label range

Downstream Generic label region: Min/Max label: 100/199
```

Task 7

Examine and describe the control plane for the 7.7.7.7/32 prefix.

This task examines the MPLS LDP control plane functions. Before MPLS label switching can be performed in the data plane, routers must assign and exchange labels for destination IP prefixes that should be label-switched. To do so, LSRs perform the following functions:

1. Assign local labels to each prefix in their routing table. The combination of local label and IP prefix is called a local label Binding.
2. Advertise their local bindings via LDP to their LDP peers over the LDP TCP session.
3. Store received LDP bindings from their LDP peers. These are called remote bindings.

LSRs store all local and remote label bindings in their label information base or LIB. The LIB is analogous to the routing information base also known as the RIB. It is from the LIB that the router generates what is known as the label forwarding information base or LFIB. The LFIB is analogous to the forwarding information base or FIB used by CEF.

To demonstrate the exchange of this information, the task states the label bindings each router has for the

7.7.7.7 /32 prefix should be examined. This prefix is configured on the loopback 0 interface on R7 and is advertised into OSPF area 0. The following outputs verify the same:

On R7:

```
R7#show run interface lo0

interface Loopback0
 ip address 7.7.7.7 255.255.255.255
end

R7#show ip ospf interface brief

Interface      PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Lo0            1     0               7.7.7.7/32         1     LOOP  0/0
Gi0/6          1     0               67.1.1.7/24        1     DR    1/1
```

From the output above, R7 is configured for MPLS LDP and has an LDP adjacency with R6. Its local label binding for the 7.7.7.7/32 prefix can be seen with the show mpls ldp binding 7.7.7.7 32 :

```
R7#show mpls ldp bindings 7.7.7.7 32

  lib entry: 7.7.7.7/32, rev 2
     local binding:  label: imp-null
     remote binding: lsr: 6.6.6.6:0, label: 600
```

The output above is the label information entry for the 7.7.7.7/32 prefix. Since this prefix exists in R7's routing table, R7 will locally bind a label to this prefix. As seen above, R7 binds or assigns the imp-null (implicit null) label to the 7.7.7.7/32 prefix. The implicit null label is a label with label ID 3. When bound to a prefix, it indicates to upstream LSRs that the topmost label on the IP packet should be removed before forwarding.

Notice the output highlighted in blue. This is the remote binding (Label 600 for prefix 6.6.6.6) that R7 receives from R6. As mentioned earlier, every LSR will assign labels to prefixes in their routing. Because the prefix 7.7.7.7/32 appears in R6's routing table, R6 also assigns to it a local label and advertises it to R7. When an LSR receives a binding for a prefix from its downstream LSR, the upstream LSR checks if the downstream LSR is the next hop for that prefix. If the advertising LSR isn't the next hop, the LSR has two choices:

1. Keep track of such bindings

2. Discard such bindings

If the LSR decides to keep track of bindings for which the downstream LSR is not the next hop, the LSR is functioning in what is known as the Liberal label retention mode. If it decides to discard the binding, the LSR is functioning in a Conservative label retention mode. Seeing how R7 decides to keep a label for a prefix that exits on itself indicates that R7 is functioning in a Liberal label retention mode. It however will not have any use of this particular entry because the prefix 7.7.7.7 is configured on its own loopback interface. R7's behavior is the default behavior in Cisco IOS for LDP. By default, IOS operates LDP in liberal label retention mode.

R7 as a downstream LSR will advertise this label mapping to the upstream LSR R6. The show mpls ldp bindings 7.7.7.7 32 command output shows the details of this binding on R6:

On R6:

```
R6#show mpls ldp bindings 7.7.7.7 32

  lib entry: 7.7.7.7/32, rev 10
     local binding:  label: 600
     remote binding: lsr: 5.5.5.5:0, label: 502
     remote binding: lsr: 7.7.7.7:0, label: imp-null
```

The label information entry for the 7.7.7.7 prefix on R6 shows a single local binding and two remote bindings. R6 has locally assigned the label 600 the the 7.7.7.7 prefix. Note, this label value was picked from the label range (600 - 699) configured in the earlier task. R6 advertises this label-to-IP mapping to its neighboring LSRs R5 and R7. The earlier output from R7 confirms it has received the label 600 mapped to the 7.7.7.7 address from R6. The remote binding R6 receives from R7 is the imp-null label R7 advertises. It also receives the label 502 for the prefix from R5. This entry will not be used because R5 is not the next hop for the 7.7.7.7 prefix. It will exist in it's LIB because R6 is functioning in a liberal label retention mode.

The advertisement process described above continues down the line for every LSR in the topology. The following lists the output of the show mpls ldp binding 7.7.7.7 32 command along with an explanation of what each entry means.

The following shows the label binding for the 7.7.7.7 prefix on R5:

On R5:

```
R5#show mpls ldp bindings 7.7.7.7 32

  lib entry: 7.7.7.7/32, rev 12
     local binding:  label: 502
     remote binding: lsr: 6.6.6.6:0, label: 600
     remote binding: lsr: 4.4.4.4:0, label: 405
```

From the above:

1. R5 locally assigns the label 502 to 7.7.7.7
2. A remote binding with label 600 from R6
3. A remote binding with label 405 from R4

On R4:

```
R4#show mpls ldp bindings 7.7.7.7 32

  lib entry: 7.7.7.7/32, rev 21
     local binding:  label: 405
     remote binding: lsr: 5.5.5.5:0, label: 502
     remote binding: lsr: 3.3.3.3:0, label: 307
```

From the above:

1. R4 locally assigns the label 405 to 7.7.7.7
2. A remote binding with label 502 from R5
3. A remote binding with label 307 from R3

On R3:

```
R3#show mpls ldp bindings 7.7.7.7 32

  lib entry: 7.7.7.7/32, rev 24
     local binding:  label: 307
     remote binding: lsr: 4.4.4.4:0, label: 405
     remote binding: lsr: 2.2.2.2:0, label: 205
```

From the above:

1. R3 locally assigns the label 307 to 7.7.7.7
2. A remote binding with label 405 from R4
3. A remote binding with label 205 from R2

On R2:

```
R2#show mpls ldp bindings 7.7.7.7 32

  lib entry: 7.7.7.7/32, rev 21
     local binding:  label: 205
     remote binding: lsr: 11.1.1.1:0, label: 100
     remote binding: lsr: 3.3.3.3:0, label: 307
```

From the above:

1. R2 locally assigns the label 205 to 7.7.7.7
2. A remote binding with label 307 from R3
3. A remote binding with label 100 from R1

On R1:

```
R1#show mpls ldp bindings 7.7.7.7 32

  lib entry: 7.7.7.7/32, rev 8
     local binding:  label: 100
     remote binding: lsr: 2.2.2.2:0, label: 205
```

From the above:

1. R1 locally assigns the label 100 to 7.7.7.7
2. A remote binding with label 205 from R2

Task 8

Examine and describe the data plane for the 7.7.7.7/32 prefix, starting from R1.

The previous task demonstrated the MPLS LDP control plane function. It showed how each LSR advertised its local label bindings for the 7.7.7.7/32 prefix to its neighboring LSRs. This task examines how LSRs use these labels for packet forwarding.

MPLS label switching relies on the concept of a label stack. This label stack is simply the total list of MPLS labels that have been attached to a particular packet. LSRs make forwarding decisions based on the value of the topmost label on the label stack. They match the label ID of the topmost label to a matching entry in its label information base to determine how the packet should be forwarded. There are three main operations:

1. Push: the LSR adds one or more labels onto the label stack. If the packet does not have a label stack, this operation is called imposing a label stack onto the packet.
2. Pop: the LSR removes one or more labels from the label stack.
3. Swap: the LSR swaps the topmost label on the label stack with a new label.

MPLS dataplane forwarding has the following basic flow:

1. An LSR receives an unlabelled packet that should be label switched. It imposes a label stack on to the packet using the push operation based on the contents of its LIB and then forwards it out the designated outgoing label-switched interface.
2. The receiving LSR examines the topmost label on the label and performs a lookup in its own LIB for the label operation it should use. If it is not the final destination, it swaps the topmost label with the appropriate label and forwards the new labelled packet. This step repeats until the final destination LSR is reached.
3. The next receiving LSR receives the labeled packet and examines the topmost label. It determines that the packet should be forwarded unlabeled. It pops the label stack from the packet and forwards it as a native packet towards the end host.

In step 1, the router that first imposed the label stack on the packet is called the imposing or ingress LSR. In step 2, all routers that simply perform swap operation and aren't directly-connected to the destination are called intermediate LSRs. Finally, the router at step 3 that pops the label stack and forwards the packet natively towards the end host is called the disposing or egress LSR. The path from ingress LSR to egress LSR is called the label switched path or LSP.

The workflow above indicated that LSRs make their decisions based on the LIB. In truth, the contents of the LIB are downloaded into a format that makes it faster to perform label operations called the LFIB. The contents of the LFIB can be seen using the show mpls forwarding-table command. The task asks to examine the data path from R1 to the 7.7.7.7/32 prefix. As such, the show mpls forwarding-table 7.7.7.7 32 command is used to examine only the LFIB information for the 7.7.7.7/32 prefix.