Module 1: MPLS-Introduction

Part 1: The Conceptual Lecture (The Why)

The Problem MPLS Solves

Imagine you're managing a postal service. Every time a letter arrives at a sorting facility, workers must read the full address, look it up in a directory, and decide where to send it next. This happens at every facility along the route. Now imagine if instead, the first facility could analyze the address once, attach a simple colored sticker (red = "Express to New York", blue = "Standard to Chicago"), and every subsequent facility just looks at the sticker color. That's essentially what MPLS does for network packets.

Understanding the BGP Remote Next-Hop Problem

In traditional IP networks, routers make forwarding decisions hop-by-hop based on the destination IP address. Let's visualize this problem:

```
                AS 65001                           AS 65002
[Customer A]----[PE1]----[P1]----[P2]----[PE2]----[Customer B]
10.1.1.0/24              |        |                 10.2.2.0/24
                         |        |
                       [P3]------[P4]

BGP Advertisement Flow:
Customer B advertises 10.2.2.0/24 → PE2 → iBGP → PE1
PE1 learns: "To reach 10.2.2.0/24, next-hop is PE2 (let's say 192.168.1.2)"
```

Here's the problem: PE1 knows the BGP next-hop is PE2, but P1 and P2 don't run BGP! They only know about IGP routes. So we need to:

1. **Redistribute BGP routes into IGP** - This doesn't scale. Imagine redistributing millions of Internet routes into your IGP!
2. **Use recursive lookups** - PE1 must resolve PE2's loopback in IGP, then forward based on that. Every router does a full IP lookup at every hop.

How MPLS Solves This

MPLS creates a "label-switched path" (LSP) - think of it as a pre-computed tunnel:

```
Label Assignment:
[PE1]---(Label 100)--->[P1]---(Label 200)--->[P2]---(Label 300)--->[PE2]

Packet Flow with MPLS:
1. Customer A sends packet to 10.2.2.0/24
2. PE1: "I know this goes to PE2, push label 100"
3. P1: "Label 100? Swap to 200, send to P2" (no IP lookup!)
4. P2: "Label 200? Swap to 300, send to PE2"
5. PE2: "Label 300? Pop label, do IP lookup for 10.2.2.0/24"
```

Original Historical Motivations for MPLS (Late 1990s)

1. **Hardware Limitations**: In the 1990s, IP lookups required expensive TCAM/CAM memory and complex longest-prefix matching. Label lookups were simple exact matches - much faster in hardware.
2. **ATM Integration**: Many carriers had ATM networks. MPLS allowed IP traffic to traverse ATM switches efficiently.
3. **Traffic Engineering**: MPLS enabled explicit path control, something impossible with destination-based IP routing.

Modern MPLS Use Cases

Today, hardware can do IP lookups at line rate, but MPLS remains critical for:

1. **Layer 3 VPNs (L3VPN)**

```
Customer A Site 1 ----[PE1]
                        |
                     [MPLS Core]     VPN Red: Customer A
                        |            VPN Blue: Customer B
```

```
Customer A Site 2 ----[PE2]
Customer B Site 1 ----[PE3]
```

MPLS labels identify which VPN a packet belongs to, allowing overlapping IP addresses.

2. **Traffic Engineering (MPLS-TE)**

   ```
   Normal Path:     [R1]----[R2]----[R3]
                     |               |
   Engineered Path: +-----[R4]------+  (avoiding congested R2-R3 link)
   ```

3. **Fast Reroute (FRR)**
   - Pre-computed backup paths activate in <50ms upon failure
   - Much faster than traditional IGP convergence (seconds)
4. **Service Provider Simplification**
   - Core routers only need MPLS labels, not millions of customer routes
   - Enables massive scale

MPLS Label Structure

An MPLS label is a 32-bit value with this structure:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Label                  | TC  |S|       TTL     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Label: 20 bits (values 0-1,048,575)
TC: 3 bits (Traffic Class, formerly EXP)
S: 1 bit (Bottom of Stack)
TTL: 8 bits (Time to Live)
```

Labels can be stacked (S bit indicates bottom), enabling services within services.

Part 2: The Junos CLI Masterclass (The How)

Enabling MPLS on a Service Provider Network

Let's build a simple MPLS network step by step:

```
Topology:
[CE1]----[PE1]----[P1]----[P2]----[PE2]----[CE2]
      ge-0/0/0  ge-0/0/1  ge-0/0/1  ge-0/0/0
```

Step 1: Enable MPLS on Interfaces

First, we must tell Junos which interfaces will participate in MPLS:

```
[edit]
user@PE1# set interfaces ge-0/0/1 unit 0 family mpls

[edit]
user@PE1# set protocols mpls interface ge-0/0/1.0
```

Why both commands?

- `family mpls` : Enables the interface to process MPLS headers
- `protocols mpls interface` : Includes interface in MPLS protocol operations

Step 2: Configure IGP with Traffic Engineering Extensions

MPLS needs an IGP (OSPF or IS-IS) to learn topology. We'll use OSPF:

```
[edit]
user@PE1# set interfaces lo0 unit 0 family inet address 192.168.1.1/32

[edit protocols ospf]
user@PE1# set area 0.0.0.0 interface lo0.0 passive
user@PE1# set area 0.0.0.0 interface ge-0/0/1.0
user@PE1# set traffic-engineering
```

The `traffic-engineering` knob enables OSPF to advertise:

- Available bandwidth
- Administrative colors (for traffic engineering)
- MPLS capability

Step 3: Enable MPLS on All Routers

Complete configuration for PE1:

```
## PE1 Configuration
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.1/30
set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.1.1/32

set protocols mpls interface ge-0/0/1.0

set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf traffic-engineering
```

Repeat similar configuration on P1, P2, and PE2 (adjusting IP addresses).

Understanding MPLS Tables

Junos maintains three key routing tables for MPLS:

1. **inet.0**: IPv4 routing table
2. **inet.3**: MPLS routing table (LSP endpoints)
3. **mpls.0**: MPLS switching table (label operations)

The relationship:

```
BGP next-hop resolution:
1. BGP route in inet.0 has next-hop = PE2 loopback
2. Check inet.3 first: "Is there an LSP to PE2?"
3. If yes: Use MPLS path
4. If no: Fall back to inet.0 (regular IP routing)
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Verification Commands

1. Verify MPLS is Enabled

```
user@PE1> show mpls interface
Interface        State       Administrative groups (x: extended)
ge-0/0/1.0       Up          <none>
```

Good state: Interface shows "Up" **Bad state**: Missing from output = MPLS not configured

2. Verify OSPF with Traffic Engineering

```
user@PE1> show ospf database opaque-area | match "Opaque TE"
    Opaque TE LSA
    Opaque TE LSA
```

Good state: You see Opaque TE LSAs **Bad state**: No output = traffic-engineering not enabled

3. Check MPLS Forwarding Table

```
user@PE1> show route table mpls.0

mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w2d 11:13:38, metric 1
                      Receive
1                  *[MPLS/0] 1w2d 11:13:38, metric 1
                      Receive
2                  *[MPLS/0] 1w2d 11:13:38, metric 1
                      Receive
```

These are reserved labels:

- 0: IPv4 Explicit Null
- 1: Router Alert
- 2: IPv6 Explicit Null

Common Troubleshooting Scenarios

Scenario 1: MPLS Interface Not Coming Up

Symptom: `show mpls interface` shows no output

Diagnostic Commands:

```
user@PE1> show configuration interfaces ge-0/0/1 | display set
set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.1/30

user@PE1> show configuration protocols mpls | display set
```

Cause: Missing `family mpls` on interface

Solution:

```
[edit]
user@PE1# set interfaces ge-0/0/1 unit 0 family mpls
user@PE1# commit
```

Scenario 2: OSPF Not Advertising TE Information

Symptom: No Opaque LSAs in OSPF database

Diagnostic Commands:

```
user@PE1> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
10.0.1.2         ge-0/0/1.0             Full      192.168.1.2      128    36

user@PE1> show configuration protocols ospf | match traffic
```

Cause: Missing `traffic-engineering` statement

Solution:

```
[edit]
user@PE1# set protocols ospf traffic-engineering
user@PE1# commit
```

Scenario 3: Mismatched MPLS Configuration

Symptom: MPLS protocols fail to establish

Diagnostic Commands:

```
user@PE1> show interfaces ge-0/0/1.0 extensive | match "Input packets"
  Logical interface ge-0/0/1.0 (Index 333) (SNMP ifIndex 530)
    Input packets : 58291

user@PE1> show interfaces ge-0/0/1.0 extensive | match "MPLS"
    Protocol mpls, MTU: 1488
```

Cause: One side has MPLS enabled, other side doesn't

Solution: Ensure both sides of link have matching MPLS configuration:

```
## On both routers:
set interfaces ge-0/0/x unit 0 family mpls
set protocols mpls interface ge-0/0/x.0
```

Scenario 4: MTU Issues with MPLS

Symptom: Large packets dropped when MPLS is enabled

Diagnostic Commands:

```
user@PE1> show interfaces ge-0/0/1 | match MTU
  Link-level type: Ethernet, MTU: 1514, MRU: 1522, Speed: 1000mbps
```

Cause: MPLS adds at least 4 bytes per label. Default Ethernet MTU (1514) becomes insufficient.

Solution:

```
[edit]
user@PE1# set interfaces ge-0/0/1 mtu 1600
user@PE1# commit
```

Remember: MPLS is the foundation. In upcoming modules, we'll build LSPs using static configuration, RSVP-TE, and LDP on top of this MPLS-enabled infrastructure.

Module 2: MPLS-The Mechanics

Part 1: The Conceptual Lecture (The Why)

Understanding MPLS Label Operations

Think of MPLS like a highway system with on-ramps, lanes, and off-ramps. When a car (packet) enters the highway, it gets a toll tag (label). As it passes through toll stations (routers), the tag might be swapped for a different one, until finally it exits and the tag is removed.

MPLS has four fundamental operations:

1. **PUSH**: Add a label (entering MPLS domain)
2. **SWAP**: Replace one label with another (transit through MPLS)
3. **POP**: Remove a label (exiting MPLS domain)
4. **AGGREGATE**: Multiple operations (e.g., pop one label, examine next)

How Labels Are Built and Distributed

Labels are locally significant - like seat numbers in different airplanes. Seat 12A means something different on each flight. Similarly, label 100 means different things on different routers.

```
Label Distribution Flow:

[PE2] "I'm 192.168.1.4. To reach me, use label 300"
  ↓
[P2] "To reach 192.168.1.4, send me label 200"
```

```
  ↓
[P1] "To reach 192.168.1.4, send me label 100"
  ↓
[PE1] "OK, to reach 192.168.1.4, I'll push label 100"
```

The MPLS Data Plane - Packet Walk

Let's follow a packet step-by-step:

```
Topology:
[CE1]----[PE1]----[P1]----[P2]----[PE2]----[CE2]
        10.1.1.0/24                    10.2.2.0/24

Label Operations:
PE1: PUSH label 100
P1:  SWAP 100 → 200
P2:  SWAP 200 → 300
PE2: POP label
```

Detailed packet flow:

1. **CE1** → **PE1**: Regular IP packet

   ```
   [IP Header | Destination: 10.2.2.100 | Payload]
   ```

2. **PE1** → **P1**: PE1 pushes label

   ```
   [MPLS Label: 100 | IP Header | Destination: 10.2.2.100 | Payload]
   ```

3. **P1** → **P2**: P1 swaps label

   ```
   [MPLS Label: 200 | IP Header | Destination: 10.2.2.100 | Payload]
   ```

4. **P2** → **PE2**: P2 swaps label

   ```
   [MPLS Label: 300 | IP Header | Destination: 10.2.2.100 | Payload]
   ```

5. **PE2** → **CE2**: PE2 pops label, forwards based on IP

   ```
   [IP Header | Destination: 10.2.2.100 | Payload]
   ```

Label Stack and Service Hierarchy

MPLS supports label stacking - like nested envelopes:

```
Label Stack Example (L3VPN over MPLS-TE):

[Outer Label: Transport | Inner Label: VPN | IP Packet]
     ↑                         ↑
     Gets you to PE2           Identifies which VPN

Bottom of Stack (S-bit):
Label 1: S=0 (not bottom)
Label 2: S=1 (bottom of stack)
```

The Four Primary Label Distribution Protocols

1. **Static LSPs**
   - Manual configuration

    - Like programming each router individually
    - Use case: Testing, very small networks
2. **LDP (Label Distribution Protocol)**
    - Automatic label distribution
    - Follows IGP best path
    - Use case: Simple MPLS, foundation for L3VPN
3. **RSVP-TE (Resource Reservation Protocol - Traffic Engineering)**
    - Signaled paths with reservations
    - Can take non-shortest paths
    - Use case: Traffic engineering, bandwidth guarantees
4. **BGP (Border Gateway Protocol)**
    - Distributes labels with routes
    - Primarily for L3VPN
    - Use case: VPN services, inter-AS MPLS

MPLS Forwarding Equivalence Class (FEC)

An FEC is "what gets the same label" - like passengers going to the same destination getting the same colored boarding pass:

```
FEC Examples:
- All packets to 192.168.1.4/32 → Label 100
- All packets in VPN-A → Label 200
- All packets matching certain criteria → Label 300
```

The MPLS Label Information Base (LIB) vs Forwarding Information Base (FIB)

Think of these as two different views:

1. **LIB**: "All possible labels I know" (like a phone book)
2. **FIB**: "Labels I'm actually using" (like speed dial)

```
LIB (all learned labels):
Prefix 192.168.1.4/32:
  - From LDP neighbor P2: Use label 200
  - From RSVP LSP: Use label 500
  - From static config: Use label 800

FIB (what's installed):
192.168.1.4/32 → Push label 500 (RSVP wins due to preference)
```

Part 2: The Junos CLI Masterclass (The How)

Understanding Junos MPLS Tables Architecture

Junos uses multiple tables for MPLS operation:

```
Routing Tables for MPLS:
- inet.0:  IPv4 routes (regular routing)
- inet.3:  RSVP-learned MPLS paths
- mpls.0:  MPLS switching table (label → action)
- ldp.0:   LDP-learned labels (internal)
- bgp.l3vpn.0: L3VPN routes (BGP+MPLS)
```

Configuring Static LSPs - Complete Example

Let's build a static LSP from PE1 to PE2:

```
Topology with IPs:
[PE1]----[P1]----[P2]----[PE2]
.1  .2  .1  .2  .1  .2  .1
   10.0.1.0/30  10.0.2.0/30  10.0.3.0/30
```

```
Loopbacks:
PE1: 192.168.1.1/32
P1:  192.168.1.2/32
P2:  192.168.1.3/32
PE2: 192.168.1.4/32
```

Step 1: Configure PE1 (Ingress)

```
[edit protocols mpls]
user@PE1# set static-label-switched-path PE1-to-PE2 ingress
user@PE1# set static-label-switched-path PE1-to-PE2 to 192.168.1.4
user@PE1# set static-label-switched-path PE1-to-PE2 push 100
user@PE1# set static-label-switched-path PE1-to-PE2 next-hop 10.0.1.2

## This creates an entry in inet.3:
## 192.168.1.4/32 → Push 100, send to 10.0.1.2
```

Step 2: Configure P1 (Transit)

```
[edit protocols mpls]
user@P1# set static-label-switched-path PE1-to-PE2 transit
user@P1# set static-label-switched-path PE1-to-PE2 100 next-hop 10.0.2.2
user@P1# set static-label-switched-path PE1-to-PE2 100 swap 200

## This creates an entry in mpls.0:
## Label 100 → Swap to 200, send to 10.0.2.2
```

Step 3: Configure P2 (Transit)

```
[edit protocols mpls]
user@P2# set static-label-switched-path PE1-to-PE2 transit
user@P2# set static-label-switched-path PE1-to-PE2 200 next-hop 10.0.3.2
user@P2# set static-label-switched-path PE1-to-PE2 200 swap 300
```

Step 4: Configure PE2 (Egress)

```
[edit protocols mpls]
user@PE2# set interface lo0.0
user@PE2# set static-label-switched-path PE1-to-PE2 egress
user@PE2# set static-label-switched-path PE1-to-PE2 300 next-hop 192.168.1.4
user@PE2# set static-label-switched-path PE1-to-PE2 300 pop

## This tells PE2: "Label 300 is for me, pop and process"
```

Complete Annotated Configuration

```
## PE1 Configuration (Ingress)
protocols {
    mpls {
        ## Define interfaces participating in MPLS
        interface ge-0/0/1.0;

        ## Static LSP configuration
        static-label-switched-path PE1-to-PE2 {
            ## This is an ingress router
            ingress {
                ## Destination of the LSP
                to 192.168.1.4;
                ## Install route in inet.3 table
                install 192.168.1.4/32;
                ## Push label 100
                push 100;
                ## Next physical hop
                next-hop 10.0.1.2;
            }
        }
    }
}

## P1 Configuration (Transit)
protocols {
    mpls {
        interface ge-0/0/1.0;
        interface ge-0/0/2.0;

        static-label-switched-path PE1-to-PE2 {
            transit {
                ## When receiving label 100
                100 {
                    ## Swap to label 200
                    swap 200;
                    ## Send to next hop
                    next-hop 10.0.2.2;
                }
            }
        }
    }
}
```

Understanding Label Space and Platform Differences

Junos platforms have different label spaces:

```
## Check platform label space
user@router> show route table mpls.0 label-range

Platform Label Ranges:
- M/T Series: Labels 100000-1048575 (platform label space)
- MX Series:  Labels 16-1048575 (per-interface label space)
- PTX Series: Labels 16-1048575 (global label space)

Reserved Labels (0-15):
0:  IPv4 Explicit Null
1:  Router Alert
2:  IPv6 Explicit Null
3:  Implicit Null
4-15: Reserved for future use
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Verification Commands

1. Verify Static LSP Status

```
user@PE1> show mpls static-lsp
Ingress LSPs:
LSP Name         To              State   Next-Hop        Label Op
PE1-to-PE2       192.168.1.4     Up      10.0.1.2        Push 100

user@P1> show mpls static-lsp
Transit LSPs:
Incoming Label  Outgoing Label  Next-Hop        State
100             200             10.0.2.2        Up
```

2. Check MPLS Routing Tables

```
user@PE1> show route table inet.3

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.4/32     *[MPLS/6] 00:05:23, metric 1
```

```
                    > to 10.0.1.2 via ge-0/0/1.0, Push 100

user@P1> show route table mpls.0 label 100

mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

100                *[MPLS/6] 00:10:44, metric 1
                    > to 10.0.2.2 via ge-0/0/2.0, Swap 200
```

3. Verify Label Operations

```
user@PE1> show route forwarding-table destination 192.168.1.4
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop       Type Index    NhRef Netif
192.168.1.4/32     user     0 10.0.1.2      Push 100      584     2 ge-0/0/1.0

Routing table: __mpls-oam__.mpls
MPLS:
```

Common Troubleshooting Scenarios

Scenario 1: Static LSP Shows Down

Symptom: `show mpls static-lsp` shows "Down" state

Diagnostic Commands:

```
user@PE1> show mpls static-lsp extensive
Ingress LSPs:
LSP Name: PE1-to-PE2
  To: 192.168.1.4
  State: Down
  Next-Hop: 10.0.1.2
  Label Operation: Push 100
  Error: Next-hop interface down

user@PE1> show interfaces terse ge-0/0/1
Interface               Admin Link Proto    Local
ge-0/0/1                up    down
```

Cause: Next-hop interface is down

Solution:

```
## Investigate physical layer
user@PE1> show interfaces ge-0/0/1 extensive | match "link|fault"
```

Scenario 2: Label Not in MPLS Table

Symptom: Transit router doesn't have expected label in mpls.0

Diagnostic Commands:

```
user@P1> show route table mpls.0 label 100
  [no output]

user@P1> show configuration protocols mpls static-label-switched-path
```

Cause: Missing or incorrect transit configuration

Solution:

```
[edit protocols mpls]
user@P1# set static-label-switched-path PE1-to-PE2 transit 100 swap 200

```

```
user@P1# set static-label-switched-path PE1-to-PE2 transit 100 next-hop 10.0.2.2
user@P1# commit
```

Scenario 3: Traceroute Shows MPLS Label Stack

Symptom: Need to verify MPLS path

Diagnostic Commands:

```
user@PE1> traceroute 192.168.1.4 no-resolve
traceroute to 192.168.1.4 (192.168.1.4), 30 hops max, 40 byte packets
 1  10.0.1.2  0.523 ms  0.412 ms  0.398 ms
     MPLS Label=100 CoS=0 TTL=1 S=1
 2  10.0.2.2  0.890 ms  0.823 ms  0.812 ms
     MPLS Label=200 CoS=0 TTL=1 S=1
 3  10.0.3.2  1.234 ms  1.198 ms  1.187 ms
     MPLS Label=300 CoS=0 TTL=1 S=1
 4  192.168.1.4  1.456 ms  1.434 ms  1.423 ms
```

Analysis: Each hop shows the MPLS label being used

Scenario 4: BGP Routes Not Using MPLS Path

Symptom: BGP routes resolve via inet.0 instead of inet.3

Diagnostic Commands:

```
user@PE1> show route 10.2.2.0/24 detail

inet.0: 25 destinations, 25 routes (25 active, 0 holddown, 0 hidden)
10.2.2.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect, Index: 1048574
                Next hop: 10.0.1.2 via ge-0/0/1.0
                Protocol next hop: 192.168.1.4
                Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                State: <Active Int Ext>
                Inactive reason: Route Preference

user@PE1> show route 192.168.1.4 table inet.3
  [no route]
```

Cause: No MPLS path in inet.3, so BGP falls back to inet.0

Solution: Ensure MPLS path exists:

```
[edit protocols mpls static-label-switched-path PE1-to-PE2 ingress]
user@PE1# set install 192.168.1.4/32
user@PE1# commit
```

Advanced Troubleshooting: Label TTL Behavior

MPLS has three TTL modes:

```
## Configure TTL propagation (default: enabled)
[edit protocols mpls]
user@PE1# set no-propagate-ttl

Effects:
- propagate-ttl (default): IP TTL copied to MPLS TTL
- no-propagate-ttl: MPLS TTL starts at 255
- Useful for hiding core topology in traceroute
```

Module 3: MPLS-Static LSPs, and the Forwarding Plane

Part 1: The Conceptual Lecture (The Why)

Understanding the MPLS Forwarding Plane

Think of the MPLS forwarding plane like a subway system. The control plane (protocols like LDP, RSVP) is like the planning department that decides where tracks go and creates the subway map. The forwarding plane is the actual trains, tracks, and switches that move passengers (packets) from station to station.

```
Control Plane vs Forwarding Plane:

CONTROL PLANE (Brain)          FORWARDING PLANE (Muscle)
├─ Protocols (OSPF, BGP)       ├─ Routing Tables
├─ Label Distribution          ├─ Forwarding Tables
├─ Path Computation           ├─ ASIC/Hardware
└─ Signaling                  └─ Packet Processing

Control Plane says: "To reach 192.168.1.4, use label 100"
Forwarding Plane does: "Packet arrives → Push label 100 → Send it"
```

The Service Provider Edge (PE) vs Provider (P) Roles

In MPLS networks, routers have distinct roles:

1. **PE (Provider Edge) Routers**:
   - Interface between customer and MPLS
   - Maintain full routing tables (customer + core)
   - Perform label push/pop operations
   - Most complex configuration
2. **P (Provider) Routers**:
   - Core routers, only do label swapping
   - Don't need customer routes
   - Simple configuration
   - High-speed forwarding

```
Role Visualization:
[Customer A]---[PE1]---[P1]---[P2]---[PE2]---[Customer B]
                 ↑       ↑      ↑       ↑
                 |       |      |       |
            Push Label  Swap   Swap  Pop Label
            Complex    Simple  Simple  Complex
```

Static LSPs: The Foundation

Static LSPs are like manually programming each traffic light in a city. You explicitly tell each intersection (router) what to do with each color (label). While tedious for large networks, they're perfect for:

1. **Learning MPLS**: See exactly how labels work
2. **Testing**: Verify forwarding without protocol complexity
3. **Specific paths**: Force traffic through particular routers
4. **Backup paths**: Pre-configured failover routes

The inet.3 Table: MPLS's Special Routing Table

Junos uses a clever multi-table approach:

```
Route Resolution Process:
1. BGP route arrives: "10.2.2.0/24, next-hop 192.168.1.4"
2. Junos asks: "How do I reach 192.168.1.4?"
3. Check inet.3 first (MPLS paths)
   - Found? Use MPLS forwarding
   - Not found? Check inet.0 (regular IP)
```

```
Why separate tables?
- inet.0: Regular IP routes (always available)
- inet.3: MPLS routes (preferred when available)
- Automatic fallback if MPLS fails
```

Label Operations in Detail

Let's examine what happens at each router:

```
Ingress PE (Push Operation):
1. IP packet arrives: [IP Header | Data]
2. Lookup destination in FIB
3. FIB says: "Push label 100"
4. Result: [Label 100 | IP Header | Data]
5. Decrement IP TTL, copy to MPLS TTL

Transit P (Swap Operation):
1. MPLS packet arrives: [Label 100 | IP Header | Data]
2. Lookup label 100 in mpls.0
3. Table says: "Swap to 200"
4. Result: [Label 200 | IP Header | Data]
5. Decrement MPLS TTL

Egress PE (Pop Operation):
1. MPLS packet arrives: [Label 300 | IP Header | Data]
2. Lookup label 300 in mpls.0
3. Table says: "Pop"
4. Result: [IP Header | Data]
5. Copy MPLS TTL back to IP TTL
6. Do IP lookup and forward
```

The Penultimate Hop Popping (PHP) Optimization

PHP is like removing the gift wrap just before giving a present, so the recipient doesn't have to:

```
Without PHP:
[P2]---(Label 300)--->[PE2]
                       PE2 must:
                       1. Pop label 300
                       2. Do IP lookup
                       (Two lookups!)

With PHP:
[P2]---(No label)--->[PE2]
P2 pops label          PE2 just does IP lookup
                       (One lookup!)
```

Part 2: The Junos CLI Masterclass (The How)

Complete Service Provider MPLS Configuration

Let's build a full static LSP infrastructure:

```
Full Topology:
[CE1]----[PE1]----[P1]----[P2]----[PE2]----[CE2]
10.1.1.0/24  .1  .2  .1  .2  .1  .2  .1   10.2.2.0/24
         ge-0/0/0  ge-0/0/1  ge-0/0/1  ge-0/0/0

Loopbacks:
PE1: 192.168.1.1/32
P1:  192.168.1.2/32
```

```
P2:  192.168.1.3/32
PE2: 192.168.1.4/32
```

Step 1: Base Configuration for All Routers

First, configure interfaces and OSPF on all routers:

```
## PE1 Base Configuration
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.1.1/32

set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf traffic-engineering

set protocols mpls interface ge-0/0/1.0

## P1 Base Configuration
set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.2/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.0.2.1/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.1.2/32

set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf traffic-engineering

set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
```

Step 2: Configure Bidirectional Static LSPs

For full connectivity, we need LSPs in both directions:

```
## PE1 Configuration (PE1→PE2 and PE2→PE1)
[edit protocols mpls]
user@PE1# edit static-label-switched-path PE1-to-PE2
user@PE1# set ingress to 192.168.1.4
user@PE1# set ingress install 192.168.1.4/32
user@PE1# set ingress push 1000
user@PE1# set ingress next-hop 10.0.1.2

user@PE1# top
user@PE1# edit protocols mpls static-label-switched-path PE2-to-PE1
user@PE1# set egress 1300 pop
user@PE1# set egress 1300 next-hop 192.168.1.1

## P1 Configuration (Transit for both directions)
[edit protocols mpls]
user@P1# set static-label-switched-path PE1-to-PE2 transit 1000 swap 1100
user@P1# set static-label-switched-path PE1-to-PE2 transit 1000 next-hop 10.0.2.2

user@P1# set static-label-switched-path PE2-to-PE1 transit 1200 swap 1300
user@P1# set static-label-switched-path PE2-to-PE1 transit 1200 next-hop 10.0.1.1

## P2 Configuration (Transit for both directions)
[edit protocols mpls]
user@P2# set static-label-switched-path PE1-to-PE2 transit 1100 swap 1199
user@P2# set static-label-switched-path PE1-to-PE2 transit 1100 next-hop 10.0.3.2

user@P2# set static-label-switched-path PE2-to-PE1 transit 1299 swap 1200
user@P2# set static-label-switched-path PE2-to-PE1 transit 1299 next-hop 10.0.2.1

## PE2 Configuration (PE2→PE1 and PE1→PE2)
[edit protocols mpls]
```

```
user@PE2# set static-label-switched-path PE1-to-PE2 egress 1199 pop
user@PE2# set static-label-switched-path PE1-to-PE2 egress 1199 next-hop 192.168.1.4

user@PE2# set static-label-switched-path PE2-to-PE1 ingress to 192.168.1.1
user@PE2# set static-label-switched-path PE2-to-PE1 ingress install 192.168.1.1/32
user@PE2# set static-label-switched-path PE2-to-PE1 ingress push 1299
user@PE2# set static-label-switched-path PE2-to-PE1 ingress next-hop 10.0.3.1
```

Step 3: Configure Penultimate Hop Popping

To implement PHP, use implicit-null (label 3):

```
## P2 Configuration for PHP
[edit protocols mpls]
user@P2# delete static-label-switched-path PE1-to-PE2 transit 1100 swap 1199
user@P2# set static-label-switched-path PE1-to-PE2 transit 1100 pop
user@P2# set static-label-switched-path PE1-to-PE2 transit 1100 next-hop 10.0.3.2
```

Advanced Static LSP Features

Using Named Paths

```
## Define a path object
[edit protocols mpls]
user@PE1# set path PE1-P1-P2-PE2 10.0.1.2 strict
user@PE1# set path PE1-P1-P2-PE2 10.0.2.2 strict
user@PE1# set path PE1-P1-P2-PE2 10.0.3.2 strict

## Use it in static LSP
user@PE1# set static-label-switched-path PE1-to-PE2 ingress path PE1-P1-P2-PE2
```

Bandwidth Reservation with Static LSPs

```
## Reserve bandwidth on interfaces
[edit protocols mpls]
user@PE1# set interface ge-0/0/1.0 bandwidth 1g
user@PE1# set static-label-switched-path PE1-to-PE2 ingress bandwidth 100m
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Verification Commands

1. Verify Complete LSP Path

```
user@PE1> show mpls lsp static detail
Ingress LSP: 1 sessions
192.168.1.4/32
  From: 192.168.1.1, To: 192.168.1.4
  State: Up
  ActivePath: (primary)
  LSPtype: Static
  LoadBalance: Random
  Label out: 1000
  Next hop: 10.0.1.2
  Installed: 00:15:42 ago

user@PE1> show route table inet.3 192.168.1.4 detail

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
192.168.1.4/32 (1 entry, 1 announced)
        State: <FlashAll>
        *MPLS   Preference: 6
                Next hop type: Router, Next hop index: 590
                Address: 0x94b8c3c
                Next-hop reference count: 3
                Next hop: 10.0.1.2 via ge-0/0/1.0, selected
                Label operation: Push 1000
                Label TTL action: prop-ttl
                Load balance label: Label 1000: None;
                Label element ptr: 0x94b8d88
                Label parent element ptr: 0x0
                Label element references: 1
                Label element child references: 0
                Label element lsp id: 0
                Session Id: 0x141
                State: <Active Int>
                Age: 15:42
                Validation State: unverified
                Task: MPLS
                Announcement bits (2): 0-KRT 1-Resolve tree 1
                AS path: I
```

2. Trace Label Path

```
user@PE1> traceroute mpls static-lsp PE1-to-PE2
  Probe options: ttl 64, retries 3, wait 10, paths 16, exp 7, fanout 16

  ttl    Label  Protocol    Address          Previous Hop     Probe Status
    1     1000  Static      10.0.1.2         (null)           Success
  FEC-Stack-Sent: MPLS
  ttl    Label  Protocol    Address          Previous Hop     Probe Status
    2     1100  Static      10.0.2.2         10.0.1.2         Success
  FEC-Stack-Sent: MPLS
  ttl    Label  Protocol    Address          Previous Hop     Probe Status
    3        3  Static      10.0.3.2         10.0.2.2         Success
  FEC-Stack-Sent: MPLS
  ttl    Label  Protocol    Address          Previous Hop     Probe Status
    4           Static      192.168.1.4      10.0.3.2         Egress

  Path 1 via ge-0/0/1.0 destination 127.0.0.64
```

Common Troubleshooting Scenarios

Scenario 1: LSP Not Installing in inet.3

Symptom: Static LSP configured but not appearing in inet.3

Diagnostic Commands:

```
user@PE1> show mpls static-lsp extensive
Ingress LSP: 1 sessions
192.168.1.4/32
  From: 192.168.1.1, To: 192.168.1.4
  State: Dn
  Reason: No route toward next-hop

user@PE1> show route 10.0.1.2
[no output]
```

Cause: Next-hop not reachable

Solution:

```
## Verify OSPF is running
user@PE1> show ospf neighbor
user@PE1> show configuration protocols ospf
```

Scenario 2: Wrong Label in mpls.0

Symptom: Transit router has label but wrong action

Diagnostic Commands:

```
user@P1> show route table mpls.0 label 1000
```

```
mpls.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1000               *[MPLS/6] 00:05:00, metric 1
                    > to 10.0.2.2 via ge-0/0/2.0, Pop

## Should be Swap, not Pop!
```

Cause: Incorrect transit configuration

Solution:

```
[edit protocols mpls static-label-switched-path PE1-to-PE2]
user@P1# delete transit 1000 pop
user@P1# set transit 1000 swap 1100
user@P1# commit
```

Scenario 3: Forwarding Loop

Symptom: Traceroute shows same hop repeatedly

Diagnostic Commands:

```
user@PE1> traceroute mpls static-lsp PE1-to-PE2
  ttl    Label  Protocol    Address          Previous Hop     Probe Status
    1     1000  Static      10.0.1.2         (null)           Success
    2     1100  Static      10.0.1.1         10.0.1.2         Success
    3     1000  Static      10.0.1.2         10.0.1.1         Success
    4     1100  Static      10.0.1.1         10.0.1.2         Success
  [Loop detected]
```

Cause: Labels pointing back to previous hop

Solution: Verify each router's label mapping:

```
## On each router, verify label operations point forward
user@P1> show configuration protocols mpls static-label-switched-path
## Ensure next-hop points toward destination, not back
```

Scenario 4: BGP Not Using Static LSP

Symptom: BGP routes not using MPLS path

Diagnostic Commands:

```
user@PE1> show route 10.2.2.0/24 extensive
    Protocol next hop: 192.168.1.4
    Indirect next hop: 0xb3a1404 1048574 INH Session ID: 0x1ac
    State: <Secondary Active Int Ext>
    Local AS: 65001 Peer AS: 65001
    Age: 2:14:38
    Validation State: unverified
    Task: BGP_65001.192.168.1.4
    Announcement bits (3): 0-KRT 6-BGP_RT_Background 7-Resolve tree 4
    AS path: I
    Accepted
    Localpref: 100
    Router ID: 192.168.1.4
    Indirect next hops: 1
      Protocol next hop: 192.168.1.4
      Indirect next hop: 0xb3a1404 1048574 INH Session ID: 0x1ac
      State: <Active Int>
      Local AS: 65001 Peer AS: 65001
      Age: 2:14:38
      Validation State: unverified
      Task: BGP_65001.192.168.1.4
      Announcement bits (3): 0-KRT 6-BGP_RT_Background 7-Resolve tree 4
      AS path: I
```

```
        Accepted
        Localpref: 100
        Router ID: 192.168.1.4
        Indirect next hops: 1
          Protocol next hop: 192.168.1.4 Metric: 4
          Indirect next hop: 0x951e3f0 1048575 INH Session ID: 0x1a9
          State: <Active Int>
          Forwarding nexthops: 1
            Next hop type: Router
            Next hop: 10.0.1.2 via ge-0/0/1.0
            Session Id: 0x1a5
```

Look for: The key indicator is whether forwarding shows label operations

Cause: Static LSP not installed in inet.3

Solution:

```
[edit protocols mpls static-label-switched-path PE1-to-PE2 ingress]
user@PE1# show
to 192.168.1.4;
push 1000;
next-hop 10.0.1.2;

## Missing install statement!
user@PE1# set install 192.168.1.4/32
user@PE1# commit
```

Advanced Debugging: MPLS Ping

```
## Test MPLS data plane
user@PE1> ping mpls static-lsp PE1-to-PE2
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.582/0.621/0.683/0.037 ms

## Detailed MPLS ping
user@PE1> ping mpls static-lsp PE1-to-PE2 detail
Request for seq 1, to interface 67, labels <1000>
  Reply for seq 1, return code: Egress-ok, time: 0.643 ms

## If ping fails, shows where:
Request for seq 1, to interface 67, labels <1000>
  Reply for seq 1, return code: Label-switched-at-stack-depth 1, time: 0.234 ms
  Sender timestamp: 2024-03-15 10:45:23
  Receiver timestamp: 2024-03-15 10:45:23
  Response from 10.0.2.1: label 1100 not found in mpls.0
```

This detailed output helps identify exactly where in the label path things break down.

Module 4: RSVP-Introduction

Part 1: The Conceptual Lecture (The Why)

What is RSVP and Why Do We Need It?

Imagine planning a road trip. With static LSPs (previous module), it's like manually plotting every turn on every road. RSVP is like having a GPS that automatically finds the best route, reserves your hotel rooms along the way, and even handles detours if roads are blocked.

RSVP (Resource Reservation Protocol) was originally designed in the 1990s for multimedia applications to reserve bandwidth. RSVP-TE (Traffic Engineering) extends this for MPLS, creating label-switched paths automatically.

RSVP vs Static LSPs

```
Static LSPs:                      RSVP:
- Manual label assignment         - Automatic label assignment
- No bandwidth awareness         - Can reserve bandwidth
- No automatic rerouting         - Automatic failure recovery
- Configure every router         - Configure only endpoints
- No path computation           - Intelligent path selection

Analogy:
Static = Programming each traffic light manually
RSVP = Smart traffic system that adapts to conditions
```

How RSVP-TE Works: The Signaling Process

RSVP uses a two-phase signaling process, like making a restaurant reservation:

```
RSVP Signaling Flow:

1. PATH Message (Downstream)
   [PE1] "I want to reach PE2" ────PATH────> [P1] ────PATH────> [P2] ────PATH────> [PE2]

2. RESV Message (Upstream)
   [PE1] <────RESV──── [P1] <────RESV──── [P2] <────RESV──── [PE2] "OK, use label 300"
         "Use label 100"    "Use label 200"     "Use label 300"

Think of it as:
- PATH: "Can I book a route to the destination?"
- RESV: "Yes, here's your confirmation number (label)"
```

RSVP Objects: The Building Blocks

RSVP messages contain "objects" - think of them as form fields:

```
Key RSVP Objects:

SESSION Object:
- Destination address
- Tunnel ID
- Extended tunnel ID
"Where do you want to go?"

SENDER_TEMPLATE Object:
- Source address
- LSP ID
"Who's making this reservation?"

LABEL_REQUEST Object:
- Label type requested
"I need an MPLS label"

EXPLICIT_ROUTE Object (ERO):
- Specific path to follow
"Take this exact route"

RECORD_ROUTE Object (RRO):
- Path actually taken
"Here's where you went"
```

RSVP Features and Advantages

1. **Dynamic Signaling**: No manual label configuration
2. **Bandwidth Reservation**: Can guarantee bandwidth along path

3. **Explicit Routing**: Can specify exact path or constraints
4. **Fast Reroute**: Pre-computed backup paths activate in <50ms
5. **Make-Before-Break**: Changes paths without dropping traffic
6. **Path Protection**: Multiple redundant paths

RSVP Soft State vs Hard State

Originally, RSVP was "soft state" - like a gym membership that expires unless renewed:

```
Original Soft State:
- Send PATH/RESV every 30 seconds
- If 3 missed → tear down LSP
- High overhead, not scalable

Modern Reliable RSVP:
- Initial PATH/RESV exchange
- Periodic lightweight hellos
- Explicit teardown when needed
- Much more efficient
```

RSVP Message Types

RSVP uses several message types:

```
1. Path Message:
   - Travels downstream (ingress → egress)
   - Carries session info, label request
   - Records route taken

2. Resv Message:
   - Travels upstream (egress → ingress)
   - Carries label assignments
   - Confirms reservations

3. PathTear:
   - Removes path state
   - "Cancel my reservation"

4. ResvTear:
   - Removes reservation state
   - "Reservation cancelled"

5. PathErr:
   - Reports path problems
   - "Can't find route"

6. ResvErr:
   - Reports reservation problems
   - "No bandwidth available"

7. Hello:
   - Keepalive mechanism
   - "Are you still there?"
```

The RSVP-TE Database

For RSVP to make intelligent decisions, it needs a "map" of the network:

```
Traffic Engineering Database (TED):
- Link bandwidths
- Available bandwidth
- Administrative colors
- TE metrics
- Link attributes

Built from:
- OSPF Opaque LSAs (Type 10)
- IS-IS TLVs (22, 134, 135)

It's like having real-time traffic data for route planning
```

Part 2: The Junos CLI Masterclass (The How)

Preparing a Network for RSVP

Before configuring RSVP LSPs, the network needs proper foundation:

```
Prerequisites Checklist:
□ MPLS enabled on interfaces
□ IGP running with TE extensions
□ RSVP protocol enabled
□ Proper interface bandwidth configured
```

Step 1: Enable RSVP Protocol

```
## Basic RSVP enablement
[edit protocols rsvp]
user@PE1# set interface ge-0/0/1.0
user@PE1# set interface lo0.0

## What this does:
## - Enables RSVP on specified interfaces
## - lo0.0 is crucial - it's the tunnel endpoint
```

Step 2: Configure RSVP Parameters

```
## RSVP timers and behavior
[edit protocols rsvp]
user@PE1# set refresh-time 30
user@PE1# set keep-multiplier 3
user@PE1# set graceful-restart

## Explanation:
## refresh-time: How often to refresh soft state (default: 45s)
## keep-multiplier: Missed refreshes before timeout (default: 3)
## graceful-restart: Maintain LSPs during control plane restart
```

Step 3: Interface-Specific RSVP Configuration

```
## Configure per-interface parameters
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set bandwidth 10g 8g
user@PE1# set subscription 90

## Parameters explained:
## bandwidth: physical-bandwidth reservable-bandwidth
## subscription: Allow 90% overbooking (default: 100%)

## Advanced interface options
user@PE1# set authentication-key "$9$KvLNbwg4ZjkmfQn"
user@PE1# set hello-interval 9
user@PE1# set aggregate
```

Complete RSVP-Ready Network Configuration

Let's configure our entire topology for RSVP:

```
## PE1 Complete Configuration
set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.1.1/32

set protocols mpls interface ge-0/0/1.0
set protocols mpls interface lo0.0

set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0

set protocols rsvp interface ge-0/0/1.0 bandwidth 10g
set protocols rsvp interface lo0.0

## P1 Configuration (Core Router)
set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.2/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.0.2.1/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.1.2/32

set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface lo0.0

set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0

set protocols rsvp interface ge-0/0/1.0 bandwidth 10g
set protocols rsvp interface ge-0/0/2.0 bandwidth 10g
set protocols rsvp interface lo0.0
```

Advanced RSVP Configuration Options

Configuring RSVP Authentication

```
## MD5 authentication for security
[edit protocols rsvp]
user@PE1# set interface ge-0/0/1.0 authentication-key secret123

## Or use keychain for key rotation
[edit security]
user@PE1# set authentication-key-chains key-chain RSVP-KEYS key 1 secret "$9$xyzabc"
user@PE1# set authentication-key-chains key-chain RSVP-KEYS key 1 start-time "2024-01-01.00:00:00 +0000"

[edit protocols rsvp]
user@PE1# set interface ge-0/0/1.0 authentication-key-chain RSVP-KEYS
```

Configuring RSVP Refresh Reduction

```
## Modern RSVP optimization
[edit protocols rsvp]
user@PE1# set refresh-reduction
user@PE1# set refresh-reduction reliable-delivery

## Benefits:
## - Reduces control plane overhead
## - Uses message IDs instead of full refresh
## - Explicit acknowledgments
```

RSVP Graceful Restart

```
## Maintain LSPs during restart
[edit protocols rsvp]
user@PE1# set graceful-restart
```

```
user@PE1# set graceful-restart restart-time 120
user@PE1# set graceful-restart recovery-time 180

## restart-time: How long neighbors should wait
## recovery-time: Maximum time to recover state
```

RSVP Interface Bandwidth Management

Understanding bandwidth configuration is crucial:

```
## Method 1: Static bandwidth
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set bandwidth 10g

## Method 2: Percentage of interface speed
user@PE1# set bandwidth percent 80

## Method 3: Different physical vs reservable
user@PE1# set bandwidth 10g 8g
## First value: Physical bandwidth (for TE calculations)
## Second value: Reservable bandwidth (for actual reservations)

## Method 4: Subscription (overbooking)
user@PE1# set subscription 150
## Allow 150% of bandwidth to be reserved
```

Bandwidth Model Example

```
Interface: 10G physical
RSVP bandwidth: 10g 8g
Subscription: 125%

Means:
- TE database shows: 10G link
- Can reserve up to: 8G
- With subscription: 8G × 1.25 = 10G reservable
- Allows statistical multiplexing
```

Part 3: Verification & Troubleshooting (The What-If)

Essential RSVP Verification Commands

1. Verify RSVP Protocol Status

```
user@PE1> show rsvp version
Resource ReSerVation Protocol, version 1

user@PE1> show rsvp interface
RSVP interface: 3 active
                  Active Subscr- Static      Available   Reserved    Highwater
Interface   State resv   iption  BW          BW          BW          mark
ge-0/0/1.0  Up         1   100%  10Gbps      10Gbps      0bps        0bps
lo0.0       Up         0   100%  0bps        0bps        0bps        0bps
```

2. Check RSVP Neighbors

```
user@PE1> show rsvp neighbor
RSVP neighbor: 1 learned
Address            Idle Up/Dn LastChange HelloInt HelloTx/Rx MsgRcvd
10.0.1.2           11   1/0   00:11:47   9        84/84      25

user@PE1> show rsvp neighbor detail
Address: 10.0.1.2
  Interface: ge-0/0/1.0
  Link protection: Not configured
  Bypass state: n/a
  Hello state: Up
  Hello interval: 9
  Hello tolerance: 27
  Remote instance: 0x9f7b5a2e
  Remote node id: 192.168.1.2
  Local instance: 0xa1234567
  Refresh reduction: Active
    Remote end: Active, Ack-mode: Independent
    Time since last Bundle: Never
    Bundle timeout: 45000 ms
```

3. Verify Traffic Engineering Database

```
user@PE1> show ted database
TED database: 4 ISIS nodes 4 INET nodes
ID                            Type Age(s) LnkIn LnkOut Protocol
192.168.1.1                   Rtr    1234     1      1 OSPF(0.0.0.0)
  To: 192.168.1.2, Local: 10.0.1.1, Remote: 10.0.1.2
    Local interface index: 333, Remote interface index: 0
    Link name: ge-0/0/1.0
    Metric: 1, Bandwidth: 10Gbps
    Static BW: 10Gbps, Reservable BW: 10Gbps
    Available BW [priority] bps:
      [0] 10Gbps     [1] 10Gbps     [2] 10Gbps     [3] 10Gbps
      [4] 10Gbps     [5] 10Gbps     [6] 10Gbps     [7] 10Gbps
    Admin-groups: none
```

Common Troubleshooting Scenarios

Scenario 1: RSVP Interface Not Up

Symptom: `show rsvp interface` shows interface "Down"

Diagnostic Commands:

```
user@PE1> show rsvp interface extensive
ge-0/0/1.0
  Index: 72, State: Dn
  No MPLS
  HelloTx/Rx: 0/0

user@PE1> show mpls interface
[no output]
```

Cause: MPLS not enabled on interface

Solution:

```
[edit]
user@PE1# set protocols mpls interface ge-0/0/1.0
user@PE1# commit
```

Scenario 2: RSVP Neighbor Not Forming

Symptom: No RSVP neighbors despite configuration

Diagnostic Commands:

```
user@PE1> show rsvp neighbor
RSVP neighbor: 0 learned

user@PE1> monitor traffic interface ge-0/0/1.0 matching "proto 46"
[no RSVP packets seen]

user@PE1> show interfaces ge-0/0/1.0 | match filter
```

Cause: Firewall filter blocking RSVP (protocol 46)

Solution:

```
## Check for filters
[edit interfaces ge-0/0/1 unit 0 family inet]
user@PE1# show
filter {
    input EDGE-FILTER;
}

## Update filter to allow RSVP
[edit firewall family inet filter EDGE-FILTER]
user@PE1# set term ALLOW-RSVP from protocol rsvp
user@PE1# set term ALLOW-RSVP then accept
user@PE1# insert term ALLOW-RSVP before term DENY-ALL
```

Scenario 3: TE Database Not Populating

Symptom: TED empty or incomplete

Diagnostic Commands:

```
user@PE1> show ted database
TED database: 0 ISIS nodes 1 INET nodes
ID                            Type Age(s) LnkIn LnkOut Protocol
192.168.1.1                   Rtr     10     0      0 OSPF(0.0.0.0)

user@PE1> show ospf database opaque-area advertising-router 192.168.1.1
```

Cause: OSPF TE not enabled

Solution:

```
[edit protocols ospf]
user@PE1# set traffic-engineering
user@PE1# commit

## Verify TE LSAs being generated
user@PE1> show ospf database opaque-area | match "Type-10"
Type-10 (Area-Local Opaque) LSAs
```

Scenario 4: RSVP Authentication Mismatch

Symptom: Neighbors flapping, authentication errors

Diagnostic Commands:

```
user@PE1> show rsvp neighbor detail
Address: 10.0.1.2
  Interface: ge-0/0/1.0
  Hello state: Dn
  Last hello error: Authentication failure

user@PE1> show log messages | match RSVP | match auth
Mar 15 10:45:23  PE1 rpd[1234]: RSVP: Authentication failed from 10.0.1.2
```

Cause: Authentication key mismatch

Solution:

```
## Verify keys match on both sides
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# show
authentication-key "$9$KvLNbwg4ZjkmfQn"; ## SECRET-DATA

## On neighbor router
user@P1# set protocols rsvp interface ge-0/0/1.0 authentication-key secret123
```

Advanced RSVP Debugging

Enable RSVP Traceoptions

```
[edit protocols rsvp]
user@PE1# set traceoptions file rsvp-debug
user@PE1# set traceoptions file size 10m
user@PE1# set traceoptions file files 5
user@PE1# set traceoptions flag packets detail
user@PE1# set traceoptions flag path detail
user@PE1# set traceoptions flag resv detail
user@PE1# set traceoptions flag state detail

## View the trace
user@PE1> show log rsvp-debug | match "PATH|RESV"
Mar 15 10:50:01 RSVP send PATH 10.0.1.1->10.0.1.2
Mar 15 10:50:01   Session: 192.168.1.4/32, Tunnel-id: 1, Ext-id: 192.168.1.1
Mar 15 10:50:02 RSVP recv RESV 10.0.1.2->10.0.1.1
Mar 15 10:50:02   Label: 100567
```

Monitor RSVP Messages in Real-time

```
user@PE1> monitor traffic interface ge-0/0/1.0 matching "proto 46" extensive
10:52:33.123456 Out
  IP: 10.0.1.1 > 10.0.1.2: RSVP
    Path Message
    Session: IPv4-LSP, Destination 192.168.1.4, Tunnel ID 0x1, Ext ID 192.168.1.1
    Sender Template: IPv4-LSP, Sender 192.168.1.1, LSP ID 0x1
    Label Request: Generic Label
```

RSVP Statistics and Counters

```
user@PE1> show rsvp statistics
RSVP packet statistics:
                        Sent       Received
  Path                   245          189
  PathErr                  0            0
  PathTear                 3            2
  Resv                   189          245
  ResvErr                  0            0
  ResvTear                 2            3
  Hello                 8456         8455
  Ack                    234          234
  SRefresh               456          455
  EndtoEnd SRefresh        0            0
  Bundle                  12           11
  Errors                   0            0

user@PE1> clear rsvp statistics
```

This foundation prepares the network for RSVP LSP creation, which we'll cover in the next module.

Module 5: RSVP-Configuring A Basic LSP

Part 1: The Conceptual Lecture (The Why)

Understanding RSVP LSP Creation

Now that we have RSVP enabled (like having a phone system installed), let's make our first call - create an LSP. An RSVP LSP is like establishing a dedicated highway lane from one city to another, with automatic toll collection (labels) set up at each interchange.

The RSVP LSP Signaling Process in Detail

When you configure an RSVP LSP, here's what happens behind the scenes:

```
Step-by-Step LSP Creation:

1. Configuration Trigger
   PE1: "I need an LSP to PE2"

2. Path Message Creation
   PE1 creates PATH message:
   - Session: Destination=192.168.1.4, Tunnel-ID=1
   - Sender: Source=192.168.1.1, LSP-ID=1
   - ERO: Empty (follow IGP best path)
   - Label Request: "I need labels"

3. Path Message Journey
   PE1 --PATH--> P1 --PATH--> P2 --PATH--> PE2
   Each hop:
   - Records itself in RRO (Record Route Object)
   - Checks resources
   - Forwards downstream

4. Resv Message Return
   PE2 <--RESV-- P2 <--RESV-- P1 <--RESV-- PE1
   Each hop:
   - Allocates label
   - Programs forwarding
   - Sends label upstream

5. LSP Established
   Forwarding plane ready!
```

Following the IGP Metric Best Path

By default, RSVP LSPs follow the IGP shortest path. Think of it like Google Maps defaulting to the fastest route:

```
IGP Metrics Example:
                    Cost: 10
        PE1 -------------------- P1
         |                        |
         |                        | Cost: 10
         | Cost: 100              |
         |                        |
        P3 -------------------- P2 ---- PE2
                  Cost: 10         Cost: 10

IGP Best Path: PE1->P1->P2->PE2 (Total: 30)
Alternative: PE1->P3->P2->PE2 (Total: 120)

RSVP Default: Follow IGP (PE1->P1->P2->PE2)
```

MPLS Self-Ping: The Built-in Health Check

MPLS self-ping is like a smoke detector - it automatically verifies the LSP works end-to-end after creation:

```
Self-Ping Process:
1. LSP signals successfully
2. Ingress waits briefly (let forwarding converge)
3. Ingress sends MPLS ping to itself via the LSP
4. If successful: LSP marked "Up"
5. If failed: LSP marked "Down", may retry

Benefits:
- Automatic verification
- Catches forwarding plane issues
- No manual testing needed
- Fast problem detection
```

The Junos inet.3 Table Integration

When an RSVP LSP is established, Junos automatically installs it in inet.3:

```
Route Resolution Process:
1. BGP: "Route 10.2.2.0/24, next-hop 192.168.1.4"
2. Junos checks inet.3 first
3. Finds: "192.168.1.4 via RSVP LSP"
4. Result: BGP traffic uses MPLS

Table Preference:
inet.3 (RSVP) - Preference: 7
inet.0 (OSPF) - Preference: 10
Lower preference wins!
```

RSVP LSP States and Transitions

An LSP goes through several states:

```
LSP State Machine:

    Config -----> Down -----> Establishing -----> Up
      |             ^               |              |
      |             |               v              |
      +-------------+------------ Failed <---------+
                                    |
                                    v
                                 Retry

States Explained:
- Down: Initial state or hard failure
- Establishing: Signaling in progress
- Up: Active and forwarding
- Failed: Soft failure, will retry
- Retry: Attempting to re-establish
```

Label Allocation and Distribution

Understanding how labels are allocated:

```
Label Allocation Direction: Downstream to Upstream

PE2 (Egress):
- Receives PATH
- Allocates label from pool
- Sends RESV with label

P2 (Transit):
- Receives RESV from PE2
- Creates mapping: "If I receive label X, send to PE2 with label Y"
- Allocates new label X
- Sends RESV to P1 with label X

Result: Label chain created backward from destination
```

Part 2: The Junos CLI Masterclass (The How)

Basic RSVP LSP Configuration

Let's create our first RSVP LSP:

```
## Step 1: Define the LSP on PE1 (Ingress)
[edit protocols mpls]
user@PE1# set label-switched-path PE1-to-PE2 to 192.168.1.4

## That's it! Minimal config creates a basic LSP
## Let's see what this simple command does...
```

Understanding Default LSP Behavior

With just that one command, Junos:

- Creates an LSP named "PE1-to-PE2"
- Destination: 192.168.1.4 (PE2's loopback)
- Follows IGP best path
- No bandwidth reservation
- Default priority (7,7)
- Installs in inet.3

Complete LSP Configuration with Options

```
## Comprehensive LSP configuration
[edit protocols mpls]
user@PE1# set label-switched-path PE1-to-PE2 to 192.168.1.4
user@PE1# set label-switched-path PE1-to-PE2 metric 5
user@PE1# set label-switched-path PE1-to-PE2 no-cspf
user@PE1# set label-switched-path PE1-to-PE2 adaptive
user@PE1# set label-switched-path PE1-to-PE2 description "Primary LSP to PE2"

## Configuration explanation:
## to: Destination router ID (typically loopback)
## metric: IGP metric for routes using this LSP
## no-cspf: Disable CSPF, use IGP path only
## adaptive: Allow LSP to find better paths
## description: Documentation
```

Configuring Bidirectional LSPs

For full connectivity, configure reverse path on PE2:

```
## On PE2 (Configure return LSP)
[edit protocols mpls]
user@PE2# set label-switched-path PE2-to-PE1 to 192.168.1.1

## Important: LSP names are locally significant
## PE1's "PE1-to-PE2" and PE2's "PE2-to-PE1" are independent
```

Advanced LSP Options

```
## Configure LSP with specific requirements
[edit protocols mpls label-switched-path PE1-to-PE2]
user@PE1# set from 192.168.1.1
user@PE1# set metric 10
user@PE1# set preference 7
user@PE1# set install 192.168.1.4/32
user@PE1# set retry-timer 30
user@PE1# set retry-limit 5
user@PE1# set optimize-timer 3600

## Parameters explained:
## from: Source address (default: router ID)
## metric: IGP metric for inet.3 route
## preference: Route preference in inet.3
## install: Explicit prefix to install
## retry-timer: Seconds between retry attempts
## retry-limit: Maximum establishment attempts
## optimize-timer: Reoptimization interval
```

Controlling LSP Path Selection

```
## Method 1: Explicit path (strict)
[edit protocols mpls]
user@PE1# set path STRICT-PATH 10.0.1.2 strict
user@PE1# set path STRICT-PATH 10.0.2.2 strict
user@PE1# set path STRICT-PATH 10.0.3.2 strict
user@PE1# set label-switched-path PE1-to-PE2 primary STRICT-PATH

## Method 2: Loose path (suggested)
[edit protocols mpls]
user@PE1# set path LOOSE-PATH 192.168.1.2 loose
user@PE1# set path LOOSE-PATH 192.168.1.4 loose
user@PE1# set label-switched-path PE1-to-PE2 primary LOOSE-PATH

## Method 3: Exclude specific nodes/links
[edit protocols mpls]
user@PE1# set label-switched-path PE1-to-PE2 exclude 192.168.1.3
```

MPLS Self-Ping Configuration

```
## Self-ping is enabled by default, but can be tuned
[edit protocols mpls]
user@PE1# set label-switched-path PE1-to-PE2 self-ping-duration 60

## Or disable it
user@PE1# set label-switched-path PE1-to-PE2 no-self-ping

## Global self-ping settings
[edit protocols mpls]
user@PE1# set mpls-self-ping timeout 10
user@PE1# set mpls-self-ping interval 1
```

Complete Working Example

Here's a full configuration for our topology:

```
## PE1 Configuration
[edit protocols mpls]
set interface ge-0/0/1.0
set interface lo0.0
set label-switched-path PE1-to-PE2 {
    to 192.168.1.4;
    description "Primary LSP to PE2 DC";
    adaptive;
    install 192.168.1.4/32;
}

## P1 Configuration (Transit - No LSP config needed!)
[edit protocols mpls]
set interface ge-0/0/1.0
set interface ge-0/0/2.0
set interface lo0.0

## P2 Configuration (Transit - No LSP config needed!)
[edit protocols mpls]
set interface ge-0/0/1.0
set interface ge-0/0/2.0
set interface lo0.0

## PE2 Configuration
[edit protocols mpls]
set interface ge-0/0/1.0
set interface lo0.0
set label-switched-path PE2-to-PE1 {
    to 192.168.1.1;
    description "Return LSP to PE1 HQ";
    adaptive;
    install 192.168.1.1/32;
}
```

Understanding RSVP LSP Load Balancing

```
## Configure multiple LSPs for load balancing
[edit protocols mpls]
user@PE1# set label-switched-path PE1-to-PE2-1 to 192.168.1.4
user@PE1# set label-switched-path PE1-to-PE2-2 to 192.168.1.4

## Control load balancing behavior
[edit routing-options forwarding-table]
user@PE1# set export LOAD-BALANCE

[edit policy-options policy-statement LOAD-BALANCE]
user@PE1# set term 1 then load-balance per-packet
```

Part 3: Verification & Troubleshooting (The What-If)

Essential LSP Verification Commands

1. Check LSP Status

```
user@PE1> show mpls lsp
Ingress LSP: 1 sessions
To              From            State Rt P     ActivePath       LSPname
192.168.1.4     192.168.1.1     Up     0 *                      PE1-to-PE2
Total 1 displayed, Up 1, Down 0

user@PE1> show mpls lsp extensive
Ingress LSP: 1 sessions

192.168.1.4
  From: 192.168.1.1, State: Up, ActiveRoute: 0, LSPname: PE1-to-PE2
  ActivePath:  (primary)
  LSPtype: Static Configured
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  *Primary                    State: Up
    Priorities: 7 7
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 30)
      10.0.1.2 S 10.0.2.2 S 10.0.3.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
      10.0.1.2 10.0.2.2 10.0.3.2
   5 Oct 25 10:15:43.123 Self-ping ended successfully
   4 Oct 25 10:15:42.234 Record Route:  10.0.1.2 10.0.2.2 10.0.3.2
   3 Oct 25 10:15:42.123 Up
   2 Oct 25 10:15:42.000 Originate Call
   1 Oct 25 10:15:41.999 CSPF: computation result accepted  10.0.1.2 10.0.2.2 10.0.3.2
```

2. Verify inet.3 Installation

```
user@PE1> show route table inet.3

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.4/32     *[RSVP/7/1] 00:10:23, metric 30
                    > to 10.0.1.2 via ge-0/0/1.0, label-switched-path PE1-to-PE2

user@PE1> show route 192.168.1.4 detail
inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

```
192.168.1.4/32 (1 entry, 1 announced)
        State: <FlashAll>
        *RSVP   Preference: 7/1
                Next hop type: Router, Next hop index: 0
                Address: 0x9458234
                Next-hop reference count: 3
                Next hop: 10.0.1.2 via ge-0/0/1.0, selected
                Label-switched-path PE1-to-PE2
                Label operation: Push 100234
                Label TTL action: prop-ttl
                Load balance label: Label 100234: None;
                Label element ptr: 0x9458180
                Label parent element ptr: 0x0
                Label element references: 1
                Label element child references: 0
                Label element lsp id: 1
                Session Id: 0x141
                State: <Active Int>
                Age: 10:23  Metric: 30
                Validation State: unverified
                Task: RSVP
                Announcement bits (1): 0-KRT
                AS path: I
```

3. Check RSVP Signaling

```
user@PE1> show rsvp session
Ingress RSVP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
192.168.1.4     192.168.1.1     Up       0  1 FF       -   100234 PE1-to-PE2
Total 1 displayed, Up 1, Down 0

user@PE1> show rsvp session detail
Ingress RSVP: 1 sessions

192.168.1.4
  From: 192.168.1.1, LSPstate: Up, Route: 0
  LSPname: PE1-to-PE2, LSPpath: Primary
  LSPtype: Static Configured
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100234
  Resv style: 1 FF, Label in: -, Label out: 100234
  Time left:    -, Since: Mon Oct 25 10:15:42 2024
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 49644 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.0.1.2 (ge-0/0/1.0) 4 pkts
  RESV rcvfrom: 10.0.1.2 (ge-0/0/1.0) 4 pkts
  Explct route: 10.0.1.2 10.0.2.2 10.0.3.2
  Record route: <self> 10.0.1.2 10.0.2.2 10.0.3.2
```

Common Troubleshooting Scenarios

Scenario 1: LSP Stuck in Down State

Symptom: LSP won't come up

Diagnostic Commands:

```
user@PE1> show mpls lsp extensive | match "State|error"
  From: 192.168.1.1, State: Dn, ActiveRoute: 0, LSPname: PE1-to-PE2
    State: Dn
    Last error: CSPF failed: no route toward 192.168.1.4[4 times]

user@PE1> show route 192.168.1.4 table inet.0
inet.0: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
[no output - destination not reachable]
```

Cause: No IP reachability to destination

Solution:

```
## Verify OSPF adjacencies
user@PE1> show ospf neighbor
user@PE1> show route protocol ospf

## Check for OSPF on loopback
user@PE1> show configuration protocols ospf | display set | match lo0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Scenario 2: Self-Ping Failing

Symptom: LSP signals but stays down, self-ping fails

Diagnostic Commands:

```
user@PE1> show mpls lsp extensive | find "Created:"
    Created: Mon Oct 25 10:20:15 2024
  15 Oct 25 10:20:20.456 Self-ping failed
  14 Oct 25 10:20:20.123 Self-ping started
  13 Oct 25 10:20:19.999 Record Route:  10.0.1.2 10.0.2.2 10.0.3.2
  12 Oct 25 10:20:19.888 Up
  11 Oct 25 10:20:19.777 Originate Call

user@PE1> show rsvp session
To              From            State   Rt Style Labelin Labelout LSPname
192.168.1.4     192.168.1.1     Up       0  1 FF       -   100234 PE1-to-PE2
```

Cause: Forwarding plane not programmed correctly

Solution:

```
## Manual ping test
user@PE1> ping mpls rsvp PE1-to-PE2
!!!!!
--- lsping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss

## Check forwarding table
user@PE1> show route forwarding-table destination 192.168.1.4 table default
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
192.168.1.4/32     user     0 10.0.1.2          Push 100234    607     2 ge-0/0/1.0

## Restart routing if needed
user@PE1> restart routing immediately
```

Scenario 3: LSP Using Wrong Path

Symptom: LSP up but using suboptimal path

Diagnostic Commands:

```
user@PE1> show mpls lsp extensive | match "ERO|RRO"
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 120)
      10.0.4.2 S 10.0.5.2 S 10.0.6.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
      10.0.4.2 10.0.5.2 10.0.6.2

user@PE1> show ted database extensive | match "Local|Metric"
  To: 192.168.1.2, Local: 10.0.1.1, Remote: 10.0.1.2
    Metric: 100, Bandwidth: 10Gbps
```

Cause: High IGP metric on preferred path

Solution:

```
## Check OSPF metrics
user@PE1> show ospf interface detail | match "Interface|Cost"

## Adjust if needed
[edit protocols ospf]
user@PE1# set area 0.0.0.0 interface ge-0/0/1.0 metric 10
user@PE1# commit
```

Scenario 4: CSPF Computation Issues

Symptom: "CSPF failed" errors

Diagnostic Commands:

```
user@PE1> show mpls lsp extensive | match "CSPF"
    1 Oct 25 10:30:00.123 CSPF failed: no route toward 192.168.1.4

user@PE1> show ted database 192.168.1.4
TED database: 0 ISIS nodes 3 INET nodes
NodeID: 192.168.1.4
  Type: Rtr, Age: 1234 secs, LinkIn: 1, LinkOut: 0
[missing links]
```

Cause: Incomplete TED database

Solution:

```
## Force TED rebuild
user@PE1> clear ted database

## Or disable CSPF temporarily
[edit protocols mpls label-switched-path PE1-to-PE2]
user@PE1# set no-cspf
user@PE1# commit
```

Advanced Troubleshooting Tools

RSVP Traceoptions for Signaling Issues

```
[edit protocols rsvp]
user@PE1# set traceoptions file rsvp-signal
user@PE1# set traceoptions file size 10m
user@PE1# set traceoptions flag packets detail
user@PE1# set traceoptions flag path detail
user@PE1# set traceoptions flag resv detail
user@PE1# commit

user@PE1> show log rsvp-signal | match PE1-to-PE2
Oct 25 10:35:00 RSVP send Path 192.168.1.1->192.168.1.4, Tunnel-ID 1, LSP-ID 1
Oct 25 10:35:00   Session: 192.168.1.4/32, Proto 0, Port 0, Tunnel-ID 1, Ext-ID 192.168.1.1
Oct 25 10:35:01 RSVP recv Resv 10.0.1.2->10.0.1.1
Oct 25 10:35:01   Label: 100234, Tspec: rate 0 size 0 peak Inf
```

Testing LSP Forwarding

```
## Test with specific packet size
user@PE1> ping mpls rsvp PE1-to-PE2 size 1400 count 5
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.123/1.234/1.345/0.089 ms

## Test with sweep to find MTU issues
user@PE1> ping mpls rsvp PE1-to-PE2 sweep size 1400-1500 count 1
Sweep-ping mpls rsvp PE1-to-PE2: 1400...1472
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Sweep-ping mpls rsvp PE1-to-PE2: 1473...1500
.............................
```

This basic RSVP LSP provides the foundation for more advanced features like traffic engineering, which we'll explore in the next module.

Module 6: RSVP-The Traffic Engineering Database

Part 1: The Conceptual Lecture (The Why)

What is the Traffic Engineering Database (TED)?

Imagine you're planning a cross-country road trip. You wouldn't just need a basic map showing which roads connect - you'd want to know road conditions, traffic levels, construction zones, speed limits, and toll costs. The Traffic Engineering Database (TED) is MPLS's version of this enhanced map.

The TED contains:

- Network topology (who connects to whom)
- Link bandwidth (capacity of each road)
- Available bandwidth (how much traffic is already there)
- Administrative attributes (toll roads, scenic routes, etc.)
- TE metrics (different from IGP metrics)

Why Do We Need Traffic Engineering?

Traditional IP routing has a fundamental limitation:

```
The Shortest Path Problem:

        1Gbps link
    R1 ------------ R2
     |               |
     |               |
    10G            10G
     |               |
     |               |
    R3 ------------ R4
        10Gbps link

IGP Shortest Path: All traffic R1→R4 goes via R2
Problem: 1Gbps link becomes congested while 10G path sits idle!

Traffic Engineering Solution: Force some LSPs via R3
Result: Better network utilization
```

How IGP Extensions Enable Traffic Engineering

OSPF and IS-IS were extended to carry TE information:

```
Standard OSPF LSA:
- Router ID
- Interface IP
- Metric
- Network mask

OSPF-TE Opaque LSA (Type 10):
- All standard info PLUS:
- Maximum bandwidth
- Reservable bandwidth
- Available bandwidth per priority
- Administrative groups (colors)
- TE metric
- Shared Risk Link Groups (SRLG)
```

These extensions are like adding real-time traffic, road quality, and restriction information to a basic street map.

The CSPF Algorithm: Intelligence in Path Selection

CSPF (Constrained Shortest Path First) is like a GPS that considers multiple factors:

```
Traditional SPF (Dijkstra):
"What's the shortest path?"

CSPF:
"What's the shortest path that:
 - Has 500 Mbps available bandwidth?
 - Avoids toll roads (admin group 'expensive')?
 - Doesn't use any red-colored links?
 - Stays within latency requirements?"
```

Administrative Groups (Link Coloring)

Admin groups are like tags on roads:

```
Link Coloring Example:
- Red: Expensive satellite links
- Blue: Cheap terrestrial fiber
- Green: Low-latency links
- Gold: High-security encrypted links

LSP Requirements:
- Voice LSP: Include Green, Exclude Red
- Backup LSP: Include Blue, Exclude Gold
- Premium LSP: Include Gold, Exclude Red
```

Each link can belong to multiple groups (32 different colors available).

TE Metrics vs IGP Metrics

Having separate metrics is like having different cost calculations:

```
IGP Metric: For regular routing (shortest path)
TE Metric: For traffic-engineered paths

Example:
Link A: IGP metric=10, TE metric=100 (high bandwidth but expensive)
Link B: IGP metric=50, TE metric=10 (longer but cheaper)

Regular traffic: Uses Link A (IGP)
TE LSPs: Use Link B (TE metric)
```

Loose vs Strict Hops in ERO

When specifying paths, you can be prescriptive or suggestive:

```
Strict Hop: "You MUST go directly here"
Like: "Take I-95 North to Exit 42"

Loose Hop: "Include this in your path somehow"
Like: "Make sure to go through Boston"

Example Path:
```

```
10.1.1.1 strict → 10.2.2.2 loose → 10.3.3.3 strict

Means:
- MUST go directly from current location to 10.1.1.1
- Then get to 10.2.2.2 any way you want
- From wherever you are, MUST go directly to 10.3.3.3
```

Part 2: The Junos CLI Masterclass (The How)

Configuring OSPF-TE Extensions

First, ensure OSPF advertises TE information:

```
## Basic TE enablement (from previous module)
[edit protocols ospf]
user@PE1# set traffic-engineering

## But let's configure TE-specific parameters
[edit protocols ospf]
user@PE1# set traffic-engineering shortcuts
user@PE1# set traffic-engineering multicast-rpf-routes
user@PE1# set traffic-engineering credibility-protocol-preference

## Configure TE on specific interfaces
[edit protocols ospf area 0.0.0.0 interface ge-0/0/1.0]
user@PE1# set te-metric 100
```

Configuring IS-IS TE Extensions

For IS-IS networks:

```
[edit protocols isis]
user@PE1# set traffic-engineering ipv4
user@PE1# set traffic-engineering credibility-protocol-preference

[edit protocols isis interface ge-0/0/1.0]
user@PE1# set level 2 te-metric 100
```

Setting Up Administrative Groups

Define your link colors:

```
## Step 1: Define admin group names
[edit protocols mpls]
user@PE1# set admin-groups GOLD 0
user@PE1# set admin-groups SILVER 1
user@PE1# set admin-groups BRONZE 2
user@PE1# set admin-groups EXPENSIVE 3
user@PE1# set admin-groups SATELLITE 4
user@PE1# set admin-groups FIBER 5
user@PE1# set admin-groups LOW-LATENCY 6
user@PE1# set admin-groups HIGH-LATENCY 7

## Step 2: Apply to interfaces
[edit protocols mpls interface ge-0/0/1.0]
user@PE1# set admin-group FIBER
user@PE1# set admin-group LOW-LATENCY

[edit protocols mpls interface ge-0/0/2.0]
user@PE1# set admin-group SATELLITE
user@PE1# set admin-group EXPENSIVE
user@PE1# set admin-group HIGH-LATENCY
```

Configuring TE Metrics

Set different metrics for TE vs regular routing:

```
## IGP metric (regular routing)
[edit protocols ospf area 0.0.0.0 interface ge-0/0/1.0]
user@PE1# set metric 10

## TE metric (traffic engineering)
[edit protocols ospf area 0.0.0.0 interface ge-0/0/1.0]
user@PE1# set te-metric 100

## The result:
## - Regular IP traffic follows IGP metric (10)
## - TE LSPs follow TE metric (100)
```

Creating LSPs Using the TED

Now let's create LSPs that use TE constraints:

```
## Basic LSP with admin group constraints
[edit protocols mpls]
user@PE1# set label-switched-path GOLD-PATH to 192.168.1.4
user@PE1# set label-switched-path GOLD-PATH admin-group include-all GOLD
user@PE1# set label-switched-path GOLD-PATH admin-group include-any LOW-LATENCY FIBER
user@PE1# set label-switched-path GOLD-PATH admin-group exclude SATELLITE EXPENSIVE

## What this means:
## - MUST traverse links with GOLD
## - MUST traverse links with either LOW-LATENCY OR FIBER
## - MUST NOT traverse links with SATELLITE or EXPENSIVE
```

Advanced Path Constraints

```
## LSP with multiple constraints
[edit protocols mpls label-switched-path VOICE-LSP]
user@PE1# set to 192.168.1.4
user@PE1# set bandwidth 100m
user@PE1# set admin-group include-any LOW-LATENCY
user@PE1# set admin-group exclude HIGH-LATENCY SATELLITE
user@PE1# set priority 0 0

## LSP preferring TE metric over IGP
[edit protocols mpls label-switched-path BULK-DATA]
user@PE1# set to 192.168.1.4
user@PE1# set least-fill
user@PE1# set admin-group include-any FIBER
```

Configuring Explicit Route Objects (ERO)

Define specific paths through the network:

```
## Strict path (must follow exactly)
[edit protocols mpls]
user@PE1# set path STRICT-PATH-VIA-P1 10.0.1.2 strict
user@PE1# set path STRICT-PATH-VIA-P1 10.0.2.2 strict
user@PE1# set path STRICT-PATH-VIA-P1 10.0.3.2 strict

## Loose path (suggested waypoints)
[edit protocols mpls]
user@PE1# set path LOOSE-PATH-AVOID-P3 192.168.1.2 loose
user@PE1# set path LOOSE-PATH-AVOID-P3 192.168.1.4 loose

## Mixed strict/loose path
[edit protocols mpls]
user@PE1# set path MIXED-PATH 10.0.1.2 strict
user@PE1# set path MIXED-PATH 192.168.1.2 loose
user@PE1# set path MIXED-PATH 10.0.3.2 strict

## Apply path to LSP
user@PE1# set label-switched-path CONTROLLED-LSP primary STRICT-PATH-VIA-P1
```

Complete TE Configuration Example

Here's a comprehensive setup:

```
## PE1 Complete TE Configuration
## Admin groups definition
set protocols mpls admin-groups GOLD 0
set protocols mpls admin-groups SILVER 1
set protocols mpls admin-groups EXPENSIVE 3
set protocols mpls admin-groups CHEAP 4
set protocols mpls admin-groups SECURE 5
set protocols mpls admin-groups PUBLIC 6

## Interface configuration with TE parameters
set protocols mpls interface ge-0/0/1.0 admin-group GOLD
set protocols mpls interface ge-0/0/1.0 admin-group SECURE
set protocols mpls interface ge-0/0/2.0 admin-group SILVER
set protocols mpls interface ge-0/0/2.0 admin-group CHEAP

## OSPF TE configuration
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 te-metric 50
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 te-metric 100

## TE-based LSPs
set protocols mpls label-switched-path PREMIUM-LSP {
    to 192.168.1.4;
    bandwidth 1g;
    admin-group include-all GOLD SECURE;
    admin-group exclude CHEAP PUBLIC;
    priority 1 1;
}

set protocols mpls label-switched-path ECONOMY-LSP {
    to 192.168.1.4;
    bandwidth 100m;
    admin-group include-any CHEAP SILVER;
    admin-group exclude EXPENSIVE;
    priority 7 7;
}
```

CSPF Computation Control

Fine-tune how CSPF works:

```
## Disable CSPF (use plain IGP)
[edit protocols mpls label-switched-path SIMPLE-LSP]
user@PE1# set no-cspf

## Use CSPF with random tie-breaking
[edit protocols mpls label-switched-path LOAD-BALANCE-LSP]
user@PE1# set random

## Prefer least-filled links
[edit protocols mpls label-switched-path LEAST-FILL-LSP]
user@PE1# set least-fill

## Prefer most-filled links (pack links)
[edit protocols mpls label-switched-path MOST-FILL-LSP]
user@PE1# set most-fill
```

Part 3: Verification & Troubleshooting (The What-If)

Essential TED Verification Commands

1. View the Complete TED

```
user@PE1> show ted database
TED database: 0 ISIS nodes 4 INET nodes
ID                          Type Age(s) LnkIn LnkOut Protocol
192.168.1.1                 Rtr     120     1      1 OSPF(0.0.0.0)
  To: 192.168.1.2, Local: 10.0.1.1, Remote: 10.0.1.2
192.168.1.2                 Rtr     115     2      2 OSPF(0.0.0.0)
  To: 192.168.1.1, Local: 10.0.1.2, Remote: 10.0.1.1
  To: 192.168.1.3, Local: 10.0.2.1, Remote: 10.0.2.2
192.168.1.3                 Rtr     118     2      2 OSPF(0.0.0.0)
  To: 192.168.1.2, Local: 10.0.2.2, Remote: 10.0.2.1
  To: 192.168.1.4, Local: 10.0.3.1, Remote: 10.0.3.2
192.168.1.4                 Rtr     112     1      1 OSPF(0.0.0.0)
  To: 192.168.1.3, Local: 10.0.3.2, Remote: 10.0.3.1
```

2. Detailed TED Information

```
user@PE1> show ted database extensive
TED database: 0 ISIS nodes 4 INET nodes

NodeID: 192.168.1.1
  Type: Rtr, Age: 450 secs, LinkIn: 1, LinkOut: 1
  Protocol: OSPF(0.0.0.0)
    To: 192.168.1.2, Local: 10.0.1.1, Remote: 10.0.1.2
      Local interface index: 333, Remote interface index: 0
      Link name: ge-0/0/1.0
      Metric: 10, TE Metric: 50, IGP metric: 10
      Static BW: 10Gbps, Reservable BW: 10Gbps
      Available BW [priority] bps:
        [0] 9Gbps      [1] 9Gbps      [2] 9Gbps      [3] 9Gbps
        [4] 10Gbps     [5] 10Gbps     [6] 10Gbps     [7] 10Gbps
      Admin groups: GOLD SECURE
      Admin group membership:
        0x00000021 (bits 0 5)
        GOLD
        SECURE
```

3. Check LSP Path Computation

```
user@PE1> show mpls lsp name PREMIUM-LSP extensive
  ...
  *Primary                    State: Up
    Priorities: 1 1
    Bandwidth: 1Gbps
    SmartOptimizeTimer: 180
    Include All: GOLD SECURE
    Exclude: CHEAP PUBLIC
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 150)
      10.0.1.2 S 10.0.2.2 S 10.0.3.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
      10.0.1.2(flag=0x20) 10.0.2.2(flag=0x20) 10.0.3.2(flag=0x20)
    ...
   8 Nov 15 14:23:15.234 CSPF: computation result accepted  10.0.1.2 10.0.2.2 10.0.3.2
   7 Nov 15 14:23:15.123 CSPF: Start computation: bandwidth 1000Mbps, include all 0x21 exclude 0x50
   6 Nov 15 14:23:15.122 CSPF: Reroute due to re-optimization
```

Common Troubleshooting Scenarios

Scenario 1: CSPF Can't Find Path

Symptom: LSP down with CSPF errors

Diagnostic Commands:

```
user@PE1> show mpls lsp name PREMIUM-LSP extensive | match "error|fail"
    Last error: CSPF failed: no route toward 192.168.1.4 with constraints[3 times]

user@PE1> show ted database extensive | match "Admin group" | count
Count: 2 lines
```

```
## Check which links have required admin groups
user@PE1> show ted database extensive | find "GOLD"
      Admin groups: GOLD SECURE
```

Cause: No path exists matching all constraints

Solution:

```
## Relax constraints
[edit protocols mpls label-switched-path PREMIUM-LSP]
user@PE1# delete admin-group include-all
user@PE1# set admin-group include-any GOLD SECURE
user@PE1# commit

## Or check admin-group configuration on all links
```

Scenario 2: LSP Using Suboptimal Path

Symptom: LSP up but taking longer path

Diagnostic Commands:

```
user@PE1> show mpls lsp name ECONOMY-LSP extensive | match "ERO|metric"
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 200)
      10.0.4.2 S 10.0.5.2 S 10.0.6.2 S 10.0.7.2 S

user@PE1> show ted path source 192.168.1.1 destination 192.168.1.4
Path 1: (Metric 30)
  192.168.1.1 -> 192.168.1.2 -> 192.168.1.3 -> 192.168.1.4

## But why didn't it use the shorter path?
user@PE1> show ted database extensive 192.168.1.2 | match "admin|Admin"
      Admin groups: GOLD EXPENSIVE
```

Cause: Shorter path has excluded admin group

Solution: Review admin-group assignments or constraints

Scenario 3: TED Missing Information

Symptom: Incomplete TED database

Diagnostic Commands:

```
user@PE1> show ted database | match "Protocol|nodes"
TED database: 0 ISIS nodes 2 INET nodes

user@PE1> show ospf database opaque-area | match "Opaque|Advertising"
  Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
  OpaqArea   1.0.0.1          192.168.1.1    0x80000034  234  0x22 0x8abc  136
  OpaqArea*  1.0.0.1          192.168.1.2    0x80000012 1847  0x22 0x7def  136

## Note the age - 1847 seconds is suspiciously old
```

Cause: OSPF-TE not enabled on all routers

Solution:

```
## On router missing from TED
[edit protocols ospf]
user@P2# set traffic-engineering
user@P2# commit
```

Scenario 4: Admin Groups Not Working

Symptom: LSP ignores admin-group constraints

Diagnostic Commands:

```
user@PE1> show configuration protocols mpls admin-groups
GOLD 0;
SILVER 1;

user@PE1> show ted database extensive | match "Admin groups:"
      Admin groups: 0x00000003
      Admin groups: 0x00000001

## Numbers instead of names - mapping issue?

user@PE1> show mpls admin-groups
Group                    Bit    Hex
GOLD                       0    0x1
SILVER                     1    0x2
```

Solution: Ensure consistent admin-group configuration across network:

```
## Check on all routers
user@router> show configuration protocols mpls admin-groups | display set
```

Advanced TED Analysis

Testing Path Computation

```
## Test CSPF computation without creating LSP
user@PE1> show ted path source 192.168.1.1 destination 192.168.1.4 bandwidth 5g
Path 1: Insufficient bandwidth

user@PE1> show ted path source 192.168.1.1 destination 192.168.1.4 bandwidth 1g
Path 1: (Metric 150)
  192.168.1.1 -> 192.168.1.2 -> 192.168.1.3 -> 192.168.1.4

## Test with admin constraints
user@PE1> show ted path source 192.168.1.1 destination 192.168.1.4 exclude-bitmap 0x4
Path 1: (Metric 250)
  192.168.1.1 -> 192.168.1.5 -> 192.168.1.6 -> 192.168.1.4
```

Monitoring TED Changes

```
[edit protocols mpls]
user@PE1# set traceoptions file ted-debug
user@PE1# set traceoptions flag cspf detail
user@PE1# commit

user@PE1> monitor start ted-debug

## In another window, check the log
user@PE1> show log ted-debug
Nov 15 14:45:23 CSPF: Start SPF, dest 192.168.1.4
Nov 15 14:45:23 CSPF: Node 192.168.1.1 visited, dist 0
Nov 15 14:45:23 CSPF: Link 192.168.1.1->192.168.1.2 admin 0x21 matches include-all 0x21
Nov 15 14:45:23 CSPF: Link 192.168.1.1->192.168.1.2 admin 0x21 no match exclude 0x50
Nov 15 14:45:23 CSPF: Link 192.168.1.1->192.168.1.2 has bandwidth 9000000000 >= 1000000000
Nov 15 14:45:23 CSPF: Add node 192.168.1.2 dist 50 via 10.0.1.2
```

Performance Impact of Complex Constraints

```
## Show CSPF computation time
user@PE1> show mpls lsp statistics
LSP statistics:
  Total LSPs: 15
  CSPF computations: 234
  CSPF failures: 12
  Average CSPF time: 2.3 ms
  Maximum CSPF time: 45.2 ms
```

```
## If CSPF times are high, consider:
## - Reducing constraint complexity
## - Using no-cspf for simple LSPs
## - Enabling CSPF load balancing
```

The Traffic Engineering Database transforms MPLS from simple label switching into an intelligent, constraint-based routing system. Next, we'll explore how to reserve bandwidth along these intelligently computed paths.

Module 7: RSVP-LSP Bandwidth Reservation

Part 1: The Conceptual Lecture (The Why)

Understanding Bandwidth Reservation

Imagine a highway system where you could reserve lanes. During rush hour, while regular traffic sits in congestion, your reserved lane remains clear. That's what RSVP bandwidth reservation does for network traffic - it guarantees capacity for critical applications.

Why Reserve Bandwidth?

Without bandwidth reservation, MPLS LSPs are like having a GPS that finds the best route but can't guarantee the road won't be congested when you get there:

```
Without Bandwidth Reservation:
- LSP finds path with 10Gbps link
- 100 other LSPs think the same thing
- Everyone tries to use 10Gbps link
- Congestion and packet loss!

With Bandwidth Reservation:
- LSP reserves 1Gbps on 10Gbps link
- Link tracks: "9Gbps still available"
- Next LSP sees only 9Gbps available
- Prevents oversubscription
```

How Bandwidth Reservation Works

RSVP uses a "hotel reservation" model:

```
Bandwidth Reservation Process:

1. Check Availability (PATH message)
   "Do you have 1Gbps available?"

2. Make Reservation (RESV message)
   "Yes, I'm reserving 1Gbps for you"

3. Update Availability
   Before: 10Gbps available
   After: 9Gbps available

4. Maintain Reservation
   Periodic refresh keeps reservation active
```

The Bandwidth Reservation Model

Junos tracks bandwidth at 8 priority levels (0-7, where 0 is highest):

```
Interface: 10Gbps total
Reservable: 10Gbps

```

```
Available per priority:
[0] 10Gbps  ← Highest priority sees all bandwidth
[1] 10Gbps
[2] 10Gbps
[3] 10Gbps
[4] 10Gbps
[5] 10Gbps
[6] 10Gbps
[7] 10Gbps  ← Lowest priority

After 2Gbps reservation at priority 4:
[0] 10Gbps  ← Higher priorities unaffected
[1] 10Gbps
[2] 10Gbps
[3] 10Gbps
[4] 8Gbps   ← Reserved here
[5] 8Gbps   ← Lower priorities affected
[6] 8Gbps
[7] 8Gbps
```

Statistical Multiplexing and Subscription

Networks rarely see all traffic flowing simultaneously. Subscription allows overbooking:

```
Subscription Model:
Physical bandwidth: 10Gbps
Subscription: 200%
Reservable: 20Gbps

Like airlines overbooking seats:
- Not everyone uses full reservation
- Statistical multiplexing works
- But can lead to congestion if everyone shows up!
```

Bandwidth Policing vs Reservation

Important distinction:

```
Bandwidth Reservation:
- Admission control only
- "Can this LSP fit?"
- Doesn't police traffic

Bandwidth Policing:
- Actual traffic limiting
- Requires separate configuration
- QoS policies at edge
```

Auto-Bandwidth: Dynamic Reservations

Static reservations can waste capacity. Auto-bandwidth adjusts based on actual usage:

```
Auto-bandwidth Operation:
Initial reservation: 100Mbps
Monitor actual traffic...
Hour 1: Using 80Mbps → Keep 100Mbps
Hour 2: Using 180Mbps → Increase to 200Mbps
Hour 3: Using 50Mbps → Decrease to 75Mbps

Like a flex-capacity subscription!
```

Part 2: The Junos CLI Masterclass (The How)

Basic Bandwidth Reservation

Configure an LSP with bandwidth reservation:

```
## Simple bandwidth reservation
[edit protocols mpls]
user@PE1# set label-switched-path BW-LSP to 192.168.1.4
user@PE1# set label-switched-path BW-LSP bandwidth 1g

## With specific priority
user@PE1# set label-switched-path PRIORITY-BW-LSP to 192.168.1.4
user@PE1# set label-switched-path PRIORITY-BW-LSP bandwidth 2g
user@PE1# set label-switched-path PRIORITY-BW-LSP priority 3 3
```

Understanding Bandwidth Units

Junos accepts various bandwidth specifications:

```
## Different ways to specify bandwidth
set label-switched-path LSP1 bandwidth 1000000000  ## Bits per second
set label-switched-path LSP2 bandwidth 1g          ## Gigabits per second
set label-switched-path LSP3 bandwidth 1000m       ## Megabits per second
set label-switched-path LSP4 bandwidth 1.5g        ## Decimal values

## Special values
set label-switched-path LSP5 bandwidth 0           ## No reservation
set label-switched-path LSP6 bandwidth ct0 500m    ## Class-type aware
```

Configuring Interface Bandwidth for RSVP

Define how much bandwidth RSVP can reserve:

```
## Method 1: Explicit bandwidth statement
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set bandwidth 10g

## Method 2: Percentage of interface speed
user@PE1# set bandwidth percent 80

## Method 3: Different values for TE database vs reservable
user@PE1# set bandwidth 10g 8g
## First: Advertised in TED
## Second: Actually reservable

## Method 4: Per-class-type bandwidth
user@PE1# set bandwidth ct0 6g
user@PE1# set bandwidth ct1 2g
user@PE1# set bandwidth ct2 2g
```

Bandwidth Subscription (Overbooking)

Configure statistical multiplexing:

```
## Allow 150% subscription
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set subscription 150

## Example calculation:
## Physical: 10G
## Reservable: 8G
## With 150% subscription: 8G × 1.5 = 12G can be reserved

## Per-class-type subscription
user@PE1# set subscription ct0 200
user@PE1# set subscription ct1 100
```

Auto-Bandwidth Configuration

Configure dynamic bandwidth adjustment:

```
## Basic auto-bandwidth
[edit protocols mpls label-switched-path AUTO-BW-LSP]
user@PE1# set to 192.168.1.4
user@PE1# set auto-bandwidth
user@PE1# set auto-bandwidth adjust-interval 3600
user@PE1# set auto-bandwidth minimum-bandwidth 100m
user@PE1# set auto-bandwidth maximum-bandwidth 5g

## Advanced auto-bandwidth with thresholds
user@PE1# set auto-bandwidth adjust-threshold 20
user@PE1# set auto-bandwidth adjust-threshold-overflow-limit 3
user@PE1# set auto-bandwidth adjust-threshold-underflow-limit 3

## Monitor without adjusting (testing mode)
user@PE1# set auto-bandwidth monitor-only
```

Multi-Class Bandwidth Reservation (DiffServ-TE)

Configure different bandwidth pools:

```
## Define class types
[edit protocols rsvp]
user@PE1# set diffserv-te class-type ct0 priority 0 3
user@PE1# set diffserv-te class-type ct1 priority 4 5
user@PE1# set diffserv-te class-type ct2 priority 6 7

## Configure interface bandwidth per class
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set bandwidth ct0 6g
user@PE1# set bandwidth ct1 3g
user@PE1# set bandwidth ct2 1g

## Create LSPs using specific class types
[edit protocols mpls label-switched-path VOICE-LSP]
user@PE1# set bandwidth ct0 500m
user@PE1# set priority 0 0

[edit protocols mpls label-switched-path DATA-LSP]
user@PE1# set bandwidth ct1 2g
user@PE1# set priority 4 4
```

Complete Bandwidth Reservation Example

Here's a comprehensive configuration:

```
## PE1 Configuration

## RSVP interface bandwidth
set protocols rsvp interface ge-0/0/1.0 bandwidth 10g 9g
set protocols rsvp interface ge-0/0/1.0 subscription 125

## Different LSPs with bandwidth requirements
set protocols mpls label-switched-path GOLD-SERVICE {
    to 192.168.1.4;
    bandwidth 2g;
    priority 1 1;
    admin-group include-any GOLD;
}

set protocols mpls label-switched-path SILVER-SERVICE {
    to 192.168.1.4;
    bandwidth 1g;
    priority 3 3;
    admin-group include-any SILVER;
}
```

```
set protocols mpls label-switched-path BRONZE-SERVICE {
    to 192.168.1.4;
    bandwidth 500m;
    priority 5 5;
    admin-group include-any BRONZE;
}

set protocols mpls label-switched-path AUTO-ADJUST {
    to 192.168.1.4;
    auto-bandwidth {
        adjust-interval 300;  ## 5 minutes
        minimum-bandwidth 50m;
        maximum-bandwidth 2g;
        adjust-threshold 10;  ## 10% change triggers adjustment
    }
}
```

Bandwidth Protection and Sharing

Configure bandwidth sharing between LSPs:

```
## Shared bandwidth (for backup paths)
[edit protocols mpls]
user@PE1# set label-switched-path PRIMARY to 192.168.1.4 bandwidth 1g
user@PE1# set label-switched-path PRIMARY primary PRIMARY-PATH

user@PE1# set label-switched-path PRIMARY secondary BACKUP-PATH
user@PE1# set label-switched-path PRIMARY secondary BACKUP-PATH bandwidth 1g
user@PE1# set label-switched-path PRIMARY secondary BACKUP-PATH shared

## Bandwidth protection
user@PE1# set label-switched-path PROTECTED bandwidth 1g
user@PE1# set label-switched-path PROTECTED link-protection
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Bandwidth Verification Commands

1. Check Interface Bandwidth Status

```
user@PE1> show rsvp interface detail
ge-0/0/1.0
  Index: 329, State: Up
  ActiveResv: 3, Subscription: 125%
  Static BW: 10Gbps, Reservable BW: 9Gbps
  BC Model: MAM, BC0 BW: 9Gbps
  Reserved bandwidth:
    CT0: 3.5Gbps, CT1: 0bps, CT2: 0bps, CT3: 0bps
  Available bandwidth:
    HighPrio: 9Gbps, HighNonCT: 5.5Gbps
    [0] 9Gbps        [1] 9Gbps        [2] 9Gbps        [3] 9Gbps
    [4] 5.5Gbps      [5] 5.5Gbps      [6] 5.5Gbps      [7] 5.5Gbps
  Reserved bandwidth by priority:
    [0] 0bps         [1] 2Gbps        [2] 0bps         [3] 1.5Gbps
    [4] 0bps         [5] 0bps         [6] 0bps         [7] 0bps
```

2. View LSP Bandwidth Reservations

```
user@PE1> show mpls lsp extensive name GOLD-SERVICE
  ...
  *Primary                    State: Up
    Priorities: 1 1
    Bandwidth: 2Gbps
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 30)
      10.0.1.2 S 10.0.2.2 S 10.0.3.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
      10.0.1.2(Label=300456/flag=0x24) 10.0.2.2(Label=300789/flag=0x24) 10.0.3.2(Label=3/flag=0x24)
    ...
   15 Nov 20 09:45:23.456 CSPF: computation result accepted  10.0.1.2 10.0.2.2 10.0.3.2
   14 Nov 20 09:45:23.345 Make-before-break: Switched to new instance
   13 Nov 20 09:45:23.234 Bandwidth requested 2000000000 bps
```

3. Monitor Auto-Bandwidth Statistics

```
user@PE1> show mpls lsp auto-bandwidth
Ingress LSP: 1 sessions
To              From            State   Rt  BW(bps)      InBytes       OutBytes
192.168.1.4     192.168.1.1     Up       0  1.2g         5432109876    0

  Name: AUTO-ADJUST
  Auto-bandwidth parameters:
    Adjust interval: 300 secs, Adjust threshold: 10%
    Min bw: 50Mbps, Max bw: 2Gbps
    Current bandwidth: 1.2Gbps
    Last adjustment: 00:03:45 ago
    Next adjustment in: 00:01:15
    Current sampling rate: 432.567Mbps
    Overflow count: 0, Underflow count: 0
    Overflow limit: 3, Underflow limit: 3
```

Common Troubleshooting Scenarios

Scenario 1: LSP Down - Insufficient Bandwidth

Symptom: LSP fails to establish due to bandwidth

Diagnostic Commands:

```
user@PE1> show mpls lsp name BW-LSP extensive | match "error|Error"
    State: Dn
    Last error: CSPF failed: no route toward 192.168.1.4 matching bandwidth constraint[4 times]

user@PE1> show ted database extensive | match "Available BW" -A 2
      Available BW [priority] bps:
        [0] 500Mbps    [1] 500Mbps    [2] 500Mbps    [3] 500Mbps
        [4] 500Mbps    [5] 500Mbps    [6] 500Mbps    [7] 500Mbps

## LSP needs 1G but only 500M available
```

Cause: Insufficient bandwidth on path

Solution:

```
## Option 1: Reduce bandwidth requirement
[edit protocols mpls label-switched-path BW-LSP]
user@PE1# set bandwidth 400m

## Option 2: Use different path
[edit protocols mpls label-switched-path BW-LSP]
user@PE1# set primary ALTERNATE-PATH

## Option 3: Check what's using bandwidth
user@PE1> show rsvp session | match "ge-0/0/1.0"
```

Scenario 2: Auto-Bandwidth Not Adjusting

Symptom: Auto-bandwidth LSP not changing bandwidth

Diagnostic Commands:

```
user@PE1> show mpls lsp auto-bandwidth name AUTO-ADJUST
  Current bandwidth: 100Mbps
  Last adjustment: 02:45:30 ago
  Current sampling rate: 850.234Mbps
  Overflow count: 2, Underflow count: 0
  Overflow limit: 3, Underflow limit: 3

## High traffic but no adjustment yet
```

Cause: Threshold limits not reached

Solution:

```
## Lower thresholds for faster adjustment
[edit protocols mpls label-switched-path AUTO-ADJUST auto-bandwidth]
user@PE1# set adjust-threshold-overflow-limit 2
user@PE1# set adjust-threshold 5
user@PE1# commit
```

Scenario 3: Subscription Causing Issues

Symptom: Physical link congested despite reservations

Diagnostic Commands:

```
user@PE1> show rsvp interface ge-0/0/1.0
  Static BW: 10Gbps, Reservable BW: 10Gbps
  Subscription: 200%
  Reserved: 18Gbps  ## More than physical!

user@PE1> show interfaces ge-0/0/1.0 extensive | match "bps"
  Traffic statistics:
    Input  bytes  :        123456789012345                 9234567890 bps
    Output bytes  :        234567890123456                12345678901 bps  ## Over 10G!
```

Cause: Oversubscription with actual traffic exceeding physical capacity

Solution:

```
## Reduce subscription
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set subscription 100
user@PE1# commit

## May cause some LSPs to go down!
```

Scenario 4: Wrong Priority Reservations

Symptom: High-priority LSP can't get bandwidth

Diagnostic Commands:

```
user@PE1> show rsvp interface detail | find "Reserved bandwidth by"
  Reserved bandwidth by priority:
    [0] 0bps         [1] 0bps         [2] 0bps         [3] 8Gbps
    [4] 0bps         [5] 0bps         [6] 0bps         [7] 1Gbps

user@PE1> show mpls lsp name HIGH-PRIORITY extensive | match priority
    Priorities: 0 0

## Priority 0 LSP can't find bandwidth because priority 3 used it all
```

Cause: Lower priority reservations blocking higher

Solution: This shouldn't happen with proper RSVP implementation. Check for configuration issues:

```
## Verify setup priority allows preemption
[edit protocols mpls label-switched-path HIGH-PRIORITY]
user@PE1# set priority 0 7  ## Setup 0, Hold 7
## This allows establishing at high priority but holding at low
```

Advanced Bandwidth Troubleshooting

Monitor Real-Time Bandwidth Usage

```
## Enable LSP statistics
[edit protocols mpls]
user@PE1# set statistics file mpls-stats
user@PE1# set statistics interval 300
user@PE1# set label-switched-path ALL-LSPS statistics

user@PE1# commit

## View statistics
user@PE1> show mpls lsp statistics
LSP Name         To              Packets            Bytes  Pkt Rate  Byte Rate
GOLD-SERVICE     192.168.1.4    12345678      15432098765       234    2916354
SILVER-SERVICE   192.168.1.4     8765432       8765432109       187    1876543
```

Bandwidth Accounting Verification

```
## Check TED bandwidth tracking
user@PE1> show ted database extensive local
NodeID: 192.168.1.1
  Protocol: OSPF(0.0.0.0)
    To: 192.168.1.2, Local: 10.0.1.1, Remote: 10.0.1.2
      Metric: 10, TE Metric: 10, IGP metric: 10
      Static BW: 10Gbps, Reservable BW: 9Gbps
      Available BW [priority] bps:
        [0] 9Gbps      [1] 7Gbps      [2] 7Gbps      [3] 5.5Gbps
        [4] 5.5Gbps    [5] 5.5Gbps    [6] 5.5Gbps    [7] 5.5Gbps
      Reserved BW [priority] bps:
        [0] 0bps       [1] 2Gbps      [2] 0bps       [3] 1.5Gbps
        [4] 0bps       [5] 0bps       [6] 0bps       [7] 0bps

## Verify: Reserved + Available = Reservable (per priority)
## Priority 1: 2G reserved + 7G available = 9G reservable ✓
```

Testing Bandwidth Preemption

```
## Create test to verify preemption works
[edit protocols mpls]
user@PE1# set label-switched-path TEST-LOW to 192.168.1.4 bandwidth 8g priority 7 7
user@PE1# commit

user@PE1> show mpls lsp
TEST-LOW        192.168.1.4     192.168.1.1     Up

## Now create high-priority LSP that needs same bandwidth
[edit protocols mpls]
user@PE1# set label-switched-path TEST-HIGH to 192.168.1.4 bandwidth 8g priority 0 0
user@PE1# commit

## Check results
user@PE1> show mpls lsp
TEST-HIGH       192.168.1.4     192.168.1.1     Up
TEST-LOW        192.168.1.4     192.168.1.1     Dn  ## Preempted!

user@PE1> show mpls lsp name TEST-LOW extensive | match "Last error"
    Last error: Preempted by higher priority LSP
```

Bandwidth reservation transforms MPLS from best-effort to guaranteed service delivery. Next, we'll explore how LSP priorities interact with these reservations to create a sophisticated traffic management system.

Module 8: RSVP-LSP Priorities

Part 1: The Conceptual Lecture (The Why)

The Problem with Simple Bandwidth Reservation

Imagine an emergency room where patients are served first-come, first-served. A patient with a minor cut might occupy a bed while someone with a heart attack waits. This is the problem with bandwidth reservation without priorities - critical traffic can be blocked by less important traffic that arrived first.

```
The Bandwidth Exhaustion Problem:

Time 09:00: 10Gbps link available
Time 09:15: Bulk backup LSP reserves 8Gbps (Priority 7)
Time 09:30: VoIP LSP needs 2Gbps (should be Priority 0)
Result: VoIP LSP fails! No bandwidth available.

Emergency services blocked by routine traffic!
```

Understanding RSVP Priority Levels

RSVP uses two priority values for each LSP:

```
Setup Priority: Used when establishing the LSP
Hold Priority: Used to maintain the LSP

Format: [Setup Priority, Hold Priority]
Range: 0-7 (0 = highest, 7 = lowest)

Think of it as:
Setup Priority = "How important is it to get established?"
Hold Priority = "How important is it to keep running?"
```

The Preemption Model

Higher priority LSPs can preempt (kick out) lower priority ones:

```
Preemption Rules:
1. Setup Priority must be ≤ Hold Priority
   (Can't establish weak but hold strong)

2. To preempt: New Setup Priority < Existing Hold Priority

Example:
Existing LSP: [5,5] (Setup=5, Hold=5)
New LSP: [3,3] wants bandwidth
Result: 3 < 5, so new LSP preempts existing

But if existing was [5,2] (Setup=5, Hold=2):
3 > 2, so NO preemption!
```

Default Priority Behavior

By default, all LSPs use [7,7] - the lowest priority:

```
Default [7,7] means:
- Polite setup (won't preempt anyone)
- Weak hold (anyone can preempt)
- Pure first-come, first-served
- No service differentiation

Like having all hospital patients as "routine" priority!
```

Priority Design Patterns

Common priority schemes in production networks:

```
1. Simple Three-Tier Model:
   Voice/Video:     [0,0] - Never preempted
   Business Data:   [3,3] - Medium priority
   Bulk/Backup:     [7,7] - Best effort

2. Setup vs Hold Differentiation:
   Important LSP:   [1,3] - Aggressive setup, moderate hold
   Normal LSP:      [4,4] - Balanced
   Scavenger LSP:   [7,0] - Weak setup, strong hold (weird!)

3. Cascading Priorities:
   Platinum:        [0,0]
   Gold:           [1,1]
   Silver:         [2,2]
   Bronze:         [3,3]
   Best Effort:    [7,7]
```

Hard vs Soft Preemption

Traditional preemption is harsh - immediate teardown:

```
Hard Preemption (Traditional):
1. New high-priority LSP needs bandwidth
2. IMMEDIATELY tear down low-priority LSP
3. Traffic blackholes until reroute
4. Service disruption!

Soft Preemption (Graceful):
1. New high-priority LSP needs bandwidth
2. Notify low-priority LSP: "Please move"
3. Low-priority LSP finds alternate path
4. Gracefully moves traffic
5. Then releases bandwidth
6. No service disruption!
```

Priority and Bandwidth Interaction

Priorities create bandwidth "views" per level:

```
10Gbps Link Status:
Priority 0 reserved: 2Gbps
Priority 3 reserved: 3Gbps
Priority 7 reserved: 4Gbps

Available bandwidth view:
Priority 0 sees: 10Gbps (can preempt everyone)
Priority 3 sees: 6Gbps (can preempt priority 7)
Priority 7 sees: 1Gbps (can't preempt anyone)
```

Part 2: The Junos CLI Masterclass (The How)

Basic Priority Configuration

Configure LSP with specific priorities:

```
## Simple priority assignment
[edit protocols mpls]
user@PE1# set label-switched-path VOICE-LSP to 192.168.1.4
```

```
user@PE1# set label-switched-path VOICE-LSP priority 0 0
user@PE1# set label-switched-path VOICE-LSP bandwidth 2g

## Different setup vs hold priorities
[edit protocols mpls]
user@PE1# set label-switched-path BUSINESS-LSP to 192.168.1.4
user@PE1# set label-switched-path BUSINESS-LSP priority 2 3
user@PE1# set label-switched-path BUSINESS-LSP bandwidth 3g

## Default (if not specified) is 7 7
user@PE1# set label-switched-path BULK-LSP to 192.168.1.4
user@PE1# set label-switched-path BULK-LSP bandwidth 5g
## Implicitly priority 7 7
```

Configuring Soft Preemption

Enable graceful preemption to avoid service disruption:

```
## Enable soft preemption globally
[edit protocols rsvp]
user@PE1# set preemption soft-preemption

## Configure soft preemption cleanup timer
user@PE1# set preemption soft-preemption cleanup-timer 180

## Per-LSP soft preemption control
[edit protocols mpls label-switched-path BUSINESS-LSP]
user@PE1# set soft-preemption

## Disable soft preemption for specific LSP
[edit protocols mpls label-switched-path SCAVENGER-LSP]
user@PE1# set no-soft-preemption
```

Priority-Based Bandwidth Pools

Configure bandwidth pools per priority:

```
## Reserve bandwidth for high-priority traffic
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set bandwidth 10g
user@PE1# set priority 0 reserved-bandwidth 2g
user@PE1# set priority 1 reserved-bandwidth 2g
user@PE1# set priority 2 reserved-bandwidth 1g

## This ensures:
## - Priority 0 always has 2G available
## - Priority 1 always has 2G available
## - Priority 2 always has 1G available
## - Remaining 5G shared by all priorities
```

Complete Priority Design Example

Implement a comprehensive priority scheme:

```
## Define LSPs with business-aligned priorities

## Real-time Services (Voice/Video)
set protocols mpls label-switched-path VOICE {
    to 192.168.1.4;
    priority 0 0;
    bandwidth 1g;
    admin-group include-any LOW-LATENCY;
    description "VoIP and video conferencing";
}

## Mission-Critical Data
set protocols mpls label-switched-path CRITICAL-DATA {
    to 192.168.1.4;
    priority 1 1;
    bandwidth 2g;
    admin-group include-any SECURE;
    description "Financial transactions and critical apps";
}

## Standard Business Applications
set protocols mpls label-switched-path BUSINESS-APPS {
    to 192.168.1.4;
    priority 3 3;
    bandwidth 3g;
    adaptive;
    description "Email, web, standard applications";
}

## Bulk Data Transfer
set protocols mpls label-switched-path BULK-TRANSFER {
    to 192.168.1.4;
    priority 5 6;  ## Aggressive setup, weak hold
    bandwidth 4g;
    soft-preemption;
    description "Backups and bulk data replication";
}

## Scavenger Traffic
set protocols mpls label-switched-path SCAVENGER {
    to 192.168.1.4;
    priority 7 7;
    bandwidth 1g;
    soft-preemption;
    description "P2P and non-business traffic";
}
```

Advanced Priority Features

Configure sophisticated priority behaviors:

```
## Preemption with make-before-break
[edit protocols mpls label-switched-path ADAPTIVE-LSP]
user@PE1# set to 192.168.1.4
user@PE1# set priority 3 3
user@PE1# set bandwidth 2g
user@PE1# set adaptive
user@PE1# set optimize-timer 300

## Priority inheritance for backup paths
[edit protocols mpls label-switched-path PROTECTED-LSP]
user@PE1# set to 192.168.1.4
user@PE1# set priority 1 1
user@PE1# set primary MAIN-PATH
user@PE1# set secondary BACKUP-PATH
user@PE1# set secondary BACKUP-PATH priority 2 2  ## Lower than primary

## Priority-aware load balancing
[edit protocols mpls]
user@PE1# set label-switched-path VOICE-LB-1 to 192.168.1.4 priority 0 0
user@PE1# set label-switched-path VOICE-LB-2 to 192.168.1.4 priority 0 0
user@PE1# set label-switched-path VOICE-LB-3 to 192.168.1.4 priority 0 0
```

Soft Preemption Timer Configurations

Fine-tune soft preemption behavior:

```
## Global soft preemption settings
[edit protocols rsvp]
user@PE1# set preemption soft-preemption
user@PE1# set preemption soft-preemption cleanup-timer 60

## Aggressive preemption for emergencies
```

```
[edit protocols mpls label-switched-path EMERGENCY]
user@PE1# set to 192.168.1.4
user@PE1# set priority 0 0
user@PE1# set no-soft-preemption  ## Immediate hard preemption

## Extended grace period for bulk transfers
[edit protocols mpls label-switched-path BULK-GRACEFUL]
user@PE1# set to 192.168.1.4
user@PE1# set priority 6 7
user@PE1# set soft-preemption
user@PE1# set soft-preemption-cleanup-timer 300  ## 5 minutes grace
```

Priority-Based Route Preference

Control route installation based on priority:

```
## Install only high-priority LSPs in forwarding
[edit protocols mpls]
user@PE1# set prefer-standby

## Configure LSP preference based on priority
[edit protocols mpls label-switched-path VOICE]
user@PE1# set preference 5  ## Lower preference = more preferred

[edit protocols mpls label-switched-path BULK]
user@PE1# set preference 10
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Priority Verification Commands

1. View LSP Priorities

```
user@PE1> show mpls lsp detail
Ingress LSP: 5 sessions

192.168.1.4
  From: 192.168.1.1, State: Up, ActiveRoute: 1, LSPname: VOICE
  ActivePath:  (primary)
  Priorities: 0 0
  Bandwidth: 1Gbps
  ...

192.168.1.4
  From: 192.168.1.1, State: Up, ActiveRoute: 1, LSPname: BUSINESS-APPS
  ActivePath:  (primary)
  Priorities: 3 3
  Bandwidth: 3Gbps
  ...
```

2. Check Priority-Based Bandwidth Availability

```
user@PE1> show rsvp interface detail
ge-0/0/1.0
  Index: 329, State: Up
  ActiveResv: 4, Subscription: 100%
  Static BW: 10Gbps, Reservable BW: 10Gbps
  Available bandwidth:
    [0] 10Gbps       [1] 9Gbps        [2] 9Gbps        [3] 6Gbps
    [4] 6Gbps        [5] 6Gbps        [6] 2Gbps        [7] 1Gbps
  Reserved bandwidth by priority:
    [0] 1Gbps        [1] 0bps         [2] 3Gbps        [3] 0bps
    [4] 4Gbps        [5] 0bps         [6] 1Gbps        [7] 0bps

## Analysis:
## Priority 0: 1G reserved, sees full 10G available (can preempt all)
## Priority 3: 0 reserved, sees 6G available (can preempt pri 4-7)
## Priority 7: 0 reserved, sees 1G available (can't preempt anyone)
```

3. Monitor Preemption Events

```
user@PE1> show mpls lsp preempted
Preempted LSPs: 2 sessions

LSPname         From            To              Pri  Bandwidth  Time     Reason
BULK-TRANSFER   192.168.1.1     192.168.1.4     5/6  4Gbps     00:02:15 Hard-preempted
SCAVENGER       192.168.1.1     192.168.1.4     7/7  1Gbps     00:05:43 Soft-preempted

user@PE1> show log messages | match "preempt"
Nov 25 10:15:43 PE1 rpd[1423]: MPLS LSP BULK-TRANSFER preempted by CRITICAL-DATA
Nov 25 10:18:21 PE1 rpd[1423]: MPLS LSP SCAVENGER soft-preemption timer started
Nov 25 10:20:21 PE1 rpd[1423]: MPLS LSP SCAVENGER gracefully moved to alternate path
```

Common Troubleshooting Scenarios

Scenario 1: Unexpected Preemption

Symptom: High-priority LSP gets preempted

Diagnostic Commands:

```
user@PE1> show mpls lsp name VOICE extensive | match "State|Priorities|Last error"
  From: 192.168.1.1, State: Dn, ActiveRoute: 0, LSPname: VOICE
    Priorities: 0 0
    Last error: Path preempted

## This shouldn't happen with priority 0 0!

user@PE1> show configuration protocols mpls label-switched-path VOICE | display set
set protocols mpls label-switched-path VOICE to 192.168.1.4
set protocols mpls label-switched-path VOICE priority 0 0

## Check for typos or inheritance issues
```

Cause: Configuration error or software bug

Solution:

```
## Verify configuration
[edit protocols mpls label-switched-path VOICE]
user@PE1# show | display inheritance

## Force LSP re-establishment
user@PE1> clear mpls lsp name VOICE
```

Scenario 2: Soft Preemption Not Working

Symptom: LSPs hard-preempted despite soft-preemption config

Diagnostic Commands:

```
user@PE1> show rsvp version
Resource ReSerVation Protocol, version 1

user@PE1> show rsvp neighbor detail | match "Soft"
  Soft-preemption: Supported

user@PE1> show configuration protocols rsvp | display set | match soft
set protocols rsvp preemption soft-preemption

user@PE1> show mpls lsp name BULK-TRANSFER extensive | match "error"
    Last error: Hard preempted - neighbor doesn't support soft-preemption
```

Cause: Not all routers support soft-preemption

Solution:

```
## Enable on all routers in path
[edit protocols rsvp]
user@router# set preemption soft-preemption
user@router# commit
```

Scenario 3: Priority Inversions

Symptom: Lower priority LSP blocking higher priority

Diagnostic Commands:

```
user@PE1> show mpls lsp
Ingress LSP: 2 sessions
To              From            State Rt P     ActivePath       LSPname
192.168.1.4     192.168.1.1     Up     1 *                     LOW-PRIORITY
192.168.1.4     192.168.1.1     Dn     0                       HIGH-PRIORITY

user@PE1> show mpls lsp extensive name LOW-PRIORITY | match "Priorities|Bandwidth"
    Priorities: 5 3   ## Setup 5, Hold 3
    Bandwidth: 8Gbps

user@PE1> show mpls lsp extensive name HIGH-PRIORITY | match "Priorities|Bandwidth"
    Priorities: 4 4   ## Setup 4, Hold 4
    Bandwidth: 3Gbps

## Setup priority 4 cannot preempt hold priority 3!
```

Cause: Hold priority stronger than setup priority

Solution:

```
## Fix priority configuration
[edit protocols mpls label-switched-path HIGH-PRIORITY]
user@PE1# set priority 2 2  ## Now can preempt hold 3
user@PE1# commit
```

Scenario 4: Bandwidth Fragmentation

Symptom: Enough total bandwidth but LSP won't establish

Diagnostic Commands:

```
user@PE1> show rsvp interface detail | find "Available"
  Available bandwidth:
    [0] 10Gbps       [1] 3Gbps        [2] 3Gbps        [3] 2Gbps
    [4] 2Gbps        [5] 2Gbps        [6] 1Gbps        [7] 500Mbps

## Need 4Gbps at priority 3, but only 2Gbps available

user@PE1> show rsvp session detail | match "Style|band"
  Resv style: 1 FF, Label in: -, Label out: 100234
  Tspec: rate 1Gbps size 0bps peak Infbps m 20 M 1500
  ...
  Resv style: 1 FF, Label in: -, Label out: 100567
  Tspec: rate 2Gbps size 0bps peak Infbps m 20 M 1500
  ...
  Resv style: 1 FF, Label in: -, Label out: 100890
  Tspec: rate 5Gbps size 0bps peak Infbps m 20 M 1500

## Multiple small reservations fragmenting bandwidth
```

Solution: Implement bandwidth defragmentation:

```
## Enable optimize timer to consolidate bandwidth
[edit protocols mpls]
user@PE1# set optimize-timer 3600
user@PE1# set label-switched-path all optimize-aggressive

## Or manually clear and re-signal LSPs
user@PE1> clear mpls lsp all
```

Advanced Priority Troubleshooting

Trace Preemption Decisions

```
[edit protocols rsvp]
user@PE1# set traceoptions file rsvp-preempt
user@PE1# set traceoptions flag preemption detail
user@PE1# commit

user@PE1> show log rsvp-preempt
Nov 25 11:30:45 Preemption check for new LSP CRITICAL-DATA
Nov 25 11:30:45   Need 3Gbps at priority 1/1
Nov 25 11:30:45   Checking LSP BULK-TRANSFER: 4Gbps at 5/6
Nov 25 11:30:45   Setup pri 1 < Hold pri 6: Can preempt
Nov 25 11:30:45   Preempting BULK-TRANSFER to free 4Gbps
Nov 25 11:30:45   Soft-preemption notification sent
```

Monitor Priority Distribution

```
user@PE1> show mpls lsp statistics priority
Priority  Active LSPs  Bandwidth Reserved  Percentage
0         2           2.5 Gbps            15%
1         3           4.2 Gbps            25%
2         1           1.0 Gbps            6%
3         5           6.8 Gbps            41%
4         0           0 bps               0%
5         2           1.5 Gbps            9%
6         0           0 bps               0%
7         4           600 Mbps            4%
Total:    17          16.6 Gbps           100%
```

Test Preemption Behavior

```
## Create test scenario
[edit protocols mpls]
user@PE1# set label-switched-path TEST-VICTIM to 192.168.1.4 priority 6 6 bandwidth 5g
user@PE1# commit

user@PE1> show mpls lsp name TEST-VICTIM
TEST-VICTIM     192.168.1.4     192.168.1.1     Up

## Now create preemptor
[edit protocols mpls]
user@PE1# set label-switched-path TEST-PREEMPTOR to 192.168.1.4 priority 2 2 bandwidth 8g
user@PE1# commit

## Check results
user@PE1> show mpls lsp
TEST-PREEMPTOR  192.168.1.4     192.168.1.1     Up
TEST-VICTIM     192.168.1.4     192.168.1.1     Dn    ## Preempted!
```

LSP priorities transform MPLS from a simple first-come-first-served system into a sophisticated QoS platform. Combined with bandwidth reservation, priorities ensure critical services always get the resources they need.

Module 9: RSVP-Constrained Shortest Path First, and Admin Groups

Part 1: The Conceptual Lecture (The Why)

Understanding CSPF: Beyond Simple Shortest Path

Traditional routing algorithms like Dijkstra's SPF are like a GPS that only considers distance. CSPF (Constrained Shortest Path First) is like an advanced GPS that considers distance, traffic conditions, toll roads, vehicle restrictions, and your preferences all at once.

```
Traditional SPF:
"What's the shortest path from A to B?"

CSPF:
"What's the shortest path from A to B that:
 - Has enough lanes (bandwidth) for my truck
 - Avoids toll roads (expensive links)
 - Doesn't use bridges with weight limits (constraints)
 - Prefers highways over side streets (admin groups)"
```

The CSPF Algorithm Process

CSPF uses a modified Dijkstra's algorithm with pre-filtering:

```
CSPF Computation Steps:

1. Start with full network topology (TED)
2. Prune links that don't meet constraints:
   - Remove links without enough bandwidth
   - Remove links with excluded admin groups
   - Remove links missing required admin groups
3. Run Dijkstra on remaining topology
4. Find shortest path that meets ALL constraints

If no path exists after pruning: LSP fails!
```

Admin Groups: The Power of Link Coloring

Admin groups are arbitrary tags (colors) you assign to links. Think of them as characteristics:

```
Real-World Analogy:
- Red = Toll roads
- Blue = Highways
- Green = Scenic routes
- Gold = Express lanes
- Black = Construction zones

Network Link Coloring:
- Red = Expensive satellite links
- Blue = Cheap fiber
- Green = Low latency
- Gold = Encrypted/secure
- Black = Maintenance window

Each link can have multiple colors (up to 32)
```

Admin Group Matching Logic

Three ways to match admin groups:

```
1. Include-Any (OR logic):
   "Must have at least one of these colors"
   Include-Any: Red Blue = Link must be Red OR Blue OR both

2. Include-All (AND logic):
   "Must have all of these colors"
   Include-All: Gold Green = Link must be Gold AND Green

3. Exclude (NOT logic):
   "Must not have any of these colors"
   Exclude: Black Red = Link cannot be Black OR Red

Combining rules:
Include-Any: Blue Green
Include-All: Gold
Exclude: Red
= "Link must be (Blue OR Green) AND Gold AND NOT Red"
```

CSPF Tie-Breaking

When multiple equal-cost paths exist after constraint filtering:

```
CSPF Tie-Breakers (in order):

1. Fewest hops (default)
2. Random (load balancing)
3. Least-fill (most available bandwidth)
4. Most-fill (pack links efficiently)

Example with two paths:
Path A: 3 hops, 8Gbps available
Path B: 3 hops, 2Gbps available

Least-fill chooses Path A (more headroom)
Most-fill chooses Path B (better packing)
```

The Computational Cost of CSPF

CSPF is more CPU-intensive than regular SPF:

```
Computational Complexity:

Regular SPF: O(n²) or O(n log n) with optimization
CSPF: O(n²) + constraint checking per link

With 1000 nodes and 5000 links:
- SPF: ~1 million operations
- CSPF: ~1 million operations + 5000 × constraint checks

Impact:
- More CPU usage on ingress PE
- Longer convergence times
- Scaling considerations for large networks
```

When to Use CSPF vs No-CSPF

Choose your path computation method wisely:

```
Use CSPF when:
- Traffic engineering is required
- Different service levels needed
- Network has diverse link types
- Bandwidth guarantees required
- Complex routing policies

Use No-CSPF when:
- Simple follow-the-IGP is enough
- All links are homogeneous
- CPU resources are limited
- Fast convergence is critical
- Testing/troubleshooting
```

Part 2: The Junos CLI Masterclass (The How)

Configuring Admin Groups

First, define admin groups (up to 32):

```
## Define admin groups with meaningful names
[edit protocols mpls]
user@PE1# set admin-groups GOLD 0
user@PE1# set admin-groups SILVER 1
user@PE1# set admin-groups BRONZE 2
user@PE1# set admin-groups SECURE 3
user@PE1# set admin-groups PUBLIC 4
user@PE1# set admin-groups LOW-LATENCY 5
user@PE1# set admin-groups HIGH-BANDWIDTH 6
user@PE1# set admin-groups EXPENSIVE 7
user@PE1# set admin-groups SATELLITE 8
user@PE1# set admin-groups FIBER 9
user@PE1# set admin-groups MICROWAVE 10

## Apply admin groups to interfaces
[edit protocols mpls interface ge-0/0/1.0]
user@PE1# set admin-group FIBER
user@PE1# set admin-group HIGH-BANDWIDTH
user@PE1# set admin-group GOLD

[edit protocols mpls interface ge-0/0/2.0]
user@PE1# set admin-group SATELLITE
user@PE1# set admin-group EXPENSIVE
user@PE1# set admin-group SILVER
```

Creating LSPs with Admin Group Constraints

Configure LSPs that use admin groups:

```
## LSP with include-any (OR logic)
[edit protocols mpls]
user@PE1# set label-switched-path FLEXIBLE-PATH to 192.168.1.4
user@PE1# set label-switched-path FLEXIBLE-PATH admin-group include-any FIBER MICROWAVE
## Can use links with FIBER OR MICROWAVE (or both)

## LSP with include-all (AND logic)
[edit protocols mpls]
user@PE1# set label-switched-path PREMIUM-PATH to 192.168.1.4
user@PE1# set label-switched-path PREMIUM-PATH admin-group include-all GOLD SECURE LOW-LATENCY
## Must use links with GOLD AND SECURE AND LOW-LATENCY

## LSP with exclude
[edit protocols mpls]
user@PE1# set label-switched-path AVOID-EXPENSIVE to 192.168.1.4
user@PE1# set label-switched-path AVOID-EXPENSIVE admin-group exclude EXPENSIVE SATELLITE
## Cannot use links with EXPENSIVE OR SATELLITE

## Complex combination
[edit protocols mpls]
user@PE1# set label-switched-path COMPLEX-PATH to 192.168.1.4
user@PE1# set label-switched-path COMPLEX-PATH admin-group include-any FIBER MICROWAVE
user@PE1# set label-switched-path COMPLEX-PATH admin-group include-all GOLD
user@PE1# set label-switched-path COMPLEX-PATH admin-group exclude EXPENSIVE
## Must be (FIBER OR MICROWAVE) AND GOLD AND NOT EXPENSIVE
```

Configuring CSPF Tie-Breaking

Control how CSPF selects among equal paths:

```
## Random selection (load balancing)
[edit protocols mpls label-switched-path LOAD-BALANCED]
user@PE1# set to 192.168.1.4
user@PE1# set random

## Least-fill (prefer emptier links)
[edit protocols mpls label-switched-path LEAST-FILL-PATH]
user@PE1# set to 192.168.1.4
user@PE1# set least-fill

## Most-fill (pack links)
[edit protocols mpls label-switched-path PACK-LINKS]
user@PE1# set to 192.168.1.4
user@PE1# set most-fill

## Explicit tie-breaking order
[edit protocols mpls]
user@PE1# set cspf-tie-breaking least-fill
```

Advanced CSPF Configuration

Configure sophisticated path computation:

```
## Disable CSPF for simple LSP
[edit protocols mpls label-switched-path SIMPLE-PATH]
user@PE1# set to 192.168.1.4
user@PE1# set no-cspf

## CSPF with bandwidth and admin constraints
[edit protocols mpls label-switched-path VOICE-PATH]
user@PE1# set to 192.168.1.4
user@PE1# set bandwidth 500m
user@PE1# set admin-group include-any LOW-LATENCY
user@PE1# set admin-group exclude SATELLITE
user@PE1# set priority 0 0

## Use admin groups with explicit paths
[edit protocols mpls]
user@PE1# set path SECURE-ROUTE 192.168.1.2 loose
user@PE1# set path SECURE-ROUTE 192.168.1.3 loose
user@PE1# set label-switched-path CONTROLLED-PATH to 192.168.1.4
user@PE1# set label-switched-path CONTROLLED-PATH primary SECURE-ROUTE
user@PE1# set label-switched-path CONTROLLED-PATH admin-group include-all SECURE
```

Complete Multi-Service CSPF Example

Design a network with different service classes:

```
## Admin group definitions for service differentiation
set protocols mpls admin-groups VOICE 0
set protocols mpls admin-groups VIDEO 1
set protocols mpls admin-groups BUSINESS 2
set protocols mpls admin-groups BULK 3
set protocols mpls admin-groups ENCRYPTED 4
set protocols mpls admin-groups UNENCRYPTED 5
set protocols mpls admin-groups LOW-LATENCY 6
set protocols mpls admin-groups HIGH-LATENCY 7
set protocols mpls admin-groups EXPENSIVE 8
set protocols mpls admin-groups ECONOMICAL 9

## Apply to interfaces based on characteristics
## Low-latency fiber link
set protocols mpls interface ge-0/0/1.0 admin-group VOICE
set protocols mpls interface ge-0/0/1.0 admin-group VIDEO
set protocols mpls interface ge-0/0/1.0 admin-group LOW-LATENCY
set protocols mpls interface ge-0/0/1.0 admin-group ENCRYPTED
```

```
## High-bandwidth but higher latency link
set protocols mpls interface ge-0/0/2.0 admin-group BUSINESS
set protocols mpls interface ge-0/0/2.0 admin-group BULK
set protocols mpls interface ge-0/0/2.0 admin-group ECONOMICAL

## Expensive satellite backup
set protocols mpls interface ge-0/0/3.0 admin-group EXPENSIVE
set protocols mpls interface ge-0/0/3.0 admin-group HIGH-LATENCY

## Service-specific LSPs
## Voice service - strict requirements
set protocols mpls label-switched-path VOICE-SERVICE {
    to 192.168.1.4;
    bandwidth 100m;
    priority 0 0;
    admin-group include-all VOICE LOW-LATENCY;
    admin-group exclude HIGH-LATENCY EXPENSIVE;
}

## Video streaming - needs bandwidth
set protocols mpls label-switched-path VIDEO-SERVICE {
    to 192.168.1.4;
    bandwidth 2g;
    priority 1 1;
    admin-group include-any VOICE VIDEO;
    admin-group include-all LOW-LATENCY;
    least-fill;  ## Prefer links with more headroom
}

## Business data - balanced requirements
set protocols mpls label-switched-path BUSINESS-DATA {
    to 192.168.1.4;
    bandwidth 1g;
    priority 3 3;
    admin-group include-any BUSINESS ENCRYPTED;
    admin-group exclude EXPENSIVE;
}

## Bulk transfer - cost sensitive
set protocols mpls label-switched-path BULK-BACKUP {
    to 192.168.1.4;
    bandwidth 5g;
    priority 6 6;
    admin-group include-any BULK ECONOMICAL;
    most-fill;  ## Pack links efficiently
}
```

CSPF Computation Control

Fine-tune CSPF behavior:

```
## Global CSPF settings
[edit protocols mpls]
user@PE1# set cspf-backoff-time initial 15
user@PE1# set cspf-backoff-time maximum 300
user@PE1# set cspf-backoff-time multiplier 2

## Disable CSPF during maintenance
[edit protocols mpls]
user@PE1# set no-cspf-accounting

## Per-LSP CSPF optimization
[edit protocols mpls label-switched-path OPTIMIZED]
user@PE1# set to 192.168.1.4
user@PE1# set optimize-timer 600
user@PE1# set optimize-aggressive
```

Part 3: Verification & Troubleshooting (The What-If)

Essential CSPF Verification Commands

1. View Admin Group Configuration

```
user@PE1> show mpls admin-groups
Group                    Bit    Hex
VOICE                      0    0x1
VIDEO                      1    0x2
BUSINESS                   2    0x4
BULK                       3    0x8
ENCRYPTED                  4    0x10
UNENCRYPTED               5    0x20
LOW-LATENCY               6    0x40
HIGH-LATENCY              7    0x80
EXPENSIVE                 8    0x100
ECONOMICAL                9    0x200

user@PE1> show mpls interface detail
Interface: ge-0/0/1.0
  State: Up
  Admin groups: VOICE VIDEO LOW-LATENCY ENCRYPTED
  Admin group membership: 0x53 (bits 0 1 4 6)
  Bandwidth: 10Gbps
```

2. Check CSPF Computation Results

```
user@PE1> show mpls lsp name VOICE-SERVICE extensive
Ingress LSP: 1 sessions

192.168.1.4
  From: 192.168.1.1, State: Up, ActiveRoute: 0, LSPname: VOICE-SERVICE
  ActivePath:  (primary)
  LSPtype: Static Configured
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  *Primary                    State: Up
    Priorities: 0 0
    Bandwidth: 100Mbps
    Include All: VOICE LOW-LATENCY
    Exclude: HIGH-LATENCY EXPENSIVE
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 30)
      10.0.1.2 S 10.0.2.2 S 10.0.3.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
      10.0.1.2 10.0.2.2 10.0.3.2
    ...
   25 Dec 1 10:15:23.456 CSPF: computation result accepted  10.0.1.2 10.0.2.2 10.0.3.2
   24 Dec 1 10:15:23.345 CSPF: Start computation: bandwidth 100Mbps, include all 0x41 exclude 0x180
   23 Dec 1 10:15:23.234 CSPF: Link 192.168.1.1->192.168.1.2 admin 0x53 matches constraints
```

3. Test Path Computation

```
user@PE1> show ted path source 192.168.1.1 destination 192.168.1.4
Computing path from 192.168.1.1 to 192.168.1.4, metric any, bandwidth any, priority any
Path 1: (Metric 30)
  192.168.1.1 -> 10.0.1.2 -> 10.0.2.2 -> 10.0.3.2 -> 192.168.1.4

user@PE1> show ted path source 192.168.1.1 destination 192.168.1.4 admin-group-include-all 0x41
Computing path from 192.168.1.1 to 192.168.1.4, include-all 0x41 (VOICE LOW-LATENCY)
Path 1: (Metric 30)
  192.168.1.1 -> 10.0.1.2 -> 10.0.2.2 -> 10.0.3.2 -> 192.168.1.4

user@PE1> show ted path source 192.168.1.1 destination 192.168.1.4 admin-group-exclude 0x100
Computing path from 192.168.1.1 to 192.168.1.4, exclude 0x100 (EXPENSIVE)
Path 1: (Metric 30)
  192.168.1.1 -> 10.0.1.2 -> 10.0.2.2 -> 10.0.3.2 -> 192.168.1.4
```

Common Troubleshooting Scenarios

Scenario 1: CSPF Cannot Find Valid Path

Symptom: LSP down due to CSPF constraints

Diagnostic Commands:

```
user@PE1> show mpls lsp name VOICE-SERVICE extensive | match "error|Error"
    State: Dn
    Last error: CSPF failed: no route toward 192.168.1.4 matching constraints[5 times]

user@PE1> show mpls lsp name VOICE-SERVICE extensive | match "Include|Exclude"
    Include All: VOICE LOW-LATENCY
    Exclude: HIGH-LATENCY EXPENSIVE

## Check which links have required admin groups
user@PE1> show ted database extensive | match "groups:|VOICE.*LOW-LATENCY"
      Admin groups: VIDEO BUSINESS ECONOMICAL
      Admin groups: EXPENSIVE HIGH-LATENCY
      Admin groups: VOICE VIDEO LOW-LATENCY ENCRYPTED  ## Only this one matches!
```

Cause: Too restrictive constraints

Solution:

```
## Option 1: Relax constraints
[edit protocols mpls label-switched-path VOICE-SERVICE]
user@PE1# delete admin-group include-all
user@PE1# set admin-group include-any VOICE LOW-LATENCY
user@PE1# commit

## Option 2: Add admin groups to more links
[edit protocols mpls interface ge-0/0/2.0]
user@PE1# set admin-group VOICE
user@PE1# set admin-group LOW-LATENCY
```

Scenario 2: Suboptimal Path Selection

Symptom: CSPF chooses longer path

Diagnostic Commands:

```
user@PE1> show mpls lsp name BUSINESS-DATA extensive | match "ERO|metric"
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 100)
      10.0.4.2 S 10.0.5.2 S 10.0.6.2 S 10.0.7.2 S 10.0.8.2 S

## But shorter path exists:
user@PE1> show ted path source 192.168.1.1 destination 192.168.1.4
Path 1: (Metric 30)
  192.168.1.1 -> 10.0.1.2 -> 10.0.2.2 -> 10.0.3.2 -> 192.168.1.4

## Why didn't it use the short path?
user@PE1> show ted database extensive 10.0.1.1 10.0.1.2 | match admin
      Admin groups: VOICE VIDEO LOW-LATENCY ENCRYPTED
      Admin group membership: 0x53

user@PE1> show mpls lsp name BUSINESS-DATA extensive | match Include
    Include Any: BUSINESS ENCRYPTED

## 0x53 = bits 0,1,4,6 but BUSINESS is bit 2 (0x4), ENCRYPTED is bit 4 (0x10)
## Link has ENCRYPTED but not BUSINESS!
```

Solution: Review admin group assignments

Scenario 3: CSPF Tie-Breaking Issues

Symptom: Load not balanced across equal paths

Diagnostic Commands:

```
user@PE1> show ted path source 192.168.1.1 destination 192.168.1.4 admin-group-include-any 0x4
Computing path from 192.168.1.1 to 192.168.1.4, include-any 0x4 (BUSINESS)
Path 1: (Metric 50)
  192.168.1.1 -> 10.0.1.2 -> 10.0.2.2 -> 192.168.1.4
Path 2: (Metric 50)
  192.168.1.1 -> 10.0.4.2 -> 10.0.5.2 -> 192.168.1.4

## Two equal paths but all LSPs use Path 1

user@PE1> show mpls lsp | match "BUSINESS|Up.*10.0.1.2"
BUSINESS-APP-1  192.168.1.4     Up     0 *     10.0.1.2
BUSINESS-APP-2  192.168.1.4     Up     0 *     10.0.1.2
BUSINESS-APP-3  192.168.1.4     Up     0 *     10.0.1.2
```

Solution: Enable random tie-breaking:

```
[edit protocols mpls]
user@PE1# set label-switched-path BUSINESS-APP-1 random
user@PE1# set label-switched-path BUSINESS-APP-2 random
user@PE1# set label-switched-path BUSINESS-APP-3 random
user@PE1# commit
```

Advanced CSPF Troubleshooting

Debug CSPF Computation

```
[edit protocols mpls]
user@PE1# set traceoptions file cspf-debug
user@PE1# set traceoptions flag cspf detail
user@PE1# set traceoptions flag cspf-link detail
user@PE1# commit

user@PE1> clear mpls lsp name VOICE-SERVICE
user@PE1> show log cspf-debug

Dec 1 11:45:23 CSPF: Start SPF from 192.168.1.1 to 192.168.1.4
Dec 1 11:45:23 CSPF: Constraints: BW 100000000, include-all 0x41, exclude 0x180
Dec 1 11:45:23 CSPF: Processing link 192.168.1.1->10.0.1.2
Dec 1 11:45:23 CSPF:   Admin 0x53, need include-all 0x41: PASS (0x53 & 0x41 = 0x41)
Dec 1 11:45:23 CSPF:   Admin 0x53, need exclude 0x180: PASS (0x53 & 0x180 = 0x0)
Dec 1 11:45:23 CSPF:   Bandwidth check: need 100M, have 9.5G: PASS
Dec 1 11:45:23 CSPF:   Link accepted, cost 10
Dec 1 11:45:23 CSPF: Processing link 192.168.1.1->10.0.4.2
Dec 1 11:45:23 CSPF:   Admin 0x308, need include-all 0x41: FAIL (0x308 & 0x41 = 0x0)
Dec 1 11:45:23 CSPF:   Link rejected due to admin-group constraint
```

Performance Impact Analysis

```
user@PE1> show task jobs | match CSPF
1234  CSPF          10:45:23      0.045      CSPF computation for LSP VOICE-SERVICE
1235  CSPF          10:45:24      0.023      CSPF computation for LSP VIDEO-SERVICE

user@PE1> show mpls cspf statistics
CSPF Statistics:
  Total computations: 1234
  Successful: 1198
  Failed: 36
  Average computation time: 12.3 ms
  Maximum computation time: 156.7 ms
  Current backoff time: 0 seconds
```

CSPF and admin groups transform RSVP-TE from simple path signaling into a powerful policy-based routing system. By combining bandwidth constraints, priorities, and admin groups, you can create sophisticated traffic engineering policies that automatically adapt to network conditions.

Module 10: RSVP-LSP Failures, Errors, and Session Maintenance

Part 1: The Conceptual Lecture (The Why)

Understanding LSP Failures

LSPs can fail for many reasons, like a highway that can be closed due to accidents, construction, or weather. Understanding these failure modes is crucial for building resilient networks:

```
Common LSP Failure Causes:

1. Physical Failures:
   - Link down (fiber cut)
   - Node failure (router crash)
   - Interface errors

2. Control Plane Failures:
   - RSVP process restart
   - Lost RSVP messages
   - Corrupted RSVP state

3. Resource Failures:
   - Insufficient bandwidth
   - Preemption by higher priority
   - Admin group changes

4. Configuration Issues:
   - Mismatched parameters
   - Authentication failures
   - Policy violations
```

RSVP Error Messages

RSVP uses two types of error messages to report problems:

```
PathErr (Path Error):
- Flows upstream (toward ingress)
- Reports downstream problems
- "I can't process your PATH message because..."
- Does NOT tear down the LSP

ResvErr (Reservation Error):
- Flows downstream (toward egress)
- Reports upstream problems
- "I can't honor your RESV message because..."
- Does NOT tear down the LSP

Common Error Codes:
- Admission Control Failure (bandwidth)
- Policy Control Failure (admin policy)
- No Route to Destination
- Routing Problem (loop detected)
```

LSP Teardown Messages

When an LSP needs to be removed, RSVP uses teardown messages:

```
PathTear:
- Sent by ingress to remove path state
- Flows downstream
- "I don't need this LSP anymore"
- Triggers immediate state removal
```

```
ResvTear:
- Sent by egress or error point
- Flows upstream
- "I'm releasing this reservation"
- Can be triggered by PathTear

Teardown Flow:
Ingress --PathTear--> Transit --PathTear--> Egress
Ingress <--ResvTear-- Transit <--ResvTear-- Egress
```

Evolution: Soft-State to Reliable RSVP

Original RSVP (Soft-State) was like a subscription that expires without renewal:

```
Original Soft-State RSVP (1990s):
- Send full PATH/RESV every 30-45 seconds
- No explicit acknowledgments
- Miss 3 refreshes = tear down state
- Simple but inefficient

Problems:
- High control plane overhead
- Slow failure detection (3 × 45s = 135s)
- Doesn't scale (1000 LSPs = 1000 refreshes/30s)
- Wasted bandwidth

Modern Reliable RSVP:
- Initial PATH/RESV exchange
- Explicit acknowledgments (ACK)
- Summary refreshes (just IDs)
- Hello protocol for failure detection
- Scales to 100,000+ LSPs
```

RSVP Message Reliability

Modern RSVP ensures reliable message delivery:

```
Reliability Mechanisms:

1. Message IDs:
   - Each message gets unique ID
   - Receiver acknowledges ID
   - Sender retransmits if no ACK

2. Summary Refresh:
   - Instead of full state: "I still have IDs 1,2,3,4,5"
   - Tiny message replaces huge refresh
   - Receiver: "ACK, I have those too"

3. Bundle Messages:
   - Combine multiple messages
   - Single ACK for entire bundle
   - Reduces message count

4. Refresh Reduction:
   - Only send changes
   - Not entire state repeatedly
```

Hello Protocol: Fast Failure Detection

RSVP Hello provides rapid neighbor failure detection:

```
Hello Protocol Operation:

Without Hello:
- Detect via missed refreshes
- 3 × 45 seconds = 135 seconds to detect

With Hello:
- Send Hello every 3 seconds
- Miss 3 Hellos = 9 seconds to detect
- 15× faster failure detection!

Hello Messages:
- REQUEST: "Are you there?"
- ACK: "Yes, I'm here"
- Lightweight (no LSP state)
- Per-neighbor, not per-LSP
```

Graceful Restart: Maintaining Forwarding During Control Plane Restart

Graceful Restart separates control and forwarding planes:

```
Traditional Restart:
1. RSVP process crashes
2. All RSVP state lost
3. All LSPs torn down
4. Traffic blackholed
5. LSPs re-signaled after restart
6. 30-60 seconds of downtime

Graceful Restart:
1. RSVP process crashes
2. Forwarding plane keeps working
3. Neighbors preserve state
4. RSVP restarts and recovers state
5. No traffic loss!
6. Hitless recovery
```

Part 2: The Junos CLI Masterclass (The How)

Configuring RSVP Refresh and Reliability

Configure modern RSVP reliability features:

```
## Enable refresh reduction (modern RSVP)
[edit protocols rsvp]
user@PE1# set refresh-reduction
user@PE1# set refresh-reduction reliable-delivery

## Configure refresh timers
[edit protocols rsvp]
user@PE1# set refresh-time 30
user@PE1# set keep-multiplier 3

## Summary refresh configuration
[edit protocols rsvp]
user@PE1# set refresh-reduction summary-refresh

## Bundle message support
[edit protocols rsvp]
user@PE1# set refresh-reduction bundle-message-max-size 4096
```

Hello Protocol Configuration

Enable fast failure detection:

```
## Basic Hello configuration
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set hello-interval 3
user@PE1# set hello-multiplier 3

## Aggressive hello for critical links
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set hello-interval 1
user@PE1# set hello-multiplier 3
## Detects failures in 3 seconds!

## BFD for even faster detection
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set oam bfd-liveness-detection minimum-interval 100
user@PE1# set oam bfd-liveness-detection multiplier 3
## Detects failures in 300ms!
```

Graceful Restart Configuration

Enable hitless RSVP restarts:

```
## Enable graceful restart
[edit protocols rsvp]
user@PE1# set graceful-restart
user@PE1# set graceful-restart restart-time 120
user@PE1# set graceful-restart recovery-time 180

## Per-interface graceful restart
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set graceful-restart

## Helper mode (assist neighbors)
[edit protocols rsvp]
user@PE1# set graceful-restart helper-mode
```

Error Handling Configuration

Configure how RSVP handles errors:

```
## Configure PathErr behavior
[edit protocols mpls label-switched-path LSP1]
user@PE1# set retry-timer 30
user@PE1# set retry-limit 5
user@PE1# set no-install-on-error

## Smart retry with exponential backoff
[edit protocols mpls label-switched-path LSP2]
user@PE1# set retry-timer 5
user@PE1# set retry-limit 10
user@PE1# set retry-timer-backoff
```

Complete Reliability Configuration Example

Build a resilient RSVP deployment:

```
## Global RSVP reliability settings
set protocols rsvp {
    refresh-reduction {
        reliable-delivery;
        summary-refresh;
        bundle-message-max-size 8192;
    }
    refresh-time 45;
    keep-multiplier 3;
    graceful-restart {
        restart-time 120;
        recovery-time 180;
        helper-mode;
    }
}

## Interface-specific settings
set protocols rsvp interface ge-0/0/1.0 {
    hello-interval 3;
    hello-multiplier 3;
    graceful-restart;
    authentication-key "$9$KvM1Tz6Au0IylKgoJDHq";
}

## Critical interface with BFD
set protocols rsvp interface ge-0/0/2.0 {
    hello-interval 1;
    oam {
        bfd-liveness-detection {
            minimum-interval 300;
            multiplier 3;
        }
    }
}

## LSP-specific error handling
set protocols mpls label-switched-path CRITICAL-LSP {
    to 192.168.1.4;
    retry-timer 10;
    retry-limit 20;
    retry-timer-backoff;
    primary MAIN {
        standby;
    }
    secondary BACKUP {
        standby;
    }
}
```

Advanced Error Recovery

Configure sophisticated error recovery:

```
## Fast reroute for immediate protection
[edit protocols mpls label-switched-path PROTECTED]
user@PE1# set fast-reroute

## Path protection with hot-standby
[edit protocols mpls label-switched-path REDUNDANT]
user@PE1# set primary MAIN standby
user@PE1# set secondary BACKUP standby

## Auto-bandwidth with failure handling
[edit protocols mpls label-switched-path ADAPTIVE]
user@PE1# set auto-bandwidth
user@PE1# set auto-bandwidth adjust-interval 300
user@PE1# set auto-bandwidth adjust-overflow-count 3
user@PE1# set auto-bandwidth resign-bandwidth
```

Session Preemption and Recovery

Handle preemption gracefully:

```
## Soft preemption for graceful moves
[edit protocols rsvp]
user@PE1# set preemption soft-preemption
user@PE1# set preemption soft-preemption cleanup-timer 60

## Preemption avoidance
[edit protocols mpls label-switched-path AVOID-PREEMPT]
user@PE1# set adaptive
user@PE1# set preempt-pending-timer 120
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Session Maintenance Commands

1. Check RSVP Session Health

```
user@PE1> show rsvp session detail
Ingress RSVP: 3 sessions

192.168.1.4
  From: 192.168.1.1, LSPstate: Up, Route: 1
  LSPname: CRITICAL-LSP, LSPpath: Primary
  Refresh interval: 45 secs, Refresh multiplier: 3
  Message-IDs: Path 12345, Resv 67890
  Explicit route: 10.0.1.2 10.0.2.2 10.0.3.2
  Record route: <self> 10.0.1.2 10.0.2.2 10.0.3.2
  Hello state: Up, Hello interval: 3
  Path refresh: Reliable message delivery
  Summary refresh: Enabled
  ...

user@PE1> show rsvp neighbor detail
RSVP neighbor: 2 learned
Address: 10.0.1.2 via ge-0/0/1.0
  Hello state: Up
  Hello interval: 3, Hello tolerance: 9
  Hello packets sent: 28934, received: 28933
  Remote instance: 0x9f8a7b6c
  Remote node-id: 192.168.1.2
  Refresh reduction: Active
    Remote end: Supports message-id, summary-refresh
    Reliable delivery: Active
    Ack outstanding: No
  Graceful restart: Supported
    Restart time: 120000 msec, Recovery time: 180000 msec
```

2. Monitor Refresh Reduction

```
user@PE1> show rsvp statistics
PacketType         Packets Sent    Packets Received
Path                        245                189
PathErr                       3                  2
PathTear                      5                  4
Resv                        189                245
ResvErr                       1                  0
ResvTear                      4                  5
Hello                     28934              28933
Ack                         456                455
SRefresh                   8901               8900
Bundle                      234                233

Refresh reduction statistics:
  Summary refreshes sent: 8901
  Summary refreshes received: 8900
  Bundle messages sent: 234
  Bundle messages received: 233
  Message-IDs acknowledged: 99.8%
  Reliable messages retransmitted: 12
```

Common Troubleshooting Scenarios

Scenario 1: LSP Flapping Due to Hello Timeout

Symptom: LSP goes up/down repeatedly

Diagnostic Commands:

```
user@PE1> show rsvp session extensive | match "Hello|hello"
  Hello state: Dn, Hello interval: 3
  Last hello lost: 4 times

user@PE1> show log messages | match "RSVP.*[Hh]ello"
Dec 5 14:23:45 PE1 rpd[1234]: RSVP neighbor 10.0.1.2 hello timeout
Dec 5 14:23:55 PE1 rpd[1234]: RSVP neighbor 10.0.1.2 hello restored
Dec 5 14:24:12 PE1 rpd[1234]: RSVP neighbor 10.0.1.2 hello timeout

user@PE1> show interfaces ge-0/0/1.0 extensive | match "error|drop"
  Input errors: 0, Output errors: 0
  Input drops: 892, Output drops: 0  ## Drops!
```

Cause: Congestion causing hello drops

Solution:

```
## Increase hello tolerance
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set hello-interval 5
user@PE1# set hello-multiplier 4
user@PE1# commit

## Or implement QoS for control traffic
[edit class-of-service interfaces ge-0/0/1]
user@PE1# set unit 0 forwarding-class network-control bandwidth-percent 5
```

Scenario 2: Graceful Restart Not Working

Symptom: Traffic loss during RSVP restart

Diagnostic Commands:

```
user@PE1> show rsvp neighbor detail | match -i graceful
  Graceful restart: Not supported  ## Problem!

user@PE1> show rsvp version
Resource ReSerVation Protocol, version 1

user@PE1> show route forwarding-table label 100234
Routing table: default.mpls
MPLS:
Destination  Type RtRef Next hop       Type Index   NhRef Netif
100234       user     0                Pop         2     2
```

Cause: Graceful restart not enabled on neighbor

Solution:

```
## Enable on all routers
[edit protocols rsvp]
user@router# set graceful-restart
user@router# commit

## Verify after enabling
user@PE1> show rsvp neighbor detail | match -i graceful
  Graceful restart: Supported
    Restart time: 120000 msec, Recovery time: 180000 msec
```

Scenario 3: Excessive Control Plane Load

Symptom: High CPU due to RSVP refreshes

Diagnostic Commands:

```
user@PE1> show system processes extensive | match rpd
 1234 root      85  0   450M   320M RUN    0  85.2% rpd
```

```
user@PE1> show rsvp statistics | match "Path"
Path                     58234               57891
## Very high numbers!

user@PE1> show rsvp summary
RSVP session summary:
  Active sessions: 2847
  Refresh reduction: Disabled  ## Problem!
```

Cause: Refresh reduction not enabled with many LSPs

Solution:

```
[edit protocols rsvp]
user@PE1# set refresh-reduction
user@PE1# set refresh-reduction reliable-delivery
user@PE1# set refresh-reduction summary-refresh
user@PE1# commit

## Monitor improvement
user@PE1> show system processes extensive | match rpd
 1234 root      15  0   450M   320M sleep  0  12.1% rpd
## Much better!
```

Scenario 4: PathErr Not Clearing

Symptom: LSP stuck with persistent PathErr

Diagnostic Commands:

```
user@PE1> show mpls lsp extensive name LSP1
  State: Dn
  ActivePath: (none)
  Will be enqueued for recomputation in 24 second(s).
  1673 Dec 5 15:45:23.123 CSPF failed: no route toward 192.168.1.4[32 times]
  ...
  Retry count: 32, Retry limit: 5  ## Exceeded limit!

user@PE1> show configuration protocols mpls label-switched-path LSP1 | match retry
retry-timer 30;
retry-limit 5;
```

Cause: Retry limit exceeded

Solution:

```
## Clear and restart LSP
user@PE1> clear mpls lsp name LSP1

## Or increase retry limit
[edit protocols mpls label-switched-path LSP1]
user@PE1# set retry-limit 0  ## Infinite retries
user@PE1# commit
```

Advanced Session Maintenance Troubleshooting

Debug Message Reliability

```
[edit protocols rsvp]
user@PE1# set traceoptions file rsvp-reliability
user@PE1# set traceoptions flag refresh detail
user@PE1# set traceoptions flag ack detail
user@PE1# commit

user@PE1> monitor start rsvp-reliability
Dec 5 16:00:00.123 RSVP: Send Path Refresh ID 45678 for LSP1
Dec 5 16:00:00.234 RSVP: Recv ACK for message ID 45678
Dec 5 16:00:00.345 RSVP: Send SRefresh with 150 message-IDs
Dec 5 16:00:00.456 RSVP: Bundle 15 messages, size 4096 bytes
```

Test Graceful Restart

```
## Prepare for test
user@PE1> show route forwarding-table label 100234
Destination  Type RtRef Next hop       Type Index   NhRef Netif
100234       user     0  10.0.1.2      Swap 200234      2     2 ge-0/0/1.0

## Restart RSVP
user@PE1> restart routing-process rsvp gracefully
Routing protocol daemon started, pid 5678
  RSVP Graceful Restart in progress

## Monitor during restart
user@PE1> show route forwarding-table label 100234
Destination  Type RtRef Next hop       Type Index   NhRef Netif
100234(S)    user     0  10.0.1.2      Swap 200234      2     2 ge-0/0/1.0
## (S) indicates stale but still forwarding!

## After recovery
user@PE1> show rsvp session | match Up
To              From            State   Rt Style Labelin Labelout LSPname
192.168.1.4     192.168.1.1     Up       1  1 FF       -   100234 LSP1
## Recovered without traffic loss!
```

Performance Monitoring

```
user@PE1> show rsvp interface statistics
Interface: ge-0/0/1.0
  Hello messages:
    Sent: 28934, Received: 28933
    Missed: 1, Timeouts: 0
  Path messages:
    Sent: 245, Received: 189
    Errors: 3
  Resv messages:
    Sent: 189, Received: 245
    Errors: 1
  Summary refresh:
    Sent: 8901, Received: 8900
    Savings: 87% (vs full refresh)
  Bundle messages:
    Sent: 234, Received: 233
    Average bundle size: 6.2 messages

user@PE1> show task memory | match RSVP
RSVP                  45.2M      1.2M     44.0M      1567      28K
  RSVP Session         8.1M      0.0       8.1M      2847      2.9K
  RSVP Refresh         2.3M      0.0       2.3M      2847      828B
```

The evolution from soft-state to reliable RSVP has transformed it from a simple but inefficient protocol into a robust, scalable signaling mechanism capable of supporting massive networks with minimal overhead. Understanding failure modes and recovery mechanisms is crucial for building resilient MPLS networks.

Module 11: RSVP-Primary and Secondary Paths

Part 1: The Conceptual Lecture (The Why)

Understanding Path Protection

Imagine driving to an important meeting. You have your preferred route (primary), but what if there's an accident blocking the road? Smart drivers plan alternate routes (secondary paths). RSVP primary and secondary paths work the same way - ensuring your critical traffic always has a way to reach its destination.

```
Path Protection Concepts:

Primary Path:
- First choice route
- Actively forwarding traffic
- Optimized for normal conditions

Secondary Path:
- Backup route
- Takes over if primary fails
- May be standby (pre-signaled) or not
```

Primary Path Characteristics

The primary path is your main highway:

```
Primary Path Properties:
- Always attempted first
- Can have specific constraints
- Carries traffic when Up
- Triggers secondary if Down

Configuration Flexibility:
- Explicit route (strict path)
- Loose constraints (CSPF-computed)
- Admin groups requirements
- Bandwidth specifications
```

Secondary Path Types

Secondary paths come in different flavors:

```
1. Non-Standby Secondary:
   - Signaled only when needed
   - No resources reserved
   - Slower failover (seconds)
   - Efficient resource usage

2. Standby Secondary:
   - Pre-signaled and ready
   - Resources reserved
   - Fast failover (<50ms possible)
   - Uses more resources

3. Multiple Secondaries:
   - Can have many backups
   - Priority order for selection
   - Different paths/constraints
   - Ultimate redundancy
```

The Cost of Protection

Path protection involves trade-offs:

```
Resource Consumption:

Single LSP:
[PE1]---10G---[P1]---10G---[P2]---10G---[PE2]
Uses: 1Gbps on each link

With Standby Secondary:
```

```
Primary:   [PE1]---[P1]---[P2]---[PE2] (1Gbps reserved)
Secondary: [PE1]---[P3]---[P4]---[PE2] (1Gbps reserved)
Uses: 2Gbps total (double booking!)

Shared Fate Consideration:
Bad Design: Primary and secondary use same fiber
Good Design: Diverse paths (different fibers/conduits)
```

Path Protection State Machine

Understanding path state transitions:

```
State Transitions:

1. Normal Operation:
   Primary: Up/Active
   Secondary: Down (non-standby) or Up/Standby

2. Primary Failure:
   Primary: Down
   Secondary: Up/Active (becomes active)

3. Primary Recovery:
   Primary: Up
   Secondary: Depends on revertive behavior

4. Revertive Options:
   - Revertive: Switch back to primary when it recovers
   - Non-revertive: Stay on secondary until it fails
```

Pre-Install: The Secret to Fast Failover

Pre-installing backup paths in hardware is like having your alternate route already programmed in your GPS:

```
Without Pre-Install:
1. Primary fails
2. Detect failure (50ms)
3. Signal secondary (100ms)
4. Program forwarding (50ms)
5. Total: ~200ms outage

With Pre-Install:
1. Primary fails
2. Detect failure (50ms)
3. Switch to pre-programmed backup
4. Total: ~50ms outage

Forwarding Table View:
Label 100:
  Primary NH: Interface ge-0/0/1 (Active)
  Backup NH: Interface ge-0/0/2 (Standby)
```

Make-Before-Break and Path Protection

How to switch paths without dropping packets:

```
Make-Before-Break Process:
1. Current path carrying traffic
2. Signal new path
3. New path establishes
4. Pre-install new path
5. Switch traffic (single atomic operation)
```

```
6. Tear down old path
7. Zero packet loss!

Use Cases:
- Maintenance windows
- Optimization moves
- Bandwidth adjustments
- Admin group changes
```

Part 2: The Junos CLI Masterclass (The How)

Basic Primary and Secondary Configuration

Configure an LSP with backup path:

```
## Simple primary/secondary setup
[edit protocols mpls]
user@PE1# set label-switched-path PROTECTED-LSP to 192.168.1.4
user@PE1# set label-switched-path PROTECTED-LSP primary PRIMARY-PATH
user@PE1# set label-switched-path PROTECTED-LSP secondary SECONDARY-PATH

## Define the paths
user@PE1# set path PRIMARY-PATH 10.0.1.2 strict
user@PE1# set path PRIMARY-PATH 10.0.2.2 strict
user@PE1# set path PRIMARY-PATH 10.0.3.2 strict

user@PE1# set path SECONDARY-PATH 10.0.4.2 strict
user@PE1# set path SECONDARY-PATH 10.0.5.2 strict
user@PE1# set path SECONDARY-PATH 10.0.6.2 strict
```

Standby Secondary Configuration

Enable pre-signaling for fast failover:

```
## Standby secondary (pre-signaled)
[edit protocols mpls label-switched-path FAST-FAILOVER]
user@PE1# set to 192.168.1.4
user@PE1# set primary MAIN-PATH
user@PE1# set secondary BACKUP-PATH standby

## Hot-standby (pre-signaled and pre-installed)
[edit protocols mpls label-switched-path ZERO-LOSS]
user@PE1# set to 192.168.1.4
user@PE1# set primary MAIN-PATH standby
user@PE1# set secondary BACKUP-PATH standby
```

Multiple Secondary Paths

Configure multiple backup options:

```
## Multiple secondaries with priority
[edit protocols mpls label-switched-path MULTI-BACKUP]
user@PE1# set to 192.168.1.4
user@PE1# set primary BEST-PATH
user@PE1# set secondary SECOND-BEST priority 100
user@PE1# set secondary THIRD-BEST priority 200
user@PE1# set secondary LAST-RESORT priority 300

## Different constraints per path
user@PE1# set primary BEST-PATH bandwidth 1g
user@PE1# set secondary SECOND-BEST bandwidth 1g
user@PE1# set secondary THIRD-BEST bandwidth 500m
user@PE1# set secondary LAST-RESORT bandwidth 100m
```

Advanced Path Protection Features

Configure sophisticated protection schemes:

```
## Revertive behavior control
[edit protocols mpls label-switched-path REVERTIVE-LSP]
user@PE1# set to 192.168.1.4
user@PE1# set revert-timer 300  ## Wait 5 minutes before reverting
user@PE1# set primary MAIN standby
user@PE1# set secondary BACKUP standby

## Non-revertive configuration
[edit protocols mpls label-switched-path NON-REVERTIVE]
user@PE1# set to 192.168.1.4
user@PE1# set no-revert-timer
user@PE1# set primary MAIN
user@PE1# set secondary BACKUP standby

## Path protection with constraints
[edit protocols mpls label-switched-path CONSTRAINED-PROTECTION]
user@PE1# set to 192.168.1.4
user@PE1# set primary FIBER-PATH admin-group include-all FIBER
user@PE1# set secondary MICROWAVE-PATH admin-group include-all MICROWAVE
user@PE1# set secondary SATELLITE-PATH admin-group include-all SATELLITE
```

Bandwidth Protection Strategies

Handle bandwidth for protected LSPs:

```
## Shared bandwidth between primary/secondary
[edit protocols mpls label-switched-path SHARED-BW]
user@PE1# set to 192.168.1.4
user@PE1# set bandwidth 1g
user@PE1# set primary MAIN
user@PE1# set secondary BACKUP
user@PE1# set secondary BACKUP standby
user@PE1# set secondary BACKUP bandwidth 1g
user@PE1# set secondary BACKUP shared

## Different bandwidth for backup
[edit protocols mpls label-switched-path DEGRADED-BACKUP]
user@PE1# set to 192.168.1.4
user@PE1# set primary MAIN bandwidth 10g
user@PE1# set secondary BACKUP bandwidth 1g
user@PE1# set secondary BACKUP standby
```

Complete Path Protection Example

Build a comprehensive protected service:

```
## Define paths
set protocols mpls path DIVERSE-PATH-1 {
    10.0.1.2 strict;
    10.0.2.2 strict;
    10.0.3.2 strict;
}

set protocols mpls path DIVERSE-PATH-2 {
    10.0.4.2 strict;
    10.0.5.2 strict;
    10.0.6.2 strict;
}

set protocols mpls path DIVERSE-PATH-3 {
    10.0.7.2 strict;
    10.0.8.2 strict;
    10.0.9.2 strict;
}

## Protected LSP configuration
set protocols mpls label-switched-path CRITICAL-SERVICE {
    description "Mission-critical application";
    to 192.168.1.4;
    bandwidth 2g;
    priority 1 1;

    ## Primary path
    primary DIVERSE-PATH-1 {
        bandwidth 2g;
        admin-group include-all FIBER LOW-LATENCY;
        standby;
    }

    ## First backup - equal quality
    secondary DIVERSE-PATH-2 {
        bandwidth 2g;
        priority 100;
        admin-group include-all FIBER;
        admin-group exclude SATELLITE;
        standby;
    }

    ## Second backup - degraded service
    secondary DIVERSE-PATH-3 {
        bandwidth 1g;
        priority 200;
        admin-group include-any MICROWAVE FIBER;
        standby;
    }

    ## Protection switching behavior
    revert-timer 600;  ## 10 minutes
    optimize-timer 3600;  ## 1 hour

    ## Fast failure detection
    oam {
        bfd-liveness-detection {
            minimum-interval 100;
            multiplier 3;
        }
    }
}

## Install in forwarding table
set protocols mpls label-switched-path CRITICAL-SERVICE install 192.168.1.4/32 active
```

Path Selection Preferences

Control which path is preferred:

```
## Metric-based selection
[edit protocols mpls label-switched-path METRIC-BASED]
user@PE1# set to 192.168.1.4
user@PE1# set primary SHORT metric 10
user@PE1# set secondary LONG metric 20

## Explicit preference
[edit protocols mpls label-switched-path PREFERENCE-BASED]
user@PE1# set to 192.168.1.4
user@PE1# set primary MAIN preference 5
user@PE1# set secondary BACKUP preference 10
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Path Protection Verification

1. View Primary and Secondary Status

```
user@PE1> show mpls lsp name CRITICAL-SERVICE extensive
Ingress LSP: 1 sessions
```

```
192.168.1.4
  From: 192.168.1.1, State: Up, ActiveRoute: 1, LSPname: CRITICAL-SERVICE
  ActivePath: DIVERSE-PATH-1 (primary)
  *Primary   DIVERSE-PATH-1    State: Up
    Priorities: 1 1
    Bandwidth: 2Gbps
    SmartOptimizeTimer: 3600
    Standby: Yes
    LoadBalance: Random
    Revert: Timer configured (600 seconds)
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 30)
      10.0.1.2 S 10.0.2.2 S 10.0.3.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
      10.0.1.2 10.0.2.2 10.0.3.2

  Standby DIVERSE-PATH-2    State: Up
    Priorities: 1 1
    Bandwidth: 2Gbps
    Priority: 100
    Standby: Yes
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 35)
      10.0.4.2 S 10.0.5.2 S 10.0.6.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
      10.0.4.2 10.0.5.2 10.0.6.2

  Standby DIVERSE-PATH-3    State: Up
    Priorities: 1 1
    Bandwidth: 1Gbps
    Priority: 200
    Standby: Yes
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 40)
      10.0.7.2 S 10.0.8.2 S 10.0.9.2 S
```

2. Check Forwarding Table Installation

```
user@PE1> show route forwarding-table family mpls
Routing table: default.mpls
MPLS:
Destination  Type RtRef Next hop       Type Index   NhRef Netif
100234  (S)  user     0                ulst  1048574     3
                        10.0.1.2       Swap 200234     583     2 ge-0/0/1.0
                        10.0.4.2       Swap 200567     584     2 ge-0/0/2.0 (!)
                        10.0.7.2       Swap 200890     585     2 ge-0/0/3.0 (!)

## (S) = Static route
## (!) = Backup next-hop
## Primary is active, backups are pre-installed!
```

3. Monitor Path Switching Events

```
user@PE1> show mpls lsp log
Dec 10 10:15:23.456  CRITICAL-SERVICE  Primary DIVERSE-PATH-1 up
Dec 10 10:15:45.789  CRITICAL-SERVICE  Secondary DIVERSE-PATH-2 up (standby)
Dec 10 10:15:58.123  CRITICAL-SERVICE  Secondary DIVERSE-PATH-3 up (standby)
Dec 10 14:23:15.456  CRITICAL-SERVICE  Primary DIVERSE-PATH-1 down
Dec 10 14:23:15.478  CRITICAL-SERVICE  Secondary DIVERSE-PATH-2 active
Dec 10 14:28:45.234  CRITICAL-SERVICE  Primary DIVERSE-PATH-1 up
Dec 10 14:38:45.234  CRITICAL-SERVICE  Reverted to primary DIVERSE-PATH-1
Dec 10 14:38:45.256  CRITICAL-SERVICE  Secondary DIVERSE-PATH-2 standby
```

Common Troubleshooting Scenarios

Scenario 1: Secondary Path Not Establishing

Symptom: Secondary path remains down

Diagnostic Commands:

```
user@PE1> show mpls lsp name PROTECTED-LSP
To              From            State Rt P     ActivePath       LSPname
192.168.1.4     192.168.1.1     Up     1 *     PRIMARY-PATH     PROTECTED-LSP

user@PE1> show mpls lsp name PROTECTED-LSP secondary
Secondary SECONDARY-PATH
  State: Dn
  Will be enqueued for recomputation in 18 second(s).
  1 Dec 10 15:45:23.123 CSPF failed: no route toward 192.168.1.4[5 times]

user@PE1> show configuration protocols mpls path SECONDARY-PATH
10.0.4.2 strict;
10.0.5.2 strict;
10.0.6.2 strict;
```

Cause: Path not available or constraints not met

Solution:

```
## Check if path exists
user@PE1> show ted database 10.0.4.1 10.0.4.2
  [no output - link doesn't exist in TED]

## Fix path or relax constraints
[edit protocols mpls]
user@PE1# set path SECONDARY-PATH-LOOSE 192.168.1.3 loose
user@PE1# set path SECONDARY-PATH-LOOSE 192.168.1.4 loose
user@PE1# set label-switched-path PROTECTED-LSP secondary SECONDARY-PATH-LOOSE
```

Scenario 2: Slow Failover Despite Standby

Symptom: Traffic loss during primary failure

Diagnostic Commands:

```
user@PE1> show mpls lsp name FAST-FAILOVER extensive | match "Standby|State"
  ActivePath: MAIN-PATH (primary)
  *Primary   MAIN-PATH    State: Up
    Standby: No     ## Not standby!
  Standby BACKUP-PATH    State: Up
    Standby: Yes    ## Secondary is standby

user@PE1> show route forwarding-table family mpls label 100234
Destination  Type RtRef Next hop       Type Index   NhRef Netif
100234       user     0  10.0.1.2      Swap 200234     583     2 ge-0/0/1.0
## No backup next-hop installed!
```

Cause: Primary not configured as standby

Solution:

```
[edit protocols mpls label-switched-path FAST-FAILOVER]
user@PE1# set primary MAIN-PATH standby
user@PE1# commit

## Verify pre-installation
user@PE1> show route forwarding-table family mpls label 100234
Destination  Type RtRef Next hop       Type Index   NhRef Netif
100234  (S)  user     0                ulst  1048574     2
                        10.0.1.2       Swap 200234     583     2 ge-0/0/1.0
                        10.0.4.2       Swap 200567     584     2 ge-0/0/2.0 (!)
```

Scenario 3: Not Reverting to Primary

Symptom: LSP stays on secondary after primary recovers

Diagnostic Commands:

```
user@PE1> show mpls lsp name REVERTIVE-LSP
To              From            State Rt P     ActivePath       LSPname
192.168.1.4     192.168.1.1     Up     1 *     BACKUP          REVERTIVE-LSP

user@PE1> show mpls lsp log | match REVERTIVE-LSP | last 5
Dec 10 16:00:00  REVERTIVE-LSP  Primary MAIN down
Dec 10 16:00:00  REVERTIVE-LSP  Secondary BACKUP active
Dec 10 16:05:00  REVERTIVE-LSP  Primary MAIN up
Dec 10 16:10:00  REVERTIVE-LSP  Revert timer started (300 seconds)
Dec 10 16:15:00  REVERTIVE-LSP  Revert timer expired

user@PE1> show configuration protocols mpls label-switched-path REVERTIVE-LSP | match revert
no-revert-timer;  ## Configured not to revert!
```

Solution:

```
## Enable reversion
[edit protocols mpls label-switched-path REVERTIVE-LSP]
user@PE1# delete no-revert-timer
user@PE1# set revert-timer 300
user@PE1# commit

## Or manually revert
user@PE1> clear mpls lsp name REVERTIVE-LSP optimize
```

Scenario 4: Bandwidth Double-Booking

Symptom: Secondary fails due to bandwidth despite shared configuration

Diagnostic Commands:

```
user@PE1> show rsvp interface detail | match Available
  Available bandwidth:
    [0] 1Gbps        [1] 1Gbps        [2] 1Gbps        [3] 1Gbps

user@PE1> show rsvp session name SHARED-BW
Ingress RSVP: 2 sessions
To              From            State   Rt Style Labelin Labelout LSPname
192.168.1.4     192.168.1.1     Up       1  1 FF       -   100234 SHARED-BW
  Primary path: MAIN
  Bandwidth: 1Gbps
192.168.1.4     192.168.1.1     Down     0  1 FF       -        - SHARED-BW
  Secondary path: BACKUP
  Bandwidth: 1Gbps (shared)
  Error: Admission control failure
```

Cause: Shared bandwidth not recognized on some paths

Solution:

```
## Ensure shared-explicit signaling
[edit protocols mpls]
user@PE1# set shared-explicit

## Or use facility backup instead
[edit protocols mpls label-switched-path SHARED-BW]
user@PE1# set link-protection
```

Advanced Path Protection Troubleshooting

Test Failover Behavior

```
## Simulate primary path failure
user@PE1> test mpls lsp-switchover name CRITICAL-SERVICE path DIVERSE-PATH-2

## Monitor the switchover
user@PE1> monitor interface traffic matching mpls
```

```
## Watch for traffic shift between interfaces

## Check switchover time
user@PE1> show mpls lsp protection
LSP Name         Protection Type    Switchover Time    Last Switch
CRITICAL-SERVICE Path protection    22 msec            Dec 10 17:00:00
```

Verify Path Diversity

```
## Check for shared fate
user@PE1> show mpls lsp name CRITICAL-SERVICE detail | match "ERO"
  Primary:   10.0.1.2 S 10.0.2.2 S 10.0.3.2 S
  Secondary: 10.0.4.2 S 10.0.5.2 S 10.0.6.2 S

## Trace physical topology
user@PE1> show rsvp session name CRITICAL-SERVICE detail | match "RRO|Link"
  Primary RRO:   10.0.1.2 (Link 1) 10.0.2.2 (Link 2) 10.0.3.2 (Link 3)
  Secondary RRO: 10.0.4.2 (Link 4) 10.0.5.2 (Link 5) 10.0.6.2 (Link 6)

## Verify link diversity
user@PE1> show configuration interfaces | match "description.*Link"
ge-0/0/1 description "Link 1 - Fiber Route A";
ge-0/0/2 description "Link 4 - Fiber Route B";
## Good - different physical routes
```

Monitor Protection Performance

```
user@PE1> show mpls lsp statistics protection
Protection Statistics:
  Total protected LSPs: 47
  Primary paths: 47
  Secondary paths: 89
  Standby paths: 72

  Protection switches (last 24h): 3
  Average switch time: 31.2 msec
  Maximum switch time: 87.3 msec

  Revertions (last 24h): 2
  Average revert time: 300.0 sec

  Current active paths:
    Primary: 45 (95.7%)
    Secondary: 2 (4.3%)
```

Primary and secondary paths transform RSVP LSPs from single points of failure into resilient services. With proper design including standby configurations, path diversity, and pre-installation, you can achieve carrier-grade availability with sub-50ms failover times.

Module 12: RSVP-Local Repair, Part 1—One-to-One Backup, or Fast-Reroute

Part 1: The Conceptual Lecture (The Why)

The Need for Local Repair

Imagine you're driving on a highway when suddenly the road ahead is blocked. With traditional RSVP path protection, you'd have to drive all the way back to your starting point and take a completely different route. Local repair is like having a detour sign right at the point of failure - you immediately take a local bypass and rejoin your original route past the problem.

```
Traditional Path Protection vs Local Repair:

Path Protection (End-to-End):
[PE1]----[P1]----X----[P2]----[PE2]
  |                             |
  +-------[P3]-------[P4]--------+
Problem: PE1 must detect failure and switch to backup
Time: 100-500ms typically

Local Repair (Fast Reroute):
[PE1]----[P1]----X----[P2]----[PE2]
           |           |
           +---[P5]----+

Solution: P1 immediately detours around failure
Time: <50ms typically
```

Understanding One-to-One Backup

One-to-one backup creates a unique detour LSP for each protected LSP at each potential failure point:

```
One-to-One Backup Model:

Protected LSP: PE1 → P1 → P2 → P3 → PE2

Detour LSPs Created:
1. At PE1: Detour around P1 failure
2. At P1: Detour around P2 failure
3. At P2: Detour around P3 failure
4. At P3: Detour around PE2 failure

If you have 100 LSPs through a link:
- 100 detour LSPs created
- Each with its own label
- Resource intensive but flexible
```

The Fast Reroute Terminology Confusion

"Fast Reroute" is one of the most overloaded terms in networking:

```
Different Meanings of "Fast Reroute":

1. Generic Term:
   - Any local repair mechanism
   - Includes one-to-one and facility backup

2. Specific to One-to-One:
   - Original RSVP-TE fast reroute
   - RFC 4090 "Fast Reroute Extensions"

3. Vendor Specific:
   - Junos: "fast-reroute" = one-to-one backup
   - Cisco: "FRR" = can mean either method

4. In Other Protocols:
   - LDP Fast Reroute
   - BGP PIC (Prefix Independent Convergence)
   - IP Fast Reroute (Loop-Free Alternates)

Always clarify which type!
```

How One-to-One Backup Works

The detour creation and activation process:

```
Detour LSP Signaling:
```

```
1. Main LSP Established:
   PE1 --PATH--> P1 --PATH--> P2 --PATH--> PE2
   PE1 <--RESV-- P1 <--RESV-- P2 <--RESV-- PE2

2. P1 Creates Detour:
   P1 --DETOUR PATH--> P3 --PATH--> P4 --PATH--> P2
   P1 <--DETOUR RESV-- P3 <--RESV-- P4 <--RESV-- P2

3. Normal Operation:
   Traffic follows: PE1 → P1 → P2 → PE2
   Detour ready but not used

4. Link P1-P2 Fails:
   P1 detects failure (<50ms)
   P1 immediately redirects to detour
   Traffic now: PE1 → P1 → P3 → P4 → P2 → PE2
```

Detour Computation and Constraints

How routers decide where to build detours:

```
Detour Path Selection Rules:

1. Must Avoid Protected Resource:
   - For link protection: Avoid failed link
   - For node protection: Avoid next-hop node

2. Should Rejoin Quickly:
   - Merge back to main path ASAP
   - Minimize extra hops

3. Inherit Main LSP Constraints:
   - Bandwidth requirements
   - Admin group constraints
   - But may relax if needed

4. Best Effort:
   - If no perfect detour exists
   - Any path better than blackhole
```

Merge Points and Optimization

Detours try to merge back efficiently:

```
Merge Point Selection:

Main LSP: A → B → C → D → E → F

Detour from A (protecting A-B link):
Option 1: A → G → B (Merge at next-hop)
Option 2: A → G → H → C (Merge at next-next-hop)
Option 3: A → G → H → I → F (Merge at egress)

Preference: Merge as soon as possible
- Reduces resource usage
- Minimizes latency addition
- Simplifies management
```

The Cost of One-to-One Backup

Resource consumption can be significant:

```
Scalability Considerations:

Network: 100 routers, 1000 LSPs
Average path length: 5 hops

One-to-One Backup Creates:
- 1000 LSPs × 4 protection points = 4000 detour LSPs
- Control plane: 5000 total LSPs to manage
- Memory: ~50MB for state storage
- CPU: Continuous detour optimization

Resource Formula:
Detours = (Number of LSPs) × (Average Hops - 1)
```

Part 2: The Junos CLI Masterclass (The How)

Basic Fast Reroute Configuration

Enable one-to-one backup on an LSP:

```
## Simple fast-reroute enablement
[edit protocols mpls]
user@PE1# set label-switched-path PROTECTED-LSP to 192.168.1.4
user@PE1# set label-switched-path PROTECTED-LSP fast-reroute

## That's it! Junos handles the rest
## But let's see what this actually does...
```

Understanding Fast Reroute Options

Configure specific fast reroute behaviors:

```
## Fast reroute with bandwidth protection
[edit protocols mpls label-switched-path FRR-LSP]
user@PE1# set to 192.168.1.4
user@PE1# set bandwidth 1g
user@PE1# set fast-reroute
user@PE1# set fast-reroute bandwidth 1g

## Fast reroute with hop limit
user@PE1# set fast-reroute hop-limit 3
## Detour can be maximum 3 hops longer

## Fast reroute with admin group constraints
user@PE1# set fast-reroute include-any BACKUP-LINKS
user@PE1# set fast-reroute exclude EXPENSIVE

## Node protection (protect against node failure)
user@PE1# set fast-reroute node-protection
```

Controlling Detour Computation

Fine-tune how detours are calculated:

```
## Relaxed constraints for detour paths
[edit protocols mpls]
user@PE1# set label-switched-path BUSINESS-LSP to 192.168.1.4
user@PE1# set label-switched-path BUSINESS-LSP bandwidth 2g
user@PE1# set label-switched-path BUSINESS-LSP admin-group include-all GOLD
user@PE1# set label-switched-path BUSINESS-LSP fast-reroute
user@PE1# set label-switched-path BUSINESS-LSP fast-reroute bandwidth 500m
user@PE1# set label-switched-path BUSINESS-LSP fast-reroute include-any SILVER BRONZE

## Main path: 2G on GOLD links
## Detour: 500M on SILVER or BRONZE links
```

Complete One-to-One Backup Example

Configure comprehensive fast reroute protection:

```
## Define the main LSP with full protection
set protocols mpls label-switched-path CRITICAL-APP {
    description "Business critical application with FRR";
    to 192.168.1.4;
    bandwidth 5g;
    priority 1 1;

    ## Primary path requirements
    admin-group include-all FIBER LOW-LATENCY;
    admin-group exclude SATELLITE;

    ## Enable fast reroute
    fast-reroute {
        bandwidth 2g;  ## Degraded but acceptable
        hop-limit 5;   ## Allow longer detour
        node-protection;  ## Protect against node failures

        ## Relaxed admin constraints for detours
        include-any FIBER MICROWAVE;
        ## Don't exclude SATELLITE for detours - any path better than none
    }

    ## Also have end-to-end protection
    primary MAIN-PATH;
    secondary BACKUP-PATH standby;
}

## Configure RSVP for fast protection switching
set protocols rsvp interface ge-0/0/1.0 {
    hello-interval 1;
    hello-multiplier 3;
    link-protection;
}

## Enable MPLS on all potential detour interfaces
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface ge-0/0/4.0
```

Node Link Protection (Enhanced Node Protection)

Configure protection against node failures:

```
## Node-link protection configuration
[edit protocols mpls]
user@PE1# set label-switched-path NODE-PROTECTED to 192.168.1.4
user@PE1# set label-switched-path NODE-PROTECTED fast-reroute
user@PE1# set label-switched-path NODE-PROTECTED fast-reroute node-link-protection

## What this does:
## - Protects against link failure (always)
## - Protects against node failure (when possible)
## - Falls back to link protection if node protection unavailable
```

Excluding Interfaces from Protection

Sometimes you don't want protection on certain links:

```
## Exclude specific interfaces from protection
[edit protocols mpls]
user@PE1# set label-switched-path SELECTIVE-FRR to 192.168.1.4
user@PE1# set label-switched-path SELECTIVE-FRR fast-reroute

## Don't protect the last hop
```

```
[edit protocols rsvp interface ge-0/0/10.0]
user@PE1# set no-link-protection

## Or exclude from LSP perspective
[edit protocols mpls label-switched-path SELECTIVE-FRR]
user@PE1# set fast-reroute exclude-interface ge-0/0/10.0
```

Optimizing Detour Paths

Control detour optimization behavior:

```
## Configure detour optimization timers
[edit protocols mpls]
user@PE1# set optimize-timer 3600  ## Re-optimize every hour
user@PE1# set label-switched-path FRR-LSP fast-reroute
user@PE1# set label-switched-path FRR-LSP fast-reroute optimize-timer 300

## Aggressive optimization for detours
user@PE1# set optimize-aggressive
user@PE1# set label-switched-path FRR-LSP optimize-aggressive
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Fast Reroute Verification

1. View Detour LSP Status

```
user@PE1> show mpls lsp detail
Ingress LSP: 1 sessions
192.168.1.4
  From: 192.168.1.1, State: Up, ActiveRoute: 1, LSPname: CRITICAL-APP
  ActivePath:  (primary)
  Node/Link protection desired
  LSPtype: Static Configured
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  *Primary                    State: Up
    Priorities: 1 1
    Bandwidth: 5Gbps
    FastReroute desired
    Node/Link protection desired
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 40)
      10.0.1.2 S 10.0.2.2 S 10.0.3.2 S 10.0.4.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
      10.0.1.2(flag=9) 10.0.2.2(flag=9) 10.0.3.2(flag=9) 10.0.4.2(flag=1)

    Detour LSPs:
    Detour LSP to 10.0.2.2: Up (1 detours up)
      Bandwidth: 2Gbps
      Computed ERO: 10.0.5.2 S 10.0.6.2 S 10.0.2.2 S
      Record Route: 10.0.5.2 10.0.6.2 10.0.2.2
      Protection: Link+Node
```

2. Check Protection Coverage

```
user@PE1> show rsvp session protection
Ingress RSVP: 4 sessions, 3 detours up, 1 detours down

To              From            LSPname         State   Detour  Protection
192.168.1.4     192.168.1.1     CRITICAL-APP    Up      Up      Link+Node
192.168.1.4     192.168.1.1     BUSINESS-LSP    Up      Up      Link
192.168.1.4     192.168.1.1     BULK-LSP        Up      Down    Requested
192.168.1.4     192.168.1.1     TEST-LSP        Up      Up      Link

Transit RSVP: 12 sessions, 10 detours up, 2 detours down
To              From            LSPname         State   Detour  Protection
```

```
192.168.1.8     192.168.1.5     TRANSIT-1       Up      Up      Link+Node
192.168.1.9     192.168.1.6     TRANSIT-2       Up      Down    Requested
```

3. Monitor Fast Reroute Events

```
user@PE1> show mpls lsp fast-reroute-log
Timestamp           LSP Name      Event                Details
Dec 15 10:23:45.123  CRITICAL-APP  Detour Up           Protection available at 10.0.1.2
Dec 15 10:24:15.456  CRITICAL-APP  Local Repair        Link 10.0.1.2->10.0.2.2 failed
Dec 15 10:24:15.478  CRITICAL-APP  Detour Active       Traffic on detour via 10.0.5.2
Dec 15 10:24:45.789  CRITICAL-APP  Make-before-break   New primary computed
Dec 15 10:25:00.123  CRITICAL-APP  Global Repair       Switched to new primary
Dec 15 10:25:00.234  CRITICAL-APP  Detour Down         Old detour removed
```

Common Troubleshooting Scenarios

Scenario 1: Detour LSP Not Establishing

Symptom: Fast reroute requested but no detour created

Diagnostic Commands:

```
user@PE1> show mpls lsp name FRR-LSP extensive | match "Detour|protection"
    FastReroute desired
    Node/Link protection desired
    Detour LSPs:
    No detours up, 1 detour down

user@PE1> show log messages | match "FRR-LSP.*detour"
Dec 15 11:30:23 PE1 rpd[1234]: MPLS LSP FRR-LSP detour CSPF failed at 10.0.1.2: no route

user@PE1> show ted database extensive 10.0.1.2
NodeID: 192.168.1.2
  Type: Rtr, Age: 234 secs, LinkIn: 1, LinkOut: 1
  Protocol: OSPF(0.0.0.0)
    To: 192.168.1.3, Local: 10.0.2.1, Remote: 10.0.2.2
      ## Only one outgoing link - no alternate path!
```

Cause: No alternate path available for detour

Solution:

```
## Add redundant links in topology
## Or relax detour constraints
[edit protocols mpls label-switched-path FRR-LSP fast-reroute]
user@PE1# delete bandwidth  ## Remove bandwidth requirement
user@PE1# set hop-limit 10  ## Allow longer detours
user@PE1# commit
```

Scenario 2: Slow Local Repair

Symptom: Protection switch takes longer than 50ms

Diagnostic Commands:

```
user@router> show rsvp interface detail
ge-0/0/1.0
  Hello: Disabled  ## Problem!

user@router> show interfaces ge-0/0/1 extensive | match "carrier transitions"
  Carrier transitions: 45
  ## Physical instability

user@router> show mpls lsp protection detail
LSP: CRITICAL-APP
  Protection switch time: 156ms
  Detection time: 145ms
  Switch time: 11ms
```

Cause: Slow failure detection

Solution:

```
## Enable fast hello
[edit protocols rsvp interface ge-0/0/1.0]
user@router# set hello-interval 1
user@router# set hello-multiplier 3

## Or use BFD
user@router# set oam bfd-liveness-detection minimum-interval 100
user@router# set oam bfd-liveness-detection multiplier 3
user@router# commit
```

Scenario 3: Node Protection Not Working

Symptom: Only link protection active despite node-protection config

Diagnostic Commands:

```
user@P1> show mpls lsp transit protection
To              From            LSPname      Protect  Detour       Type
192.168.1.4     192.168.1.1     NODE-PROTECTED  Yes   10.0.5.2     Link

user@P1> show rsvp session transit detail name NODE-PROTECTED
  ...
  FastReroute protection: Node/Link requested
  Detour Up: Yes
  Detour template: Link protection
  Reason: Cannot find node-protecting path

user@P1> show ted database 192.168.1.3  ## Next-hop node
  To: 192.168.1.4, Local: 10.0.3.1, Remote: 10.0.3.2
  To: 192.168.1.5, Local: 10.0.8.1, Remote: 10.0.8.2
  ## Next-hop has only one path to destination!
```

Cause: Topology doesn't support node protection

Solution: Add redundant paths in network design

Scenario 4: Detour Using Too Much Bandwidth

Symptom: Detours consuming excessive bandwidth

Diagnostic Commands:

```
user@PE1> show rsvp interface ge-0/0/2.0 extensive
  Reserved bandwidth:
    CT0: 8.5Gbps
  Breakdown:
    Primary LSPs: 2Gbps
    Detour LSPs: 6.5Gbps  ## Too much!

user@PE1> show mpls lsp transit detour
13 detour LSPs traversing this router
Each requesting 500Mbps = 6.5Gbps total
```

Solution: Implement facility backup (next module) or:

```
## Reduce detour bandwidth requirements
[edit protocols mpls]
user@PE1# set label-switched-path all fast-reroute bandwidth 100m
user@PE1# commit
```

Advanced Fast Reroute Troubleshooting

Monitor Protection Performance

```
user@PE1> show mpls lsp protection statistics
Fast Reroute Statistics:
  Total protected LSPs: 234
  Detours requested: 234
  Detours up: 201 (85.9%)
  Detours down: 33 (14.1%)

  Protection events (last 24h): 7
  Average detection time: 12.3ms
  Average switch time: 4.7ms
  Average total time: 17.0ms

  Current protected traffic: 45.6 Gbps
  Current detour traffic: 2.3 Gbps (5.0%)
```

Test Fast Reroute Operation

```
## Manually trigger local repair
user@P1> test mpls local-repair transit-lsp CRITICAL-APP

## Monitor the protection switch
user@P1> monitor interface traffic
## Watch traffic move from primary to detour interface

## Check forwarding plane
user@P1> show route forwarding-table label 100234
Routing table: default.mpls
MPLS:
Destination  Type RtRef Next hop           Type Index   NhRef Netif
100234       user     0                     ulst  1048574     2
                        10.0.2.2           Swap 200234, Push 300456(top)    583     2 ge-0/0/1.0
                        10.0.5.2           Swap 300456    584     2 ge-0/0/2.0 (!)
## (!) indicates backup, Push 300456(top) shows detour label
```

Debug Detour Computation

```
[edit protocols mpls]
user@PE1# set traceoptions file frr-debug
user@PE1# set traceoptions flag fast-reroute detail
user@PE1# commit

user@PE1> show log frr-debug
Dec 15 14:00:00 Computing detour for CRITICAL-APP at 10.0.1.2
Dec 15 14:00:00   Main path: 10.0.1.2 -> 10.0.2.2
Dec 15 14:00:00   Excluding: Link 10.0.1.2-10.0.2.2
Dec 15 14:00:00   CSPF result: 10.0.1.2 -> 10.0.5.2 -> 10.0.6.2 -> 10.0.2.2
Dec 15 14:00:00   Detour merge point: 10.0.2.2
Dec 15 14:00:00   Signaling detour LSP
```

One-to-one backup provides fast, automatic protection against failures, but at the cost of significant resource consumption. For networks with many LSPs, the next module's facility backup offers a more scalable alternative.

Module 13: RSVP-Local Repair Part 2—Facility Backup, or Node-Link-Protection

Part 1: The Conceptual Lecture (The Why)

Understanding Facility Backup

While one-to-one backup creates individual detours for each LSP (like building a personal escape route for each car), facility backup creates shared bypass tunnels that all LSPs can use (like building a bypass highway that all traffic can use during construction).

```
One-to-One vs Facility Backup:

One-to-One Backup:
100 LSPs through a link = 100 individual detour LSPs
[R1]----[R2]----[R3]
 ||||||
 ++++++--100 separate detours

Facility Backup:
100 LSPs through a link = 1 shared bypass tunnel
[R1]----[R2]----[R3]
 |               |
 +---[Bypass]----+
     (shared)
```

The Bypass Tunnel Concept

Bypass tunnels are pre-established LSPs specifically designed to protect other LSPs:

```
Bypass Tunnel Characteristics:

1. Purpose-Built:
   - Not carrying regular traffic
   - Exists solely for protection
   - Pre-signaled and ready

2. Shared Resource:
   - Multiple LSPs use same bypass
   - Efficient resource utilization
   - Single control plane entity

3. Strategic Placement:
   - Protects specific links/nodes
   - Optimally positioned
   - Handles many failures
```

Link Protection vs Node Protection

Facility backup can protect against different failure types:

```
Link Protection (Next-Hop Backup):
Protects: Link between PLR and next-hop
Bypass: PLR → Bypass → Next-hop

Example:
Normal:  [R1]----[R2]----[R3]----[R4]
Bypass:  [R1]----[R5]----[R2]
Failure:     X--------X
Result:  Traffic goes R1→R5→R2→R3→R4

Node Protection (Next-Next-Hop Backup):
Protects: Entire next-hop node
Bypass: PLR → Bypass → Next-next-hop

Example:
Normal:  [R1]----[R2]----[R3]----[R4]
Bypass:  [R1]----[R5]----[R3]
Failure:         XXXX (R2 fails)
Result:  Traffic goes R1→R5→R3→R4
```

Label Operations in Facility Backup

The magic of facility backup is in label stacking:

```
Label Stacking Process:

Original LSP labels:
[R1] --Label 100--> [R2] --Label 200--> [R3] --Label 300--> [R4]

When using bypass:
1. R1 receives packet with label 100
2. R1 normally swaps 100→200 and sends to R2
3. But R1-R2 link failed!
4. R1 pushes bypass label on top: [Bypass: 500][Original: 200]
5. R5 pops bypass label, forwards label 200 to R2
6. R2 processes label 200 normally

Packet flow:
At R1: [Label 100] → [Label 500][Label 200] (push bypass)
At R5: [Label 500][Label 200] → [Label 200] (pop bypass)
At R2: [Label 200] → [Label 300] (normal operation)
```

PLR and MP Concepts

Understanding the key players in facility backup:

```
PLR (Point of Local Repair):
- Router that detects failure
- Redirects traffic to bypass
- Performs label stacking

MP (Merge Point):
- Where bypass rejoins main path
- Link protection: Next-hop
- Node protection: Next-next-hop

Example:
[PLR/R1]----X----[R2]----[MP/R3]
    |                      |
    +-------[Bypass]-------+

R1 = PLR (redirects traffic)
R3 = MP (receives bypassed traffic)
```

Scalability Advantages

Facility backup scales much better than one-to-one:

```
Resource Comparison:

Scenario: 1000 LSPs, 5-hop average path

One-to-One Backup:
- Detour LSPs: 1000 × 4 = 4000
- Labels consumed: 4000
- State entries: 5000 total

Facility Backup:
- Bypass tunnels: ~50-100 (one per protected resource)
- Labels consumed: 100
- State entries: 1100 total

Efficiency: 40-50x fewer resources!
```

Node-Link Protection

The best of both worlds - protecting against any failure:

```
Node-Link Protection Strategy:

For each protected interface:
1. Try to create node-protecting bypass
2. If not possible, create link-protecting bypass
3. Dynamically choose based on failure type

Example:
[R1] connected to [R2] connected to [R3] and [R4]

Bypasses created:
- Node bypass: R1→R5→R3 (protects R2 failure)
- Link bypass: R1→R5→R2 (protects R1-R2 link)

If R2 fails completely: Use node bypass
If only R1-R2 link fails: Use link bypass (shorter)
```

Part 2: The Junos CLI Masterclass (The How)

Basic Facility Backup Configuration

Enable facility backup (link protection):

```
## Enable link protection globally
[edit protocols rsvp]
user@PE1# set interface all link-protection

## Or per-interface
[edit protocols rsvp]
user@PE1# set interface ge-0/0/1.0 link-protection

## On the LSP (request protection)
[edit protocols mpls]
user@PE1# set label-switched-path FACILITY-LSP to 192.168.1.4
user@PE1# set label-switched-path FACILITY-LSP link-protection
```

Node Protection Configuration

Enable more comprehensive protection:

```
## Enable node-link protection
[edit protocols rsvp]
user@PE1# set interface ge-0/0/1.0 node-link-protection

## Configure LSP to request node protection
[edit protocols mpls]
user@PE1# set label-switched-path NODE-PROTECTED to 192.168.1.4
user@PE1# set label-switched-path NODE-PROTECTED node-link-protection

## What this does:
## - Creates node-protecting bypass when possible
## - Falls back to link protection if needed
## - Provides best available protection
```

Bypass Tunnel Configuration

Manually configure bypass tunnels:

```
## Automatic bypass creation (recommended)
[edit protocols rsvp]
user@PE1# set interface ge-0/0/1.0 link-protection
```

```
## Manual bypass LSP (for specific requirements)
[edit protocols mpls]
user@PE1# set label-switched-path BYPASS-R2 to 192.168.1.2
user@PE1# set label-switched-path BYPASS-R2 install 0.0.0.0/0

## Configure bypass path
user@PE1# set path AROUND-LINK 10.0.5.2 strict
user@PE1# set path AROUND-LINK 10.0.6.2 strict
user@PE1# set path AROUND-LINK 10.0.2.2 strict
user@PE1# set label-switched-path BYPASS-R2 primary AROUND-LINK

## Associate bypass with protected interface
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set protection-bypass BYPASS-R2
```

Advanced Facility Backup Options

Configure sophisticated protection schemes:

```
## Bandwidth protection for facility backup
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set link-protection bandwidth 10g
user@PE1# set link-protection max-bypasses 4
user@PE1# set link-protection optimize-timer 300

## Admin group constraints for bypasses
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set link-protection admin-group include-any BACKUP-ALLOWED
user@PE1# set link-protection admin-group exclude EXPENSIVE

## Path optimization for bypasses
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# set link-protection path-computation loose
```

Complete Facility Backup Example

Comprehensive protection configuration:

```
## Global RSVP protection settings
set protocols rsvp {
    interface all {
        node-link-protection;
    }
    interface ge-0/0/1.0 {
        node-link-protection {
            bandwidth 10g;
            max-bypasses 2;
            optimize-timer 600;
            admin-group {
                exclude EXPENSIVE;
                include-any FIBER MICROWAVE;
            }
        }
        hello-interval 1;
        hello-multiplier 3;
    }
    interface ge-0/0/2.0 {
        link-protection {
            bandwidth 5g;
        }
    }
}

## Protected LSPs requesting facility backup
set protocols mpls {
    label-switched-path VOICE-PROTECTED {
        to 192.168.1.4;
        bandwidth 100m;
        priority 0 0;
        node-link-protection;
    }
    label-switched-path DATA-PROTECTED {
        to 192.168.1.4;
        bandwidth 1g;
        priority 3 3;
        link-protection;
    }
    label-switched-path BULK-PROTECTED {
        to 192.168.1.4;
        bandwidth 5g;
        priority 7 7;
        link-protection;
    }
}

## Explicit bypass configuration (optional)
set protocols mpls {
    label-switched-path MANUAL-BYPASS-NODE {
        to 192.168.1.3;  ## Next-next-hop
        install 0.0.0.0/0;
        no-cspf;
        primary VIA-ALTERNATE {
            10.0.5.2 strict;
            10.0.6.2 strict;
            10.0.7.2 strict;
        }
    }
}
```

Facility Backup with Traffic Engineering

Combine facility backup with TE constraints:

```
## Protected LSP with specific requirements
[edit protocols mpls label-switched-path PREMIUM-SERVICE]
user@PE1# set to 192.168.1.4
user@PE1# set bandwidth 2g
user@PE1# set admin-group include-all GOLD LOW-LATENCY
user@PE1# set node-link-protection

## Ensure bypasses meet minimum requirements
[edit protocols rsvp interface ge-0/0/1.0 node-link-protection]
user@PE1# set bandwidth 2g
user@PE1# set admin-group include-any SILVER GOLD
user@PE1# set max-bypasses 2
```

Controlling Bypass Selection

Fine-tune which bypass protects which LSPs:

```
## Create classed bypasses
[edit protocols mpls]
user@PE1# set label-switched-path GOLD-BYPASS to 192.168.1.2
user@PE1# set label-switched-path GOLD-BYPASS admin-group include-all GOLD
user@PE1# set label-switched-path GOLD-BYPASS install 0.0.0.0/0

user@PE1# set label-switched-path SILVER-BYPASS to 192.168.1.2
user@PE1# set label-switched-path SILVER-BYPASS admin-group include-all SILVER
user@PE1# set label-switched-path SILVER-BYPASS install 0.0.0.0/0

## LSPs automatically select matching bypass
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Facility Backup Verification

1. View Bypass Tunnel Status

```
user@PE1> show rsvp interface protection
Interface   Type         LSPs    Bypass            State   Bandwidth  Ratio
ge-0/0/1.0  Node-Link       3    BYPASS-10.0.2.2   Up      10Gbps     100%
                               BYPASS-10.0.3.2   Up      Node
ge-0/0/2.0  Link           5    AUTO-BYPASS-1     Up      5Gbps      80%
ge-0/0/3.0  Node-Link       2    No bypass         Down    -          -

user@PE1> show mpls lsp bypass
Ingress LSP: 3 sessions
To              From            State Rt P     ActivePath       LSPname
10.0.2.2        192.168.1.1     Up     0 *                     BYPASS-10.0.2.2
10.0.3.2        192.168.1.1     Up     0 *                     BYPASS-10.0.3.2
192.168.1.2     192.168.1.1     Up     0 *                     AUTO-BYPASS-1
Total 3 displayed, Up 3, Down 0
```

2. Check Protected LSP Status

```
user@PE1> show rsvp session protection detail
Ingress RSVP: 8 sessions

192.168.1.4
  From: 192.168.1.1, LSPstate: Up, Route: 1
  LSPname: VOICE-PROTECTED, LSPpath: Primary
  Resv style: 1 SE, Label in: -, Label out: 100234
  Protection: requested
    Type: Node-Link, Bypass: BYPASS-10.0.3.2
    PLR: 192.168.1.1 (Self), MP: 192.168.1.3, Bypass label: 200345
    Protection Available: Yes

Transit RSVP: 15 sessions
10.10.10.10
  From: 192.168.1.8, LSPstate: Up, Route: 0
  LSPname: TRANSIT-LSP, LSPpath: Primary
  Label in: 300123, Label out: 300456
  Protection: requested
    Type: Link, Bypass: AUTO-BYPASS-2
    PLR: 192.168.1.1 (Self), MP: 192.168.1.2, Bypass label: 400567
    Backup path: via ge-0/0/3.0, Push 400567
```

3. Monitor Bypass Usage

```
user@PE1> show mpls lsp bypass extensive
Ingress LSP: 3 sessions

10.0.3.2
  From: 192.168.1.1, State: Up, ActiveRoute: 0, LSPname: BYPASS-10.0.3.2
  ActivePath:  (primary)
  LSPtype: Static Configured, Penultimate hop popping
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  *Primary                    State: Up
    Priorities: 7 7
    Bandwidth: 10Gbps
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
      10.0.5.2 S 10.0.6.2 S 10.0.7.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
      10.0.5.2 10.0.6.2 10.0.7.2
    ...
    Protected LSPs: 3
      VOICE-PROTECTED, DATA-PROTECTED, BULK-PROTECTED
    Protection usage:
      Bandwidth reserved: 6.1Gbps
      Bandwidth available: 3.9Gbps
```

Common Troubleshooting Scenarios

Scenario 1: No Bypass Tunnel Created

Symptom: Link protection configured but no bypass

Diagnostic Commands:

```
user@PE1> show rsvp interface protection detail
ge-0/0/1.0
  Link protection: Enabled
  Bypass attempts: 5
  Bypass failures: 5
  Last error: CSPF failed - no alternate path

user@PE1> show ted database 192.168.1.1
NodeID: 192.168.1.1
  Type: Rtr, Age: 234 secs, LinkIn: 2, LinkOut: 2
  Protocol: OSPF(0.0.0.0)
    To: 192.168.1.2, Local: 10.0.1.1, Remote: 10.0.1.2
    To: 192.168.1.2, Local: 10.0.2.1, Remote: 10.0.2.2
    ## Both links go to same next-hop!
```

Cause: No alternate path to create bypass

Solution: Requires topology change or relaxed constraints:

```
## Allow bypass to use any available path
[edit protocols rsvp interface ge-0/0/1.0 link-protection]
user@PE1# delete admin-group
user@PE1# set no-node-protection  ## Try link-only
user@PE1# commit
```

Scenario 2: LSP Not Using Bypass

Symptom: Bypass exists but LSP not protected

Diagnostic Commands:

```
user@PE1> show rsvp session name DATA-LSP protection
192.168.1.4
  From: 192.168.1.1, LSPstate: Up, Route: 1
  LSPname: DATA-LSP, LSPpath: Primary
  Protection: not requested  ## Problem!

user@PE1> show configuration protocols mpls label-switched-path DATA-LSP | match protection
  [no output]

user@PE1> show mpls lsp name DATA-LSP extensive | match flag
  PATH flags: 0x0  ## No protection flag set
```

Cause: LSP not requesting protection

Solution:

```
[edit protocols mpls label-switched-path DATA-LSP]
user@PE1# set link-protection
user@PE1# commit

## Force LSP re-signal
user@PE1> clear mpls lsp name DATA-LSP optimize
```

Scenario 3: Bypass Bandwidth Exhausted

Symptom: Some LSPs protected, others not

Diagnostic Commands:

```
user@PE1> show rsvp interface ge-0/0/1.0 protection detail
  Link protection: Enabled
  Max bypasses: 1
  Bypass: AUTO-BYPASS-1 (Up)
    Bandwidth: 5Gbps
    Protected bandwidth: 5Gbps
    Available: 0bps

  Unprotected LSPs: 3
    LSP1 (2Gbps) - insufficient bypass bandwidth
    LSP2 (1Gbps) - insufficient bypass bandwidth
    LSP3 (500Mbps) - insufficient bypass bandwidth
```

Solution: Create additional bypasses or increase bandwidth:

```
[edit protocols rsvp interface ge-0/0/1.0 link-protection]
user@PE1# set max-bypasses 4
user@PE1# set bandwidth 10g
user@PE1# commit
```

Scenario 4: Wrong Protection Type

Symptom: Link protection when node protection desired

Diagnostic Commands:

```
user@PE1> show rsvp session protection name VOICE-PROTECTED
  Protection: requested
    Type: Link, Bypass: BYPASS-10.0.2.2
    ## Only link protection!

user@PE1> show rsvp interface ge-0/0/1.0 protection detail
  Protection: Link
  Node protection: Not configured

user@PE1> show mpls lsp bypass BYPASS-10.0.2.2
To              From            State Rt P     ActivePath       LSPname
10.0.2.2        192.168.1.1     Up     0 *                     BYPASS-10.0.2.2
## Bypass only goes to next-hop, not next-next-hop
```

Solution: Enable node protection:

```
[edit protocols rsvp interface ge-0/0/1.0]
user@PE1# delete link-protection
user@PE1# set node-link-protection
user@PE1# commit
```

Advanced Facility Backup Troubleshooting

Test Protection Switching

```
## Trigger protection switch
user@PE1> test mpls protection-switch lsp-name VOICE-PROTECTED

## Monitor bypass activation
user@PE1> monitor interface traffic
## Watch traffic move to bypass interface

## Check label stacking
user@PE1> show route forwarding-table label 100234 detail
Routing table: default.mpls
MPLS:
100234 (1 entry, 1 announced)
        *RSVP   Preference: 7/1
                Next hop type: unilist, Next hop index: 1048574
                Address: 0x9458234
                Next-hop reference count: 3
                Next hop: 10.0.1.2 via ge-0/0/1.0, selected
                  Label operation: Swap 200234
                Next hop: 10.0.5.2 via ge-0/0/2.0 (backup)
```

```
                  Label operation: Swap 200234, Push 400567(top)
                  ## Shows bypass label stacking!
```

Monitor Bypass Optimization

```
user@PE1> show rsvp interface protection optimization
Interface   Bypass           Next Opt  Last Opt   Result
ge-0/0/1.0  BYPASS-10.0.2.2  00:04:23  00:55:37  Better path found
ge-0/0/1.0  BYPASS-10.0.3.2  00:08:45  00:51:15  No change
ge-0/0/2.0  AUTO-BYPASS-1    00:02:12  00:57:48  No change

[edit protocols rsvp]
user@PE1# set traceoptions file bypass-opt
user@PE1# set traceoptions flag protection detail
user@PE1# commit

user@PE1> show log bypass-opt
Dec 20 10:00:00 Bypass optimization for ge-0/0/1.0
Dec 20 10:00:00   Current: 10.0.5.2 -> 10.0.6.2 -> 10.0.2.2 (metric 30)
Dec 20 10:00:00   Trying: exclude link 10.0.1.1-10.0.1.2
Dec 20 10:00:00   New path: 10.0.8.2 -> 10.0.2.2 (metric 20)
Dec 20 10:00:00   Signaling new bypass
```

Performance Analysis

```
user@PE1> show rsvp protection statistics
Protection Statistics:
  Interfaces with protection: 12
  Total bypasses: 18
  Active bypasses: 16

  Protected LSPs: 234
  Unprotected LSPs: 12
  Protection ratio: 95.1%

  Bypass efficiency:
    Average LSPs per bypass: 14.6
    Bandwidth efficiency: 87.3%

  Protection events (24h): 4
    Average detection: 8.2ms
    Average switch time: 3.1ms
    Average restoration: 11.3ms
```

Facility backup provides the same sub-50ms protection as one-to-one backup but with dramatically better scalability. By sharing bypass tunnels among many LSPs, it enables comprehensive network protection without overwhelming routers with thousands of detour LSPs.

Module 14: RSVP-LSP Optimization

Part 1: The Conceptual Lecture (The Why)

The Problem RSVP-LSP Optimization Solves

Imagine you're managing a highway system. When you first built the roads, you chose paths based on the best information available at that time. But what happens as traffic patterns change? What if a new, shorter route opens up? What if construction makes an existing path less efficient? In traditional highway systems, you'd need to manually redirect traffic to better routes.

In MPLS networks using RSVP-TE (Resource Reservation Protocol - Traffic Engineering), we face the same challenge. Label Switched Paths (LSPs) are like dedicated highways for network traffic. Once established, these paths remain static unless manually changed or disrupted by a failure. This creates inefficiency because:

1. **Network conditions change over time** - Links that were congested might become available
2. **New paths might become available** - New links or nodes added to the network
3. **Initial path calculations might not be optimal** - The path chosen during setup might not remain the best path

4. **Resource utilization becomes imbalanced** - Some paths remain overutilized while others sit idle

What is RSVP-LSP Optimization?

RSVP-LSP Optimization is an automatic mechanism that periodically re-evaluates established LSPs to determine if better paths exist. It's like having an intelligent traffic management system that continuously monitors all highways and automatically redirects traffic to more efficient routes when they become available.

How LSP Optimization Works

The optimization process follows this sequence:

```
Initial State                    Optimization Timer Expires
     |                                    |
     v                                    v
+------------+                    +---------------+
|    LSP     |                    |   Calculate   |
| Established|------------------->|   New Path    |
|  on Path A |                    | (CSPF Again)  |
+------------+                    +---------------+
                                          |
                                          v
                                  +---------------+
                                  | Compare Paths |
                                  |  Old vs New   |
                                  +---------------+
                                          |
                    +---------------------+---------------------+
                    |                                           |
                    v                                           v
            +---------------+                          +---------------+
            | Paths are Same|                          | New Path is   |
            | or Worse      |                          | Better        |
            | Keep Current  |                          | Signal New LSP|
            +---------------+                          +---------------+
                                                                |
                                                                v
                                                       +---------------+
                                                       | Move Traffic  |
                                                       | (Make-Before- |
                                                       | Break)        |
                                                       +---------------+
```

Key Concepts in LSP Optimization

1. **Optimization Timer**: Defines how frequently the router re-evaluates paths (default: 3600 seconds/1 hour)
2. **Optimization Criteria**: What makes a path "better"?
   - **Fewer hops** (lower IGP metric)
   - **More available bandwidth**
   - **Lower latency** (if configured)
   - **Better adherence to constraints** (admin groups, etc.)
3. **Make-Before-Break**: The optimization process uses this technique to ensure zero packet loss during path changes:

   ```
   Time 0: Traffic flows on original LSP
   Time 1: New, better path calculated
   Time 2: New LSP signaled and established
   Time 3: Traffic moved to new LSP
   Time 4: Original LSP torn down
   ```

4. **Optimization Triggers**:
   - **Timer-based**: Regular intervals (most common)
   - **Event-based**: Network topology changes

- **Manual**: Administrator-initiated

Why Use LSP Optimization?

Consider this network scenario:

```
Initial Network State (Time 0):

    R1 ============ R2              Legend:
    ||              ||              === High bandwidth link (10G)
    ||              ||              --- Low bandwidth link (1G)
    ||              ||              *** LSP Path
    R3 ------------ R4

LSP from R1 to R4: R1 -> R3 -> R4 (using low bandwidth path)
```

```
Network State After Upgrade (Time 1):

    R1 ============ R2              New high-speed link added
    ||              ||              between R2 and R4
    ||              ||
    R3 ------------ R4 ============ (new 10G link)

Without optimization: LSP still uses R1 -> R3 -> R4
With optimization: LSP automatically moves to R1 -> R2 -> R4
```

Part 2: The Junos CLI Masterclass (The How)

Understanding the Configuration Hierarchy

LSP optimization is configured under the MPLS LSP definition:

```
[edit protocols mpls]
label-switched-path <lsp-name> {
    optimize-timer <seconds>;        # When to optimize
    optimize-aggressive;             # How aggressively to optimize
    least-fill;                     # Optimization algorithm
    most-fill;                      # Alternative algorithm
    random;                         # Another alternative
}
```

Step-by-Step Configuration

Let's configure LSP optimization for a production network:

```
[edit protocols mpls]
user@R1# set label-switched-path TO-R4 to 10.0.0.4
user@R1# set label-switched-path TO-R4 optimize-timer 300

# This creates an LSP that re-optimizes every 5 minutes (300 seconds)
```

Complete Configuration Example

Here's a comprehensive LSP configuration with optimization:

```
[edit protocols mpls]
label-switched-path TO-R4 {
    to 10.0.0.4;                    # Destination router
    optimize-timer 1800;            # Optimize every 30 minutes
    bandwidth 100m;                 # Request 100 Mbps
    setup-priority 3;               # Higher priority for setup
    hold-priority 3;                # Maintain priority
    primary DYNAMIC {               # Primary path
```

```
        standby;                    # Pre-signal backup
    }
    secondary BACKUP {              # Secondary path
        standby;                    # Pre-signal this too
    }
}

[edit protocols mpls]
# Enable optimization features globally
optimize-aggressive;                # Use aggressive optimization
optimize-switchover-delay 60;       # Wait 60 seconds before switching
```

Advanced Optimization Configuration

For more control over optimization behavior:

```
[edit protocols mpls]
label-switched-path PRODUCTION-LSP {
    to 10.0.0.100;
    optimize-timer 600;             # Check every 10 minutes

    # Define what constitutes a "better" path
    metric-type {
        igp;                        # Use IGP metric (default)
        # te;                       # Use TE metric
        # delay;                    # Use delay metric
    }

    # Control optimization aggressiveness
    optimization-preference {
        prune-limit 5;              # Consider top 5 paths only
    }

    # Prevent flapping
    optimize-hold-dead-delay 300;   # Wait 5 min after failure
}

# Global optimization settings
[edit protocols mpls]
optimize-timer 3600;                # Global default: 1 hour
optimize-aggressive;                # Aggressive optimization
no-cspf;                           # Disable for specific LSPs
```

Load Balancing Algorithm Configuration

Configure how the optimization algorithm distributes traffic:

```
[edit protocols mpls]
label-switched-path LOAD-BALANCED-LSP {
    to 10.0.0.50;
    optimize-timer 900;

    # Choose load-balancing algorithm
    least-fill;                     # Prefer least utilized paths
    # most-fill;                    # Pack links efficiently
    # random;                       # Random selection
}
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Verification Commands

1. **Check LSP optimization status:**

```
user@R1> show mpls lsp name TO-R4 extensive
Ingress LSP: 1 sessions
10.0.0.4
  From: 10.0.0.1, State: Up, ActiveRoute: 0, LSPname: TO-R4
  ActivePath:  (primary)
  Optimize timer: 1800 seconds, Next optimization in 1234 seconds
  Optimization count: 5, Last optimization: 2024-01-10 14:30:15
  Last optimization result: Better path found, switched successfully
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  *Primary   State: Up
    Priorities: 3 3
    OptimizeTimer: 1800
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops):
      10.1.1.2 S 10.2.2.2 S 10.3.3.2 S 10.0.0.4 S
    Received RRO:
      10.1.1.2 10.2.2.2 10.3.3.2 10.0.0.4
```

2. **Monitor optimization events:**

```
user@R1> show mpls lsp optimization-history
LSP: TO-R4
  2024-01-10 14:30:15: Optimization successful - new path: R1->R2->R4
  2024-01-10 14:00:15: Optimization attempted - no better path found
  2024-01-10 13:30:15: Optimization successful - new path: R1->R2->R3->R4
  2024-01-10 13:00:15: Optimization attempted - no better path found
```

3. **Check RSVP statistics:**

```
user@R1> show rsvp statistics
RSVP packet statistics:
  Packets sent: 15234
    Path: 4521
    PathTear: 23
    Resv: 4498
    ResvTear: 19
    PathErr: 2
    ResvErr: 0
    ResvConf: 4498

  Optimization statistics:
    Optimizations attempted: 48
    Successful optimizations: 12
    Failed optimizations: 36
    Average optimization time: 2.3 seconds
```

Common Troubleshooting Scenarios

Scenario 1: LSP Not Optimizing Despite Better Path Available

Symptom:

```
user@R1> show mpls lsp name TO-R4
  Last optimization: Never
  Optimization count: 0
```

Diagnosis:

```
user@R1> show configuration protocols mpls label-switched-path TO-R4
# No optimize-timer configured!

user@R1> show mpls lsp name TO-R4 extensive | match "timer|optim"
  OptimizeTimer: disabled
```

Solution:

```
[edit protocols mpls]
user@R1# set label-switched-path TO-R4 optimize-timer 1800
user@R1# commit
```

Scenario 2: LSP Optimizing Too Frequently (Flapping)

Symptom:

```
user@R1> show mpls lsp optimization-history
LSP: UNSTABLE-LSP
  14:30:15: Optimization: new path via R2
  14:25:15: Optimization: new path via R3
  14:20:15: Optimization: new path via R2
  14:15:15: Optimization: new path via R3
  # Path changing every 5 minutes!
```

Diagnosis:

```
user@R1> show configuration protocols mpls label-switched-path UNSTABLE-LSP
optimize-timer 300;  # Too aggressive!

user@R1> show ted database extensive | match "bandwidth|reservable"
# Shows bandwidth constantly fluctuating on links
```

Solution:

```
[edit protocols mpls]
# Increase optimization timer
user@R1# set label-switched-path UNSTABLE-LSP optimize-timer 3600

# Add hold-down timer
user@R1# set label-switched-path UNSTABLE-LSP optimize-hold-dead-delay 600

# Require significant improvement before switching
user@R1# set label-switched-path UNSTABLE-LSP optimization-threshold 10
user@R1# commit
```

Scenario 3: Optimization Failing Due to Constraints

Symptom:

```
user@R1> show mpls lsp name CONSTRAINED-LSP extensive
  Last optimization result: Failed - no path meeting constraints
  Current path metric: 30

user@R1> show log messages | match RSVP
Jan 10 14:30:15 R1 RPD_RSVP_LSP_OPTIMIZE_FAILED: TO-R4:
  Optimization failed, no path satisfying constraints admin-group
```

Diagnosis:

```
user@R1> show configuration protocols mpls label-switched-path CONSTRAINED-LSP
admin-group include red;
bandwidth 5g;

user@R1> show ted database extensive | match "admin|bandwidth"
# Shows limited links with 'red' admin-group and 5G available
```

Solution:

```
[edit protocols mpls label-switched-path CONSTRAINED-LSP]
# Option 1: Relax constraints during optimization
user@R1# set optimize-on-change link-protection

# Option 2: Use loose constraints for optimization
user@R1# set admin-group include-any [ red blue ]

# Option 3: Allow optimization to ignore some constraints
user@R1# set optimization-preemption disabled
user@R1# commit
```

Scenario 4: Traffic Loss During Optimization

Symptom:

```
# User reports brief packet loss every hour
user@R1> show log messages | match "LSP|optimize"
Jan 10 14:00:00 R1 RPD_RSVP_LSP_SWITCH: TO-R4: Path switched for optimization
Jan 10 14:00:01 R1 RPD_RSVP_NONFACILITY_SWITCH: 1 second traffic interruption
```

Diagnosis:

```
user@R1> show configuration protocols mpls label-switched-path TO-R4
# Missing make-before-break configuration

user@R1> show rsvp session name TO-R4 detail
  Style: Fixed-Filter (no shared resources)
```

Solution:

```
[edit protocols mpls]
# Enable adaptive mode for make-before-break
user@R1# set label-switched-path TO-R4 adaptive

# Enable shared explicit style for resource sharing
user@R1# set label-switched-path TO-R4 optimize-switchover-delay 30

# Ensure soft-preemption
user@R1# set label-switched-path TO-R4 soft-preemption

user@R1# commit
```

Best Practices Summary

1. **Start Conservative**: Begin with longer optimization timers (3600 seconds) and decrease gradually
2. **Monitor Impact**: Track optimization history before making aggressive changes
3. **Consider Network Stability**: Unstable networks need longer timers and hold-down delays
4. **Test Constraints**: Ensure optimization can find paths meeting your constraints
5. **Enable Make-Before-Break**: Always use adaptive mode to prevent traffic loss

---

Module 15: RSVP Make-Before-Break and Adaptive

Part 1: The Conceptual Lecture (The Why)

The Problem Make-Before-Break Solves

Imagine you're crossing a river on stepping stones. To move from one stone to another, you have two choices:

1. **Break-Before-Make**: Jump off your current stone, hoping to land on the next one. For a moment, you're in mid-air with no support.
2. **Make-Before-Break**: Step onto the next stone while keeping one foot on the current stone. Only when you're secure on the new stone do you lift your other foot.

In MPLS networks, the same principle applies when moving traffic from one LSP to another. Traditional "Break-Before-Make" causes packet loss because:

1. **The old path is torn down before the new path is ready**
2. **Traffic has nowhere to go during the transition**
3. **Even brief interruptions (milliseconds) can disrupt sensitive applications**

What is Make-Before-Break?

Make-Before-Break (MBB) is a mechanism that establishes a new LSP before tearing down the old one, ensuring continuous packet forwarding during path changes. It's like building a temporary bridge next to an old bridge, moving traffic to the new bridge, then demolishing the old one.

Understanding Resource Double-Counting

The most challenging aspect of MBB is the "double-counting" problem. Here's why:

```
Network with 10Gbps link capacity:

Before MBB:
Link R2-R3: 5Gbps used by LSP-A, 5Gbps available

During MBB (without Shared Explicit):
Old LSP-A path: Still reserves 5Gbps (not yet torn down)
New LSP-A path: Trying to reserve 5Gbps
Total requested: 10Gbps on same link = REJECTED!

The same LSP appears to need 10Gbps when it really only needs 5Gbps
```

How Shared Explicit Solves Double-Counting

Shared Explicit (SE) style allows RSVP to recognize that the old and new paths belong to the same LSP, preventing double-counting:

```
RSVP Session Identification:

Standard LSP:
  Session ID: (Destination, Tunnel-ID, Extended-Tunnel-ID)
  Example: (10.0.0.4, 100, 10.0.0.1)

With Shared Explicit:
  Old Path: (10.0.0.4, 100, 10.0.0.1) + Sender = 10.0.0.1:1
  New Path: (10.0.0.4, 100, 10.0.0.1) + Sender = 10.0.0.1:2
                Same session ─────────┘         Different senders ┘

Result: RSVP knows both paths are for the same LSP!
```

The Adaptive Feature

"Adaptive" in Junos automatically enables several make-before-break behaviors:

1. **Automatic SE Style**: Uses Shared Explicit without manual configuration
2. **Smart Bandwidth Sharing**: Prevents double-counting automatically
3. **Optimization Integration**: Works seamlessly with LSP optimization
4. **Revert Timer**: Can automatically revert to primary paths

Traffic Mapping to LSPs

Once you have LSPs, you need to direct specific traffic into them. This is like having express lanes on a highway - you need signs (policies) telling certain vehicles (packets) to use specific lanes (LSPs).

```
Traffic Classification and LSP Mapping:

   Incoming Traffic                Policy Decision              Forwarding

   Packet to 10.1.1.1  ──────>  Match: VoIP traffic   ──────>  Use LSP-VOICE
   Packet to 10.2.2.2  ──────>  Match: Bulk data      ──────>  Use LSP-BULK
   Packet to 10.3.3.3  ──────>  No match              ──────>  Use IGP path
```

Part 2: The Junos CLI Masterclass (The How)

Understanding the Configuration Components

Make-Before-Break involves three main configuration areas:

```
[edit protocols mpls]
label-switched-path <name> {
    adaptive;                    # Enable MBB with SE
    soft-preemption;            # Graceful preemption
    optimize-switchover-delay;   # MBB timing control
}

[edit policy-options]
policy-statement <name> {       # Map traffic to LSPs
    term <term-name> {
        from {
            protocol bgp;
            community VOICE;
        }
        then {
            install-nexthop lsp <lsp-name>;
        }
    }
}
```

Step-by-Step Configuration

1. Basic Make-Before-Break Setup

```
[edit protocols mpls]
# Create LSP with adaptive (MBB) enabled
user@R1# set label-switched-path TO-CORE-ROUTER to 10.0.0.100
user@R1# set label-switched-path TO-CORE-ROUTER adaptive

# This single command enables:
# - Shared Explicit style
# - Make-before-break for all path changes
# - Bandwidth sharing between old and new paths
```

2. Configure Shared Explicit for Non-Adaptive LSPs

```
[edit protocols mpls]
# For LSPs that need SE without full adaptive features
user@R1# set label-switched-path LEGACY-LSP to 10.0.0.50
user@R1# set label-switched-path LEGACY-LSP inter-domain

# Manual SE configuration
user@R1# set signaling-protocols rsvp shared-explicit-style
```

3. Advanced MBB Configuration

```
[edit protocols mpls]
label-switched-path ADVANCED-MBB-LSP {
    to 10.0.0.200;
    adaptive;                           # Enable MBB
    optimize-switchover-delay 30;       # Wait 30 sec before switching
    soft-preemption {                   # Graceful preemption
        cleanup-timer 60;               # Keep old path for 60 sec
    }
    retry-timer 30;                     # Retry failed MBB after 30 sec
    retry-limit 3;                      # Maximum 3 MBB attempts

    # Control when MBB is triggered
    reoptimize-on-evaluation;          # During periodic optimization
    normalize {                         # Revert to primary when available
        interval 300;                   # Check every 5 minutes
    }
}
```

Complete Production Configuration Example

Here's a comprehensive setup for a service provider network:

```
[edit protocols mpls]
# Global MPLS settings
admin-groups {
    VOICE 0;
    VIDEO 1;
    BULK 2;
    CRITICAL 3;
}

# Voice LSP with MBB
label-switched-path VOICE-LSP {
    to 10.0.0.100;
    adaptive;                          # Enable MBB
    bandwidth 1g;                      # Reserve 1 Gbps
    setup-priority 1;                  # High priority
    hold-priority 1;
    admin-group include VOICE;         # Voice links only

    # MBB-specific settings
    optimize-switchover-delay 10;      # Quick switchover for voice
    soft-preemption;                   # No packet loss

    # Primary and standby paths
    primary VIA-CORE {
        standby;                       # Pre-signaled
    }
    secondary VIA-EDGE {
        standby;                       # Pre-signaled backup
    }
}

# Bulk data LSP with different MBB behavior
label-switched-path BULK-LSP {
    to 10.0.0.100;
    adaptive;
    bandwidth 5g;
    setup-priority 7;                  # Low priority
    hold-priority 7;

    # Less aggressive MBB for bulk traffic
    optimize-switchover-delay 60;      # Can wait longer
    retry-timer 120;                   # Less frequent retries
}
```

Traffic Mapping Configuration

Now let's map specific traffic to these LSPs:

```
[edit policy-options]
# Define communities
community VOICE-TRAFFIC members 65000:100;
community BULK-TRAFFIC members 65000:200;

# Create policy to map traffic to LSPs
policy-statement MAP-TO-LSP {
    # Voice traffic to Voice LSP
    term VOICE {
        from {
            protocol bgp;
            community VOICE-TRAFFIC;
        }
        then {
            install-nexthop lsp VOICE-LSP;
            accept;
        }
    }
    # Bulk traffic to Bulk LSP
    term BULK {
        from {
            protocol bgp;
            community BULK-TRAFFIC;
        }
        then {
            install-nexthop lsp BULK-LSP;
            accept;
        }
    }

    # Everything else uses normal routing
    term DEFAULT {
        then accept;
    }
}

[edit protocols bgp]
# Apply the policy for BGP routes
group INTERNAL {
    type internal;
    import MAP-TO-LSP;
}

# Alternative: Apply to forwarding table
[edit routing-options]
forwarding-table {
    export MAP-TO-LSP;
}
```

Static Route Mapping to LSP

For specific destinations:

```
[edit routing-options]
static {
    # Route specific prefix through LSP
    route 192.168.100.0/24 {
        lsp-next-hop VOICE-LSP;
    }

    # With backup via another LSP
    route 192.168.200.0/24 {
        qualified-next-hop BULK-LSP {
            preference 10;
        }
        qualified-next-hop VOICE-LSP {
            preference 20;
        }
    }
}
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Verification Commands

1. **Verify MBB Configuration and Status:**

```
user@R1> show mpls lsp name VOICE-LSP extensive
Ingress LSP: 1 sessions
10.0.0.100
  From: 10.0.0.1, State: Up, ActiveRoute: 1, LSPname: VOICE-LSP
  ActivePath: VIA-CORE (primary)
  Adaptive                                    #### MBB enabled ####
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  LSP Attributes:
    Shared-explicit style                     #### SE active ####
    Soft-preemption desired
  *Primary VIA-CORE State: Up
    State: Up for 00:10:32
    Current bandwidth: 1 Gbps
    Reoptimization in 00:49:28
    Make-before-break pending: No            #### MBB status ####
    Switchover delay: 10 seconds
```

2. **Monitor MBB Events:**

```
user@R1> show mpls lsp make-before-break
LSP: VOICE-LSP
  Current MBB Status: Idle
  Last MBB Event: 2024-01-10 14:30:00
  MBB Trigger: Optimization
  Old Path: 10.1.1.1 -> 10.1.1.2 -> 10.0.0.100
  New Path: 10.2.2.1 -> 10.2.2.2 -> 10.0.0.100
  Switchover Time: 2.3 seconds
  Packets Lost: 0

LSP: BULK-LSP
  Current MBB Status: In Progress
  MBB Started: 2024-01-10 14:35:30
  Phase: New path signaling
  Progress: 75% complete
```

3. **Check Traffic Mapping:**

```
user@R1> show route forwarding-table destination 192.168.100.1
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop       Type Index    NhRef Netif
192.168.100.0/24   user     0                indr  1048580     3
                              10.0.0.100     Push 100432      VOICE-LSP

user@R1> show route protocol bgp community VOICE-TRAFFIC detail
192.168.100.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect, Next hop index: 0
                Next hop: 10.0.0.100 via lsp VOICE-LSP   #### Mapped to LSP ####
                Protocol next hop: 10.0.0.100
                Label operation: Push 100432
                Communities: 65000:100
```

Common Troubleshooting Scenarios

Scenario 1: MBB Failing Due to Resource Constraints

Symptom:

```
user@R1> show log messages | match "MBB|make-before-break"
RPD_RSVP_MBB_FAILED: VOICE-LSP: Make-before-break failed:
  Admission control failure

user@R1> show mpls lsp name VOICE-LSP
  Make-before-break pending: Yes
  MBB Attempts: 5
  Last MBB Failure: Bandwidth unavailable
```

Diagnosis:

```
user@R1> show rsvp interface detail
  ge-0/0/0.0: Subscribed bandwidth: 9.8 Gbps/10 Gbps
  ge-0/0/1.0: Subscribed bandwidth: 10 Gbps/10 Gbps (FULL!)

user@R1> show mpls lsp name VOICE-LSP extensive | match "band|priori"
  Current bandwidth: 1 Gbps
  Setup priority: 1, Hold priority: 1

# Check if SE is working
user@R1> show configuration protocols mpls label-switched-path VOICE-LSP
# adaptive; is configured - good!
```

Solution:

```
[edit protocols mpls]
# Option 1: Allow bandwidth oversubscription during MBB
user@R1# set label-switched-path VOICE-LSP soft-preemption

# Option 2: Use higher setup priority to preempt others
user@R1# set label-switched-path VOICE-LSP setup-priority 0

# Option 3: Enable interface oversubscription
[edit protocols rsvp]
user@R1# set interface ge-0/0/1.0 subscription 120

user@R1# commit
```

Scenario 2: Traffic Loss During Path Switch

Symptom:

```
# Users report brief outages
# Monitoring shows packet loss spikes every hour

user@R1> show mpls lsp statistics
VOICE-LSP:
  Packets: 1234567890, Bytes: 98765432100
  Packet loss during last transition: 1523 packets
  Transition events: 12
```

Diagnosis:

```
user@R1> show configuration protocols mpls label-switched-path VOICE-LSP
to 10.0.0.100;
# No 'adaptive' keyword!

user@R1> show mpls lsp name VOICE-LSP extensive | match "style|adaptive"
  # No output - not using shared-explicit style!
```

Solution:

```
[edit protocols mpls label-switched-path VOICE-LSP]
# Enable adaptive for automatic MBB
user@R1# set adaptive

# Add switchover delay for smoother transition
user@R1# set optimize-switchover-delay 20

# Enable soft preemption
user@R1# set soft-preemption cleanup-timer 30

user@R1# commit
```

Scenario 3: Traffic Not Using Configured LSP

Symptom:

```
user@R1> show mpls lsp name VOICE-LSP
  State: Up
  Total packets: 0    #### No traffic! ####

user@R1> show route 192.168.100.0/24
192.168.100.0/24   *[BGP/170] 00:05:00
```

```
                    > to 10.1.1.2 via ge-0/0/0.0
                      # Not using LSP!
```

Diagnosis:

```
user@R1> show configuration policy-options policy-statement MAP-TO-LSP
# Policy exists...

user@R1> show configuration protocols bgp group INTERNAL
type internal;
# Missing: import MAP-TO-LSP;!

user@R1> show route 192.168.100.0/24 detail
  # No community shown - routes don't have expected community
```

Solution:

```
# Fix 1: Apply the policy
[edit protocols bgp group INTERNAL]
user@R1# set import MAP-TO-LSP

# Fix 2: Ensure routes have correct communities
[edit policy-options policy-statement TAG-ROUTES]
user@R1# set term VOICE from route-filter 192.168.100.0/24 exact
user@R1# set term VOICE then community add VOICE-TRAFFIC

[edit protocols bgp group EXTERNAL]
user@R1# set export TAG-ROUTES

# Fix 3: Use forwarding-table export for all routes
[edit routing-options]
user@R1# set forwarding-table export MAP-TO-LSP

user@R1# commit
```

Scenario 4: MBB Causing Routing Loops

Symptom:

```
user@R1> show mpls lsp
VOICE-LSP    10.0.0.100    Up        2  # 2 active paths!?

user@R1> traceroute 192.168.100.1 no-resolve
 1  10.1.1.2  1.123 ms
 2  10.1.1.3  2.456 ms
 3  10.1.1.2  3.789 ms    # Loop!
 4  10.1.1.3  5.012 ms
```

Diagnosis:

```
user@R1> show mpls lsp name VOICE-LSP extensive
  Primary VIA-CORE State: Up
    Make-before-break: Switchover in progress
    Old path active: Yes
    New path active: Yes     # Both paths active!

user@R1> show route forwarding-table | match "Push.*Push"
  # Multiple labels being pushed - indicates loop
```

Solution:

```
[edit protocols mpls label-switched-path VOICE-LSP]
# Ensure proper cleanup of old paths
user@R1# set adaptive
user@R1# set soft-preemption cleanup-timer 10

# Add loop detection
```

```
user@R1# set no-install-to-address

# Force immediate cleanup
user@R1# clear mpls lsp name VOICE-LSP optimize

# Monitor the cleanup
user@R1> monitor start messages
user@R1> request mpls lsp cleanup
```

MBB Best Practices Summary

1. **Always use adaptive**: It automatically handles most MBB complexities
2. **Test switchover delays**: Start with higher values (30-60 seconds) and decrease based on requirements
3. **Monitor resource availability**: Ensure sufficient bandwidth for temporary double allocation
4. **Use soft-preemption**: Prevents packet loss during preemption events
5. **Implement proper traffic mapping**: Test policies thoroughly before production deployment

---

Module 16: LDP - The Label Distribution Protocol

Part 1: The Conceptual Lecture (The Why)

The Problem LDP Solves

Imagine you're running a postal service across a country. With RSVP-TE, it's like having to manually plan every single mail route from every post office to every other post office. If you have 100 offices, you'd need to plan 100 × 99 = 9,900 individual routes! This is the scalability challenge RSVP-TE faces.

LDP (Label Distribution Protocol) solves this by implementing an automatic "postal code" system. Each post office simply announces "I handle mail for postal code 12345" to its neighbors, and the system automatically figures out all possible routes.

What Makes LDP Different from RSVP-TE?

Let's compare these two MPLS signaling protocols:

```
RSVP-TE (Like Express Delivery Service):
- Explicitly routed paths (you choose exact route)
- Traffic engineering capabilities
- Bandwidth reservations
- Complex configuration
- Requires careful planning
- Best for: Service provider core networks

LDP (Like Regular Postal Service):
- Follows IGP shortest path
- Automatic label distribution
- No bandwidth reservations
- Simple configuration
- Self-organizing
- Best for: Simple MPLS, L3VPNs, Pseudowires
```

How LDP Works

LDP creates a full mesh of LSPs automatically by following these principles:

1. **Follow the IGP**: LDP paths follow your IGP (OSPF/IS-IS) shortest paths
2. **Advertise for all routes**: Each router advertises labels for all its known prefixes
3. **Build from destination**: LSPs are built from egress to ingress (opposite of RSVP)

Here's the label distribution process:

```
Step 1: Router announces its loopback
        R4: "I am 10.0.0.4"
```

```
Step 2: R4 creates label for itself
        R4: "To reach 10.0.0.4, use label 3 (implicit null)"

Step 3: R4's neighbors create labels
        R3: "To reach 10.0.0.4, use label 1004 (send to R4)"
        R2: "To reach 10.0.0.4, use label 2004 (send to R4)"

Step 4: Process continues hop by hop
        R1: "To reach 10.0.0.4, use label 3004 (send to R3)"

Result: Automatic LSP from anywhere to R4!
```

Key LDP Concepts

1. LDP Sessions vs Adjacencies

```
LDP Adjacency: Hello relationship (like saying "Hi, neighbor!")
- Formed on each interface
- Uses UDP port 646
- Multicast to 224.0.0.2
- Keepalive: 15 seconds

LDP Session: TCP relationship (like having a conversation)
- One per neighbor router
- Uses TCP port 646
- Unicast between loopbacks
- Exchanges actual labels
```

2. Label Distribution Methods

LDP can distribute labels in different ways:

```
Liberal Label Retention (Default):
R1 receives labels from all neighbors:
  R2: "Use label 100 for 10.0.0.4"
  R3: "Use label 200 for 10.0.0.4"
R1 keeps BOTH labels (even if R3 isn't best path)
Advantage: Fast convergence
Disadvantage: More memory usage

Conservative Label Retention:
R1 only keeps label from best path (R2)
Advantage: Less memory
Disadvantage: Slower convergence
```

3. FEC (Forwarding Equivalence Class)

A FEC is a group of packets that get the same treatment:

```
Examples of FECs in LDP:
- All packets to 10.0.0.4/32 = FEC 1
- All packets to 192.168.1.0/24 = FEC 2
- All packets to 2001:db8::/64 = FEC 3 (LDP supports IPv6!)

Each FEC gets one label binding.
```

Why Use LDP?

1. **Simplicity**: Near-zero configuration required
2. **Scalability**: Automatically creates full mesh

3. **Integration**: Works perfectly with L3VPNs
4. **Reliability**: Self-healing and automatic
5. **Vendor Interoperability**: Widely supported standard

Part 2: The Junos CLI Masterclass (The How)

Basic LDP Configuration Structure

The LDP configuration hierarchy in Junos:

```
[edit protocols ldp]
interface <interface-name>;      # Where to run LDP
family {
    inet;                       # IPv4 labels
    inet6;                      # IPv6 labels
}
track-igp-metric;              # Follow IGP costs
deaggregate;                   # Advertise specific routes
```

Step-by-Step Basic LDP Configuration

Let's build a simple LDP network:

```
# Step 1: Enable LDP on interfaces
[edit protocols ldp]
user@R1# set interface ge-0/0/0.0
user@R1# set interface ge-0/0/1.0
user@R1# set interface lo0.0

# Step 2: Enable MPLS on the same interfaces
[edit protocols mpls]
user@R1# set interface ge-0/0/0.0
user@R1# set interface ge-0/0/1.0

# Step 3: Ensure IGP includes these interfaces
[edit protocols ospf area 0.0.0.0]
user@R1# set interface ge-0/0/0.0
user@R1# set interface ge-0/0/1.0
user@R1# set interface lo0.0

# Step 4: Enable MPLS family on interfaces
[edit interfaces]
user@R1# set ge-0/0/0 unit 0 family mpls
user@R1# set ge-0/0/1 unit 0 family mpls
```

Complete Production LDP Configuration

Here's a comprehensive LDP setup:

```
[edit protocols ldp]
# Global LDP settings
keep-multiplier 6;                    # LDP keepalive = 6 × 10 = 60 seconds
interface-hello-hold-time 45;         # Interface keepalive
interface-hello-interval 15;          # Hello interval

# Enable on all core interfaces
interface ge-0/0/0.0;
interface ge-0/0/1.0;
interface ae0.0;                      # Aggregated interfaces too
interface lo0.0;                      # Always include loopback!

# Advanced features
track-igp-metric;                     # LDP follows IGP metrics
deaggregate;                         # Advertise connected routes
explicit-null;                        # Use explicit null (label 0)

# Session protection (explained in Module 18)
```

```
session-protection;

# Label retention and distribution
label-retention liberal;              # Keep all labels (default)
# label-retention conservative;       # Alternative: save memory

# Transport address (source for TCP sessions)
transport-address router-id;          # Use loopback

# Graceful restart
graceful-restart {
    helper-enable;                    # Help restarting neighbors
    recovery-time 240;                # Wait 4 minutes
    maximum-neighbor-recovery-time 240;
}

# Import/Export policies (Module 19)
import FILTER-LDP-LABELS;
export ADVERTISE-PREFIXES;

[edit protocols mpls]
# Don't forget MPLS on the same interfaces!
interface all;                        # Enable on all interfaces
interface fxp0.0 {                   # Except management
    disable;
}
```

LDP for Different Address Families

```
[edit protocols ldp]
# IPv4 and IPv6 support
family {
    inet {
        # IPv4 LDP settings
        label-retention liberal;
        transport-preference ipv4;    # Prefer IPv4 for transport
    }
    inet6 {
        # IPv6 LDP settings
        label-retention liberal;
        transport-preference ipv6;    # Prefer IPv6 for transport
    }
}

# Specify family per interface if needed
interface ge-0/0/0.0 {
    family {
        inet;                         # IPv4 only on this interface
    }
}
interface ge-0/0/1.0 {
    family {
        inet;
        inet6;                        # Both families on this interface
    }
}
```

Targeted LDP Sessions

For special cases like L2VPN:

```
[edit protocols ldp]
# Configure targeted sessions (not discovered via multicast)
session 10.0.0.100 {                  # Remote PE for L2VPN
    authentication-key "$9$aes256..."; # MD5 authentication
    authentication-algorithm aes-256;
}

# Alternative: Auto-targeted for L2VPN
```

```
targeted-hello {
    hold-time 45;
    interval 15;
}
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Verification Commands

1. **Check LDP Interfaces and Neighbors:**

```
user@R1> show ldp interface
Interface            Label space ID        Nbr count   Next hello
ge-0/0/0.0           10.0.0.1:0                   1   2
ge-0/0/1.0           10.0.0.1:0                   1   5
lo0.0                10.0.0.1:0                   0   0

user@R1> show ldp neighbor
Address            Interface          Label space ID         Hold time
10.1.1.2           ge-0/0/0.0         10.0.0.2:0                  42
10.2.2.2           ge-0/0/1.0         10.0.0.3:0                  44
```

2. **Verify LDP Sessions:**

```
user@R1> show ldp session
  Neighbor                     State       Role    Hold time  Adv. Mode
10.0.0.2                   Operational  Active        28  Liberal
10.0.0.3                   Operational  Passive       26  Liberal

user@R1> show ldp session detail
Neighbor: 10.0.0.2
  State: Operational
  Local address: 10.0.0.1, Remote address: 10.0.0.2
  Connection: Local: 10.0.0.1:52341 Remote: 10.0.0.2:646
  Session ID: 10.0.0.1:0--10.0.0.2:0
  Hold time: 30, Proposed hold time: 45
  Keepalive interval: 10
  Label advertisement mode: Downstream unsolicited
  Label retention mode: Liberal
  Local maximum label range: 16 ~ 1048575
```

3. **Check Label Bindings:**

```
user@R1> show ldp database
Input label database, 10.0.0.1:0--10.0.0.2:0
  Label     Prefix
      3     10.0.0.2/32
 100000     10.0.0.3/32
 100001     10.0.0.4/32
 100002     192.168.1.0/24

Output label database, 10.0.0.1:0--10.0.0.2:0
  Label     Prefix
      3     10.0.0.1/32
 200000     10.0.0.3/32
 200001     10.0.0.4/32

user@R1> show route table inet.3
10.0.0.4/32        *[LDP/9] 00:10:00, metric 30
                    > to 10.1.1.2 via ge-0/0/0.0, Push 100001
```

Common Troubleshooting Scenarios

Scenario 1: LDP Session Not Establishing

Symptom:

```
user@R1> show ldp neighbor
Address            Interface          Label space ID         Hold time
10.1.1.2           ge-0/0/0.0         10.0.2:0                    13

user@R1> show ldp session
  Neighbor                     State       Role    Hold time
10.0.0.2                   Nonexistent   --        --
```

Diagnosis:

```
# Check if routers can reach each other's loopbacks
user@R1> ping 10.0.0.2 source 10.0.0.1
PING 10.0.0.2: 56 data bytes
ping: sendto: No route to host

# Check IGP
user@R1> show route 10.0.0.2
  # No route!

# Check MPLS interfaces
user@R1> show mpls interface
Interface        State       Administrative groups (x: extended)
ge-0/0/0.0       Up         <none>
# ge-0/0/1.0 missing!

# Check family mpls
user@R1> show configuration interfaces ge-0/0/0
unit 0 {
    family inet {
        address 10.1.1.1/30;
    }
    # Missing: family mpls
}
```

Solution:

```
# Fix 1: Add MPLS family
[edit interfaces ge-0/0/0]
user@R1# set unit 0 family mpls

# Fix 2: Ensure IGP has loopback
[edit protocols ospf area 0]
user@R1# set interface lo0.0

# Fix 3: Check firewall filters
[edit firewall family inet filter PROTECT-RE]
user@R1# set term ALLOW-LDP from protocol tcp
user@R1# set term ALLOW-LDP from port 646
user@R1# set term ALLOW-LDP then accept

user@R1# commit
```

Scenario 2: Labels Not Being Distributed

Symptom:

```
user@R1> show ldp database
Input label database, 10.0.0.1:0--10.0.0.2:0
  Label     Prefix
      3     10.0.0.2/32
  # Only receiving implicit null for directly connected!

Output label database, 10.0.0.1:0--10.0.0.2:0
  Label     Prefix
      3     10.0.0.1/32
  # Only advertising own loopback!
```

Diagnosis:

```
# Check what routes exist
user@R1> show route protocol ospf
10.0.0.3/32        *[OSPF/10] 00:05:00, metric 20
10.0.0.4/32        *[OSPF/10] 00:05:00, metric 30

# Check LDP export policy
user@R1> show configuration protocols ldp
export BLOCK-ALL;

user@R1> show configuration policy-options policy-statement BLOCK-ALL
then reject;    # Oops! Blocking everything
```

Solution:

```
# Fix the export policy
[edit policy-options policy-statement LDP-EXPORT]
user@R1# set term LOOPBACKS from route-filter 10.0.0.0/24 prefix-length-range /32-/32
user@R1# set term LOOPBACKS then accept
user@R1# set term REJECT then reject

[edit protocols ldp]
user@R1# set export LDP-EXPORT

# Or simply remove the restrictive policy
user@R1# delete export

user@R1# commit
```

Scenario 3: LDP Not Following IGP Path

Symptom:

```
user@R1> show route 10.0.0.4
inet.0: 10.0.0.4/32
    *[OSPF/10] 00:10:00, metric 20
      > to 10.2.2.2 via ge-0/0/1.0     # Best path via R3

user@R1> show route table inet.3 10.0.0.4
inet.3: 10.0.0.4/32
    *[LDP/9] 00:10:00, metric 30
      > to 10.1.1.2 via ge-0/0/0.0, Push 100001  # LDP using different path!
```

Diagnosis:

```
# Check if track-igp-metric is enabled
user@R1> show configuration protocols ldp
interface ge-0/0/0.0;
interface ge-0/0/1.0;
# Missing: track-igp-metric

# Check metrics
user@R1> show ldp database detail
Input label database, 10.0.0.1:0--10.0.0.2:0
  Label: 100001, Prefix: 10.0.0.4/32
    Metric: 30    # Higher metric from R2

Input label database, 10.0.0.1:0--10.0.0.3:0
  Label: 200001, Prefix: 10.0.0.4/32
    Metric: 20    # Lower metric from R3, but not being used!
```

Solution:

```
[edit protocols ldp]
# Enable IGP metric tracking
user@R1# set track-igp-metric

# Verify after commit
```

```
user@R1# commit

user@R1> show route table inet.3 10.0.0.4
inet.3: 10.0.0.4/32
    *[LDP/9] 00:00:10, metric 20       # Now using correct metric
      > to 10.2.2.2 via ge-0/0/1.0, Push 200001  # And correct path!
```

Scenario 4: LDP Session Flapping

Symptom:

```
user@R1> show ldp session
  Neighbor                     State       Role   Hold time
10.0.0.2                   Operational  Active        3   # Very low!

# Few seconds later...
user@R1> show ldp session
  Neighbor                     State       Role   Hold time
10.0.0.2                   Nonexistent   --        --

user@R1> show log messages | match LDP
LDP_SESSIONDOWN: Neighbor 10.0.0.2: Hold timer expired
LDP_SESSIONUP: Neighbor 10.0.0.2: Established
LDP_SESSIONDOWN: Neighbor 10.0.0.2: Hold timer expired
```

Diagnosis:

```
# Check for routing instability
user@R1> monitor interface traffic
Interface    Link  Input packets        Output packets
ge-0/0/0       Up       1000 pps            50000 pps    # High output!

# Check CPU
user@R1> show system processes extensive | match rpd
  11 root      20   0  1216M  402M kqread  0 96.5% rpd    # Very high CPU

# Check keepalive settings
user@R1> show configuration protocols ldp
# No keepalive settings configured - using defaults
```

Solution:

```
[edit protocols ldp]
# Increase keepalive timers for stability
user@R1# set keep-multiplier 9            # 9 × 10 = 90 second hold time
user@R1# set interface-hello-hold-time 45
user@R1# set interface-hello-interval 15

# Add session protection (covered in Module 18)
user@R1# set session-protection

# Investigate high CPU/traffic separately
# May need to rate-limit LDP or fix routing loops

user@R1# commit
```

LDP Best Practices Summary

1. **Always include lo0.0**: LDP sessions use loopback addresses
2. **Enable MPLS family**: Required on all LDP interfaces
3. **Use track-igp-metric**: Ensures LDP follows IGP paths
4. **Start with defaults**: LDP works well out-of-the-box
5. **Monitor session stability**: Flapping indicates underlying issues

Module 17: LDP Configuration

Part 1: The Conceptual Lecture (The Why)

Understanding LDP Protocol Messages

Before diving into configuration, let's understand the protocol messages that make LDP work. Think of LDP like a social network for routers:

1. **Hello Messages** (Like friend requests):
   - "Hi, I speak LDP! Want to be neighbors?"
   - Sent every 5 seconds via UDP multicast
   - Creates adjacencies (not sessions yet)
2. **Initialization Messages** (Like exchanging contact info):
   - "Here are my capabilities and parameters"
   - Negotiates session parameters
   - Establishes TCP connection
3. **Label Mapping Messages** (Like sharing posts):
   - "To reach network X, use label Y"
   - The core of label distribution
   - Sent via TCP session
4. **Keepalive Messages** (Like "still there?" texts):
   - Maintains the TCP session
   - Default every 10 seconds

The LDP Message Flow

```
Step 1: Discovery Phase (UDP 646)
R1 ---[Hello: I'm 10.0.0.1:0]---> 224.0.0.2 (All routers multicast)
R2 <--[Hello: I'm 10.0.0.2:0]---- 224.0.0.2

Step 2: Session Establishment (TCP 646)
R1 ---[TCP SYN to 10.0.0.2:646]--> R2
R2 <--[TCP SYN-ACK]-------------- R1
R1 ---[TCP ACK]----------------> R2

Step 3: Initialization
R1 ---[Init: Keepalive=30, Label Range=16-100000]--> R2
R2 <--[Init: Keepalive=30, Label Range=16-100000]--- R1
R1 ---[Keepalive]--> R2
R2 <--[Keepalive]--- R1

Step 4: Label Exchange
R2 ---[Label Mapping: 10.0.0.2/32 = Label 3]------> R1
R2 ---[Label Mapping: 192.168.1.0/24 = Label 1001]-> R1
R1 <--[Label Mapping: 10.0.0.1/32 = Label 3]------- R2
```

Configuration Generates Protocol Behavior

Every configuration command triggers specific protocol messages:

```
Configuration                    Resulting Protocol Messages
─────────────────────────────────────────────────────────────
interface ge-0/0/0.0      →     Hello messages on this interface
keep-multiplier 6         →     Keepalive timeout = 6 × 10 = 60s
explicit-null            →     Label 0 instead of 3 (implicit null)
deaggregate              →     Advertise connected routes
```

Interface Messages vs Session Messages

It's crucial to understand the difference:

```
Interface-Level (Per Link):
- Hello messages
- Adjacency formation
- Link-local communication
- Uses UDP multicast

Session-Level (Per Neighbor):
- TCP connection
- Label exchanges
- Between loopbacks
- One session even with multiple links
```

Part 2: The Junos CLI Masterclass (The How)

Complete LDP Configuration Walkthrough

Let's build a full LDP configuration step by step, understanding what each command does:

```
# Step 1: Basic Interface Configuration
[edit interfaces]
user@R1# set ge-0/0/0 unit 0 family inet address 10.1.1.1/30
user@R1# set ge-0/0/0 unit 0 family mpls        # Critical for LDP!
user@R1# set ge-0/0/1 unit 0 family inet address 10.2.2.1/30
user@R1# set ge-0/0/1 unit 0 family mpls
user@R1# set lo0 unit 0 family inet address 10.0.0.1/32

# Step 2: IGP Configuration (LDP needs reachability)
[edit protocols ospf]
user@R1# set area 0.0.0.0 interface ge-0/0/0.0 interface-type p2p
user@R1# set area 0.0.0.0 interface ge-0/0/1.0 interface-type p2p
user@R1# set area 0.0.0.0 interface lo0.0 passive

# Step 3: Enable MPLS
[edit protocols mpls]
user@R1# set interface ge-0/0/0.0
user@R1# set interface ge-0/0/1.0
# Note: lo0.0 not needed here - it doesn't forward MPLS packets

# Step 4: Configure LDP
[edit protocols ldp]
user@R1# set interface ge-0/0/0.0
user@R1# set interface ge-0/0/1.0
user@R1# set interface lo0.0         # Yes, needed here for session endpoint!
```

Understanding What Happens After Configuration

After committing this configuration:

1. **Hello Messages Start**:

```
user@R1> monitor traffic interface ge-0/0/0 no-resolve
10:15:00.123456 Out IP 10.1.1.1.646 > 224.0.0.2.646: LDP Hello
  Transport Address: 10.0.0.1
  Hold Time: 15 seconds
  Label Space: 0 (platform-wide)
```

2. **Adjacencies Form**:

```
user@R1> show ldp interface detail
Interface: ge-0/0/0.0
  Label space ID: 10.0.0.1:0
  State: Enabled
  Hello interval: 5 seconds
  Hello hold time: 15 seconds
  Transport address: 10.0.0.1
  Neighbors:
    10.1.1.2
      State: Adjacent
      Last hello received: 00:00:02 ago
```

3. **TCP Sessions Establish**:

```
user@R1> show ldp session establishment
Neighbor        State           Message
10.0.0.2        Connecting      TCP SYN sent to 10.0.0.2:646
10.0.0.2        Initializing    Sent Init message
10.0.0.2        OpenRec         Received Init message
10.0.0.2        Operational     Session established!
```

Advanced LDP Configuration

Now let's add advanced features:

```
[edit protocols ldp]
# Optimize for large-scale deployment
keep-multiplier 9;                     # 90-second hold time (9 × 10)
interface-hello-hold-time 45;          # 45-second hello hold
interface-hello-interval 15;           # Send hellos every 15 seconds

# Enable on all interfaces at once
interface all;
interface fxp0.0 {                    # Except management
    disable;
}

# Advertise more than just loopbacks
deaggregate;                          # Advertise connected routes
track-igp-metric;                     # Follow IGP costs

# Use explicit null for ultimate hop popping visibility
explicit-null;

# Prefer IPv4 for dual-stack networks
family {
    inet {
        transport-preference ipv4;
    }
}

# Enable authentication
interface ge-0/0/0.0 {
    authentication-md5 0x1234567890abcdef;
}

# Session-level authentication (more secure)
session-group NEIGHBORS {
    authentication-algorithm hmac-sha-1;
    authentication-key "$9$encrypted-key-here";
}
neighbor 10.0.0.2 {
    session-group NEIGHBORS;
}
```

LDP for Service Provider Networks

Here's a production-ready configuration:

```
[edit protocols ldp]
# Global parameters
graceful-restart {
    helper-enable;
    recovery-time 180;
    reconnect-time 60;
}
```

```
# Logging for troubleshooting
log-updown {
    trap;                            # SNMP traps
}
traceoptions {
    file ldp-log size 10m files 5;
    flag error;
    flag state;
    flag event;
}

# P router configuration (core)
interface all;
interface fxp0.0 {
    disable;
}
track-igp-metric;
explicit-null;

# PE router configuration (edge)
targeted-hello {                     # For L2VPN/VPLS
    hello-interval 5;
    hold-time 15;
    accept-from 10.0.0.0/24;        # PE routers range
}

# Bandwidth allocation for L2VPN
interface ge-0/0/0.0 {
    allow-subnet-mismatch;          # For L2VPN
}
```

Verification After Configuration

Let's verify each aspect:

```
# 1. Check protocol messages
user@R1> monitor traffic interface ge-0/0/0 matching "port 646"
10:20:00.123 Out IP 10.1.1.1.646 > 224.0.0.2.646: UDP LDP Hello
10:20:01.456 In  IP 10.0.0.2.49182 > 10.0.0.1.646: TCP LDP Keepalive
10:20:02.789 Out IP 10.0.0.1.646 > 10.0.0.2.49182: TCP LDP Label Mapping

# 2. Check label generation
user@R1> show ldp database session 10.0.0.2 detail
Input label database, 10.0.0.1:0--10.0.0.2:0
  Label 3 (Implicit Null)
    Prefix 10.0.0.2/32
    Received at: 2024-01-10 10:15:30
    Attribute flags: 0x0

  Label 100000
    Prefix 10.0.0.3/32
    Received at: 2024-01-10 10:15:31
    Metric: 10 (from IGP)

Output label database, 10.0.0.1:0--10.0.0.2:0
  Label 0 (Explicit Null)          # Due to explicit-null config
    Prefix 10.0.0.1/32
    Advertised at: 2024-01-10 10:15:30

  Label 200000
    Prefix 10.1.1.0/30             # Due to deaggregate config
    Advertised at: 2024-01-10 10:15:31
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Verification Commands

1. **Complete LDP Status Overview:**

```
user@R1> show ldp overview
Instance: master
  Router ID: 10.0.0.1
  Message ID: 1234
  Configuration:
    Deaggregate: Enabled
    Explicit null: Enabled
    IPv4 transit LSP: Enabled
    Track IGP metric: Enabled
  Timers:
    Keepalive: 10 seconds
    Hello: 5 seconds
  Graceful restart:
    State: Enabled
    Reconnect time: 60 seconds
    Recovery time: 180 seconds

  Statistics:
    Sessions: 2 Operational, 0 Down
    Adjacencies: 2 Up, 0 Down
    Labels: 10 Received, 8 Advertised
```

2. **Detailed Protocol Message Analysis:**

```
user@R1> show ldp statistics
Packet types         Sent        Received
Hello                5432        5398
Keepalive           10234       10456
Init                   2           2
Label mapping        234         256
Label request          0           0
Label withdraw        12          15
Label release         15          12
Notification           0           0

Protocol errors:
  Bad LDP ID: 0
  Bad PDU length: 0
  Bad message length: 0
  Unknown message type: 0
  Session rejected: 0
```

Common Troubleshooting Scenarios

Scenario 1: Interfaces Enabled but No Adjacencies

Symptom:

```
user@R1> show ldp interface
Interface            Label space ID        Nbr count   Next hello
ge-0/0/0.0           10.0.0.1:0                   0   3
ge-0/0/1.0           10.0.0.1:0                   0   1

user@R1> show ldp neighbor
# No output!
```

Diagnosis:

```
# Check if hellos are being sent
user@R1> monitor traffic interface ge-0/0/0 no-resolve detail matching "port 646"
10:30:00.123 Out IP 10.1.1.1.646 > 224.0.0.2.646: UDP, length 42
  LDP Hello Message
  # Hellos going out but no replies!

# Check multicast
user@R1> show pim interfaces
# No output - PIM not configured but shouldn't affect LDP
# Check if MPLS family is enabled on remote router
user@R1# run ssh 10.0.0.2
user@R2> show configuration interfaces ge-0/0/0
unit 0 {
    family inet {
        address 10.1.1.2/30;
    }
    # Missing: family mpls
}
```

Solution:

```
# On R2:
[edit interfaces ge-0/0/0]
user@R2# set unit 0 family mpls

# Also check firewall filters
[edit firewall family inet filter EDGE-FILTER]
user@R2# set term ALLOW-LDP from protocol udp
user@R2# set term ALLOW-LDP from destination-port 646
user@R2# set term ALLOW-LDP from destination-address 224.0.0.2/32
user@R2# set term ALLOW-LDP then accept

user@R2# commit
```

Scenario 2: Adjacencies Up but Session Won't Establish

Symptom:

```
user@R1> show ldp neighbor
Address            Interface          Label space ID         Hold time
10.1.1.2           ge-0/0/0.0         10.0.0.2:0                  13

user@R1> show ldp session
  Neighbor                     State       Role   Hold time
10.0.0.2                   NonExistent   --        --

user@R1> show ldp session establishment
Neighbor: 10.0.0.2
  State: Connect sent
  Event: TCP connection timeout
  Attempts: 5
```

Diagnosis:

```
# Test TCP connectivity to loopback
user@R1> telnet 10.0.0.2 port 646 source 10.0.0.1
Trying 10.0.0.2...
telnet: connect to address 10.0.0.2: Connection refused

# Check if LDP is listening
user@R2> show system connections | match 646
tcp4       0      0  *.646              *.*                LISTEN

# Check routing to loopback
user@R1> show route 10.0.0.2
# No route to 10.0.0.2/32!

# Verify OSPF
user@R1> show ospf neighbor
# R2 not in OSPF!
```

Solution:

```
# On R2, add loopback to OSPF
[edit protocols ospf area 0.0.0.0]
user@R2# set interface lo0.0 passive
```

```
# If still failing, check MTU
[edit interfaces ge-0/0/0]
user@R2# set mtu 1500    # Ensure matching MTU

user@R2# commit
```

Scenario 3: Session Up but No Labels

Symptom:

```
user@R1> show ldp session
  Neighbor                     State       Role    Hold time
10.0.0.2                   Operational  Passive      28

user@R1> show ldp database
Input label database, 10.0.0.1:0--10.0.0.2:0
  Label     Prefix
  # Empty!

Output label database, 10.0.0.1:0--10.0.0.2:0
  Label     Prefix
  # Empty!

user@R1> show route table inet.3
# No LDP routes!
```

Diagnosis:

```
# Check if labels are being advertised
user@R1> show ldp traffic-statistics
Advertisement Stats:
  FECs advertised: 0          # Nothing being advertised!
  FECs received: 0
  Labels advertised: 0
  Labels received: 0

# Check configuration
user@R1> show configuration protocols ldp
interface ge-0/0/0.0;
interface ge-0/0/1.0;
# Missing lo0.0!

# Even with lo0.0 missing, should see neighbor's labels
# Check for export policy
export DENY-ALL;              # Found the problem!
```

Solution:

```
[edit protocols ldp]
# Add loopback
user@R1# set interface lo0.0

# Remove restrictive export policy
user@R1# delete export DENY-ALL

# Or create proper export policy
[edit policy-options policy-statement LDP-EXPORT]
user@R1# set term ALLOW-LOOPBACKS from protocol direct
user@R1# set term ALLOW-LOOPBACKS from route-filter 10.0.0.0/24 prefix-length-range /32-/32
user@R1# set term ALLOW-LOOPBACKS then accept

[edit protocols ldp]
user@R1# set export LDP-EXPORT

user@R1# commit
```

Scenario 4: Wrong Transport Address

Symptom:

```
user@R1> show ldp session
  Neighbor                     State       Role   Hold time
172.16.1.2                 Operational  Active       28    # Wrong address!

user@R1> show interfaces terse | match 172.16
ge-0/0/0.0              up    up   inet     172.16.1.1/30

# LDP using physical interface IP instead of loopback!
```

Diagnosis:

```
user@R1> show ldp interface detail
Interface: ge-0/0/0.0
  Transport address: 172.16.1.1    # Should be 10.0.0.1!
  Configuration: Use interface address

user@R1> show configuration protocols ldp
interface ge-0/0/0.0;
# No transport-address configured
```

Solution:

```
[edit protocols ldp]
# Force use of router-id (loopback) as transport
user@R1# set transport-address router-id

# Alternative: Set specific interface behavior
user@R1# set interface ge-0/0/0.0 transport-address 10.0.0.1

# Must clear sessions for change to take effect
user@R1# commit

user@R1> clear ldp session all
user@R1> clear ldp neighbor all

# Verify
user@R1> show ldp session
  Neighbor                     State       Role   Hold time
10.0.0.2                   Operational  Active       28    # Correct!
```

Advanced Diagnostics

For complex issues, use these commands:

```
# Enable detailed logging
[edit protocols ldp]
user@R1# set traceoptions file ldp-debug
user@R1# set traceoptions flag all
user@R1# commit

# Watch protocol messages in real-time
user@R1> monitor start ldp-debug
user@R1> monitor traffic interface ge-0/0/0 size 1500 matching "port 646" detail

# Check internal LDP state
user@R1> show ldp neighbor detail hidden
user@R1> show ldp database detail hidden
user@R1> show ldp session detail hidden
```

LDP Configuration Best Practices

1. **Always include lo0.0 interface**: Required for session establishment
2. **Configure both protocols**: MPLS and LDP on same interfaces
3. **Verify IGP first**: LDP needs loopback reachability
4. **Use explicit-null**: Better for troubleshooting

5. **Start simple**: Add advanced features incrementally

Module 18: LDP Enhancements and Best Practices

Part 1: The Conceptual Lecture (The Why)

The Problem: LDP-IGP Synchronization

Imagine this scenario: You're driving on a highway that has electronic signs showing "Route Open." You enter the highway based on this sign, but suddenly find there's no actual road surface - just the support structure! This is what happens in MPLS networks without LDP-IGP synchronization.

Here's the technical problem:

```
Timeline of a Link Coming Up:

Time 0ms:   Physical link UP
Time 10ms:  OSPF adjacency starts forming
Time 100ms: OSPF adjacency UP, routes advertised
Time 101ms: Traffic starts flowing (but MPLS not ready!)
Time 500ms: LDP session starts establishing
Time 1000ms: LDP session UP, labels exchanged
Time 1001ms: NOW it's safe for traffic

Result: 900ms of traffic blackholing!
```

Without synchronization, IGP advertises a link as usable before LDP has established labels, causing traffic to be dropped.

How LDP-IGP Synchronization Works

LDP-IGP Sync ensures IGP waits for LDP before declaring a link usable:

```
With LDP-IGP Synchronization:

1. Link comes up
2. IGP adjacency forms BUT...
3. IGP sets link metric to maximum (16777215)
4. LDP session establishes
5. LDP signals "ready" to IGP
6. IGP reduces metric to configured value
7. Traffic flows safely with MPLS labels

No packet loss!
```

BGP Next-Hop Resolution in LDP

In service provider networks, BGP carries customer routes while LDP provides transport. The challenge is ensuring BGP uses MPLS paths:

```
Default BGP Behavior (inet.0 resolution):
BGP Route: 192.168.1.0/24, Next-hop: 10.0.0.4
Resolution: Check inet.0 (no MPLS)
Result: Traffic forwarded without labels

Enhanced BGP Resolution (inet.3 resolution):
BGP Route: 192.168.1.0/24, Next-hop: 10.0.0.4
Resolution: Check inet.3 first (MPLS paths)
Result: Traffic uses MPLS LSPs
```

Session Protection

Session Protection maintains LDP sessions during link failures:

```
Without Session Protection:
1. Link fails
2. Hello messages stop
3. Hold timer expires (15 seconds)
4. LDP session torn down
5. All labels withdrawn
6. Link recovers
7. Full LDP session re-establishment (slow)
8. Labels re-advertised

With Session Protection:
1. Link fails
2. Targeted hellos take over (via alternate path)
3. LDP session stays UP
4. Labels preserved
5. Link recovers
6. Regular hellos resume
7. No re-establishment needed!
```

Part 2: The Junos CLI Masterclass (The How)

Configuring LDP-IGP Synchronization

Basic Configuration:

```
# Enable globally for all OSPF interfaces
[edit protocols ospf]
user@R1# set ldp-synchronization

# Or per-area
[edit protocols ospf area 0.0.0.0]
user@R1# set interface ge-0/0/0.0 ldp-synchronization

# For IS-IS
[edit protocols isis]
user@R1# set ldp-synchronization

# Set hold-down timer (optional)
[edit protocols ospf]
user@R1# set ldp-synchronization hold-time 10
```

Advanced Synchronization Configuration:

```
# Disable on specific interfaces (e.g., edge links)
[edit protocols ospf area 0.0.0.0]
user@R1# set interface ge-0/0/5.0 ldp-synchronization disable

# Configure what happens if LDP never comes up
[edit protocols ldp]
user@R1# set igp-synchronization holddown-interval 60

# OSPF-specific timer
[edit protocols ospf]
user@R1# set ldp-synchronization hold-time 30

# Different behavior for different areas
[edit protocols ospf]
area 0.0.0.0 {
    ldp-synchronization;
    interface ge-0/0/0.0;
    interface ge-0/0/1.0;
}
area 0.0.0.1 {
    # No LDP sync for stub area
    interface ge-0/0/2.0;
}
```

Configuring BGP Next-Hop Resolution

Method 1: Traffic Engineering bgp-igp:

```
[edit protocols mpls]
user@R1# set traffic-engineering bgp-igp

# This simple command does several things:
# 1. Copies LDP routes from inet.3 to inet.0
# 2. Gives them better preference (9 vs 10)
# 3. BGP can now resolve over MPLS
```

Method 2: Resolution RIB (Recommended):

```
# Make BGP check inet.3 for next-hop resolution
[edit routing-options]
user@R1# set resolution rib bgp.l3vpn.0 resolution-ribs [ inet.3 inet.0 ]

# For IPv6
user@R1# set resolution rib bgp.l3vpn-inet6.0 resolution-ribs [ inet6.3 inet6.0 ]
```

Method 3: RIB Groups (Most Flexible):

```
# Create RIB group to copy routes
[edit routing-options]
rib-groups {
    ldp-to-inet0 {
        import-rib [ inet.3 inet.0 ];
        import-policy ldp-routes-only;
    }
}

[edit protocols ldp]
user@R1# set rib-group ldp-to-inet0

[edit policy-options]
policy-statement ldp-routes-only {
    term 1 {
        from protocol ldp;
        then accept;
    }
    term 2 {
        then reject;
    }
}
```

Configuring Session Protection

Basic Session Protection:

```
[edit protocols ldp]
user@R1# set session-protection

# This enables session protection for all neighbors
# Default timeout is 86400 seconds (24 hours)
```

Advanced Session Protection:

```
[edit protocols ldp]
session-protection {
    # Change global timeout
    timeout 3600;            # 1 hour
    # Always maintain protection for critical neighbors
    always;
}

# Per-neighbor session protection
[edit protocols ldp]
neighbor 10.0.0.2 {
    session-protection {
        timeout 7200;          # 2 hours for this neighbor
    }
}

# Disable for specific neighbors
neighbor 10.0.0.100 {
    session-protection {
        disable;
    }
}
```

Complete Production Configuration

Here's a comprehensive LDP configuration with all enhancements:

```
# LDP Configuration
[edit protocols ldp]
# Basic settings
interface all;
interface fxp0.0 {
    disable;
}
track-igp-metric;
explicit-null;

# Session protection
session-protection {
    timeout 3600;
}

# IGP synchronization
igp-synchronization {
    holddown-interval 60;
}

# Graceful restart for additional protection
graceful-restart {
    helper-enable;
    recovery-time 180;
    reconnect-time 120;
}

# Targeted sessions for session protection
targeted-hello {
    hello-interval 5;
    hold-time 15;
}

# OSPF Configuration
[edit protocols ospf]
ldp-synchronization {
    hold-time 30;
}
area 0.0.0.0 {
    interface ge-0/0/0.0 {
        ldp-synchronization;
        metric 10;
    }
    interface ge-0/0/1.0 {
        ldp-synchronization;
        metric 10;
    }
    interface lo0.0 {
        passive;
    }
}

# BGP Configuration
[edit routing-options]
resolution {
    rib bgp.l3vpn.0 {
        resolution-ribs [ inet.3 inet.0 ];
    }
}

# Alternative: Traffic engineering bgp-igp
[edit protocols mpls]
traffic-engineering bgp-igp;
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Verification Commands

1. **Verify LDP-IGP Synchronization:**

```
user@R1> show ospf interface detail
Interface           State   Area            DR ID           BDR ID          Nbrs
ge-0/0/0.0          PtToPt  0.0.0.0         0.0.0.0         0.0.0.0            1
  Type: P2P, Address: 10.1.1.1, Mask: 255.255.255.252, MTU: 1500, Cost: 10
  LDP sync state: In sync, for 00:05:32, reason: LDP session up
  LDP-IGP Synchronization: Enabled
  Config holdtime: 30 seconds

user@R1> show ldp igp-synchronization
LDP-IGP Synchronization Information:

VRF: master
  LDP instance: master

  Interface   OSPF State    ISIS State    Sync State  Hold time
  ge-0/0/0.0  In sync       N/A           Achieved    00:00:00
  ge-0/0/1.0  In holddown   N/A           Pending     00:00:25
```

2. **Verify BGP Resolution:**

```
user@R1> show route resolution unresolved
# Should be empty if all BGP next-hops resolve

user@R1> show route protocol bgp detail | match "Protocol Next hop:"
     Protocol Next hop: 10.0.0.4 Metric: 20 (via inet.3)
                              ^^^^^^^^^^^^^^^^
                              # Shows resolution via MPLS

user@R1> show route table inet.3 hidden
# If using traffic-engineering bgp-igp, shows copied routes
```

3. **Verify Session Protection:**

```
user@R1> show ldp session detail
Address: 10.0.0.2, State: Operational, Connection: Open, Hold time: 27
  Session ID: 10.0.0.1:0--10.0.0.2:0
  Protection: Enabled, Duration: 3600 seconds
  Targeted hello: Enabled
    Hello interval: 5, Hold time: 15
  Next targetted hello in 3 seconds

user@R1> show ldp neighbor detail
Address           Interface          Label space ID      Hold time
10.1.1.2          ge-0/0/0.0         10.0.0.2:0               13
10.0.0.2          Targeted           10.0.0.2:0               14  <-- Protection
```

Common Troubleshooting Scenarios

Scenario 1: IGP Advertising But MPLS Black-Holing Traffic

Symptom:

```
user@R1> traceroute 192.168.1.1 no-resolve
 1  10.1.1.2     1 ms <MPLS:L=100001>
 2  10.2.2.2     *        *        *
 3  *            *        *        *

user@R1> show ospf interface ge-0/0/1.0 detail
  LDP sync state: Not achieved, reason: LDP not configured

user@R1> ping mpls ldp 10.0.0.4
  ! ! ! . . . . .    # Some packets lost
```

Diagnosis:

```
# Check synchronization state
user@R1> show ldp igp-synchronization
  Interface   OSPF State       Sync State
  ge-0/0/0.0  In sync         Achieved
  ge-0/0/1.0  Not in sync     LDP session down

# Check OSPF metric
user@R1> show ospf interface ge-0/0/1.0 detail | match Cost
  Cost: 16777215    # Maximum metric due to LDP sync!

# But why is traffic trying this path?
user@R1> show configuration protocols ospf
# Missing: ldp-synchronization
```

Solution:

```
[edit protocols ospf]
user@R1# set ldp-synchronization
user@R1# set ldp-synchronization hold-time 30

# Also ensure LDP is actually configured on interface
[edit protocols ldp]
user@R1# set interface ge-0/0/1.0

user@R1# commit
```

Scenario 2: BGP Routes Not Using MPLS

Symptom:

```
user@R1> show route 192.168.1.0/24 detail
192.168.1.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop: 10.0.0.4 via ge-0/0/0.0  # No MPLS label!
                Protocol next hop: 10.0.0.4

user@R1> show route table inet.3 10.0.0.4
inet.3: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/32        *[LDP/9] 00:10:00, metric 30
                    > to 10.1.1.2 via ge-0/0/0.0, Push 100432
```

Diagnosis:

```
# BGP next-hop exists in inet.3 but BGP not using it
user@R1> show route resolution table bgp.l3vpn.0
Resolution Tree for bgp.l3vpn.0 (0x7f8b290) : 0 nodes
```

```
  Resolution-ribs: inet.0
  # Not checking inet.3!

user@R1> show configuration routing-options resolution
# No configuration!

user@R1> show configuration protocols mpls
# No traffic-engineering bgp-igp
```

Solution:

```
# Option 1: Quick fix
[edit protocols mpls]
user@R1# set traffic-engineering bgp-igp

# Option 2: Proper resolution configuration
[edit routing-options]
user@R1# set resolution rib bgp.l3vpn.0 resolution-ribs [ inet.3 inet.0 ]

user@R1# commit

# Verify
user@R1> show route 192.168.1.0/24 detail | match "next hop"
                Next hop: 10.0.0.4 via ge-0/0/0.0, Push 100432  # Now has label!
```

Scenario 3: Session Protection Not Working

Symptom:

```
# Link fails
user@R1> show interfaces ge-0/0/0 terse
ge-0/0/0                up    down

# LDP session immediately goes down
user@R1> show ldp session
  Neighbor                     State
10.0.0.2                   Nonexistent

user@R1> show log messages | match LDP
Jan 10 10:00:00 LDP_ADJACENCY_DOWN: Adjacency to 10.0.0.2 down
Jan 10 10:00:15 LDP_SESSION_DOWN: Session to 10.0.0.2 down: Hold timer expired
```

Diagnosis:

```
user@R1> show configuration protocols ldp
interface ge-0/0/0.0;
# No session-protection configured!

# Even with session-protection, need alternate path
user@R1> show route 10.0.0.2
inet.0: 20 destinations, 20 routes (19 active, 0 holddown, 1 hidden)
# No route to 10.0.0.2 after link failure!
```

Solution:

```
[edit protocols ldp]
# Enable session protection
user@R1# set session-protection

# Ensure targeted hellos are enabled
user@R1# set targeted-hello hello-interval 5
user@R1# set targeted-hello hold-time 15

# Need alternate path for targeted hellos
[edit protocols ospf area 0.0.0.0]
user@R1# set interface ge-0/0/2.0    # Backup path

user@R1# commit
```

```
# Test by shutting down primary link
user@R1> request interface ge-0/0/0 disable

user@R1> show ldp session
  Neighbor                     State
10.0.0.2                   Operational    # Still up via targeted hellos!
```

Scenario 4: Synchronization Causing Persistent High Metrics

Symptom:

```
user@R1> show ospf interface ge-0/0/0.0 detail
  Cost: 16777215    # Still maximum after 5 minutes!
  LDP sync state: Not achieved, reason: No LDP adjacency

user@R1> show ldp interface
Interface            Label space ID        Nbr count
ge-0/0/0.0           10.0.0.1:0                   0    # No LDP neighbors
```

Diagnosis:

```
# Check remote router
user@R2> show configuration protocols ldp
# No LDP configured on R2!

# Check holddown behavior
user@R1> show configuration protocols ldp igp-synchronization
# No holddown-interval configured - waits forever!
```

Solution:

```
# Configure holddown to eventually give up
[edit protocols ldp]
user@R1# set igp-synchronization holddown-interval 120

# Or disable sync on interfaces to non-MPLS routers
[edit protocols ospf area 0.0.0.0]
user@R1# set interface ge-0/0/0.0 ldp-synchronization disable

# For interfaces that should have LDP, fix the remote router
user@R2# set protocols ldp interface ge-0/0/0.0
user@R2# set protocols mpls interface ge-0/0/0.0

user@R1# commit
user@R2# commit
```

Advanced Verification Scripts

For monitoring LDP health:

```
# Script to check all LDP enhancements
user@R1> op ldp-health-check
LDP Enhancement Status Report:

1. IGP Synchronization:
   - Enabled: Yes
   - Interfaces in sync: 8/10
   - Problem interfaces: ge-0/0/5.0, ge-0/0/6.0

2. Session Protection:
   - Enabled: Yes
   - Protected sessions: 5/5
   - Targeted hello active: 5

3. BGP Resolution:
   - Method: traffic-engineering bgp-igp
   - BGP routes using MPLS: 1,234/1,250
   - Unresolved next-hops: 16

4. Graceful Restart:
   - Helper mode: Enabled
   - Sessions helped: 2
```

Best Practices Summary

1. **Always enable LDP-IGP synchronization** in MPLS networks
2. **Configure holddown timers** to handle non-MPLS neighbors
3. **Use session protection** for critical neighbor relationships
4. **Enable traffic-engineering bgp-igp** for simple BGP resolution
5. **Monitor sync status** during maintenance windows

Module 19: LDP Egress, Import, and Export Policies

Part 1: The Conceptual Lecture (The Why)

Understanding FEC (Forwarding Equivalence Class)

Before diving into policies, let's truly understand what a FEC is. Think of FECs like shipping labels in a warehouse:

```
Traditional Routing (No FEC):
- Every packet examined individually
- Routing decision made per packet
- Like reading the full address on every package

MPLS with FECs:
- Packets grouped by destination
- One label per group
- Like color-coding packages: "All red labels go to Building A"

Examples of FECs:
- FEC 1: All packets destined to 10.0.0.4/32
- FEC 2: All packets destined to 192.168.0.0/16
- FEC 3: All
```

Part 1: The Conceptual Lecture (The Why) - Continued

```
Examples of FECs:
- FEC 1: All packets destined to 10.0.0.4/32
- FEC 2: All packets destined to 192.168.0.0/16
- FEC 3: All VPN traffic for Customer A
- FEC 4: All IPv6 packets to 2001:db8::/32
```

Why Control FEC Advertisement?

By default, LDP advertises labels for ALL prefixes it knows about. This can be problematic:

```
Default LDP Behavior:
Router R1 knows about:
- 10.0.0.1/32 (loopback) ✓ Should advertise
- 10.1.1.0/30 (P2P link) ✗ Unnecessary
- 10.2.2.0/30 (P2P link) ✗ Unnecessary
- 172.16.0.0/24 (Management) ✗ Definitely not!
- 192.168.1.0/24 (Customer) ✗ Not needed in core

Result: Advertising 5 FECs when only 1 is needed!
```

The Three Types of LDP Policies

1. Egress Policies (Controlling What You Advertise)

```
Purpose: Advertise FECs beyond default loopback
Use case: When you need LDP labels for non-loopback destinations

Example scenario:
- Default: Only advertise 10.0.0.1/32
- With Egress Policy: Also advertise 192.168.1.0/24
```

2. Export Policies (Limiting What You Send to Neighbors)

```
Purpose: Don't advertise certain FECs to specific neighbors
Use case: Security, scalability, or topology hiding

Example scenario:
To Core Routers: Advertise everything
To Edge Routers: Only advertise loopbacks
To Customer: Advertise nothing
```

3. Import Policies (Filtering What You Accept)

```
Purpose: Don't install labels from neighbors for certain FECs
Use case: Security, memory conservation, or loop prevention

Example scenario:
From Core: Accept all FECs
From Edge: Only accept loopbacks
From Untrusted: Accept nothing
```

Policy Evaluation Order

Understanding the order is critical:

```
Label Advertisement Flow:

1. Route exists in routing table
           ↓
2. Egress policy evaluation
   "Should I create a label for this FEC?"
           ↓
3. Label created in database
           ↓
4. Export policy evaluation (per neighbor)
   "Should I advertise this to neighbor X?"
           ↓
5. Label sent to neighbor
           ↓
6. Neighbor's import policy
   "Should I accept this label?"
           ↓
7. Label installed in neighbor's MPLS table
```

Part 2: The Junos CLI Masterclass (The How)

Configuring Egress Policies

Egress policies determine which FECs get labels in the first place:

```
# Basic Egress Policy - Advertise specific prefix
[edit policy-options]
policy-statement LDP-EGRESS {
    term CUSTOMER-ROUTES {
        from {
            protocol direct;
            route-filter 192.168.0.0/16 orlonger;
        }
        then accept;
    }
    term DEFAULT {
        then reject;  # Only advertise what we explicitly allow
    }
}

[edit protocols ldp]
user@R1# set egress-policy LDP-EGRESS

# Note: Loopback is ALWAYS advertised regardless of egress policy
```

Advanced Egress Policy Examples:

```
# Advertise all connected subnets
[edit policy-options]
policy-statement ADVERTISE-CONNECTED {
    term CONNECTED {
        from protocol direct;
        then accept;
    }
}

# Advertise specific BGP routes
policy-statement ADVERTISE-BGP-ROUTES {
    term CUSTOMER-PREFIXES {
        from {
            protocol bgp;
            community CUSTOMER-ROUTES;
        }
        then accept;
    }
    term REJECT-REST {
        then reject;
    }
}

# Advertise based on prefix length
policy-statement ADVERTISE-HOSTS-ONLY {
    term HOST-ROUTES {
        from {
            route-filter 0.0.0.0/0 prefix-length-range /32-/32;
        }
        then accept;
    }
    term REJECT {
        then reject;
    }
}
```

Configuring Export Policies

Export policies control which neighbors receive which FECs:

```
# Basic Export Policy
[edit policy-options]
policy-statement LDP-EXPORT-CORE {
    term ALLOW-LOOPBACKS {
        from {
            protocol ldp;
            route-filter 10.0.0.0/24 prefix-length-range /32-/32;
        }
        then accept;
    }
    term DENY-REST {
        then reject;
    }
}

[edit protocols ldp]
user@R1# set export LDP-EXPORT-CORE

# Per-session export policy (more specific)
[edit protocols ldp]
session 10.0.0.2 {
    export LDP-EXPORT-EDGE;  # Different policy for this neighbor
}
```

Complex Export Policy Scenarios:

```
# Export policy based on interface groups
[edit policy-options]
policy-statement LDP-EXPORT-BY-INTERFACE {
    term CORE-INTERFACES {
        from {
            interface [ ge-0/0/0.0 ge-0/0/1.0 ];
            protocol ldp;
        }
        then accept;
    }
    term EDGE-INTERFACES {
        from {
            interface [ ge-0/0/2.0 ge-0/0/3.0 ];
            protocol ldp;
            route-filter 10.0.0.0/24 prefix-length-range /32-/32;
        }
        then accept;
    }
    term DENY {
        then reject;
    }
}

# Export based on LDP neighbor
policy-statement LDP-EXPORT-BY-NEIGHBOR {
    term TO-CORE-ROUTERS {
        from neighbor [ 10.0.0.2 10.0.0.3 ];
        then accept;  # Full mesh to core
    }
    term TO-EDGE-ROUTERS {
        from {
            neighbor [ 10.0.0.100 10.0.0.101 ];
            route-filter 10.0.0.0/24 prefix-length-range /32-/32;
        }
        then accept;  # Only loopbacks to edge
    }
    term DENY {
        then reject;
    }
}
```

Configuring Import Policies

Import policies filter incoming label advertisements:

```
# Basic Import Policy
[edit policy-options]
policy-statement LDP-IMPORT-FILTER {
    term ACCEPT-INFRASTRUCTURE {
        from {
            protocol ldp;
            route-filter 10.0.0.0/24 prefix-length-range /32-/32;
        }
        then accept;
    }
    term REJECT-CUSTOMER-ROUTES {
        from {
            protocol ldp;
            route-filter 192.168.0.0/16 orlonger;
        }
        then reject;
    }
    term DEFAULT-ACCEPT {
        then accept;
    }
}

[edit protocols ldp]
user@R1# set import LDP-IMPORT-FILTER
```

Import Policy for Security:

```
# Strict import policy for edge routers
[edit policy-options]
policy-statement LDP-IMPORT-EDGE {
    term ALLOW-KNOWN-LOOPBACKS {
        from {
            protocol ldp;
            route-filter 10.0.0.0/24 prefix-length-range /32-/32;
            # Additional safety: check neighbor
            neighbor [ 10.0.0.1 10.0.0.2 10.0.0.3 ];
        }
        then accept;
    }
    term LOG-AND-REJECT {
        then {
            syslog;
            reject;
        }
    }
}
```

Complete Production Example

Here's a comprehensive policy configuration for a service provider PE router:

```
# Define prefix lists for reusability
[edit policy-options]
prefix-list INFRASTRUCTURE-LOOPBACKS {
    10.0.0.0/24;
}
prefix-list CUSTOMER-PREFIXES {
    192.168.0.0/16;
    172.16.0.0/12;
}
prefix-list P2P-LINKS {
    10.1.0.0/16;
    10.2.0.0/16;
}

# Community definitions
community CUSTOMER-ROUTES members 65000:100;
community INFRASTRUCTURE members 65000:999;

# Egress Policy - What FECs to create
policy-statement LDP-EGRESS-PE {
    term LOOPBACKS {
        from {
            prefix-list INFRASTRUCTURE-LOOPBACKS;
            route-filter 0.0.0.0/0 prefix-length-range /32-/32;
        }
        then accept;
    }
    term CUSTOMER-ROUTES {
        from {
            protocol bgp;
            community CUSTOMER-ROUTES;
        }
        then accept;
    }
    term REJECT {
        then reject;
    }
}

# Export Policy - What to advertise to neighbors
policy-statement LDP-EXPORT-PE {
    term TO-CORE {
        to neighbor [ 10.0.0.1 10.0.0.2 ];
        from prefix-list INFRASTRUCTURE-LOOPBACKS;
        then accept;
    }
    term TO-OTHER-PES {
        to neighbor 10.0.0.0/24;
        from {
            prefix-list INFRASTRUCTURE-LOOPBACKS;
            prefix-list CUSTOMER-PREFIXES;
        }
        then accept;
    }
    term REJECT {
        then reject;
    }
}

# Import Policy - What to accept from neighbors
policy-statement LDP-IMPORT-PE {
    term FROM-CORE {
        from {
            neighbor [ 10.0.0.1 10.0.0.2 ];
            prefix-list INFRASTRUCTURE-LOOPBACKS;
        }
        then accept;
    }
    term FROM-OTHER-PES {
        from {
            neighbor 10.0.0.0/24;
            route-filter 0.0.0.0/0 prefix-length-range /32-/32;
        }
        then accept;
    }
    term LOG-UNEXPECTED {
        then {
            syslog;
            reject;
        }
    }
}

# Apply the policies
[edit protocols ldp]
egress-policy LDP-EGRESS-PE;
export LDP-EXPORT-PE;
import LDP-IMPORT-PE;
```

Part 3: Verification & Troubleshooting (The What-If)

Essential Verification Commands

1. **Verify Egress Policy Effect:**

```
user@R1> show ldp database advertised-labels
Advertisement spec:
  Egress policy: LDP-EGRESS-PE

Local labels advertised:
  Label     Prefix                    State
      3     10.0.0.1/32              Active
  100001    192.168.1.0/24           Active    # Due to egress policy
  100002    192.168.2.0/24           Active    # Due to egress policy

user@R1> show policy LDP-EGRESS-PE 10.1.1.0/30
Policy LDP-EGRESS-PE: Does not match    # P2P links filtered
```

2. **Verify Export Policy:**

```
user@R1> show ldp database session 10.0.0.2
Output label database, 10.0.0.1:0--10.0.0.2:0
  Label     Prefix
      3     10.0.0.1/32     # Only loopback exported to core

user@R1> show ldp database session 10.0.0.100
Output label database, 10.0.0.1:0--10.0.0.100:0
  Label     Prefix
      3     10.0.0.1/32
  100001    192.168.1.0/24  # Customer routes exported to other PEs
  100002    192.168.2.0/24
```

3. **Verify Import Policy:**

```
user@R1> show ldp database received-labels
Input label database, 10.0.0.1:0--10.0.0.2:0
  Label     Prefix              Status
      3     10.0.0.2/32        Accepted
  200001    10.1.1.0/30        Filtered    # Import policy rejected

user@R1> show log messages | match "LDP.*reject"
LDP_IMPORT_FILTERED: Label 200001 for 10.1.1.0/30 rejected by import policy
```

Common Troubleshooting Scenarios

Scenario 1: Egress Policy Not Creating Expected Labels

Symptom:

```
user@R1> show route 192.168.1.0/24
inet.0: 192.168.1.0/24
    *[BGP/170] 00:10:00
      > to 10.1.1.2 via ge-0/0/0.0

user@R1> show ldp database advertised-labels | match 192.168
# No output - label not created!
```

Diagnosis:

```
# Test the policy
user@R1> test policy LDP-EGRESS-PE 192.168.1.0/24
Policy LDP-EGRESS-PE: Does not match

# Check policy terms
user@R1> show configuration policy-options policy-statement LDP-EGRESS-PE
term CUSTOMER-ROUTES {
    from {
        protocol bgp;
        community CUSTOMER-ROUTES;  # Requires community!
    }
}

# Check if route has community
user@R1> show route 192.168.1.0/24 detail | match community
# No community shown!
```

Solution:

```
# Option 1: Fix BGP to add community
[edit protocols bgp group CUSTOMERS]
user@R1# set import ADD-CUSTOMER-COMMUNITY

[edit policy-options policy-statement ADD-CUSTOMER-COMMUNITY]
user@R1# set term 1 from route-filter 192.168.0.0/16 orlonger
user@R1# set term 1 then community add CUSTOMER-ROUTES

# Option 2: Modify egress policy to not require community
[edit policy-options policy-statement LDP-EGRESS-PE]
user@R1# delete term CUSTOMER-ROUTES from community
user@R1# set term CUSTOMER-ROUTES from route-filter 192.168.0.0/16 orlonger

user@R1# commit
```

Scenario 2: Export Policy Blocking All Labels

Symptom:

```
user@R1> show ldp neighbor
Address            Interface          Label space ID         Hold time
10.1.1.2           ge-0/0/0.0         10.0.0.2:0                  13

user@R1> show ldp database session 10.0.0.2
Output label database, 10.0.0.1:0--10.0.0.2:0
  Label     Prefix
  # Empty! No labels being sent
```

Diagnosis:

```
# Check export policy
user@R1> show configuration protocols ldp export
export BLOCK-ALL;

user@R1> show configuration policy-options policy-statement BLOCK-ALL
then reject;    # Oops!

# Even if policy is correct, check the match conditions
user@R1> show configuration policy-options policy-statement LDP-EXPORT
term ALLOW {
    from neighbor 10.0.0.2;    # But LDP session uses loopback!
}
```

Solution:

```
# Fix the policy to match properly
[edit policy-options policy-statement LDP-EXPORT]
user@R1# delete term ALLOW from neighbor
user@R1# set term ALLOW to neighbor 10.0.0.2

# Or use a more flexible approach
user@R1# set term CORE-NEIGHBORS to neighbor 10.0.0.0/28
user@R1# set term CORE-NEIGHBORS then accept

user@R1# commit
```

Scenario 3: Import Policy Not Filtering

Symptom:

```
user@R1> show ldp database received-labels
Input label database, 10.0.0.1:0--10.0.0.2:0
  Label     Prefix
  300001    172.16.1.0/24    # Management network - shouldn't accept!
  300002    10.99.99.0/24    # Unknown network - security risk!

user@R1> show route table inet.3
172.16.1.0/24      *[LDP/9] 00:05:00
                    > to 10.1.1.2 via ge-0/0/0.0, Push 300001
```

Diagnosis:

```
# Check import policy
user@R1> show configuration protocols ldp import
# No import policy configured!

# Or policy might be too permissive
user@R1> show configuration policy-options policy-statement LDP-IMPORT
term DEFAULT {
    then accept;    # Accepts everything!
}
```

Solution:

```
[edit policy-options policy-statement LDP-IMPORT-SECURE]
user@R1# set term ALLOW-INFRASTRUCTURE from route-filter 10.0.0.0/24 orlonger
user@R1# set term ALLOW-INFRASTRUCTURE then accept
user@R1# set term DENY-MANAGEMENT from route-filter 172.16.0.0/12 orlonger
user@R1# set term DENY-MANAGEMENT then reject
user@R1# set term LOG-UNKNOWN then syslog
user@R1# set term LOG-UNKNOWN then reject

[edit protocols ldp]
user@R1# set import LDP-IMPORT-SECURE

user@R1# commit
```

Scenario 4: Policy Chain Issues

Symptom:

```
# Some labels missing to specific neighbors
user@R1> show ldp database session 10.0.0.2
  Label     Prefix
      3     10.0.0.1/32
  # Missing other expected labels

user@R1> show ldp database session 10.0.0.3
  Label     Prefix
      3     10.0.0.1/32
  100001    192.168.1.0/24    # This neighbor gets more labels
```

Diagnosis:

```
# Check for multiple policies
user@R1> show configuration protocols ldp
export POLICY1;
session 10.0.0.2 {
    export POLICY2;    # Session-specific overrides global!
}

# Check policy evaluation
user@R1> test policy POLICY2 192.168.1.0/24
Policy POLICY2: Does not match
```

Solution:

```
# Option 1: Use policy chains
[edit protocols ldp]
user@R1# set export [ POLICY1 POLICY2 ]  # Both evaluated

# Option 2: Create comprehensive per-session policies
[edit policy-options policy-statement LDP-EXPORT-CORE-ROUTER]
user@R1# set term INHERIT-GLOBAL from policy POLICY1
user@R1# set term INHERIT-GLOBAL then accept
user@R1# set term ADDITIONAL-LOGIC from ...

[edit protocols ldp session 10.0.0.2]
user@R1# set export LDP-EXPORT-CORE-ROUTER

user@R1# commit
```

Policy Testing Commands

Always test policies before applying:

```
# Test egress policy
user@R1> test policy LDP-EGRESS-PE 192.168.1.0/24

# Show policy evaluation
user@R1> show policy LDP-EXPORT-PE

# Trace policy evaluation
[edit protocols ldp]
user@R1# set traceoptions file ldp-policy
user@R1# set traceoptions flag policy detail
user@R1# commit

user@R1> show log ldp-policy | match "policy|accept|reject"
```

Best Practices Summary

1. **Default Deny**: End policies with explicit reject terms
2. **Test First**: Use `test policy` before applying to production
3. **Log Rejects**: Add syslog action to reject terms for visibility
4. **Use Prefix Lists**: More maintainable than inline route-filters
5. **Document Intent**: Use meaningful policy and term names

Lab 6: Label Distribution Protocol

Lab Overview

In this comprehensive lab, you'll implement and verify all aspects of LDP that we've covered in Modules 16-19. You'll build a service provider network with LDP, implement enhancements, and use policies to control label distribution.

Lab Topology

```
                            Customer A
                                |
                            192.168.1.0/24
                                |
                              [PE1]
                            10.0.0.10
                            /       \
                          /           \
                     ge-0/0/0      ge-0/0/1
                       /               \
                     /                   \
                  [P1]                   [P2]
                10.0.0.1               10.0.0.2
```

```
                   |  \               /  |
                   |    \           /    |
               ge-0/0/2   \       /   ge-0/0/2
                   |       \   /        |
                   |          X          |
                   |       /   \        |
               ge-0/0/3   /       \   ge-0/0/3
                   |    /           \    |
                   |  /               \  |
                  [P3]                   [P4]
                10.0.0.3               10.0.0.4
                     \                   /
                       \               /
                     ge-0/0/0      ge-0/0/1
                          \           /
                            \       /
                              [PE2]
                            10.0.0.20
                                |
                            192.168.2.0/24
                                |
                            Customer B

Physical Connections:
PE1 ge-0/0/0 <-> P1 ge-0/0/0 (10.1.0.0/30)
PE1 ge-0/0/1 <-> P2 ge-0/0/0 (10.2.0.0/30)
P1 ge-0/0/1 <-> P2 ge-0/0/1 (10.12.0.0/30)
P1 ge-0/0/2 <-> P3 ge-0/0/2 (10.13.0.0/30)
P2 ge-0/0/2 <-> P4 ge-0/0/2 (10.24.0.0/30)
P3 ge-0/0/3 <-> P4 ge-0/0/3 (10.34.0.0/30)
P3 ge-0/0/0 <-> PE2 ge-0/0/0 (10.3.0.0/30)
P4 ge-0/0/1 <-> PE2 ge-0/0/1 (10.4.0.0/30)
```

Lab Objectives

1. Configure basic LDP and verify label distribution
2. Implement LDP-IGP synchronization
3. Configure BGP next-hop resolution via LDP
4. Enable session protection and test failover
5. Implement egress policies for selective FEC advertisement
6. Configure export policies for topology hiding
7. Apply import policies for security

Part 1: Basic LDP Configuration

Step 1: Configure IP Addressing and OSPF

On all routers, configure:

```
# Example for P1
[edit interfaces]
set ge-0/0/0 unit 0 family inet address 10.1.0.1/30
set ge-0/0/0 unit 0 family mpls
set ge-0/0/1 unit 0 family inet address 10.12.0.1/30
set ge-0/0/1 unit 0 family mpls
set ge-0/0/2 unit 0 family inet address 10.13.0.1/30
set ge-0/0/2 unit 0 family mpls
set lo0 unit 0 family inet address 10.0.0.1/32

[edit protocols ospf area 0.0.0.0]
set interface ge-0/0/0.0 interface-type p2p
set interface ge-0/0/1.0 interface-type p2p
set interface ge-0/0/2.0 interface-type p2p
set interface lo0.0 passive
```

Step 2: Enable MPLS and LDP

On all routers:

```
[edit protocols mpls]
set interface all
set interface fxp0.0 disable

[edit protocols ldp]
set interface all
set interface fxp0.0 disable
```

Step 3: Verify Basic LDP Operation

```
# Verify LDP neighbors
user@P1> show ldp neighbor
Address            Interface          Label space ID         Hold time
10.1.0.2           ge-0/0/0.0         10.0.0.10:0                 13
10.12.0.2          ge-0/0/1.0         10.0.0.2:0                  14
10.13.0.2          ge-0/0/2.0         10.0.0.3:0                  12

# Verify LDP sessions
user@P1> show ldp session
  Neighbor                     State       Role    Hold time
10.0.0.10                  Operational  Active       28
10.0.0.2                   Operational  Passive      27
10.0.0.3                   Operational  Active       29

# Verify label database
user@P1> show ldp database
Input label database, 10.0.0.1:0--10.0.0.10:0
  Label     Prefix
      3     10.0.0.10/32
  16001     10.0.0.2/32
  16002     10.0.0.3/32
  16003     10.0.0.4/32
  16004     10.0.0.20/32

# Verify MPLS forwarding
user@P1> show route table inet.3
```

Part 2: LDP-IGP Synchronization

Step 1: Configure LDP-IGP Sync

On all routers:

```
[edit protocols ospf]
set ldp-synchronization hold-time 30

[edit protocols ldp]
set igp-synchronization holddown-interval 60
```

Step 2: Test Synchronization

```
# Disable LDP on one interface
user@P1> configure
user@P1# delete protocols ldp interface ge-0/0/1.0
user@P1# commit

# Check OSPF metric
user@P1> show ospf interface ge-0/0/1.0 detail | match "metric|sync"
  Type: P2P, Address: 10.12.0.1, Mask: 255.255.255.252, MTU: 1500, Cost: 16777215
  LDP sync state: Not achieved, reason: LDP session down
```

```
# Re-enable LDP and watch synchronization
user@P1> configure
user@P1# set protocols ldp interface ge-0/0/1.0
user@P1# commit

# Monitor the sync state
user@P1> show ospf interface ge-0/0/1.0 detail | match "metric|sync"
  Type: P2P, Address: 10.12.0.1, Mask: 255.255.255.252, MTU: 1500, Cost: 10
  LDP sync state: In sync, for 00:00:15
```

Part 3: BGP Next-Hop Resolution

Step 1: Configure BGP on PE Routers

```
# On PE1
[edit protocols bgp]
set group IBGP type internal
set group IBGP local-address 10.0.0.10
set group IBGP neighbor 10.0.0.20
set group IBGP family inet unicast

[edit routing-options]
set autonomous-system 65000
set router-id 10.0.0.10

# Advertise customer routes
[edit policy-options policy-statement CUSTOMER-ROUTES]
set term 1 from protocol direct
set term 1 from route-filter 192.168.1.0/24 exact
set term 1 then accept

[edit protocols bgp group IBGP]
set export CUSTOMER-ROUTES
```

Step 2: Configure BGP Resolution via LDP

```
# Method 1: Traffic Engineering BGP-IGP
[edit protocols mpls]
set traffic-engineering bgp-igp

# Verify resolution
user@PE1> show route 192.168.2.0/24 detail
192.168.2.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop: 10.0.0.20 via ge-0/0/0.0, Push 16
                Protocol next hop: 10.0.0.20 (via inet.3)
```

Part 4: Session Protection

Step 1: Configure Session Protection

On all routers:

```
[edit protocols ldp]
set session-protection timeout 300
set targeted-hello hello-interval 5
set targeted-hello hold-time 15
```

Step 2: Test Session Protection

```
# Verify targeted hellos
user@P1> show ldp neighbor detail
Address: 10.0.0.2, State: Operational, Connection: Open
  Interface: ge-0/0/1.0
  Interface: Targeted    # Session protection active
```

```
# Disable primary interface
user@P1> request interface ge-0/0/1 disable

# Verify session stays up
user@P1> show ldp session
  Neighbor                     State
10.0.0.2                   Operational    # Still up via targeted hellos!

# Check alternate path
user@P1> show route 10.0.0.2
inet.0: 10.0.0.2/32
    *[OSPF/10] 00:00:05, metric 20
      > to 10.13.0.2 via ge-0/0/2.0    # Using alternate path
```

Part 5: Egress Policies

Step 1: Configure Egress Policy on PE1

```
# Advertise customer prefixes
[edit policy-options]
policy-statement LDP-EGRESS-PE {
    term CUSTOMER {
        from {
            route-filter 192.168.1.0/24 exact;
        }
        then accept;
    }
    term REJECT {
        then reject;
    }
}

[edit protocols ldp]
set egress-policy LDP-EGRESS-PE

# Verify
user@PE1> show ldp database advertised-labels
  Label     Prefix
      3     10.0.0.10/32
  299792    192.168.1.0/24    # Customer prefix now has label
```

Part 6: Export Policies

Step 1: Configure Export Policies on P Routers

```
# On P1 - Hide internal infrastructure from PE routers
[edit policy-options]
policy-statement LDP-EXPORT-SELECTIVE {
    term TO-PE {
        to neighbor [ 10.0.0.10 10.0.0.20 ];
        from {
            route-filter 10.0.0.0/24 prefix-length-range /32-/32;
        }
        then accept;
    }
    term TO-P {
        to neighbor [ 10.0.0.2 10.0.0.3 10.0.0.4 ];
        then accept;    # Full visibility to other P routers
    }
    term REJECT {
        then reject;
    }
}

[edit protocols ldp]
set export LDP-EXPORT-SELECTIVE
```

Step 2: Verify Export Policy

```
# Check what PE1 receives from P1
user@PE1> show ldp database session 10.0.0.1
Input label database, 10.0.0.10:0--10.0.0.1:0
  Label     Prefix
      3     10.0.0.1/32
  16001     10.0.0.2/32
  16002     10.0.0.3/32
  16003     10.0.0.4/32
  # No P2P link prefixes - policy working!
```

Part 7: Import Policies

Step 1: Configure Import Policy on PE Routers

```
# On PE1 - Only accept infrastructure loopbacks
[edit policy-options]
policy-statement LDP-IMPORT-SECURE {
    term INFRASTRUCTURE {
        from {
            route-filter 10.0.0.0/24 prefix-length-range /32-/32;
        }
        then accept;
    }
    term LOG-AND-REJECT {
        then {
            syslog;
            reject;
        }
    }
}

[edit protocols ldp]
set import LDP-IMPORT-SECURE
```

Step 2: Test Import Filtering

```
# First, make P1 advertise a P2P link
user@P1# delete protocols ldp export
user@P1# commit

# Check PE1's received labels
user@PE1> show ldp database received-labels
Input label database, 10.0.0.10:0--10.0.0.1:0
  Label     Prefix              Status
      3     10.0.0.1/32        Accepted
  16010     10.1.0.0/30        Filtered    # Import policy working!

# Check logs
user@PE1> show log messages | match LDP
LDP_IMPORT_FILTERED: Prefix 10.1.0.0/30 rejected by import policy
```

Verification Tasks

1. **End-to-End LSP Verification:**

```
user@PE1> ping mpls ldp 10.0.0.20 source 10.0.0.10
```

2. **Verify LDP Synchronization:**

```
user@P1> show ldp igp-synchronization
```

3. **Test Failure Scenarios:**
   - Shut down links and verify session protection
   - Verify traffic continues flowing during LDP session protection
   - Check that OSPF doesn't advertise unsynchronized links

4. **Policy Verification:**
    - Confirm only intended FECs are advertised
    - Verify PE routers don't see infrastructure details
    - Check that import policies block unwanted labels

Lab Completion Criteria

You have successfully completed this lab when:

1. ✓ All LDP sessions are established
2. ✓ End-to-end LSPs are working (ping mpls ldp succeeds)
3. ✓ LDP-IGP synchronization prevents black holes
4. ✓ BGP routes resolve via MPLS (show Push labels)
5. ✓ Session protection maintains connectivity during failures
6. ✓ Egress policies control FEC advertisement
7. ✓ Export policies hide infrastructure from edge routers
8. ✓ Import policies filter unwanted labels

Bonus Challenges

1. **Configure LDP Authentication:**
    - Add MD5 authentication between P routers
    - Use different keys for different sessions
2. **Implement Graceful Restart:**
    - Configure and test LDP graceful restart
    - Verify forwarding continues during control plane restart
3. **Advanced Policies:**
    - Create policies that change based on time of day
    - Implement different policies for IPv4 and IPv6
4. **Monitoring and Automation:**
    - Create a script to monitor LDP health
    - Set up SNMP traps for LDP session failures