Module 1: Introduction to Service Provider Routing

Welcome to your journey toward JNCIE-SP expertise. This foundational module establishes the critical concepts that underpin all service provider routing. We'll build your understanding from absolute zero, establishing why service providers need specialized routing approaches and how Junos OS implements these solutions.

Part 1: The Conceptual Lecture (The Why)

The Fundamental Problem

Imagine you're building a road system for an entire country. You don't just need roads—you need highways, intersections, traffic signals, and signs that guide millions of vehicles efficiently from any point A to any point B. Service provider networks face this exact challenge, but with data packets instead of cars.

The Core Challenge: Service providers must interconnect thousands or millions of customers, ensuring every packet finds its optimal path through a vast network while maintaining:

- **Scale**: Managing hundreds of thousands of routes
- **Reliability**: Ensuring network availability exceeds 99.999% (five nines)
- **Performance**: Minimizing latency and maximizing throughput
- **Flexibility**: Adapting to constant topology changes
- **Security**: Protecting against attacks and misconfigurations

Understanding Routing Fundamentals

What is Routing? Routing is the process of determining the best path for data packets to travel from source to destination across interconnected networks. Think of it like GPS navigation—except instead of one car finding one route, we're managing millions of simultaneous journeys.

Service Provider vs Enterprise Routing

```
Enterprise Network:              Service Provider Network:
┌─────────────┐                  ┌──────────────────────┐
│ Small Scale  │                 │ Massive Scale        │
│ 100s routes  │                 │ 1M+ routes           │
│ Single Org   │                 │ Many Customers       │
│ Best Effort  │                 │ SLA Guarantees       │
└─────────────┘                  └──────────────────────┘
```

The Routing Table: Your Network's Map

The routing table (RIB - Routing Information Base) is the master database containing all known destinations and how to reach them:

```
Destination        Next-Hop        Interface    Metric    Protocol
10.1.1.0/24       192.168.1.1     ge-0/0/0     10        OSPF
172.16.0.0/16     192.168.2.1     ge-0/0/1     20        BGP
0.0.0.0/0         203.0.113.1     ge-0/0/2     100       Static
```

Interior vs Exterior Gateway Protocols

Service providers use two categories of routing protocols:

IGPs (Interior Gateway Protocols):

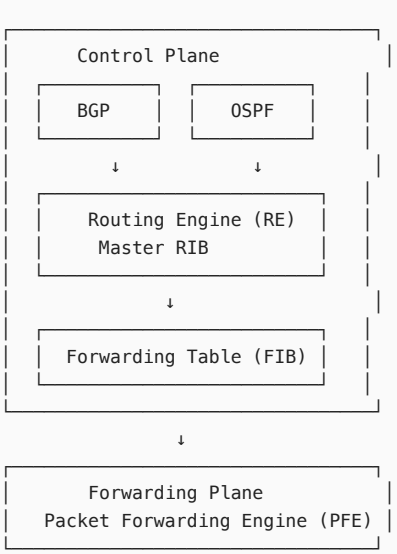
- **Purpose**: Route within your autonomous system (AS)
- **Examples**: OSPF, IS-IS
- **Characteristics**: Fast convergence, detailed topology knowledge
- **Analogy**: City street maps—detailed, frequently updated

EGPs (Exterior Gateway Protocols):

- **Purpose**: Route between autonomous systems
- **Example**: BGP
- **Characteristics**: Policy-based, scalable, slower convergence
- **Analogy**: Interstate highway system—connects cities, fewer details

The Junos OS Advantage

Junos OS implements a unique architecture optimized for service provider requirements:



Key Concepts:

- **Control Plane**: Makes routing decisions (brain)
- **Forwarding Plane**: Moves packets based on decisions (muscles)
- **Separation**: Ensures forwarding continues even during route calculations

Administrative Distance and Route Preference

When multiple protocols provide routes to the same destination, Junos uses preference values (lower is better):

Protocol	Default Preference	Purpose
Direct	0	Directly connected networks
Static	5	Manually configured routes
OSPF Internal	10	Routes within OSPF domain
IS-IS L1 Internal	15	IS-IS Level 1 routes
IS-IS L2 Internal	18	IS-IS Level 2 routes
BGP	170	External BGP routes

Part 2: The Junos CLI Masterclass (The How)

Understanding the Junos Configuration Hierarchy

Junos organizes configuration in a logical tree structure:

```
[edit]
├── system           # System-wide settings
├── interfaces       # Physical and logical interfaces
├── routing-options  # Static routes, router-id, AS number
├── protocols        # Dynamic routing protocols
└── policy-options   # Routing policies
```

Essential Initial Configuration

Let's build a complete base configuration for a service provider router:

```
## Step 1: Set the router identification
[edit routing-options]
set router-id 192.168.1.1
set autonomous-system 65001

## Step 2: Configure interfaces
[edit interfaces]
set ge-0/0/0 description "To-Core-Router-1"
set ge-0/0/0 unit 0 family inet address 10.0.0.1/30
set ge-0/0/1 description "To-Customer-1"
set ge-0/0/1 unit 0 family inet address 192.168.100.1/24
set lo0 unit 0 family inet address 192.168.1.1/32
```

```
## Step 3: Configure OSPF (basic)
[edit protocols ospf]
set area 0.0.0.0 interface ge-0/0/0.0
set area 0.0.0.0 interface lo0.0 passive

## Step 4: Configure BGP (basic)
[edit protocols bgp]
set group EBGP-CUSTOMERS type external
set group EBGP-CUSTOMERS peer-as 65100
set group EBGP-CUSTOMERS neighbor 192.168.100.2

## Step 5: Create a basic routing policy
[edit policy-options]
set policy-statement CUSTOMER-IMPORT term 1 from protocol bgp
set policy-statement CUSTOMER-IMPORT term 1 from route-filter 192.168.0.0/16 orlonger
set policy-statement CUSTOMER-IMPORT term 1 then accept
set policy-statement CUSTOMER-IMPORT term 2 then reject

## Step 6: Apply the routing policy
[edit protocols bgp group EBGP-CUSTOMERS]
set import CUSTOMER-IMPORT
```

Complete Reference Configuration

Here's a production-ready base configuration incorporating all elements:

```
## COMPLETE BASE CONFIGURATION FOR SERVICE PROVIDER ROUTER

system {
    host-name SP-EDGE-01;
    domain-name provider.net;
    root-authentication {
        encrypted-password "$ABC123..."; ## SECRET-DATA
    }
    services {
        ssh {
            protocol-version v2;
        }
        netconf {
            ssh;
        }
    }
    syslog {
        file messages {
            any info;
            authorization info;
        }
    }
    ntp {
        server 10.1.1.100;
    }
}

interfaces {
    ge-0/0/0 {
        description "CORE-LINK-1: To SP-CORE-01";
        unit 0 {
            family inet {
                address 10.0.0.1/30;
            }
            family mpls;
        }
    }
    ge-0/0/1 {
        description "CUST: ABC-Corp";
        unit 0 {
            family inet {
                address 192.168.100.1/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.1.1/32 {
                    primary;
```

```
                }
            }
        }
    }
}

routing-options {
    router-id 192.168.1.1;
    autonomous-system 65001;
    forwarding-table {
        export LOAD-BALANCE;
    }
}

protocols {
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0 {
                interface-type p2p;
                metric 10;
            }
            interface lo0.0 {
                passive;
            }
        }
    }
    bgp {
        group IBGP-CORE {
            type internal;
            local-address 192.168.1.1;
            family inet {
                unicast;
            }
            neighbor 192.168.1.2;
            neighbor 192.168.1.3;
        }
        group EBGP-CUSTOMERS {
            type external;
            peer-as 65100;
            neighbor 192.168.100.2 {
                description "Customer: ABC-Corp";
                import CUST-IN;
                export CUST-OUT;
            }
        }
    }
    lldp {
        interface all;
    }
}

policy-options {
    policy-statement CUST-IN {
        term ACCEPT-CUSTOMER-ROUTES {
            from {
                protocol bgp;
                route-filter 10.0.0.0/8 orlonger;
            }
            then accept;
        }
        term REJECT-ALL {
            then reject;
        }
    }
    policy-statement CUST-OUT {
        term ADVERTISE-DEFAULTS {
            from {
                route-filter 0.0.0.0/0 exact;
            }
            then accept;
        }
        term REJECT-ALL {
            then reject;
        }
    }
    policy-statement LOAD-BALANCE {
```

```
        then {
            load-balance per-packet;
        }
    }
}
```

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

Essential Verification Commands

Master these commands for daily operations:

```
## Check routing table
show route summary
show route protocol ospf
show route protocol bgp
show route 10.1.1.0/24 detail

## Verify protocol status
show ospf neighbor
show bgp summary
show bgp neighbor 192.168.100.2

## Check interface status
show interfaces terse
show interfaces ge-0/0/0 extensive

## Monitor real-time updates
monitor traffic interface ge-0/0/0
monitor start messages
```

Sample Output Analysis

Good State - OSPF Neighbor:

```
user@SP-EDGE-01> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
10.0.0.2         ge-0/0/0.0             Full      192.168.1.2      128    38
```

✓ **State: Full** = Adjacency established successfully ✓ **Dead timer counting down** = Keepalives being received

Good State - BGP Summary:

```
user@SP-EDGE-01> show bgp summary
Groups: 2 Peers: 3 Down peers: 0
Peer                     AS      InPkt     OutPkt    State|#Active/Received
192.168.1.2           65001      15234      15100    Establ    250/300
192.168.100.2         65100       5432       5401    Establ     10/15
```

✓ **State: Establ** = BGP session up ✓ **Active/Received routes** = Learning and selecting routes

Common Troubleshooting Scenarios

Scenario 1: No Connectivity Between Routers

Symptom: Cannot ping remote router, no routes learned

```
user@SP-EDGE-01> ping 10.0.0.2
PING 10.0.0.2: 56 data bytes
ping: sendto: No route to host
```

Diagnosis:

```
user@SP-EDGE-01> show interfaces ge-0/0/0 terse
Interface        Admin Link Proto    Local
ge-0/0/0         up    down                    ## ← Problem: Link down
```

Solution:

```
## Check physical connectivity
> show interfaces ge-0/0/0 extensive | match "link|speed|duplex"
```

```
## If negotiation issues:
[edit interfaces ge-0/0/0]
set speed 1g
set link-mode full-duplex
set gigether-options no-auto-negotiation
commit
```

Scenario 2: OSPF Neighbor Stuck in Init State

Symptom: OSPF not forming adjacency

```
user@SP-EDGE-01> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
10.0.0.2         ge-0/0/0.0             Init      192.168.1.2      128    38
```

Diagnosis:

```
## Check OSPF interface configuration
> show ospf interface detail
## Look for mismatched network types or MTU

## Check for MTU mismatch
> show interfaces ge-0/0/0 | match mtu
```

Solution:

```
[edit protocols ospf area 0.0.0.0]
set interface ge-0/0/0.0 interface-type p2p  ## Match on both sides
[edit interfaces ge-0/0/0]
set mtu 1500  ## Ensure MTU matches
commit
```

Scenario 3: BGP Routes Not Being Accepted

Symptom: BGP session up but no routes in routing table

```
user@SP-EDGE-01> show bgp summary
Peer                     AS      InPkt     OutPkt    State|#Active/Received
192.168.100.2         65100        123        110    Establ      0/15
                                                                  ## ← 0 active routes!
```

Diagnosis:

```
## Check hidden routes (policy rejections)
> show route receive-protocol bgp 192.168.100.2 hidden detail

## Check import policy
> show configuration protocols bgp group EBGP-CUSTOMERS neighbor 192.168.100.2 import

## Test policy
> test policy CUST-IN 10.50.1.0/24
```

Solution:

```
## Fix policy to accept desired routes
[edit policy-options policy-statement CUST-IN]
set term ACCEPT-CUSTOMER-ROUTES from route-filter 10.50.0.0/16 orlonger
commit
```

Scenario 4: Routing Loop Detected

Symptom: Traceroute shows packets bouncing between routers

```
user@SP-EDGE-01> traceroute 10.99.1.1
1  10.0.0.2  0.423 ms
2  10.0.0.1  0.512 ms  ## ← Loops back!
3  10.0.0.2  0.623 ms
```

Diagnosis:

```
## Check route on both routers
> show route 10.99.1.0/24

## Check for redistribution issues
> show configuration protocols ospf export
> show configuration policy-options
```

Solution:

```
## Implement loop prevention with proper policy
[edit policy-options policy-statement REDIS-TO-OSPF]
set term PREVENT-LOOPS from protocol ospf
set term PREVENT-LOOPS then reject
set term REDISTRIBUTE from protocol bgp
set term REDISTRIBUTE from route-filter 10.99.0.0/16 orlonger
set term REDISTRIBUTE then accept
commit
```

Module 1 Complete: You now understand the fundamental architecture of service provider routing, can configure basic Junos routing, and troubleshoot common issues. This foundation prepares you for the protocol-specific deep dives in upcoming modules. Remember: every complex network is just a collection of simple concepts working together. Master the basics, and complexity becomes manageable.

Module 2: OSPF (Open Shortest Path First)

Part 1: The Conceptual Lecture (The Why)

The Problem OSPF Solves

Imagine you're managing a postal service where every post office needs to know the fastest route to every other post office. Now imagine routes change constantly—roads close, new ones open, traffic varies. You need a system where every post office automatically learns about changes and recalculates the best paths. That's OSPF.

The Core Challenge: In large networks, routers need to:

- Automatically discover all available paths
- Calculate the optimal route to every destination
- Rapidly adapt when network topology changes
- Scale to hundreds or thousands of routers
- Minimize protocol overhead

Understanding Link-State vs Distance-Vector

Distance-Vector (like RIP):

- "I'll tell you how far destinations are from me"
- Shares routing table with neighbors
- Simple but slow to converge
- Prone to routing loops

Link-State (OSPF):

- "I'll tell everyone about my direct connections"
- Shares topology information with entire area
- Complex but fast convergence
- Loop-free by design

```
Distance-Vector Logic:          Link-State Logic:
"NY is 3 hops from me"         "I connect to Boston at 10Gbps"
"LA is 5 hops from me"         "I connect to Philly at 1Gbps"
                               → Everyone builds complete map
```

How OSPF Works: The Complete Picture

1. Neighbor Discovery

OSPF routers find each other using Hello packets:

```
Time 0:00  Router A               Router B
           [Hello, I'm A]  →
                          ←  [Hello, I'm B]
           "B is my neighbor!"
                             "A is my neighbor!"
```

Hello Packet Contents:

- Router ID (unique identifier)
- Area ID (must match)
- Hello/Dead intervals (must match)
- Network mask (must match on broadcast networks)
- Options (stub flag must match)

2. Database Synchronization

Routers exchange their complete topology knowledge:

```
Phase 1: Database Description (DBD)
Router A: "I have LSAs 1,2,3,5,8"
Router B: "I have LSAs 1,2,3,4,6,7"

Phase 2: Link-State Request (LSR)
Router A: "Send me LSAs 4,6,7"
Router B: "Send me LSAs 5,8"

Phase 3: Link-State Update (LSU)
[Exchanges missing LSAs]

Phase 4: Full Adjacency
Both routers have identical databases
```

3. Link-State Advertisements (LSAs)

LSAs are the building blocks of OSPF's topology database:

```
LSA Type 1 - Router LSA:
┌─────────────────────────────┐
│ Router ID: 192.168.1.1      │
│ Links:                      │
│   - 10.0.0.0/30 Cost: 10    │
│   - 10.0.1.0/30 Cost: 100   │
│   - 192.168.1.1/32 Cost:0   │
└─────────────────────────────┘

LSA Type 2 - Network LSA (for broadcast networks):
┌─────────────────────────────┐
│ DR: 192.168.1.1             │
│ Network: 10.1.1.0/24        │
│ Attached Routers:           │
│   - 192.168.1.1             │
│   - 192.168.1.2             │
│   - 192.168.1.3             │
└─────────────────────────────┘
```

Complete LSA Type Reference:

Type	Name	Generated By	Scope	Purpose
1	Router	Every router	Area	Router's links
2	Network	DR	Area	Broadcast network
3	Summary	ABR	Area	Inter-area routes
4	ASBR Summary	ABR	Area	Location of ASBR
5	AS External	ASBR	AS	External routes
7	NSSA External	ASBR in NSSA	Area	External in NSSA

4. SPF Algorithm (Dijkstra's Algorithm)

OSPF builds a shortest-path tree with itself as root:

```
Network Topology:          SPF Calculation from A:
    [A]                    Step 1: A (cost 0)
   /   \                   Step 2: A→B (cost 10)
  10    20                 Step 3: A→C (cost 20)
 /       \                 Step 4: A→B→D (cost 25)
[B]--15--[D]              Step 5: A→C→D (cost 35) - higher, ignored
 \       /
  30   40                  Result: A→D via B (cost 25)
   \   /
    [C]
```

OSPF Network Types

OSPF adapts its behavior based on network type:

```
1. Point-to-Point (P2P):
   [R1]--------[R2]
   - No DR/BDR election
   - Multicast to 224.0.0.5

2. Broadcast (Ethernet):
   [R1]--[SW]--[R2]
          |
         [R3]
   - DR/BDR election
   - Multicast to 224.0.0.5 (all routers)
   - Multicast to 224.0.0.6 (DR/BDR)

3. NBMA (Frame Relay):
   Similar to broadcast but no multicast
   - Requires neighbor statements
   - DR/BDR election

4. Point-to-Multipoint:
   Hub-and-spoke treated as P2P links
   - No DR/BDR
   - Automatic neighbor discovery
```

OSPF Authentication

OSPF supports three authentication types:

1. **Null** (Type 0): No authentication
2. **Simple Password** (Type 1): Clear text (insecure)
3. **MD5** (Type 2): Cryptographic hash

```
Authentication Process:
[OSPF Packet | MD5 Hash of (Packet + Secret Key)]
              ↓
Receiver computes hash with its key
              ↓
If hashes match = Authentic packet
```

OSPFv2 vs OSPFv3

Feature	OSPFv2	OSPFv3
IP Version	IPv4	IPv6
Transport	IP Protocol 89	IP Protocol 89
Multicast	224.0.0.5/6	FF02::5/6
Authentication	Per-area/interface	IPSec
Router ID	32-bit	32-bit (still)
Multiple Instances	No	Yes (per-link)

Part 2: The Junos CLI Masterclass (The How)

OSPF Configuration Hierarchy

```
[edit protocols ospf]
├── area <area-id>
│   ├── interface <name>
│   │   ├── metric <cost>
│   │   ├── priority <0-255>
│   │   ├── authentication
│   │   └── interface-type <type>
│   └── area-range <prefix>
├── reference-bandwidth <bandwidth>
├── export <policy>
└── traffic-engineering
```

Complete OSPF Configuration

Let's build a comprehensive OSPF configuration step-by-step:

```
## Step 1: Basic OSPF Setup
[edit protocols ospf]
set reference-bandwidth 100g  # Adjust metrics for modern speeds

## Step 2: Configure Backbone Area (0.0.0.0)
set area 0.0.0.0 interface lo0.0 passive
set area 0.0.0.0 interface ge-0/0/0.0 interface-type p2p
set area 0.0.0.0 interface ge-0/0/0.0 metric 10
set area 0.0.0.0 interface ge-0/0/0.0 hello-interval 10
set area 0.0.0.0 interface ge-0/0/0.0 dead-interval 40

## Step 3: Configure Broadcast Network
set area 0.0.0.0 interface ge-0/0/1.0 priority 150  # Higher = more likely DR
set area 0.0.0.0 interface ge-0/0/1.0 retransmit-interval 5
set area 0.0.0.0 interface ge-0/0/1.0 transit-delay 1

## Step 4: Enable Authentication
set area 0.0.0.0 interface ge-0/0/0.0 authentication md5 1 key "$9$H.5FnCu1Iyrev"
set area 0.0.0.0 interface ge-0/0/1.0 authentication md5 2 key "$9$mP5F39pOIEy"

## Step 5: Configure Traffic Engineering Extensions
set traffic-engineering
set area 0.0.0.0 interface ge-0/0/0.0 te-metric 100

## Step 6: Enable BFD for Fast Failure Detection
set area 0.0.0.0 interface ge-0/0/0.0 bfd-liveness-detection minimum-interval 300
set area 0.0.0.0 interface ge-0/0/0.0 bfd-liveness-detection multiplier 3

## Step 7: Configure Graceful Restart
set graceful-restart restart-duration 120
set graceful-restart notify-duration 30
set graceful-restart helper-disable

## Step 8: Set Protocol Preferences
[edit protocols ospf]
set preference 10         # Internal routes
set external-preference 150  # External routes

## Step 9: Configure Overload (maintenance mode)
set overload timeout 300  # Advertise max metric for 300 seconds after boot

## Step 10: Advanced Options
set spf-options delay 50 holddown 5000 rapid-runs 3
set lsa-refresh-interval 30  # Minutes (default 50)
```

Production-Ready OSPF Configuration

```
## COMPLETE PRODUCTION OSPF CONFIGURATION

protocols {
    ospf {
        reference-bandwidth 100g;
        graceful-restart {
            restart-duration 120;
            notify-duration 30;
        }
        traffic-engineering;
        spf-options {
```

```
            delay 50;
            holddown 5000;
            rapid-runs 3;
        }
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface ge-0/0/0.0 {
                interface-type p2p;
                metric 10;
                hello-interval 10;
                dead-interval 40;
                authentication {
                    md5 1 key "$9$H.5FnCu1Iyrev"; ## SECRET-DATA
                }
                bfd-liveness-detection {
                    minimum-interval 300;
                    multiplier 3;
                    transmit-interval {
                        minimum-interval 300;
                    }
                    detection-time {
                        threshold 900;
                    }
                }
            }
            interface ge-0/0/1.0 {
                priority 150;
                metric 20;
                retransmit-interval 5;
                transit-delay 1;
                authentication {
                    md5 1 key "$9$mP5F39pOIEy"; ## SECRET-DATA
                }
            }
            interface ge-0/0/2.0 {
                passive;
                metric 1000;  # Backup link - high cost
            }
        }
        export REDISTRIBUTE-CONNECTED;
    }
}

policy-options {
    policy-statement REDISTRIBUTE-CONNECTED {
        term LOOPBACKS {
            from {
                protocol direct;
                interface lo0.0;
            }
            then accept;
        }
        term CUSTOMER-NETWORKS {
            from {
                protocol direct;
                route-filter 192.168.0.0/16 prefix-length-range /24-/32;
            }
            then {
                metric 100;
                external {
                    type 1;  # Type 1 = metric increases through network
                }
                accept;
            }
        }
    }
}
```

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

Essential OSPF Verification Commands

```
## Overall OSPF Status
show ospf overview
show ospf neighbor
show ospf neighbor detail
show ospf interface
show ospf interface detail

## Database Examination
show ospf database
show ospf database router        # Type 1 LSAs
show ospf database network       # Type 2 LSAs
show ospf database extern        # Type 5 LSAs
show ospf database summary       # Overview
show ospf database advertising-router 192.168.1.1

## Route Verification
show ospf route
show route protocol ospf
show route table inet.0 protocol ospf detail

## Statistics and Troubleshooting
show ospf statistics
show ospf log
clear ospf neighbor 10.0.0.2
clear ospf database purge
```

Reading OSPF Neighbor Output

Healthy Neighbor State:

```
user@R1> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
10.0.0.2         ge-0/0/0.0             Full      192.168.1.2      128    37
10.1.1.2         ge-0/0/1.0             Full      192.168.1.3      100    35
10.1.1.3         ge-0/0/1.0             2Way      192.168.1.4       50    38
```

State Meanings:

- **Down**: No hellos received
- **Init**: Hellos received, but not two-way
- **2Way**: Bidirectional communication (normal for non-DR/BDR)
- **ExStart**: Starting database exchange
- **Exchange**: Exchanging database descriptions
- **Loading**: Requesting LSAs
- **Full**: Databases synchronized ✓

Common OSPF Problems and Solutions

Scenario 1: Stuck in ExStart State

Symptom: Neighbors never reach Full state

```
user@R1> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
10.0.0.2         ge-0/0/0.0             ExStart   192.168.1.2      128    39
```

Diagnosis:

```
## Check for MTU mismatch
show interfaces ge-0/0/0 | match mtu
show ospf interface ge-0/0/0.0 detail | match mtu

## Check for duplicate router-ids
show ospf database | match "Advertising Router"
```

Solution:

```
## Fix MTU mismatch
[edit interfaces ge-0/0/0]
set mtu 1500
```

```
## Or ignore MTU in OSPF
[edit protocols ospf area 0.0.0.0]
set interface ge-0/0/0.0 mtu-ignore
commit
```

Scenario 2: Routes Missing from Routing Table

Symptom: OSPF database has routes but routing table doesn't

```
user@R1> show ospf database extern | match 10.99.1.0
    10.99.1.0         255.255.255.0     10.0.0.2         100

user@R1> show route 10.99.1.0
(no output)
```

Diagnosis:

```
## Check for hidden routes
show route 10.99.1.0 hidden detail

## Check OSPF route table
show ospf route 10.99.1.0

## Check for import policy
show configuration protocols ospf import
```

Solution:

```
## Remove blocking import policy or fix it
[edit protocols ospf]
delete import
# OR
[edit policy-options policy-statement OSPF-IMPORT]
set term ALLOW-EXTERNALS from protocol ospf
set term ALLOW-EXTERNALS from route-type external
set term ALLOW-EXTERNALS then accept
```

Scenario 3: DR/BDR Election Issues

Symptom: Wrong router became DR on broadcast network

```
user@R1> show ospf interface ge-0/0/1.0 detail | match "DR|Priority"
  Priority: 1, Designated router: 10.1.1.3, BDR: 10.1.1.2
  # But R1 should be DR!
```

Diagnosis:

```
## Check all router priorities
show ospf neighbor detail | match "Priority|Address"
```

Solution:

```
## Set higher priority and restart OSPF process
[edit protocols ospf area 0.0.0.0]
set interface ge-0/0/1.0 priority 200
commit

## Clear OSPF to force re-election
restart routing immediately
## OR for less disruption:
clear ospf neighbor all
```

Module 3: OSPF Areas

Part 1: The Conceptual Lecture (The Why)

The Scalability Problem

As networks grow, OSPF faces challenges:

- **LSA Flooding**: Every change floods to all routers
- **SPF Calculations**: (n^2) complexity where n = routers
- **Memory Requirements**: Complete topology database
- **CPU Load**: Frequent recalculations

The Solution: Hierarchical design using areas!

```
Flat Network (Problems):        Hierarchical (Solution):
    All 100 routers              Area 0 (Backbone)
    know everything                  10 routers
         ↓                              ↓
  10,000 LSAs each              Area 1    Area 2
  100 SPF calculations           30 rtrs   30 rtrs
  Slow convergence              Isolated SPF domains
```

OSPF Area Concepts

Areas create boundaries that limit LSA propagation:

```
                Area 0 (Backbone)
                 [ABR1]---[ABR2]
                /    \    /    \
               /       \/       \
              /        /\         \
         Area 1      Area 2      Area 3
        [R1][R2]    [R3][R4]    [R5][R6]

Key Concepts:
- ABR: Area Border Router (connects areas)
- All areas MUST connect to Area 0
- Areas hide their internal topology
- ABRs create summary (Type 3) LSAs
```

Area Types Deep Dive

1. Normal Area

- Accepts all LSA types
- Full routing information
- Can have external routes

2. Stub Area

Reduces database size by blocking external routes:

```
Normal Area LSAs:          Stub Area LSAs:
Type 1: Router ✓          Type 1: Router ✓
Type 2: Network ✓         Type 2: Network ✓
Type 3: Summary ✓         Type 3: Summary ✓
Type 4: ASBR Sum ✓        Type 4: ASBR Sum ✗
Type 5: External ✓        Type 5: External ✗
                          + Default route (0.0.0.0/0)
```

Use Case: Branch offices that don't need full internet routing table

3. Totally Stubby Area (Cisco calls it "Totally Stub")

Even more restrictive:

```
Blocks:                    Allows:
Type 3: Inter-area ✗       Type 1: Router ✓
Type 4: ASBR Sum ✗         Type 2: Network ✓
Type 5: External ✗         Type 3: Default only
                          (0.0.0.0/0 from ABR)
```

Use Case: Remote sites needing only default route

4. Not-So-Stubby Area (NSSA)

Stub area that can originate external routes:

```
NSSA Special Behavior:
External route in NSSA → Type 7 LSA
                ↓
            At ABR
                ↓
    Type 7 → Type 5 conversion
                ↓
        Floods to Area 0
```

Use Case: Branch with local internet connection

5. Totally NSSA

Combines Totally Stubby + NSSA:

- No Type 3 (except default)
- No Type 4
- No Type 5
- Can originate Type 7

Route Summarization

Summarization reduces routing table size and improves stability:

```
Without Summarization:         With Summarization:
Area 1 advertises:             Area 1 advertises:
  10.1.0.0/24                    10.1.0.0/16
  10.1.1.0/24                         ↑
  10.1.2.0/24                    Single summary
  10.1.3.0/24

Benefit: If 10.1.2.0/24 flaps, Area 0 doesn't recalculate SPF!
```

Summarization Rules:

1. Only ABRs can summarize (Type 3 LSAs)
2. ASBRs can summarize external routes (Type 5/7)
3. Summary should cover all more-specific routes
4. Careful with discontiguous networks!

Part 2: The Junos CLI Masterclass (The How)

Configuring Stub Areas

```
## STUB AREA CONFIGURATION

## On all routers in the stub area:
[edit protocols ospf area 0.0.0.1]
set stub

## On ABR only - inject default route:
[edit protocols ospf area 0.0.0.1]
set stub default-metric 10

## For Totally Stubby (on ABR only):
[edit protocols ospf area 0.0.0.1]
set stub no-summaries
```

Configuring NSSA

```
## NSSA CONFIGURATION

## On all routers in NSSA:
[edit protocols ospf area 0.0.0.2]
set nssa

## On ABR - configure default route and Type 7→5 translation:
[edit protocols ospf area 0.0.0.2]
set nssa default-lsa default-metric 10
set nssa default-lsa metric-type 1
set nssa no-summaries  # For Totally NSSA
```

```
## On ASBR within NSSA - redistribute with Type 7:
[edit policy-options policy-statement REDIS-TO-NSSA]
set term 1 from protocol bgp
set term 1 then nssa-only  # Create Type 7, not Type 5
set term 1 then accept

[edit protocols ospf]
set export REDIS-TO-NSSA
```

Route Summarization Configuration

```
## AREA RANGE (Inter-Area Summarization on ABR)

[edit protocols ospf]
## Summarize Area 1's routes when advertising to Area 0
set area 0.0.0.1 area-range 10.1.0.0/16

## With options:
set area 0.0.0.1 area-range 10.1.0.0/16 restrict  # Don't advertise
set area 0.0.0.1 area-range 10.1.0.0/16 exact      # Only exact match
set area 0.0.0.1 area-range 10.1.0.0/16 override-metric 100

## EXTERNAL ROUTE SUMMARIZATION (on ASBR)
[edit policy-options policy-statement SUMMARIZE-EXTERNAL]
set term 1 from protocol bgp
set term 1 from route-filter 192.168.0.0/16 orlonger
set term 1 to protocol ospf
set term 1 then aggregate route 192.168.0.0/16
set term 1 then accept

[edit protocols ospf]
set export SUMMARIZE-EXTERNAL
```

Complete Multi-Area Configuration

```
## COMPREHENSIVE MULTI-AREA OSPF WITH ALL AREA TYPES

## AREA 0 - BACKBONE
protocols {
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface ge-0/0/0.0 {
                interface-type p2p;
                metric 10;
            }
            interface ge-0/0/1.0 {
                interface-type p2p;
                metric 10;
            }
        }

        ## AREA 1 - NORMAL AREA
        area 0.0.0.1 {
            interface ge-0/0/2.0 {
                interface-type p2p;
            }
            interface ge-0/0/3.0;
            ## Summarize when advertising to backbone
            area-range 10.1.0.0/16;
        }

        ## AREA 2 - STUB AREA
        area 0.0.0.2 {
            stub default-metric 10;
            interface ge-0/0/4.0;
            interface ge-0/0/5.0;
            ## Summarize stub networks
            area-range 10.2.0.0/16;
        }
```

```
        ## AREA 3 - TOTALLY STUBBY
        area 0.0.0.3 {
            stub default-metric 10 no-summaries;
            interface ge-0/0/6.0;
        }

        ## AREA 4 - NSSA
        area 0.0.0.4 {
            nssa {
                default-lsa {
                    default-metric 10;
                    metric-type 1;
                }
            }
            interface ge-0/0/7.0;
            interface ge-0/0/8.0;
        }

        ## AREA 5 - TOTALLY NSSA
        area 0.0.0.5 {
            nssa {
                default-lsa {
                    default-metric 10;
                    metric-type 2;
                }
                no-summaries;
            }
            interface ge-0/0/9.0;
        }
    }
}
```

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

Verification Commands for Areas

```
## Verify area configuration
show ospf interface detail | match "Area|Type"
show ospf database summary

## Check stub area operation
show ospf database netsummary area 0.0.0.2
show ospf database extern    # Should be empty in stub

## Verify NSSA operation
show ospf database nssa-extern
show ospf route 10.99.1.0 detail

## Check summarization
show ospf database netsummary advertising-router 192.168.1.1
show route protocol ospf 10.1.0.0/16 detail
```

Common Area Configuration Issues

Scenario 1: Stub Area Mismatch

Symptom: Neighbors stuck in ExStart/Exchange

```
user@R1> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
10.2.1.2         ge-0/0/4.0             ExStart   192.168.1.20     128    38
```

Diagnosis:

```
show configuration protocols ospf area 0.0.0.2 | display set
show ospf interface ge-0/0/4.0 detail | match "Options"
## E-bit = 0 means stub area
```

Solution:

```
## Ensure all routers in area have same stub config
[edit protocols ospf area 0.0.0.2]
```

```
set stub    # On ALL routers in area
commit
```

Scenario 2: No Default Route in Stub Area

Symptom: Stub area can't reach external destinations

```
user@R2> show route 0.0.0.0/0
## No output
```

Solution:

```
## On ABR only:
[edit protocols ospf area 0.0.0.2]
set stub default-metric 10
commit
```

Module 4: OSPF Case Studies and Solutions

Part 1: The Conceptual Lecture (The Why)

Virtual Links: Solving Non-Contiguous Backbone

Problem: All areas must connect to Area 0, but physical topology doesn't allow it

```
Initial Problem:              Virtual Link Solution:
Area 0---ABR1---Area 1       Area 0---ABR1---Area 1
                                        Virtual Link
         Area 2                        ============
     (Can't reach Area 0!)             ABR2---Area 2

Virtual Link = Tunnel through Area 1 to connect Area 2 to Area 0
```

How Virtual Links Work:

1. Creates a point-to-point link through transit area
2. Uses router-IDs as endpoints (not IP addresses!)
3. Inherits cost of path through transit area
4. Cannot traverse stub areas (no Type 4/5 LSAs)

Multiarea Adjacencies: Optimizing ABR Connections

Traditional Design:

```
ABR needs separate interface per area:
ge-0/0/0 → Area 0
ge-0/0/1 → Area 1
ge-0/0/2 → Area 2
```

Multiarea Adjacency:

```
Single interface in multiple areas:
ge-0/0/0 → Area 0 (primary)
         → Area 1 (secondary)
         → Area 2 (secondary)
```

Benefits:

- Fewer physical connections required
- Reduced interface count
- Maintains separate LSDBs per area
- Backward compatible

External Reachability Strategies

Service providers must carefully control external route injection:

```
1. Type 1 vs Type 2 Externals:

Type 1: Cost = External metric + Internal path cost
Type 2: Cost = External metric only (default)

Example:
ASBR advertises 8.8.8.0/24 with metric 100
                ↓
Type 1: R1 sees cost 110 (100 + 10 internal)
        R2 sees cost 120 (100 + 20 internal)
Type 2: R1 sees cost 100
        R2 sees cost 100

Use Type 1 for: Load balancing, accurate metrics
Use Type 2 for: Simple preference, administrative control
```

```
2. Forwarding Address:

Normal: Traffic to external → ASBR → External destination
With FA: Traffic → Forwarding Address (bypass ASBR)

When FA is set:
- ASBR and next-hop share common subnet
- Saves extra hop
- Reduces ASBR load
```

Part 2: The Junos CLI Masterclass (The How)

Configuring Virtual Links

```
## VIRTUAL LINK CONFIGURATION

## Identify endpoints (use router-IDs, not IPs!)
## Transit area cannot be stub!

## On both ABRs:
[edit protocols ospf area 0.0.0.1]  # Transit area
set virtual-link neighbor-id 192.168.1.2 transit-area 0.0.0.1

## With authentication:
set virtual-link neighbor-id 192.168.1.2 authentication md5 1 key "secret"

## Verify virtual link:
show ospf interface vl-192.168.1.2
show ospf neighbor detail | match Virtual
```

Configuring Multiarea Adjacencies

```
## MULTIAREA ADJACENCY CONFIGURATION

[edit protocols ospf]
## Primary area (normal)
set area 0.0.0.0 interface ge-0/0/0.0

## Secondary area (multiarea)
set area 0.0.0.1 interface ge-0/0/0.0 multiarea-adjacency

## Can have multiple secondary areas
set area 0.0.0.2 interface ge-0/0/0.0 multiarea-adjacency

## Verification
show ospf interface ge-0/0/0.0 detail
## Will show multiple areas
```

External Route Injection with Advanced Options

```
## CONTROLLING EXTERNAL REACHABILITY

## Policy for external route injection
policy-options {
    policy-statement EXTERNAL-ROUTES {
        term CUSTOMER-ROUTES {
```

```
            from {
                protocol bgp;
                as-path CUSTOMERS;
                route-filter 0.0.0.0/0 orlonger;
            }
            then {
                metric 100;
                external {
                    type 1;  # Add internal cost
                }
                tag 6500;    # For identification
                accept;
            }
        }
        term PEER-ROUTES {
            from {
                protocol bgp;
                as-path PEERS;
            }
            then {
                metric 200;
                external {
                    type 2;  # Fixed metric
                }
                tag 6501;
                accept;
            }
        }
        term SET-FORWARDING-ADDRESS {
            from {
                protocol bgp;
                next-hop 10.1.1.1;  # Shared subnet with OSPF
            }
            then {
                forwarding-address 10.1.1.1;
                accept;
            }
        }
    }

    as-path CUSTOMERS "^65[0-9][0-9][0-9]$";
    as-path PEERS "^64[0-9][0-9][0-9]$";
}

[edit protocols ospf]
set export EXTERNAL-ROUTES
```

Complete Advanced OSPF Configuration

```
## PRODUCTION CONFIGURATION WITH ALL ADVANCED FEATURES

protocols {
    ospf {
        reference-bandwidth 100g;

        ## BACKBONE AREA WITH VIRTUAL LINK
        area 0.0.0.0 {
            interface lo0.0 passive;
            interface ge-0/0/0.0 {
                interface-type p2p;
                metric 10;
            }
            ## Multiarea adjacency
            interface ge-0/0/1.0;
        }

        ## TRANSIT AREA FOR VIRTUAL LINK
        area 0.0.0.1 {
            interface ge-0/0/2.0;
            interface ge-0/0/3.0;
            ## Virtual link to connect remote area
            virtual-link neighbor-id 192.168.1.10 transit-area 0.0.0.1 {
                authentication {
                    md5 1 key "$9$hcreW87Vb2oGi"; ## SECRET-DATA
                }
```

```
                retransmit-interval 5;
                hello-interval 10;
                dead-interval 40;
            }
            ## Secondary area on shared interface
            interface ge-0/0/1.0 multiarea-adjacency;
        }

        ## REMOTE AREA (connected via virtual link)
        area 0.0.0.2 {
            interface ge-0/0/4.0;
            nssa {
                default-lsa {
                    default-metric 10;
                    metric-type 1;
                }
            }
        }

        ## External route injection
        export INJECT-EXTERNAL;

        ## Traffic engineering database
        traffic-engineering {
            database {
                import {
                    l3-unicast-topology;
                    policy IMPORT-TED;
                }
                export {
                    l3-unicast-topology;
                }
            }
        }
    }
}

policy-options {
    policy-statement INJECT-EXTERNAL {
        term BGP-TO-OSPF {
            from protocol bgp;
            then {
                metric 150;
                tag 65001;
                external {
                    type 1;
                }
                accept;
            }
        }
        term STATICS {
            from {
                protocol static;
                route-filter 192.168.0.0/16 orlonger;
            }
            then {
                metric 200;
                tag 100;
                external {
                    type 2;
                }
                accept;
            }
        }
    }
}
```

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

Virtual Link Troubleshooting

Scenario 1: Virtual Link Not Forming

Symptom: Remote area unreachable

```
user@ABR1> show ospf interface vl-192.168.1.10
## No output or "Down" state
```

Diagnosis:

```
## Check if transit area is stub
show configuration protocols ospf area 0.0.0.1 | match stub

## Verify router-IDs
show ospf overview | match "Router ID"

## Check reachability through transit area
show ospf route 192.168.1.10
```

Solution:

```
## Remove stub configuration from transit area
[edit protocols ospf area 0.0.0.1]
delete stub

## Ensure both ends configured
## On ABR1:
set area 0.0.0.1 virtual-link neighbor-id 192.168.1.10 transit-area 0.0.0.1
## On ABR2:
set area 0.0.0.1 virtual-link neighbor-id 192.168.1.1 transit-area 0.0.0.1
```

Scenario 2: External Routes Not Optimal Path

Symptom: Traffic taking longer path to external destinations

```
user@R1> traceroute 8.8.8.8
1  10.0.0.1 (ASBR1)  1ms
2  10.0.0.5 (ASBR2)  2ms  # Wrong ASBR!
3  8.8.8.8           10ms
```

Solution:

```
## Change from Type 2 to Type 1 external
[edit policy-options policy-statement EXTERNAL-ROUTES]
set term PREFERRED-EXIT then external type 1
set term PREFERRED-EXIT then metric 50  # Lower than other ASBR
```

Module 5: OSPF Troubleshooting

Part 1: The Conceptual Lecture (The Why)

Systematic Troubleshooting Methodology

OSPF problems follow patterns. Master this methodology:

```
1. LAYER 1-2 VERIFICATION
   ↓ Interfaces up?
2. OSPF NEIGHBOR STATE
   ↓ Adjacencies formed?
3. DATABASE CONSISTENCY
   ↓ LSDBs synchronized?
4. SPF CALCULATION
   ↓ Routes computed?
5. ROUTING TABLE
   ↓ Routes installed?
6. FORWARDING TABLE
   → Traffic flowing?
```

Common OSPF Failure Patterns

```
Neighbor Issues:              Database Issues:
- MTU mismatch               - LSA aging problems
- Area mismatch              - Sequence number wraps
```

```
- Hello/Dead mismatch        - Corruption
- Authentication fail        - Memory exhaustion
- Duplicate router-ID

Routing Issues:              Performance Issues:
- Route preference           - SPF throttling
- Policy blocking            - LSA flooding storms
- Summarization errors       - Slow convergence
- Metric problems            - CPU overload
```

OSPFv3 Specific Considerations

OSPFv3 for IPv6 has unique characteristics:

```
OSPFv2 (IPv4):              OSPFv3 (IPv6):
- Router-ID from IP         - Router-ID still 32-bit
- Auth in OSPF header       - IPSec authentication
- One instance               - Multiple instances
- Interface by IP           - Interface by link
```

Part 2: The Junos CLI Masterclass (The How)

Advanced Troubleshooting Commands

```
## DEEP DIAGNOSTICS

## Trace OSPF operations
[edit protocols ospf]
set traceoptions file ospf-debug.log size 10m
set traceoptions flag hello detail
set traceoptions flag error
set traceoptions flag state
set traceoptions flag lsa-update detail
set traceoptions flag spf detail
commit

## Monitor the trace
monitor start ospf-debug.log
## Perform action that triggers issue
monitor stop

## Analyze SPF calculations
show ospf spf log
show ospf spf log detail

## Check for database inconsistencies
show ospf database checksum
show ospf database detail | match "Age|Seq|Checksum"

## Memory and performance
show task memory detail | match ospf
show task io | match ospf
show system processes extensive | match rpd
```

OSPFv3 Configuration and Verification

```
## OSPFv3 FOR IPv6

## Configure OSPFv3
[edit protocols ospf3]
set area 0.0.0.0 interface ge-0/0/0.0
set area 0.0.0.0 interface lo0.0 passive

## Realm for multiple topologies
set realm ipv4-unicast area 0.0.0.0 interface ge-0/0/0.0
set realm ipv4-multicast area 0.0.0.0 interface ge-0/0/1.0

## Verification
show ospf3 neighbor
show ospf3 database
show ospf3 interface
show route table inet6.0 protocol ospf3
```

Comprehensive Troubleshooting Scenarios

```
## SCENARIO: Complete OSPF failure investigation

## Step 1: Verify physical connectivity
show interfaces ge-0/0/0 | match "Physical|Description|Link"
show interfaces ge-0/0/0 extensive | match "error|drop|collision"
ping 10.0.0.2 rapid count 100

## Step 2: Check OSPF configuration
show configuration protocols ospf | display set
show ospf interface detail

## Step 3: Examine neighbor relationships
show ospf neighbor detail
show log messages | match "OSPF|RPD_OSPF"

## Step 4: Analyze database
show ospf database summary
show ospf database detail | match "192.168.1.1"
show ospf database router lsa-id 192.168.1.1 detail

## Step 5: Verify routing
show ospf route
show route protocol ospf hidden detail
show route forwarding-table destination 10.1.1.0/24

## Step 6: Check for loops
show route 10.1.1.1 detail | match "via|metric"
traceroute 10.1.1.1 no-resolve
```

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

Complex Troubleshooting Scenarios

Scenario 1: Intermittent OSPF Flapping

Symptom: OSPF adjacency flaps every few minutes

```
Mar 10 10:15:23 RPD_OSPF_NBRDOWN: Neighbor 10.0.0.2 state changed from Full to Init
Mar 10 10:15:45 RPD_OSPF_NBRFULL: Neighbor 10.0.0.2 state changed to Full
Mar 10 10:18:11 RPD_OSPF_NBRDOWN: Neighbor 10.0.0.2 state changed from Full to Init
```

Diagnosis:

```
## Check for packet loss
ping 10.0.0.2 rapid count 1000
## Look for drops

## Check interface errors
show interfaces ge-0/0/0 extensive | match "error|drop"

## Enable debugging
set protocols ospf traceoptions flag hello detail
monitor start ospf-debug.log

## Check BFD if enabled
show bfd session address 10.0.0.2 detail
```

Solution:

```
## Tune timers if packet loss exists
[edit protocols ospf area 0.0.0.0 interface ge-0/0/0.0]
set hello-interval 10
set dead-interval 40
set retransmit-interval 5

## Disable BFD if too aggressive
delete bfd-liveness-detection

## Or tune BFD
```

```
set bfd-liveness-detection minimum-interval 1000
set bfd-liveness-detection multiplier 5
```

Scenario 2: Database Synchronization Failure

Symptom: Different LSA counts on neighbors

```
user@R1> show ospf database summary
    28 Router LSAs
    5 Network LSAs
    15 Summary LSAs
    45 Extern LSAs

user@R2> show ospf database summary
    28 Router LSAs
    5 Network LSAs
    15 Summary LSAs
    12 Extern LSAs    ## Different!
```

Diagnosis:

```
## Find missing LSAs
show ospf database extern | match "ID|Advertising"
## Compare outputs

## Check for policy filtering
show configuration policy-options
show configuration protocols ospf import
```

Solution:

```
## Clear OSPF process to force resync
restart routing immediately

## If persist, check for memory issues
request system reboot
```

Scenario 3: OSPFv3 Not Establishing

Symptom: OSPFv3 neighbors not forming despite OSPFv2 working

```
user@R1> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
10.0.0.2         ge-0/0/0.0             Full      192.168.1.2      128    38

user@R1> show ospf3 neighbor
## No output
```

Solution:

```
## Ensure IPv6 enabled on interface
[edit interfaces ge-0/0/0 unit 0]
set family inet6 address 2001:db8::1/64

## Configure OSPFv3
[edit protocols ospf3]
set area 0.0.0.0 interface ge-0/0/0.0

## Set router-ID explicitly (required if no IPv4)
[edit routing-options]
set router-id 192.168.1.1
```

Module 6: IS-IS (Intermediate System to Intermediate System)

Part 1: The Conceptual Lecture (The Why)

Why IS-IS? The Service Provider's Secret Weapon

IS-IS is the protocol most large service providers actually use for their backbone networks, despite OSPF being more famous. Here's why:

Historical Context:

- Developed for OSI networks (not IP originally)
- Adopted for IP with "Integrated IS-IS" (RFC 1195)
- Protocol number 124 (not IP protocol 89 like OSPF)
- Runs directly on Layer 2 (not on top of IP)

Key Advantages:

```
IS-IS Benefits:              OSPF Limitations:
- Incredible stability       - More complex LSA types
- Simpler design             - Tied to IP
- Larger scale (1000s)       - Area 0 requirement
- Easier IPv6 addition       - Separate OSPFv2/v3
- Flexible extensibility     - Fixed packet formats
```

IS-IS Terminology: A New Language

IS-IS uses OSI terminology that seems alien at first:

IS-IS Term	OSPF Equivalent	Meaning
Intermediate System (IS)	Router	A router
End System (ES)	Host	End device
Circuit	Interface	Network connection
System ID	Router ID	Unique identifier
NET	Router ID + Area	Network Entity Title
NSAP	—	Network Service Access Point
PDU	Packet	Protocol Data Unit
LSP	LSA	Link State PDU
CSNP	DD	Complete Sequence Number PDU
PSNP	LSR	Partial Sequence Number PDU
DIS	DR	Designated IS
L1	Intra-area	Level 1 routing
L2	Inter-area	Level 2 routing

Network Entity Title (NET) Structure

The NET is IS-IS's addressing system:

```
49.0001.1921.6800.1001.00
|    |       |         |
|    |       |         └─ NSEL (always 00)
|    |       └─────────── System ID (6 bytes)
|    └─────────────────── Area ID (variable)
└──────────────────────── AFI (49 = private)

Example breakdown:
49.0001.1921.6800.1001.00
- AFI: 49 (private addressing)
- Area: 0001
- System ID: 1921.6800.1001 (from 192.168.1.1)
- NSEL: 00 (router, not service)
```

IS-IS Levels: Hierarchical by Design

```
Level 2 (Backbone)
   [L2]-----[L2]
    |  \   /  |
    |   \ /   |
    |    X    |
    |   / \   |
    |  /   \  |
  [L1/L2] [L1/L2]  ← L1/L2 routers connect areas
    |       |
  Area 1  Area 2
  [L1]-[L1] [L1]-[L1]
```

```
Key Concepts:
- L1: Routes within area (like OSPF intra-area)
- L2: Routes between areas (like OSPF backbone)
- L1/L2: Connects L1 to L2 (like OSPF ABR)
- No Area 0 requirement!
```

IS-IS PDU Types

IS-IS uses different PDUs for different functions:

```
1. Hello PDUs:
   - IIH (IS-IS Hello)
   - Point-to-point IIH
   - LAN IIH (L1 and L2 separate)

2. Link State PDUs (LSPs):
   - Generated by every router
   - Contains all router's links
   - Flooded throughout level

3. Sequence Number PDUs:
   - CSNP: Complete list (like OSPF DD)
   - PSNP: Request specific LSPs (like LSR)
```

IS-IS Adjacency Formation

```
Point-to-Point Adjacency:
R1                          R2
│                           │
├─ P2P IIH "I'm R1" ──────→│
│                           │
│←────── P2P IIH "I'm R2" ─┤
│                           │
│        Adjacency UP!      │

LAN Adjacency (with DIS election):
R1 ──┐
     │
R2 ──┼── Shared LAN
     │
R3 ──┘
    ↓
DIS elected (highest priority/MAC)
All routers form adjacency with all others
(Unlike OSPF DR/BDR model)
```

IS-IS Metrics and Path Selection

IS-IS uses simpler metrics than OSPF:

```
Original (Narrow) Metrics:    Modern (Wide) Metrics:
- 6-bit interface (max 63)    - 24-bit interface (max 16M)
- 10-bit path (max 1023)      - 32-bit path (max 4B)
```

Default Junos Behavior:

- All interfaces: metric 10
- Reference bandwidth not used by default
- Manual configuration recommended

Part 2: The Junos CLI Masterclass (The How)

Basic IS-IS Configuration

```
## FUNDAMENTAL IS-IS SETUP

## Step 1: Configure NET address
[edit interfaces lo0 unit 0]
set family iso address 49.0001.1921.6800.1001.00

## Step 2: Enable IS-IS on interfaces
```

```
[edit protocols isis]
set interface ge-0/0/0.0
set interface ge-0/0/1.0
set interface lo0.0 passive

## Step 3: Set interface metrics
set interface ge-0/0/0.0 level 1 metric 10
set interface ge-0/0/0.0 level 2 metric 10
set interface ge-0/0/1.0 level 1 disable  # L2 only
set interface ge-0/0/1.0 level 2 metric 100

## Step 4: Configure authentication
set level 1 authentication-key "L1SecretKey"
set level 1 authentication-type md5
set level 2 authentication-key "L2SecretKey"
set level 2 authentication-type md5

## Step 5: Enable wide metrics
set level 1 wide-metrics-only
set level 2 wide-metrics-only

## Step 6: Configure DIS priority (for LANs)
set interface ge-0/0/2.0 level 1 priority 100
set interface ge-0/0/2.0 level 2 priority 100
```

Advanced IS-IS Configuration

```
## COMPREHENSIVE IS-IS CONFIGURATION

routing-options {
    router-id 192.168.1.1;
}

interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.1.1/32;
            }
            family iso {
                address 49.0001.1921.6800.1001.00;
            }
        }
    }
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.0.0.1/30;
            }
            family iso;  ## Required for IS-IS
            family mpls;
        }
    }
}

protocols {
    isis {
        ## Global options
        reference-bandwidth 100g;
        level 1 disable;  ## L2-only backbone router
        level 2 {
            authentication-key "$9$H.mfz/A0IEy"; ## SECRET-DATA
            authentication-type md5;
            wide-metrics-only;
        }

        ## Traffic engineering
        traffic-engineering {
            family inet {
                shortcuts;
            }
        }

        ## Interfaces
        interface ge-0/0/0.0 {
```

```
            point-to-point;
            level 2 metric 10;
            bfd-liveness-detection {
                minimum-interval 300;
                multiplier 3;
            }
        }

        interface ge-0/0/1.0 {
            point-to-point;
            level 2 metric 10;
            ldp-synchronization;
        }

        interface lo0.0 {
            passive;
        }

        ## Overload bit (maintenance mode)
        overload timeout 300;

        ## Export policy
        export REDISTRIBUTE-STATIC;

        ## SPF options
        spf-options {
            delay 50;
            holddown 5000;
            rapid-runs 3;
        }

        ## Graceful restart
        graceful-restart {
            restart-duration 120;
        }
    }
}

policy-options {
    policy-statement REDISTRIBUTE-STATIC {
        term 1 {
            from protocol static;
            then {
                metric 100;
                level 2;
                accept;
            }
        }
    }
}
```

IS-IS Multi-Level Configuration

```
## MULTI-LEVEL IS-IS SETUP

## L1/L2 Border Router Configuration
protocols {
    isis {
        interface ge-0/0/0.0 {
            ## L1 interface to area routers
            level 2 disable;
            level 1 metric 10;
        }

        interface ge-0/0/1.0 {
            ## L2 interface to backbone
            level 1 disable;
            level 2 metric 10;
        }

        interface ge-0/0/2.0 {
            ## Both levels on same interface
            level 1 metric 10;
            level 2 metric 100;
        }
```

```
        ## Leak L2 routes to L1 (controlled)
        export L2-TO-L1-LEAKING;
    }
}

policy-options {
    policy-statement L2-TO-L1-LEAKING {
        term DEFAULT-ROUTE {
            from {
                protocol isis;
                level 2;
                route-filter 0.0.0.0/0 exact;
            }
            to level 1;
            then accept;
        }
        term SPECIFIC-PREFIXES {
            from {
                protocol isis;
                level 2;
                route-filter 10.0.0.0/8 orlonger;
            }
            to level 1;
            then {
                metric 1000;
                accept;
            }
        }
    }
}
```

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

Essential IS-IS Verification Commands

```
## Basic verification
show isis adjacency
show isis adjacency extensive
show isis interface
show isis interface detail
show isis database
show isis database detail
show isis database extensive

## Routing verification
show isis route
show route protocol isis
show route table inet.0 protocol isis detail

## Statistics and debugging
show isis statistics
show isis spf log
show isis spf results

## Hostname mapping
show isis hostname
```

Reading IS-IS Adjacency Output

Healthy Adjacency:

```
user@R1> show isis adjacency
Interface             System         L State       Hold (secs) SNPA
ge-0/0/0.0            R2             2  Up                    23
ge-0/0/1.0            R3             2  Up                    27
```

Detailed View:

```
user@R1> show isis adjacency extensive
R2
  Interface: ge-0/0/0.0, Level: 2, State: Up, Expires in 23 secs
  Priority: 64, Up/Down transitions: 1, Last transition: 00:45:12 ago
```

```
  Circuit type: 2, Speaks: IP, IPv6, MPLS
  IP addresses: 10.0.0.2
  Transition log:
    Down  10:15:23.123 Timeout
    Up    10:15:45.456 New adjacency
```

Common IS-IS Troubleshooting Scenarios

Scenario 1: Adjacency Not Forming

Symptom: No IS-IS neighbors

```
user@R1> show isis adjacency
## No output
```

Diagnosis:

```
## Check ISO family on interface
show configuration interfaces ge-0/0/0.0 | match iso

## Verify NET address
show configuration interfaces lo0.0 family iso

## Check for area mismatch
show isis interface detail | match "Area|Level"
```

Solution:

```
## Add ISO family to interface
[edit interfaces ge-0/0/0 unit 0]
set family iso

## Fix NET address if wrong area
[edit interfaces lo0 unit 0]
set family iso address 49.0001.1921.6800.1001.00
## Area portion must match for L1 adjacency
```

Scenario 2: L1/L2 Level Mismatch

Symptom: Adjacency not forming between different levels

```
user@R1> show isis interface ge-0/0/0.0 detail
  L1 State: Disabled
  L2 State: Up

user@R2> show isis interface ge-0/0/0.0 detail
  L1 State: Up
  L2 State: Disabled
## Mismatch!
```

Solution:

```
## Enable both levels or match configuration
## Option 1: Enable L2 on R2
[edit protocols isis interface ge-0/0/0.0]
delete level 2 disable
set level 2 metric 10

## Option 2: Make both L1/L2
[edit protocols isis interface ge-0/0/0.0]
delete level 1 disable
delete level 2 disable
```

Scenario 3: Routes Missing Between Levels

Symptom: L1 routers can't reach L2 destinations

```
user@L1-Router> show route 192.168.100.0/24
## No route to L2 network
```

Solution:

```
## On L1/L2 border router, leak routes or set ATT bit
[edit protocols isis]
set level 1 attached  ## Sets attached bit for default

## Or use policy for specific leaking
[edit policy-options policy-statement LEAK-L2-TO-L1]
set term 1 from protocol isis
set term 1 from level 2
set term 1 to level 1
set term 1 then accept

[edit protocols isis]
set export LEAK-L2-TO-L1
```

Scenario 4: Wide Metrics Compatibility

Symptom: Partial reachability in mixed network

```
user@R1> show isis database extensive | match "Metric|style"
  Metric style: Narrow
  ## Old router using narrow metrics

user@R2> show isis database extensive | match "Metric|style"
  Metric style: Wide
  ## New router using wide metrics only
```

Solution:

```
## Enable both metric styles during transition
[edit protocols isis]
set level 2 wide-metrics-only  ## Eventually
## OR during transition:
delete level 2 wide-metrics-only
## This allows both narrow and wide
```

Modules 2-6 Complete: You now have comprehensive knowledge of OSPF from basic operations through advanced troubleshooting, plus a solid foundation in IS-IS. These protocols form the backbone of service provider networks worldwide. The key to mastery is understanding not just the "how" but the "why" behind each feature—this understanding lets you design and troubleshoot networks at the expert level required for JNCIE-SP certification.

Module 7: Advanced IS-IS Operations and Configuration Options

Part 1: The Conceptual Lecture (The Why)

Understanding the IS-IS Link-State Database (LSDB)

Let me start from the very beginning. IS-IS (Intermediate System to Intermediate System) is like a sophisticated mapping system for networks. Imagine you're organizing a massive city with thousands of intersections - you need a way for every traffic controller to know the complete map of the city to make optimal routing decisions.

The Fundamental Problem

Networks need to share information about their topology - who connects to whom and at what cost. IS-IS solves this by having every router maintain a complete "map" of the network called the Link-State Database (LSDB). Think of it as each router having an identical GPS map of the entire highway system.

How the LSDB Works

The LSDB contains Link-State PDUs (Protocol Data Units) - these are like detailed reports from each router about their direct connections:

```
     [Router A]                    [Router B]
          |                             |
    "I connect to B"            "I connect to A"
    "Link cost: 10"             "Link cost: 10"
    "My IP: 10.0.0.1"           "My IP: 10.0.0.2"
          |                             |
          v                             v
    +-----------+                +-----------+
    |   LSDB    |<-------------->|   LSDB    |
```

```
    +-----------+               +-----------+
    Both databases contain identical information
```

Each router creates an LSP (Link-State PDU) that contains:

- **System ID**: The router's unique identifier (like a social security number)
- **Neighbors**: Who it's directly connected to
- **IP Prefixes**: What networks it knows about
- **Metrics**: The "cost" to reach each destination

Advanced IS-IS Configurations

Beyond basic connectivity, IS-IS offers sophisticated tools:

1. **Mesh Groups**: Prevent LSP flooding storms in highly meshed networks
   - Problem: In a full mesh of 10 routers, each LSP gets replicated 90 times!
   - Solution: Group interfaces to reduce redundant flooding
2. **Overload Bit**: Temporarily remove a router from transit paths
   - Like putting a "Road Closed for Through Traffic" sign
   - Local destinations still reachable, but no transit traffic
3. **Wide Metrics**: Extended from 6-bit (max 63) to 24-bit (max 16,777,215)
   - Original IS-IS was designed when 63 was considered a large metric
   - Modern networks need more granularity
4. **BFD Integration**: Bidirectional Forwarding Detection for sub-second failure detection
   - Regular IS-IS might take 30+ seconds to detect a failure
   - BFD can detect in milliseconds

IS-IS Routing Policies

Routing policies in IS-IS control what information enters and leaves the LSDB:

```
Network Reality              Policy Filter              LSDB
---------------              -------------            --------
192.168.1.0/24    ------>   [Accept]      ------>   Installed
192.168.2.0/24    ------>   [Reject]      ---X-->   Not installed
10.0.0.0/8        ------>   [Modify metric] ----->  Installed (metric 100)
```

Policies answer questions like:

- Which routes should I advertise to neighbors?
- Which routes should I accept from neighbors?
- How should I modify route characteristics?

Part 2: The Junos CLI Masterclass (The How)

Understanding the IS-IS Configuration Hierarchy

In Junos, IS-IS configuration lives under `[edit protocols isis]`. The structure reflects the protocol's logical organization:

```
[edit protocols isis]
    |
    +-- interface <name>     # Physical interface configuration
    |     +-- level 1        # Level 1 specific settings
    |     +-- level 2        # Level 2 specific settings
    |     +-- mesh-group     # Flooding optimization
    |
    +-- level 1              # Global Level 1 settings
    +-- level 2              # Global Level 2 settings
    +-- overload             # Overload bit configuration
    +-- traffic-engineering  # MPLS-TE extensions
    +-- export               # Export routing policy
    +-- import               # Import routing policy
```

Complete Advanced IS-IS Configuration

Let's build a comprehensive IS-IS configuration with advanced features:

```
## Step 1: Enable Wide Metrics (required for modern networks)
set protocols isis level 1 wide-metrics-only
set protocols isis level 2 wide-metrics-only
```

```
## Step 2: Configure Interfaces with Different Levels
# Core interface - Level 2 only (inter-area routing)
set protocols isis interface ge-0/0/0.0 level 1 disable
set protocols isis interface ge-0/0/0.0 level 2 metric 100
set protocols isis interface ge-0/0/0.0 point-to-point  # No DIS election needed

# Access interface - Level 1 only (intra-area routing)
set protocols isis interface ge-0/0/1.0 level 2 disable
set protocols isis interface ge-0/0/1.0 level 1 metric 10
set protocols isis interface ge-0/0/1.0 level 1 hello-interval 1
set protocols isis interface ge-0/0/1.0 level 1 hold-time 3

## Step 3: Configure Mesh Groups (prevent flooding loops in mesh topologies)
set protocols isis interface ge-0/0/2.0 mesh-group 100
set protocols isis interface ge-0/0/3.0 mesh-group 100
set protocols isis interface ge-0/0/4.0 mesh-group 100
# LSPs received on one mesh-group interface won't flood to others in same group

## Step 4: Configure Overload Bit with Timeout
set protocols isis overload timeout 300  # Auto-clear after 5 minutes
set protocols isis overload advertise-high-metrics  # Advertise with high metrics

## Step 5: Configure BFD for Fast Failure Detection
set protocols isis interface ge-0/0/0.0 bfd-liveness-detection minimum-interval 100
set protocols isis interface ge-0/0/0.0 bfd-liveness-detection multiplier 3

## Step 6: Configure Loopback as Passive (advertise but don't form adjacencies)
set protocols isis interface lo0.0 passive

## Step 7: Traffic Engineering Extensions
set protocols isis traffic-engineering family inet shortcuts
set protocols isis traffic-engineering credibility-protocol-preference

## Step 8: Configure Authentication
set protocols isis level 2 authentication-key "$9$aes256$encrypted-key"
set protocols isis level 2 authentication-type md5

## Step 9: Create and Apply Routing Policies
# Export Policy - Control what routes we advertise into IS-IS
set policy-options policy-statement isis-export term direct-routes from protocol direct
set policy-options policy-statement isis-export term direct-routes from route-filter 192.168.0.0/16 orlonger
set policy-options policy-statement isis-export term direct-routes then metric 100
set policy-options policy-statement isis-export term direct-routes then tag 100
set policy-options policy-statement isis-export term direct-routes then accept

set policy-options policy-statement isis-export term bgp-routes from protocol bgp
set policy-options policy-statement isis-export term bgp-routes from community bgp-to-isis
set policy-options policy-statement isis-export term bgp-routes then metric 200
set policy-options policy-statement isis-export term bgp-routes then accept

set policy-options policy-statement isis-export term default then reject

# Import Policy - Filter incoming IS-IS routes
set policy-options policy-statement isis-import term filter-bad from route-filter 0.0.0.0/0 exact
set policy-options policy-statement isis-import term filter-bad then reject
set policy-options policy-statement isis-import term filter-bad from route-filter 224.0.0.0/4 orlonger
set policy-options policy-statement isis-import term filter-bad then reject

set policy-options policy-statement isis-import term accept-rest then accept

# Apply the policies
set protocols isis export isis-export
set protocols isis import isis-import

## Step 10: Reference Bandwidth for Auto-Cost Calculation
set protocols isis reference-bandwidth 100g

## Step 11: SPF Computation Optimization
set protocols isis spf-options delay 200
set protocols isis spf-options holddown 5000
set protocols isis spf-options rapid-runs 3
```

Complete Reference Configuration Block

```
protocols {
    isis {
        reference-bandwidth 100g;
        export isis-export;
        import isis-import;
        level 1 wide-metrics-only;
        level 2 {
            wide-metrics-only;
            authentication-key "$9$aes256$encrypted-key";
            authentication-type md5;
        }
        spf-options {
            delay 200;
            holddown 5000;
            rapid-runs 3;
        }
        overload {
            timeout 300;
            advertise-high-metrics;
        }
        traffic-engineering {
            family inet {
                shortcuts;
            }
            credibility-protocol-preference;
        }
        interface ge-0/0/0.0 {
            point-to-point;
            level 1 disable;
            level 2 metric 100;
            bfd-liveness-detection {
                minimum-interval 100;
                multiplier 3;
            }
        }
        interface ge-0/0/1.0 {
            level 2 disable;
            level 1 {
                metric 10;
                hello-interval 1;
                hold-time 3;
            }
        }
        interface ge-0/0/2.0 {
            mesh-group 100;
        }
        interface ge-0/0/3.0 {
            mesh-group 100;
        }
        interface ge-0/0/4.0 {
            mesh-group 100;
        }
        interface lo0.0 {
            passive;
        }
    }
}
```

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

Essential Verification Commands

1. Verify IS-IS Adjacencies

```
user@router> show isis adjacency
Interface             System         L State        Hold (secs) SNPA
ge-0/0/0.0            router-2       2  Up                   27 0:1a:2b:3c:4d:5e
ge-0/0/1.0            router-3       1  Up                    2 0:1a:2b:3c:4d:5f

# Detailed adjacency information
user@router> show isis adjacency extensive
router-2
  Interface: ge-0/0/0.0, Level: 2, State: Up, Expires in 27 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 00:15:43 ago
  Circuit type: 2, Speaks: IP, IPv6
```

```
  Topologies: Unicast
  Restart capable: Yes, Adjacency advertisement: Advertise
  IP addresses: 10.0.0.2
```

2. Examine the LSDB

```
user@router> show isis database
IS-IS level 1 link-state database:
LSP ID                  Sequence Checksum Lifetime Attributes
router-1.00-00               0x45   0x7d4e     1087 L1 L2
router-3.00-00               0x23   0x9a1c      945 L1
  2 LSPs

IS-IS level 2 link-state database:
LSP ID                  Sequence Checksum Lifetime Attributes
router-1.00-00               0x78   0x3d21     1087 L1 L2
router-2.00-00               0x56   0x8c4f      823 L2
  2 LSPs

# Detailed LSP examination
user@router> show isis database router-1.00-00 extensive
IS-IS level 2 link-state database:

router-1.00-00 Sequence: 0x78, Checksum: 0x3d21, Lifetime: 1087 secs
   IS neighbor: router-2.00                  Metric:      100
   IP prefix: 192.168.1.0/24                 Metric:      100 Internal Up
   IP prefix: 10.0.0.0/30                    Metric:      100 Internal Up
```

3. Verify Routing Policy Impact

```
user@router> show route advertising-protocol isis
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
Prefix                  Nexthop              MED     Lclpref    AS path
192.168.1.0/24          Self                 100                I
192.168.2.0/24          Self                 100                I

user@router> show route receive-protocol isis
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
  Prefix                 Nexthop              MED     Lclpref    AS path
* 10.1.0.0/24            10.0.0.2             200                I
```

Common Troubleshooting Scenarios

Scenario 1: Adjacency Won't Form

Symptom: `show isis adjacency` shows no neighbors

```
user@router> show isis adjacency
# No output
```

Diagnostic Commands:

```
user@router> show isis interface
Interface             L CirID Level 1 DR        Level 2 DR        L1/L2 Metric
ge-0/0/0.0            2   0x2 Disabled          router-1.02             10/10

user@router> show isis statistics | match "IIH|errors"
IIH sent: 1523
IIH received: 0          # <-- Problem: Not receiving hellos
Authentication errors: 0
```

Cause: MTU mismatch or authentication failure

```
user@router> monitor traffic interface ge-0/0/0.0 no-resolve extensive matching "isis"
# Shows IS-IS packets with "Authentication failed" in packet details
```

Solution:

```
# Fix MTU
set interfaces ge-0/0/0 mtu 9192
```

```
# Fix authentication
set protocols isis interface ge-0/0/0.0 level 2 authentication-key SecretKey123
```

Scenario 2: Routes Not Being Advertised

Symptom: Expected routes missing from neighbor's routing table

```
user@neighbor> show route protocol isis 192.168.1.0/24
# No routes found
```

Diagnostic Commands:

```
user@router> show isis database router-1 detail | match 192.168
# No output - prefix not in LSP

user@router> show route 192.168.1.0/24
192.168.1.0/24    *[Direct/0] 00:10:00
                     > via ge-0/0/1.0

user@router> test policy isis-export 192.168.1.0/24
Policy isis-export: 0 prefix accepted, 1 prefix rejected
# Rejected by policy
```

Cause: Export policy not matching the route

```
user@router> show configuration policy-options policy-statement isis-export
term direct-routes {
    from {
        protocol direct;
        route-filter 10.0.0.0/8 orlonger;  # Wrong filter!
    }
}
```

Solution:

```
delete policy-options policy-statement isis-export term direct-routes from route-filter 10.0.0.0/8
set policy-options policy-statement isis-export term direct-routes from route-filter 192.168.0.0/16 orlonger
commit
```

Scenario 3: Overload Bit Causing Traffic Loss

Symptom: Router not forwarding transit traffic

```
user@router> show isis overview | match overload
  Overload: Set, Advertise: High metrics  # <-- Problem identified
```

Diagnostic Commands:

```
user@router> show isis database router-1.00-00 extensive | match "OL|Overload"
   NLPID: 0x83, Fixed length: 27 bytes, Version: 1, Sysid length: 0 bytes
   Packet type: 20, Packet length: 1492, Remaining lifetime: 1087 secs
   Checksum: 0x3d21, Sequence: 0x78, Attributes: 0x3 <L1 L2 OL>
                                                      ^^
                                                Overload bit set!
```

Cause: Overload configured without timeout

```
user@router> show configuration protocols isis overload
set;  # Permanently set!
```

Solution:

```
delete protocols isis overload set
set protocols isis overload timeout 300
commit
```

Scenario 4: Mesh Group Preventing LSP Propagation

Symptom: LSP updates not reaching all routers in mesh

```
user@router-3> show isis database
# Missing LSPs from some routers
```

Diagnostic Commands:

```
user@router> show isis interface extensive ge-0/0/2.0 | match mesh
  Level 1 mesh group: Blocked
  Level 2 mesh group: 100

user@router> show isis mesh-group
Mesh Group 100:
  Interfaces: ge-0/0/2.0, ge-0/0/3.0, ge-0/0/4.0
  Blocked LSPs: 15  # LSPs not flooded due to mesh-group
```

Cause: All mesh interfaces in same mesh group

```
user@router> show configuration protocols isis | match "mesh-group"
interface ge-0/0/2.0 mesh-group 100;
interface ge-0/0/3.0 mesh-group 100;
interface ge-0/0/4.0 mesh-group 100;  # All in same group!
```

Solution:

```
# Remove at least one interface from mesh-group for redundancy
delete protocols isis interface ge-0/0/4.0 mesh-group
commit
```

This completes Module 7. The advanced IS-IS features we've covered - LSDB interpretation, mesh groups, overload bit, routing policies, and wide metrics - form the foundation for scalable service provider networks. These tools allow you to optimize IS-IS for networks ranging from small campus deployments to massive service provider backbones with thousands of routers.

Module 8: Multilevel IS-IS Networks

Part 1: The Conceptual Lecture (The Why)

The Fundamental Problem of Network Scalability

Imagine you're organizing communication for an entire country. If every post office had to know the detailed street address of every house in the nation, the system would collapse. Instead, we use hierarchical addressing - local post offices know local details, regional offices handle inter-city routing, and national centers manage inter-region delivery.

IS-IS solves this same problem using a two-level hierarchy:

```
        [National Backbone – Level 2]
             /            |            \
       [Region A]    [Region B]    [Region C]
        Level 1       Level 1       Level 1
        /    \         /    \         /    \
    Local  Local   Local  Local   Local  Local
    Sites  Sites   Sites  Sites   Sites  Sites
```

How Multilevel IS-IS Works

IS-IS divides the network into areas, but unlike OSPF, IS-IS routers can participate in multiple levels simultaneously:

- **Level 1 (L1)**: Intra-area routing - like local city maps
- **Level 2 (L2)**: Inter-area routing - like interstate highways
- **Level 1/2 (L1/L2)**: Border routers - like highway on-ramps

Key Concepts:

1. **Area Address**: Each area has a unique identifier (like a ZIP code)
2. **System ID**: Each router's unique identifier within the entire network
3. **NET Address**: Network Entity Title = Area + System ID + Selector

Example NET: `49.0001.1920.1680.1001.00`

- `49.0001` = Area address

- `1920.1680.1001` = System ID (often derived from IP: 192.168.0.1)
- `00` = Selector (always 00 for routers)

Default Multilevel Operation

By default, L1/L2 routers:

1. Maintain separate LSDBs for each level
2. Automatically leak L1 routes UP into L2
3. Set the "Attached bit" to tell L1 routers "I can reach other areas"
4. Do NOT leak L2 routes DOWN into L1 (by default)

This creates a problem and an opportunity:

```
Level 1 Router Perspective:
"I know everything in my area in detail"
"For everything else, go to the L1/L2 router (default route)"

Level 2 Router Perspective:
"I know how to reach all areas"
"I know summary information about each area"
```

Address Summarization in IS-IS

Summarization reduces routing table size by aggregating multiple specific routes into one summary:

```
Before Summarization:          After Summarization:
192.168.1.0/24  -->
192.168.2.0/24  -->            192.168.0.0/22
192.168.3.0/24  -->            (Covers 192.168.0.0 - 192.168.3.255)
192.168.4.0/24  -->
```

Benefits:

- Smaller routing tables
- Faster convergence
- Reduced LSP flooding
- Hidden topology changes

Route Leaking

Sometimes L1 routers need specific L2 routes (not just default). Route leaking selectively advertises L2 routes into L1:

```
Use Cases:
1. Optimal routing between L1 areas
2. Avoiding suboptimal paths through L2
3. Traffic engineering
4. Service-specific routing
```

Part 2: The Junos CLI Masterclass (The How)

Understanding Multilevel Configuration Structure

```
[edit protocols isis]
    +-- level 1
    |     +-- wide-metrics-only
    |     +-- preference          # Route preference for L1 routes
    |     +-- external-preference # External route preference
    |     +-- prefix-export-limit # Limit prefixes leaked to L2
    |
    +-- level 2
    |     +-- wide-metrics-only
    |     +-- preference
    |     +-- external-preference
    |
    +-- interface <name>
          +-- level 1
          |     +-- disable       # Make interface L2-only
          |     +-- metric
          |     +-- passive
          |
```

```
          +-- level 2
                +-- disable       # Make interface L1-only
                +-- metric
                +-- passive
```

Complete Multilevel IS-IS Configuration

```
## Step 1: Configure NET address (defines area and system ID)
# Area 49.0001, System ID from loopback IP 192.168.1.1
set interfaces lo0 unit 0 family iso address 49.0001.1921.6800.1001.00

## Step 2: Configure IS-IS with multiple levels
set protocols isis level 1 wide-metrics-only
set protocols isis level 2 wide-metrics-only

## Step 3: Configure level-specific route preferences
set protocols isis level 1 preference 15    # Prefer L1 routes
set protocols isis level 2 preference 18    # L2 routes less preferred
set protocols isis level 1 external-preference 160
set protocols isis level 2 external-preference 165

## Step 4: Configure interfaces for different levels
# Core interfaces - L2 only
set protocols isis interface ge-0/0/0.0 level 1 disable
set protocols isis interface ge-0/0/0.0 level 2 metric 100
set protocols isis interface ge-0/0/0.0 point-to-point

# Area interfaces - L1 only
set protocols isis interface ge-0/0/1.0 level 2 disable
set protocols isis interface ge-0/0/1.0 level 1 metric 10

# Border interfaces - Both L1 and L2
set protocols isis interface ge-0/0/2.0 level 1 metric 10
set protocols isis interface ge-0/0/2.0 level 2 metric 100

## Step 5: Configure route summarization from L1 to L2
# Create summary for L1 prefixes before leaking to L2
set policy-options policy-statement l1-to-l2-summary term summarize from protocol isis
set policy-options policy-statement l1-to-l2-summary term summarize from level 1
set policy-options policy-statement l1-to-l2-summary term summarize from route-filter 192.168.0.0/24 orlonger
set policy-options policy-statement l1-to-l2-summary term summarize to level 2
set policy-options policy-statement l1-to-l2-summary term summarize then accept

# Create aggregate route
set routing-options aggregate route 192.168.0.0/22

# Apply the policy
set protocols isis export l1-to-l2-summary

## Step 6: Configure route leaking from L2 to L1
# Selectively leak specific L2 routes into L1
set policy-options policy-statement l2-to-l1-leak term specific-routes from protocol isis
set policy-options policy-statement l2-to-l1-leak term specific-routes from level 2
set policy-options policy-statement l2-to-l1-leak term specific-routes from route-filter 10.0.0.0/24 exact
set policy-options policy-statement l2-to-l1-leak term specific-routes from route-filter 10.0.1.0/24 exact
set policy-options policy-statement l2-to-l1-leak term specific-routes to level 1
set policy-options policy-statement l2-to-l1-leak term specific-routes then tag 200
set policy-options policy-statement l2-to-l1-leak term specific-routes then accept

# Block everything else from leaking
set policy-options policy-statement l2-to-l1-leak term block-rest to level 1
set policy-options policy-statement l2-to-l1-leak term block-rest then reject

# Apply the leak policy
set protocols isis export l2-to-l1-leak

## Step 7: Control the Attached Bit
# Optionally suppress attached bit to prevent default route generation
set protocols isis level 1 suppress-attached-bit

## Step 8: Configure level-specific authentication
set protocols isis level 1 authentication-key "$9$level1-secret"
set protocols isis level 1 authentication-type md5
set protocols isis level 2 authentication-key "$9$level2-secret"
set protocols isis level 2 authentication-type md5
```

```
## Step 9: Configure level transition optimization
# Set different SPF delays for each level
set protocols isis level 1 spf-delay 50
set protocols isis level 2 spf-delay 200

## Step 10: Wide metrics with different reference bandwidths per level
set protocols isis level 1 reference-bandwidth 10g
set protocols isis level 2 reference-bandwidth 100g
```

Complete Reference Configuration Block

```
interfaces {
    lo0 {
        unit 0 {
            family iso {
                address 49.0001.1921.6800.1001.00;
            }
        }
    }
}
protocols {
    isis {
        export [ l1-to-l2-summary l2-to-l1-leak ];
        level 1 {
            wide-metrics-only;
            preference 15;
            external-preference 160;
            reference-bandwidth 10g;
            spf-delay 50;
            authentication-key "$9$level1-secret";
            authentication-type md5;
            suppress-attached-bit;
        }
        level 2 {
            wide-metrics-only;
            preference 18;
            external-preference 165;
            reference-bandwidth 100g;
            spf-delay 200;
            authentication-key "$9$level2-secret";
            authentication-type md5;
        }
        interface ge-0/0/0.0 {
            point-to-point;
            level 1 disable;
            level 2 metric 100;
        }
        interface ge-0/0/1.0 {
            level 2 disable;
            level 1 metric 10;
        }
        interface ge-0/0/2.0 {
            level 1 metric 10;
            level 2 metric 100;
        }
    }
}
routing-options {
    aggregate {
        route 192.168.0.0/22;
    }
}
```

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

Essential Verification Commands

```
# Verify level participation
user@router> show isis overview
Instance: master
  Router ID: 192.168.1.1
  Hostname: router-1
  Sysid: 1921.6800.1001
```

```
  Areaid: 49.0001
  Adjacency holddown: enabled
  Maximum Areas: 3
  LSP life time: 1200
  Attached bit evaluation: enabled
  SPF delay: 0.200 sec, SPF holddown: 5 sec, SPF rapid runs: 3
  IPv4 is enabled, IPv6 is enabled
  Level 1: enabled
    Internal route preference: 15
    External route preference: 160
  Level 2: enabled
    Internal route preference: 18
    External route preference: 165

# Check multilevel adjacencies
user@router> show isis adjacency detail
router-2
  Interface: ge-0/0/2.0, Level: 1, State: Up, Expires in 27 secs
  Adjacency id: 2, Up/Down transitions: 1
  Circuit type: 1, Speaks: IP, IPv6
router-2
  Interface: ge-0/0/2.0, Level: 2, State: Up, Expires in 28 secs
  Adjacency id: 3, Up/Down transitions: 1
  Circuit type: 2, Speaks: IP, IPv6

# Examine separate LSDBs
user@router> show isis database level 1
IS-IS level 1 link-state database:
LSP ID                  Sequence Checksum Lifetime Attributes
router-1.00-00               0x45   0x7d4e     1087 L1 L2 Attached
router-2.00-00               0x23   0x9a1c      945 L1 L2 Attached

user@router> show isis database level 2
IS-IS level 2 link-state database:
LSP ID                  Sequence Checksum Lifetime Attributes
router-1.00-00               0x78   0x3d21     1087 L1 L2
router-2.00-00               0x56   0x8c4f      823 L1 L2
router-core.00-00            0x34   0x2a15      756 L2

# Verify route leaking
user@router> show isis route-leaking
Level 2 to Level 1:
  Route             Metric  Tag
  10.0.0.0/24          100  200
  10.0.1.0/24          100  200

Level 1 to Level 2:
  Route             Metric
  192.168.0.0/22       10   (summary)
```

Common Troubleshooting Scenarios

Scenario 1: L1 Router Can't Reach Other Areas

Symptom: Level 1 router has no default route

```
user@l1-router> show route 0.0.0.0/0
# No route found
```

Diagnostic Commands:

```
user@l1-router> show isis adjacency | match L2
# No L1/L2 neighbors found

user@l1-router> show isis database detail | match Attached
router-1.00-00 ... Attributes: L1  # Missing "Attached" bit!
```

Cause: No L1/L2 router in the area or attached bit suppressed **Solution**:

```
# On the L1/L2 router, remove suppress-attached-bit
delete protocols isis level 1 suppress-attached-bit
commit
```

Scenario 2: Suboptimal Routing Between L1 Areas

Symptom: Traffic takes longer path through L2 backbone

```
user@router> traceroute 192.168.10.1 source 192.168.1.1
1  10.0.0.1  (L1/L2 router)
2  10.0.1.1  (L2 backbone)
3  10.0.2.1  (L2 backbone)
4  10.0.3.1  (Remote L1/L2)
5  192.168.10.1 (destination)
# Should be only 3 hops!
```

Cause: No specific route leaked, using default **Solution**: Configure route leaking for specific prefixes

```
set policy-options policy-statement l2-to-l1-leak term remote-sites from route-filter 192.168.10.0/24 exact
set policy-options policy-statement l2-to-l1-leak term remote-sites to level 1
set policy-options policy-statement l2-to-l1-leak term remote-sites then accept
```

Scenario 3: Route Summarization Not Working

Symptom: All specific routes appear in L2 instead of summary

```
user@l2-router> show route protocol isis 192.168.0.0/16
192.168.1.0/24    *[IS-IS/18] via ge-0/0/0.0
192.168.2.0/24    *[IS-IS/18] via ge-0/0/0.0
192.168.3.0/24    *[IS-IS/18] via ge-0/0/0.0
# Should see 192.168.0.0/22 summary!
```

Cause: Aggregate route not active **Solution**:

```
# Create contributing route to activate aggregate
set routing-options static route 192.168.0.0/24 discard
set routing-options aggregate route 192.168.0.0/22 preference 130
```

Module 9: IS-IS Troubleshooting

Part 1: The Conceptual Lecture (The Why)

Understanding IS-IS Problem Patterns

IS-IS problems typically fall into these categories:

1. **Adjacency Problems**: Neighbors won't form relationships
2. **Database Problems**: LSPs missing or corrupted
3. **Routing Problems**: Routes present in LSDB but not in routing table
4. **Performance Problems**: Slow convergence or instability

Think of troubleshooting IS-IS like diagnosing a city's traffic system:

- Adjacencies = Roads between intersections
- LSDB = The master map
- Routes = The actual paths traffic takes
- Performance = How quickly the system adapts to changes

The IS-IS Troubleshooting Methodology

```
Step 1: Physical/Link Layer
        |
        v
Step 2: IS-IS Adjacencies
        |
        v
Step 3: LSDB Consistency
        |
        v
Step 4: Route Calculation
        |
```

```
        v
Step 5: Policy Application
```

Each step depends on the previous one working correctly.

Part 2: The Junos CLI Masterclass (The How)

Comprehensive Troubleshooting Toolkit Configuration

```
## Enable IS-IS traceoptions for detailed debugging
set protocols isis traceoptions file isis-debug
set protocols isis traceoptions file size 10m
set protocols isis traceoptions file files 5
set protocols isis traceoptions flag error
set protocols isis traceoptions flag spf
set protocols isis traceoptions flag packets detail
set protocols isis traceoptions flag hello detail
set protocols isis traceoptions flag lsp detail
set protocols isis traceoptions flag psn detail

## Configure event logging
set event-options policy isis-events events rpd_isis_adjacency
set event-options policy isis-events events rpd_isis_lsp
set event-options policy isis-events events rpd_isis_spf
set event-options policy isis-events then log

## Configure SNMP traps for IS-IS events
set snmp trap-group isis-traps version v2
set snmp trap-group isis-traps targets 192.168.100.10
set snmp trap-group isis-traps categories link
set snmp trap-group isis-traps categories routing
```

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

Master Troubleshooting Command Reference

```
## Adjacency Troubleshooting
show isis adjacency
show isis interface
show isis statistics
show log messages | match "ISIS|isis"
monitor traffic interface <name> matching isis

## Database Troubleshooting
show isis database
show isis database extensive
show isis database <lsp-id> detail
show isis spf log
show isis spf results

## Routing Troubleshooting
show route protocol isis
show route protocol isis hidden
show route protocol isis table inet.0 detail
show isis route
show route forwarding-table

## Performance Troubleshooting
show isis statistics
show isis spf log
show isis interface extensive | match "Designated|DIS|Priority"
```

Complex Troubleshooting Scenario: Area Split

Symptom: Some routers can't reach each other

```
user@router-A> ping 192.168.50.1 source 192.168.10.1
PING failed
```

Step-by-step Diagnosis:

```
# 1. Check local adjacencies
user@router-A> show isis adjacency
Interface    System      L State
ge-0/0/0.0   router-B    1 Up
ge-0/0/1.0   router-C    1 Up

# 2. Check area configuration
user@router-A> show isis overview | match "Areaid|Sysid"
Sysid: 1921.6800.1001
Areaid: 49.0001

user@router-A> show isis database | match "router-D"
# No output - router-D LSP missing!

# 3. Check on intermediate router
user@router-B> show isis database | match "router-D"
# Still no output

# 4. Check area mismatch
user@router-D> show isis overview | match "Areaid"
Areaid: 49.0002  # Different area!

# 5. Verify L2 connectivity
user@router-A> show isis database level 2
# No L2 database - this is L1-only router!
```

Cause: Area split with no L2 connectivity **Solution**: Enable L2 on border routers or fix area configuration

---

Module 10: Border Gateway Protocol

Part 1: The Conceptual Lecture (The Why)

The Fundamental Problem BGP Solves

Imagine the Internet as millions of independent networks (Autonomous Systems) that need to exchange traffic. Each network is like a sovereign nation with its own internal rules. BGP (Border Gateway Protocol) is the diplomatic protocol that allows these networks to:

1. Announce what destinations they can reach
2. Learn paths to destinations through other networks
3. Make policy decisions about which paths to use
4. Maintain sovereignty over their routing decisions

```
  [AS 100]                    [AS 200]
Your Network               ISP Network
     |                           |
"I can reach                "I can reach
192.168.0.0/16"             the entire Internet"
     |                           |
     <-------- BGP Session ------>
             Exchange Routes
```

How BGP Works

BGP is fundamentally different from IGPs (like IS-IS or OSPF):

Aspect	IGPs (OSPF/IS-IS)	BGP
Goal	Find shortest path	Implement policy
Scope	Within one organization	Between organizations
Trust	Trust all routers	Don't trust anyone
Metric	Simple (cost/bandwidth)	Complex (multiple attributes)
Speed	Fast convergence	Stability over speed

BGP uses TCP port 179 for reliable communication and operates in two modes:

1. **iBGP** (internal BGP): Between routers in the same AS
2. **eBGP** (external BGP): Between routers in different ASes

BGP Path Selection Process

BGP uses a complex decision process (not just shortest path):

```
1. Highest Weight (Cisco-specific)
2. Highest LOCAL_PREF (local decision)
3. Locally Originated Routes
4. Shortest AS_PATH (fewest networks to cross)
5. Lowest ORIGIN (IGP < EGP < Incomplete)
6. Lowest MED (suggestion from neighbor)
7. eBGP over iBGP
8. Lowest IGP Metric to Next Hop
9. Oldest Route (most stable)
10. Lowest Router ID
11. Lowest Peer Address
```

BGP Attributes

Attributes are like shipping labels on packages - they describe characteristics of routes:

- **AS_PATH**: List of ASes the route has traversed (loop prevention)
- **NEXT_HOP**: Where to forward packets for this route
- **LOCAL_PREF**: How much we prefer this path (internal use)
- **MED**: Suggestion to external peers about entry point
- **ORIGIN**: How the route entered BGP
- **COMMUNITIES**: Tags for grouping/marking routes

Part 2: The Junos CLI Masterclass (The How)

BGP Configuration Hierarchy

```
[edit protocols bgp]
    +-- group <name>              # Peer group configuration
    |     +-- type internal/external
    |     +-- local-address       # Source for BGP session
    |     +-- peer-as            # Remote AS number
    |     +-- neighbor <address> # Specific peer
    |     +-- import             # Inbound policy
    |     +-- export             # Outbound policy
    |     +-- family             # Address families (inet, inet6)
    |
    +-- local-as                 # Our AS number
    +-- router-id                # Our BGP identifier
    +-- hold-time                # Keepalive timer
    +-- preference               # Route preference
```

Complete BGP Configuration

```
## Step 1: Configure Autonomous System and Router ID
set routing-options router-id 192.168.1.1
set routing-options autonomous-system 65001

## Step 2: Configure eBGP peer group (external neighbors)
set protocols bgp group ISP-PEERS type external
set protocols bgp group ISP-PEERS peer-as 65100
set protocols bgp group ISP-PEERS local-address 10.0.0.1
set protocols bgp group ISP-PEERS hold-time 30
set protocols bgp group ISP-PEERS family inet unicast

# Add specific external neighbors
set protocols bgp group ISP-PEERS neighbor 10.0.0.2 description "ISP-1 Primary"
set protocols bgp group ISP-PEERS neighbor 10.0.0.2 peer-as 65100

set protocols bgp group ISP-PEERS neighbor 10.0.1.2 description "ISP-2 Backup"
set protocols bgp group ISP-PEERS neighbor 10.0.1.2 peer-as 65200

# eBGP-specific settings
set protocols bgp group ISP-PEERS ttl 1  # Direct connection expected
set protocols bgp group ISP-PEERS authentication-key "$9$ebgp-secret"

## Step 3: Configure iBGP peer group (internal neighbors)
set protocols bgp group IBGP-MESH type internal
```

```
set protocols bgp group IBGP-MESH local-address 192.168.1.1
set protocols bgp group IBGP-MESH hold-time 90
set protocols bgp group IBGP-MESH family inet unicast

# Add internal neighbors
set protocols bgp group IBGP-MESH neighbor 192.168.1.2 description "Core-Router-2"
set protocols bgp group IBGP-MESH neighbor 192.168.1.3 description "Core-Router-3"

## Step 4: Configure BGP path selection preferences
set protocols bgp path-selection external-router-id
set protocols bgp path-selection always-compare-med
set protocols bgp preference 170

## Step 5: Configure BFD for fast failure detection
set protocols bgp group ISP-PEERS bfd-liveness-detection minimum-interval 300
set protocols bgp group ISP-PEERS bfd-liveness-detection multiplier 3
set protocols bgp group ISP-PEERS bfd-liveness-detection session-mode automatic

## Step 6: Configure graceful restart
set protocols bgp graceful-restart
set protocols bgp graceful-restart restart-time 120
set protocols bgp graceful-restart stale-routes-time 300

## Step 7: Configure BGP export policy (what we advertise)
set policy-options prefix-list OUR-NETWORKS 192.168.0.0/16
set policy-options prefix-list OUR-NETWORKS 172.16.0.0/12

set policy-options policy-statement BGP-EXPORT term advertise-our-networks from prefix-list OUR-NETWORKS
set policy-options policy-statement BGP-EXPORT term advertise-our-networks then origin igp
set policy-options policy-statement BGP-EXPORT term advertise-our-networks then accept
set policy-options policy-statement BGP-EXPORT term reject-rest then reject

set protocols bgp group ISP-PEERS export BGP-EXPORT

## Step 8: Configure BGP import policy (what we accept)
set policy-options prefix-list BOGONS 0.0.0.0/8
set policy-options prefix-list BOGONS 10.0.0.0/8
set policy-options prefix-list BOGONS 127.0.0.0/8
set policy-options prefix-list BOGONS 169.254.0.0/16
set policy-options prefix-list BOGONS 192.168.0.0/16
set policy-options prefix-list BOGONS 224.0.0.0/3

set policy-options policy-statement BGP-IMPORT term reject-bogons from prefix-list BOGONS
set policy-options policy-statement BGP-IMPORT term reject-bogons then reject
set policy-options policy-statement BGP-IMPORT term reject-long-prefixes from route-filter 0.0.0.0/0 prefix-length-range
/25-/32
set policy-options policy-statement BGP-IMPORT term reject-long-prefixes then reject
set policy-options policy-statement BGP-IMPORT term accept-rest then local-preference 100
set policy-options policy-statement BGP-IMPORT term accept-rest then accept

set protocols bgp group ISP-PEERS import BGP-IMPORT

## Step 9: Configure route damping (stability)
set protocols bgp group ISP-PEERS damping
set policy-options damping CUSTOM-DAMPING half-life 15
set policy-options damping CUSTOM-DAMPING reuse 750
set policy-options damping CUSTOM-DAMPING suppress 3000
set policy-options damping CUSTOM-DAMPING max-suppress 60

## Step 10: Configure BGP session options
set protocols bgp group ISP-PEERS tcp-mss 1460
set protocols bgp group ISP-PEERS passive  # Wait for peer to initiate
set protocols bgp group ISP-PEERS advertise-peer-as  # Allow peer's AS in AS_PATH
set protocols bgp group ISP-PEERS as-override  # Replace peer AS with ours
```

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

Essential Verification Commands

```
# Overall BGP status
user@router> show bgp summary
Threading mode: BGP I/O
Groups: 2 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0              8500       4250          0          0          0          0
```

```
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...
10.0.0.2               65100      45231      42156       0       1     2d3h15m Establ
  inet.0: 4000/8000/8000/0
10.0.1.2               65200       5632       5421       0       0      5h42m Establ
  inet.0: 250/500/500/0

# Detailed peer information
user@router> show bgp neighbor 10.0.0.2 extensive
Peer: 10.0.0.2+179 AS 65100 Local: 10.0.0.1+65432 AS 65001
  Description: ISP-1 Primary
  Group: ISP-PEERS               Routing-Instance: master
  Forwarding routing-instance: master
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Damping Refresh>
  Options: <BfdEnabled>
  Authentication key configured
  Holdtime: 30 Preference: 170
  NLRI inet-unicast
  Peer ID: 10.0.0.2      Local ID: 192.168.1.1        Active Holdtime: 30
  Keepalive Interval: 10         Group index: 0    Peer index: 0    SNMP index: 1
  I/O Session Thread: bgpio-0 State: Enabled
  BFD: enabled, session state: Up
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on peer: 120
  Stale routes from peer are kept for: 300
  Table inet.0 Bit: 20000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              4000
    Received prefixes:            8000
    Accepted prefixes:            8000
    Suppressed due to damping:    0
    Advertised prefixes:          50
```

Common Troubleshooting Scenarios

Scenario 1: BGP Session Won't Establish

Symptom: Session stuck in Connect or Active state

```
user@router> show bgp summary
Peer                AS      State
10.0.0.2         65100      Active  # Should be "Establ"
```

Diagnostic Commands:

```
user@router> show bgp neighbor 10.0.0.2 | match "State|Error"
  Type: External    State: Active
  Last State: Connect   Last Event: ConnectRetry
  Last Error: Connection refused

user@router> show route 10.0.0.2
# No route to peer!
```

Cause: No route to peer or TCP port 179 blocked **Solution**:

```
# Ensure route to peer exists
set routing-options static route 10.0.0.2/32 next-hop 10.0.0.254

# Check firewall filters
show configuration firewall family inet filter edge-filter
# Add term to allow BGP
set firewall family inet filter edge-filter term allow-bgp from protocol tcp
set firewall family inet filter edge-filter term allow-bgp from port 179
set firewall family inet filter edge-filter term allow-bgp then accept
```

Scenario 2: Routes Not Being Advertised

Symptom: Peer not receiving our routes

```
user@peer> show route receive-protocol bgp 10.0.0.1 192.168.0.0/16
# No output
```

Diagnostic Commands:

```
user@router> show route advertising-protocol bgp 10.0.0.2 192.168.0.0/16
# No output - not advertising!

user@router> test policy BGP-EXPORT 192.168.0.0/16
Policy BGP-EXPORT: 0 prefix accepted, 1 prefix rejected

user@router> show route 192.168.0.0/16
# Route exists but not in BGP
```

Cause: Route not in BGP or filtered by policy **Solution**:

```
# Add route to BGP
set routing-options static route 192.168.0.0/16 discard
set protocols bgp group ISP-PEERS export BGP-EXPORT
```

Scenario 3: Suboptimal Path Selection

Symptom: Traffic using backup path instead of primary

```
user@router> show route 8.8.8
8.8.8.0/32    *[BGP/170] via 10.0.1.2  # Using backup ISP!
               [BGP/170] via 10.0.0.2  # Primary not selected
```

Diagnostic Commands:

```
user@router> show route 8.8.8 detail
8.8.8.0/32 (2 entries)
  *BGP    Preference: 170
          Next hop: 10.0.1.2
          AS path: 65200 15169 I
          Local Preference: 100
   BGP    Preference: 170
          Next hop: 10.0.0.2
          AS path: 65100 3356 15169 I
          Local Preference: 100
          # Same local-pref, shorter AS-PATH wins!
```

Solution: Adjust local preference for primary path

```
set policy-options policy-statement PREFER-PRIMARY term ISP1 from neighbor 10.0.0.2
set policy-options policy-statement PREFER-PRIMARY term ISP1 then local-preference 150
set protocols bgp group ISP-PEERS import PREFER-PRIMARY
```

Module 11: BGP Attributes and Policy, Part 1

Part 1: The Conceptual Lecture (The Why)

Understanding BGP Policy

BGP policy is like international trade policy. Just as countries control imports/exports with tariffs, quotas, and regulations, BGP policies control route advertisements with filters, preferences, and modifications.

```
Inbound Policy (Import):          Outbound Policy (Export):
"What products can enter?"        "What products can we sell?"
"What tariffs to apply?"          "What price to advertise?"
"From which countries?"           "To which countries?"
```

The BGP Next-Hop Attribute

The NEXT_HOP attribute tells routers where to forward packets. It's like the "next shipping hub" in package delivery:

```
Package from China to USA:
China -> Singapore (NEXT_HOP) -> Hawaii (NEXT_HOP) -> California (NEXT_HOP) -> You

BGP Route:
AS100 -> AS200 (NEXT_HOP: 10.1.1.1) -> AS300 (NEXT_HOP: 10.2.2.2) -> Destination
```

Critical Rule: In iBGP, the NEXT_HOP doesn't change by default! This causes a common problem:

```
[Router A] ----eBGP---- [Router B] ----iBGP---- [Router C]
AS 100                    AS 200                  AS 200
                        NEXT_HOP: 10.1.1.1      NEXT_HOP: still 10.1.1.1!
                                                (C can't reach 10.1.1.1!)
```

Origin Attribute

Origin indicates how a route entered BGP:

- **IGP (i)**: Injected via `network` statement or redistribution - most trustworthy
- **EGP (e)**: Learned via EGP protocol (obsolete) - medium trust
- **Incomplete (?)**: Unclear origin (often redistribution) - least trustworthy

Think of it like product labeling:

- IGP = "Made in our factory" (highest quality assurance)
- EGP = "Made by certified partner" (good quality)
- Incomplete = "Unknown origin" (buyer beware)

MED (Multi-Exit Discriminator)

MED is a suggestion to external peers about the preferred entry point into your AS. It's like saying "Please use the front door, not the back door":

```
Your AS (65001):
        [R1] --------------- [ISP Router]
         |     MED: 100           |
         |                        | Which path
    [Internal                    | to use?
     Network]                     |
         |                        |
         |     MED: 200           |
        [R2] --------------- [ISP Router]

ISP will prefer R1 (lower MED)
```

AS_PATH Attribute

AS_PATH is the list of autonomous systems a route has traversed. It serves two critical functions:

1. **Loop Prevention**: If a router sees its own AS, it rejects the route
2. **Path Selection**: Shorter paths are preferred

```
Route: 8.8.8.8/32
Path 1: AS_PATH: 65100 3356 15169  (3 ASes)
Path 2: AS_PATH: 65200 701 174 15169 (4 ASes)
Path 1 wins (shorter)
```

Part 2: The Junos CLI Masterclass (The How)

Comprehensive BGP Attribute Manipulation

```
## Step 1: Next-Hop Configuration

# Configure next-hop-self for iBGP
set protocols bgp group IBGP-MESH next-hop-self

# Configure next-hop resolution
set routing-options resolution rib inet.0 resolution-ribs inet.3
set routing-options resolution rib inet.0 resolution-ribs inet.0

# Static next-hop mapping for multihop eBGP
set routing-options static route 100.64.0.1/32 next-hop 10.0.0.254
```

```
## Step 2: Origin Attribute Manipulation

# Set origin in export policy
set policy-options policy-statement SET-ORIGIN term internal-routes from protocol static
set policy-options policy-statement SET-ORIGIN term internal-routes then origin igp
set policy-options policy-statement SET-ORIGIN term internal-routes then accept

set policy-options policy-statement SET-ORIGIN term redistributed from protocol ospf
set policy-options policy-statement SET-ORIGIN term redistributed then origin incomplete
set policy-options policy-statement SET-ORIGIN term redistributed then accept

## Step 3: MED Configuration

# Set MED on export (to external peers)
set policy-options policy-statement SET-MED term primary-exit from route-filter 192.168.0.0/16 orlonger
set policy-options policy-statement SET-MED term primary-exit then metric 100
set policy-options policy-statement SET-MED term primary-exit then accept

set policy-options policy-statement SET-MED term backup-exit from route-filter 172.16.0.0/12 orlonger
set policy-options policy-statement SET-MED term backup-exit then metric 200
set policy-options policy-statement SET-MED term backup-exit then accept

# Configure MED comparison
set protocols bgp path-selection always-compare-med
set protocols bgp path-selection cisco-non-deterministic

# Apply to eBGP group
set protocols bgp group ISP-PEERS export SET-MED

## Step 4: AS_PATH Manipulation

# AS Path Prepending (make path look longer)
set policy-options policy-statement AS-PREPEND term prepend from route-filter 192.168.0.0/16 exact
set policy-options policy-statement AS-PREPEND term prepend then as-path-prepend "65001 65001 65001"
set policy-options policy-statement AS-PREPEND term prepend then accept

# AS Path filtering (security)
set policy-options as-path BOGON-AS ".* 64512 .*"  # Private AS
set policy-options as-path BOGON-AS ".* 65535 .*"  # Reserved
set policy-options as-path TRANSIT-FREE "^(174|701|1299|2914|3356)$"

set policy-options policy-statement AS-PATH-FILTER term reject-bogon-as from as-path BOGON-AS
set policy-options policy-statement AS-PATH-FILTER term reject-bogon-as then reject

# AS Path regular expressions
set policy-options as-path CUSTOMER-AS "^65100$"  # Directly from AS 65100
set policy-options as-path ANY-CUSTOMER "^65100 .*"  # AS 65100 and their customers
set policy-options as-path CONTAINS-65100 ".* 65100 .*"  # Path contains AS 65100

## Step 5: Complex Policy with Multiple Attributes

set policy-options policy-statement COMPLEX-BGP-POLICY term prefer-customer from as-path CUSTOMER-AS
set policy-options policy-statement COMPLEX-BGP-POLICY term prefer-customer from protocol bgp
set policy-options policy-statement COMPLEX-BGP-POLICY term prefer-customer then local-preference 150
set policy-options policy-statement COMPLEX-BGP-POLICY term prefer-customer then metric 50
set policy-options policy-statement COMPLEX-BGP-POLICY term prefer-customer then origin igp
set policy-options policy-statement COMPLEX-BGP-POLICY term prefer-customer then next-hop self
set policy-options policy-statement COMPLEX-BGP-POLICY term prefer-customer then accept

## Step 6: Conditional Advertisement Based on AS_PATH

set policy-options policy-statement CONDITIONAL-ADV term advertise-if-primary-down from protocol static
set policy-options policy-statement CONDITIONAL-ADV term advertise-if-primary-down from route-filter 192.168.0.0/16
exact
set policy-options policy-statement CONDITIONAL-ADV term advertise-if-primary-down from condition NO-PRIMARY-PATH
set policy-options policy-statement CONDITIONAL-ADV term advertise-if-primary-down then accept

set policy-options condition NO-PRIMARY-PATH if-route-exists 0.0.0.0/0
set policy-options condition NO-PRIMARY-PATH if-route-exists as-path TRANSIT-FREE

## Step 7: MED Preservation and Modification

# Preserve received MED
set protocols bgp group IBGP-MESH metric-out minimum-igp
```

```
# Add IGP cost to MED
set protocols bgp group ISP-PEERS metric-out igp
```

Complete Policy Configuration Example

```
policy-options {
    prefix-list OUR-NETWORKS {
        192.168.0.0/16;
        172.16.0.0/12;
    }
    as-path CUSTOMER-AS "^65100$";
    as-path PEER-AS "^65200$";
    as-path BOGON-AS ".* (64512|65535) .*";

    policy-statement BGP-IMPORT-POLICY {
        term reject-bogons {
            from as-path BOGON-AS;
            then reject;
        }
        term prefer-customer {
            from {
                protocol bgp;
                as-path CUSTOMER-AS;
            }
            then {
                local-preference 150;
                accept;
            }
        }
        term standard-peer {
            from {
                protocol bgp;
                as-path PEER-AS;
            }
            then {
                local-preference 100;
                accept;
            }
        }
    }

    policy-statement BGP-EXPORT-POLICY {
        term advertise-our-networks {
            from {
                prefix-list OUR-NETWORKS;
            }
            then {
                metric 100;
                origin igp;
                accept;
            }
        }
        term advertise-customer-routes {
            from {
                protocol bgp;
                as-path CUSTOMER-AS;
            }
            then {
                metric 150;
                as-path-prepend "65001";
                accept;
            }
        }
        term reject-rest {
            then reject;
        }
    }
}
```

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

Essential Verification Commands

```
# Show route with all BGP attributes
user@router> show route 8.8.8.8 detail
8.8.8.8/32 (2 entries, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Router, Next hop index: 612
                Address: 0x94d4d04
                Next-hop reference count: 150000
                Source: 10.0.0.2
                Next hop: 10.0.0.2 via ge-0/0/0.0, selected
                Session Id: 0x141
                State: <Active Ext>
                Local AS: 65001 Peer AS: 65100
                Age: 2d 3:14:27     Metric: 100     MED: 150
                Validation-state: unverified
                Task: BGP_65100.10.0.0.2
                AS path: 65100 3356 15169 I
                Origin: IGP
                LocalPref: 100
                Router ID: 10.0.0.2

# Check AS path in received routes
user@router> show route receive-protocol bgp 10.0.0.2 aspath-regex ".*701.*"
# Shows all routes with AS 701 in path
```

Troubleshooting Scenarios

Scenario 1: Next-Hop Unreachable in iBGP

Symptom: Routes hidden due to unreachable next-hop

```
user@router> show route hidden
inet.0: 5 destinations, 5 routes (0 active, 0 holddown, 5 hidden)
+ = Active Route, - = Last Active, * = Both
8.8.8.8/32   [BGP ] Next hop type: Unusable
```

Diagnostic Commands:

```
user@router> show route 10.0.0.2
# No route to next-hop!

user@router> show route receive-protocol bgp 192.168.1.2 detail | match next
                Next hop: 10.0.0.2  # External next-hop preserved!
```

Solution:

```
# Option 1: Configure next-hop-self on iBGP sender
set protocols bgp group IBGP-MESH next-hop-self

# Option 2: Redistribute connected routes into IGP
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0 passive
```

Scenario 2: MED Not Influencing Path Selection

Symptom: Higher MED path being preferred

```
user@router> show route 192.168.1.0 detail
    BGP    MED: 200    # This path selected
    BGP    MED: 100    # This should be preferred!
```

Diagnostic Commands:

```
user@router> show route 192.168.1.0 detail | match "AS path|MED|Local"
Path 1:
    AS path: 65100 I
    MED: 200
    LocalPref: 150  # Higher local preference wins!
Path 2:
    AS path: 65200 I  # Different AS!
    MED: 100
    LocalPref: 100
```

Cause: MED only compared between paths from same AS **Solution**:

```
set protocols bgp path-selection always-compare-med
# Or adjust local-preference to allow MED comparison
```

Module 12: BGP Attributes and Policy, Part 2

Part 1: The Conceptual Lecture (The Why)

Local Preference Attribute

LOCAL_PREF is your internal network's voting system for choosing exit points. Unlike MED (which is a suggestion TO you), LOCAL_PREF is your internal decision that never leaves your AS.

```
Your AS (65001):
         Internet Path A
              ↑ LOCAL_PREF: 200 (Preferred!)
         [Router A] ----iBGP---- [Router C]
              |                       |
          Internal              Internal
           Network               Network
              |                       |
         [Router B] ----iBGP---- [Router D]
              ↓ LOCAL_PREF: 100
         Internet Path B

All routers in AS 65001 will prefer Path A (higher LOCAL_PREF)
```

Think of LOCAL_PREF like employee voting:

- Higher value = more votes for that path
- Default is 100
- Only shared within your AS (iBGP)
- Takes precedence over AS_PATH length

BGP Communities

Communities are like luggage tags - they mark routes with metadata that can trigger actions anywhere in the network:

```
Standard Communities (32-bit):
[AS:Value] = [16 bits:16 bits]
65001:100 = "Customer route from AS 65001, type 100"

Well-known Communities:
NO_EXPORT (0xFFFFFF01) = "Don't advertise outside this AS"
NO_ADVERTISE (0xFFFFFF02) = "Don't advertise to any peer"
NO_EXPORT_SUBCONFED (0xFFFFFF03) = "Don't advertise outside confederation"
```

Communities enable sophisticated routing policies:

```
Route tagged with 65001:666 → Blackhole (discard)
Route tagged with 65001:100 → Set LOCAL_PREF to 100
Route tagged with 65001:200 → Set LOCAL_PREF to 200
Route tagged with 174:990 → Ask Level3 to prepend their AS
```

Routing Policy Architecture

BGP policies chain together like manufacturing assembly lines:

```
Inbound Assembly Line:
Raw Routes → [Filter] → [Modify Attributes] → [Tag] → [Accept/Reject] → RIB

Outbound Assembly Line:
RIB → [Select Routes] → [Modify Attributes] → [Filter] → [Accept/Reject] → Advertise
```

Part 2: The Junos CLI Masterclass (The How)

Advanced Local Preference Configuration

```
## Step 1: Basic Local Preference Manipulation

# Set local preference based on neighbor
set policy-options policy-statement LOCAL-PREF-POLICY term prefer-primary from neighbor 10.0.0.2
set policy-options policy-statement LOCAL-PREF-POLICY term prefer-primary then local-preference 200
set policy-options policy-statement LOCAL-PREF-POLICY term prefer-primary then accept

set policy-options policy-statement LOCAL-PREF-POLICY term standard-backup from neighbor 10.0.1.2
set policy-options policy-statement LOCAL-PREF-POLICY term standard-backup then local-preference 90
set policy-options policy-statement LOCAL-PREF-POLICY term standard-backup then accept

## Step 2: Dynamic Local Preference Based on AS_PATH

# Higher preference for direct customers
set policy-options as-path DIRECT-CUSTOMER "^65100$"
set policy-options as-path CUSTOMER-AND-THEIR-CUSTOMERS "^65100 .+$"

set policy-options policy-statement DYNAMIC-LOCAL-PREF term direct-customer from as-path DIRECT-CUSTOMER
set policy-options policy-statement DYNAMIC-LOCAL-PREF term direct-customer then local-preference 150
set policy-options policy-statement DYNAMIC-LOCAL-PREF term direct-customer then accept

set policy-options policy-statement DYNAMIC-LOCAL-PREF term customer-cone from as-path CUSTOMER-AND-THEIR-CUSTOMERS
set policy-options policy-statement DYNAMIC-LOCAL-PREF term customer-cone then local-preference 140
set policy-options policy-statement DYNAMIC-LOCAL-PREF term customer-cone then accept

## Step 3: Local Preference with Arithmetic Operations

# Add or subtract from local preference
set policy-options policy-statement LP-ARITHMETIC term add-for-backup from neighbor 10.0.2.2
set policy-options policy-statement LP-ARITHMETIC term add-for-backup then local-preference add 20
set policy-options policy-statement LP-ARITHMETIC term add-for-backup then accept

set policy-options policy-statement LP-ARITHMETIC term subtract-for-expensive from neighbor 10.0.3.2
set policy-options policy-statement LP-ARITHMETIC term subtract-for-expensive then local-preference subtract 30
set policy-options policy-statement LP-ARITHMETIC term subtract-for-expensive then accept

## Step 4: BGP Communities Configuration

# Define community values
set policy-options community CUSTOMER-ROUTE members 65001:100
set policy-options community PEER-ROUTE members 65001:200
set policy-options community TRANSIT-ROUTE members 65001:300
set policy-options community BLACKHOLE members 65001:666
set policy-options community NO-EXPORT members no-export
set policy-options community NO-ADVERTISE members no-advertise

# Regular expression communities
set policy-options community ALL-CUSTOMER-COMMUNITIES members "65001:10[0-9]"
set policy-options community GEOGRAPHIC-WEST members "65001:1[0-9][0-9]"
set policy-options community GEOGRAPHIC-EAST members "65001:2[0-9][0-9]"

## Step 5: Apply Communities on Import

set policy-options policy-statement TAG-ON-IMPORT term tag-customer from neighbor 10.1.0.2
set policy-options policy-statement TAG-ON-IMPORT term tag-customer from protocol bgp
set policy-options policy-statement TAG-ON-IMPORT term tag-customer then community add CUSTOMER-ROUTE
set policy-options policy-statement TAG-ON-IMPORT term tag-customer then accept

set policy-options policy-statement TAG-ON-IMPORT term tag-peer from neighbor 10.2.0.2
set policy-options policy-statement TAG-ON-IMPORT term tag-peer from protocol bgp
set policy-options policy-statement TAG-ON-IMPORT term tag-peer then community add PEER-ROUTE
set policy-options policy-statement TAG-ON-IMPORT term tag-peer then accept

## Step 6: Community-Based Local Preference

set policy-options policy-statement COMMUNITY-TO-LP term customer-routes from community CUSTOMER-ROUTE
set policy-options policy-statement COMMUNITY-TO-LP term customer-routes then local-preference 150
set policy-options policy-statement COMMUNITY-TO-LP term customer-routes then accept

set policy-options policy-statement COMMUNITY-TO-LP term peer-routes from community PEER-ROUTE
set policy-options policy-statement COMMUNITY-TO-LP term peer-routes then local-preference 100
set policy-options policy-statement COMMUNITY-TO-LP term peer-routes then accept

set policy-options policy-statement COMMUNITY-TO-LP term transit-routes from community TRANSIT-ROUTE
set policy-options policy-statement COMMUNITY-TO-LP term transit-routes then local-preference 80
set policy-options policy-statement COMMUNITY-TO-LP term transit-routes then accept
```

```
## Step 7: Blackhole Community Implementation

set policy-options policy-statement BLACKHOLE-POLICY term blackhole from community BLACKHOLE
set policy-options policy-statement BLACKHOLE-POLICY term blackhole then local-preference 999
set policy-options policy-statement BLACKHOLE-POLICY term blackhole then next-hop discard
set policy-options policy-statement BLACKHOLE-POLICY term blackhole then accept

# Apply to BGP
set protocols bgp group ISP-PEERS import BLACKHOLE-POLICY

## Step 8: Community Manipulation on Export

set policy-options policy-statement COMMUNITY-EXPORT term remove-internal from protocol bgp
set policy-options policy-statement COMMUNITY-EXPORT term remove-internal then community delete "65001:.*"
set policy-options policy-statement COMMUNITY-EXPORT term remove-internal then accept

set policy-options policy-statement COMMUNITY-EXPORT term mark-geographic from route-filter 192.168.0.0/24 exact
set policy-options policy-statement COMMUNITY-EXPORT term mark-geographic then community add GEOGRAPHIC-WEST
set policy-options policy-statement COMMUNITY-EXPORT term mark-geographic then accept

## Step 9: Complex Multi-Attribute Policy

set policy-options policy-statement COMPLEX-POLICY term premium-customer from community CUSTOMER-ROUTE
set policy-options policy-statement COMPLEX-POLICY term premium-customer from as-path DIRECT-CUSTOMER
set policy-options policy-statement COMPLEX-POLICY term premium-customer from route-filter 10.0.0.0/8 orlonger
set policy-options policy-statement COMPLEX-POLICY term premium-customer then local-preference 200
set policy-options policy-statement COMPLEX-POLICY term premium-customer then metric 50
set policy-options policy-statement COMPLEX-POLICY term premium-customer then community add NO-EXPORT
set policy-options policy-statement COMPLEX-POLICY term premium-customer then community add CUSTOMER-ROUTE
set policy-options policy-statement COMPLEX-POLICY term premium-customer then next-hop self
set policy-options policy-statement COMPLEX-POLICY term premium-customer then accept

## Step 10: Extended Communities (for MPLS VPNs)

set policy-options community RT-CUSTOMER-A members target:65001:100
set policy-options community RT-CUSTOMER-B members target:65001:200
set policy-options community COLOR-GOLD members color:0:100
set policy-options community COLOR-SILVER members color:0:200

set policy-options policy-statement VPN-POLICY term customer-a-routes from community RT-CUSTOMER-A
set policy-options policy-statement VPN-POLICY term customer-a-routes then local-preference 150
set policy-options policy-statement VPN-POLICY term customer-a-routes then accept
```

Complete Advanced Policy Example

```
protocols {
    bgp {
        group CUSTOMERS {
            type external;
            import [ TAG-ON-IMPORT COMMUNITY-TO-LP BLACKHOLE-POLICY ];
            export [ COMMUNITY-EXPORT ADVERTISE-SELECTED ];
            neighbor 10.1.0.2 {
                description "Customer A";
                peer-as 65100;
            }
        }
        group PEERS {
            type external;
            import [ TAG-ON-IMPORT COMMUNITY-TO-LP FILTER-BOGONS ];
            export [ COMMUNITY-EXPORT ADVERTISE-OUR-ROUTES ];
            neighbor 10.2.0.2 {
                description "Peer Exchange";
                peer-as 65200;
            }
        }
        group IBGP-MESH {
            type internal;
            local-address 192.168.1.1;
            export NEXT-HOP-SELF;
            neighbor 192.168.1.2;
            neighbor 192.168.1.3;
        }
    }
}
```

```
policy-options {
    policy-statement MASTER-IMPORT-POLICY {
        term customers {
            from {
                neighbor [ 10.1.0.0/24 ];
            }
            then {
                community add CUSTOMER-ROUTE;
                local-preference 150;
                next policy;
            }
        }
        term peers {
            from {
                neighbor [ 10.2.0.0/24 ];
            }
            then {
                community add PEER-ROUTE;
                local-preference 100;
                next policy;
            }
        }
        term transit {
            from {
                neighbor [ 10.3.0.0/24 ];
            }
            then {
                community add TRANSIT-ROUTE;
                local-preference 80;
                next policy;
            }
        }
        term filter-bogons {
            from {
                route-filter 0.0.0.0/0 prefix-length-range /25-/32 reject;
                route-filter 10.0.0.0/8 orlonger reject;
                route-filter 192.168.0.0/16 orlonger reject;
            }
        }
        term accept-rest {
            then accept;
        }
    }
}
```

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

Essential Verification Commands

```
# Show routes with communities
user@router> show route community 65001:100 detail
192.168.1.0/24 (1 entry)
        *BGP    Preference: 170/-151
                Next hop: 10.1.0.2
                Communities: 65001:100 65001:150
                Local Preference: 150

# Show routes with specific local preference
user@router> show route detail | match "Local preference: 200"
                Local preference: 200

# Test policy with specific attributes
user@router> test policy COMPLEX-POLICY 192.168.1.0/24 neighbor 10.1.0.2 community [ 65001:100 65001:200 ]
Policy COMPLEX-POLICY: 1 prefix accepted

# Show all routes with their communities
user@router> show route extensive | match "Communities:|^[0-9]"
192.168.1.0/24
    Communities: 65001:100 no-export
10.0.0.0/24
    Communities: 65001:200 65001:150
```

Troubleshooting Scenarios

Scenario 1: Local Preference Not Working

Symptom: Routes not preferring higher local preference path

```
user@router> show route 192.168.1.0
192.168.1.0/24   *[BGP/170] via 10.0.1.2  # LP: 100
                  [BGP/170] via 10.0.0.2  # LP: 200 - should be preferred!
```

Diagnostic Commands:

```
user@router> show route 192.168.1.0 extensive
Path 1: (via 10.0.1.2)
    Protocol next hop: 10.0.1.2
    Local preference: 100
    Router ID: 10.0.1.2
Path 2: (via 10.0.0.2)
    Protocol next hop: 10.0.0.2 (not resolved)  # Problem!
    Local preference: 200
```

Cause: Next-hop not resolvable **Solution**:

```
# Add route to next-hop
set routing-options static route 10.0.0.2/32 next-hop 10.0.0.254
# Or configure next-hop-self
```

Scenario 2: Communities Not Being Preserved

Symptom: Communities missing on iBGP peers

```
user@ibgp-peer> show route 192.168.1.0 detail | match Communities
# No output - communities stripped!
```

Diagnostic Commands:

```
user@router> show route advertising-protocol bgp 192.168.1.2 192.168.1.0 detail
        Communities:  # Empty!

user@router> show configuration protocols bgp group IBGP-MESH | display set | match export
set protocols bgp group IBGP-MESH export STRIP-COMMUNITIES
```

Cause: Export policy removing communities **Solution**:

```
# Preserve communities in iBGP
set policy-options policy-statement IBGP-EXPORT term preserve-all from protocol bgp
set policy-options policy-statement IBGP-EXPORT term preserve-all then next-hop self
set policy-options policy-statement IBGP-EXPORT term preserve-all then accept
# Don't modify communities in iBGP export!
```

Scenario 3: Blackhole Community Not Working

Symptom: Routes tagged with blackhole community still forwarding traffic

```
user@router> show route 192.168.100.100 community 65001:666
192.168.100.100/32 *[BGP/170] via 10.0.0.2  # Should be discarded!
```

Diagnostic Commands:

```
user@router> show route 192.168.100.100 extensive | match "Next hop|Communities"
    Next hop: 10.0.0.2  # Not discard!
    Communities: 65001:666

user@router> show configuration protocols bgp group ISP-PEERS | match import
import BASIC-IMPORT;  # Not applying BLACKHOLE-POLICY!
```

Solution:

```
set protocols bgp group ISP-PEERS import [ BLACKHOLE-POLICY BASIC-IMPORT ]
# Order matters - blackhole policy must come first!
```

This completes the comprehensive coverage of modules 8-12. We've covered multilevel IS-IS networks, IS-IS troubleshooting, BGP fundamentals, and advanced BGP attributes and policies. Each module built upon the previous ones, progressing from IS-IS (interior routing) to BGP (exterior routing), with detailed configurations, troubleshooting scenarios, and real-world applications.

Module 13: Route Reflection and Confederations - Complete Learning Guide

Part 1: The Conceptual Lecture (The Why)

The Fundamental Problem

Imagine you're building a communication network within a large company where every department needs to share information with every other department. In a small company with 3 departments, you'd need 3 connections. But what happens when you have 100 departments? You'd need $\frac{n(n-1)}{2}$ connections, which equals 4,950 connections! This is the exact problem BGP faces within an autonomous system.

Understanding IBGP Full Mesh Requirements

BGP (Border Gateway Protocol) is the protocol that routers use to exchange routing information. When BGP runs between routers in the same autonomous system (AS), we call it IBGP (Internal BGP). IBGP has a fundamental rule: **routes learned from one IBGP peer are NOT advertised to other IBGP peers**. This is called the IBGP split-horizon rule.

Why does this rule exist? To prevent routing loops. Since IBGP doesn't modify the AS_PATH attribute (which is BGP's loop prevention mechanism), it needs another way to prevent loops - hence the split-horizon rule.

```
Traditional IBGP Full Mesh Topology:

        R1 -------- R2
        |  \      /  |
        |    \  /    |
        |      X     |
        |    /  \    |
        |  /      \  |
        R3 -------- R4

Each router must peer with every other router
Number of sessions = n(n-1)/2
For 4 routers = 6 sessions
For 10 routers = 45 sessions
For 100 routers = 4,950 sessions!
```

Solution 1: Route Reflection

Route Reflection introduces a hierarchical model that breaks the split-horizon rule in a controlled manner. Think of it like creating a hub-and-spoke communication system in our company analogy - instead of everyone talking to everyone, we designate team leaders (Route Reflectors) who collect information from their team members and share it with other teams.

Route Reflector Concepts

1. **Route Reflector (RR)**: A BGP router that reflects routes between IBGP peers
2. **Client**: An IBGP peer of the Route Reflector that receives reflected routes
3. **Non-Client**: Regular IBGP peers that still need full mesh with other non-clients
4. **Cluster**: The Route Reflector and its clients form a cluster

```
Route Reflection Topology:

     [Non-Clients - Full Mesh Required]
           R5 -------- R6
            \        /
             \      /
          +---RR1---+     <-- Route Reflector
         /     |     \
        /      |      \
    Client1  Client2  Client3
      R1       R2       R3

Clients only peer with RR
RR reflects routes between clients
Non-clients still need full mesh
```

Route Reflection Rules

The Route Reflector follows these rules when receiving a route:

1. **From EBGP peer**: Advertise to all IBGP peers (clients and non-clients)
2. **From Client**: Reflect to all other clients AND all non-clients
3. **From Non-Client**: Reflect only to all clients (not to other non-clients)

Loop Prevention in Route Reflection

Route Reflection adds two BGP attributes to prevent loops:

1. **ORIGINATOR_ID**: The Router ID of the route's originator
2. **CLUSTER_LIST**: List of Cluster IDs the route has traversed

```
Route Reflection Loop Prevention:

Client1 --> RR1 --> Client2
  |          |         |
Origin    Adds       Checks
         ORIGINATOR_ID

RR1 --> RR2 --> RR3
 |       |       |
Adds   Appends  Checks
CLUSTER CLUSTER CLUSTER_LIST
  ID     ID    for loops
```

Solution 2: Confederations

Confederations take a different approach - they divide the AS into smaller sub-autonomous systems. Think of it like dividing a large country into states, where each state manages its own internal affairs but coordinates at the federal level.

Confederation Concepts

1. **Confederation AS**: The main AS number seen by external peers
2. **Sub-AS (Member AS)**: Internal AS numbers used within the confederation
3. **Confederation EBGP**: BGP sessions between sub-ASes (behaves like EBGP but with special rules)
4. **Confederation Sequence**: Special AS_PATH segment tracking sub-AS path

```
Confederation Topology:

External AS 100                 External AS 200
      |                               |
      |     [Confederation AS 65000] |
      |     +--------------------+     |
      |     |                    |     |
   [Sub-AS 65001]          [Sub-AS 65002]
   R1 --- R2                 R5 --- R6
    |  X  |                   |  X  |
   R3 --- R4                 R7 --- R8
      |                           |
      +----Confed-EBGP------------+

Within each Sub-AS: IBGP full mesh required
Between Sub-ASes: Confederation EBGP
External peers see only AS 65000
```

Confederation Operation

1. **Within Sub-AS**: Standard IBGP rules apply (full mesh required)
2. **Between Sub-ASes**: Uses Confederation EBGP which:
   - Preserves NEXT_HOP (unlike regular EBGP)
   - Preserves LOCAL_PREF
   - Preserves MED
   - Adds sub-AS to AS_CONFED_SEQUENCE
3. **To External Peers**: Removes AS_CONFED_SEQUENCE, shows only Confederation AS

Comparison: Route Reflection vs Confederations

Aspect	Route Reflection	Confederations
Complexity	Lower - Minimal config changes	Higher - Requires AS redesign

Aspect	Route Reflection	Confederations
Scalability	Excellent with hierarchy	Good but limited by sub-AS size
Migration	Easy - Can be incremental	Difficult - Requires planning
Loop Prevention	ORIGINATOR_ID, CLUSTER_LIST	AS_CONFED_SEQUENCE
Path Selection	Can cause suboptimal paths	More predictable paths
Troubleshooting	Moderate complexity	Higher complexity
Common Usage	Very common (90%+ of networks)	Less common (legacy/special cases)

Part 2: The Junos CLI Masterclass (The How)

Route Reflection Configuration

Understanding the Configuration Hierarchy

Route Reflection configuration in Junos is remarkably simple - you only configure it on the Route Reflector, not on the clients. The clients see the RR as a regular IBGP peer.

```
Configuration Hierarchy:

[edit protocols bgp group <group-name>]
  - Define the BGP group type and properties

[edit protocols bgp group <group-name> neighbor <address>]
  - Configure individual neighbors

[edit protocols bgp group <group-name> cluster <cluster-id>]
  - Define the cluster ID for route reflection
```

Step-by-Step Route Reflection Configuration

Let's build a complete Route Reflector configuration:

```
## Step 1: Configure the Route Reflector (RR1)
## RR1 has Router-ID 1.1.1.1 and is in AS 65000

[edit protocols bgp]
set group INTERNAL-CLIENTS type internal
set group INTERNAL-CLIENTS local-address 1.1.1.1
set group INTERNAL-CLIENTS family inet unicast
set group INTERNAL-CLIENTS cluster 1.1.1.1    # Cluster ID (typically Router-ID)

## Step 2: Add client neighbors
set group INTERNAL-CLIENTS neighbor 10.0.0.1  # Client1
set group INTERNAL-CLIENTS neighbor 10.0.0.2  # Client2
set group INTERNAL-CLIENTS neighbor 10.0.0.3  # Client3

## Step 3: Configure non-client peers (if any)
set group INTERNAL-NONCLIENTS type internal
set group INTERNAL-NONCLIENTS local-address 1.1.1.1
set group INTERNAL-NONCLIENTS neighbor 10.0.0.5  # Non-client peer

## Step 4: Configure the clients (Client1 example)
## Note: Clients have NO special configuration!
[edit protocols bgp]
set group TO-RR type internal
set group TO-RR local-address 10.0.0.1
set group TO-RR neighbor 1.1.1.1              # Just peer with RR
```

Advanced Route Reflection Patterns

Pattern 1: Redundant Route Reflectors

```
## Configure RR1 (Primary Route Reflector)
[edit protocols bgp group RR-CLIENTS]
set type internal
set local-address 1.1.1.1
set cluster 100.100.100.100    # Same cluster ID for both RRs
set neighbor 10.0.0.1
set neighbor 10.0.0.2
set neighbor 10.0.0.3
```

```
## Configure RR2 (Backup Route Reflector)
[edit protocols bgp group RR-CLIENTS]
set type internal
set local-address 2.2.2.2
set cluster 100.100.100.100    # Same cluster ID indicates same cluster
set neighbor 10.0.0.1
set neighbor 10.0.0.2
set neighbor 10.0.0.3

## Clients peer with both RRs
[edit protocols bgp group TO-RRS]
set type internal
set local-address 10.0.0.1
set neighbor 1.1.1.1           # Primary RR
set neighbor 2.2.2.2           # Backup RR
```

Pattern 2: Hierarchical Route Reflection

```
## Top-level RR configuration
[edit protocols bgp]
set group REGIONAL-RRS type internal
set group REGIONAL-RRS cluster 1.1.1.1
set group REGIONAL-RRS neighbor 2.2.2.2  # Regional RR1
set group REGIONAL-RRS neighbor 3.3.3.3  # Regional RR2

## Regional RR configuration
[edit protocols bgp]
set group LOCAL-CLIENTS type internal
set group LOCAL-CLIENTS cluster 2.2.2.2
set group LOCAL-CLIENTS neighbor 10.1.1.1
set group LOCAL-CLIENTS neighbor 10.1.1.2

set group TO-CORE-RR type internal
set group TO-CORE-RR neighbor 1.1.1.1    # Uplink to top-level RR
```

Confederation Configuration

Step-by-Step Confederation Configuration

```
## Step 1: Configure routing-options for confederation
[edit routing-options]
set autonomous-system 65000               # Public/Confederation AS
set confederation 65000 members [ 65001 65002 65003 ]  # List all sub-ASes

## Step 2: Configure BGP for routers in Sub-AS 65001
[edit protocols bgp]
set local-as 65001                        # Sub-AS number

## Step 3: Configure IBGP within the sub-AS
set group INTERNAL type internal
set group INTERNAL local-address 10.1.1.1
set group INTERNAL neighbor 10.1.1.2
set group INTERNAL neighbor 10.1.1.3

## Step 4: Configure Confederation EBGP to other sub-ASes
set group CONFED-PEER type external
set group CONFED-PEER local-address 10.1.1.1
set group CONFED-PEER peer-as 65002     # Peer sub-AS
set group CONFED-PEER neighbor 10.2.1.1

## Step 5: Configure regular EBGP to external ASes
set group EXTERNAL type external
set group EXTERNAL local-address 192.168.1.1
set group EXTERNAL peer-as 100          # Real external AS
set group EXTERNAL neighbor 192.168.1.2
```

Complete Reference Configuration

Route Reflector Complete Configuration:

```
## Route Reflector (1.1.1.1)
system {
    host-name RR1;
```

```
}

interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 1.1.1.1/32;
            }
        }
    }
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.0.0.254/24;
            }
        }
    }
}

routing-options {
    router-id 1.1.1.1;
    autonomous-system 65000;
}

protocols {
    bgp {
        group RR-CLIENTS {
            type internal;
            local-address 1.1.1.1;
            family inet {
                unicast;
            }
            cluster 1.1.1.1;        ## This makes it a Route Reflector
            neighbor 10.0.0.1;      ## Client 1
            neighbor 10.0.0.2;      ## Client 2
            neighbor 10.0.0.3;      ## Client 3
        }
        group INTERNAL-MESH {
            type internal;
            local-address 1.1.1.1;
            neighbor 2.2.2.2;       ## Non-client peer (another RR)
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0 passive;
            interface ge-0/0/0.0;
        }
    }
}
```

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

Essential Verification Commands

For Route Reflection:

```
## 1. Verify BGP neighbor status
user@RR1> show bgp summary
Threading mode: BGP I/O
Groups: 2 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                10          5          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State
10.0.0.1              65000        142        145       0       0       1:03:27 4/4/1/0
10.0.0.2              65000        141        145       0       0       1:03:25 3/3/1/0
10.0.0.3              65000        140        145       0       0       1:03:23 3/3/1/0
2.2.2.2               65000        150        148       0       0       1:05:00 0/0/0/0

## 2. Verify route reflection is working
user@RR1> show route advertising-protocol bgp 10.0.0.1 detail
inet.0: 15 destinations, 20 routes (15 active, 0 holddown, 0 hidden)
* 192.168.1.0/24 (1 entry, 1 announced)
 BGP group RR-CLIENTS type Internal
     Nexthop: 10.0.0.2
```

```
     Flags: Nexthop Change
     Localpref: 100
     AS path: [65000] I
     Originator ID: 10.0.0.2      ## Shows route was originated by client2
     Cluster list:  1.1.1.1        ## Shows this RR added its cluster ID

## 3. Check received routes with attributes
user@Client1> show route receive-protocol bgp 1.1.1.1 detail
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
* 192.168.2.0/24 (1 entry, 1 announced)
     Accepted
     Nexthop: 10.0.0.3
     Localpref: 100
     AS path: I
     Originator ID: 10.0.0.3      ## Different originator
     Cluster list:  1.1.1.1        ## RR's cluster ID

## 4. Verify cluster list for loop detection
user@RR1> show route 192.168.5.0/24 detail
inet.0: 15 destinations, 18 routes (15 active, 0 holddown, 0 hidden)
192.168.5.0/24 (3 entries, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0xb3d4d04
                Next-hop reference count: 5
                Source: 2.2.2.2
                Protocol next hop: 10.0.0.5
                Indirect next hop: 0xb4e1080 1048574 INH Session ID: 0x149
                Local AS: 65000 Peer AS: 65000
                Age: 27:13      Metric2: 0
                Validation State: unverified
                Task: BGP_65000.2.2.2.2
                AS path: I
                Originator ID: 10.0.0.5
                Cluster list:  2.2.2.2 1.1.1.1  ## Multiple clusters - hierarchical RR
                Accepted
                Localpref: 100
                Router ID: 2.2.2.2
```

For Confederations:

```
## 1. Verify confederation configuration
user@R1> show bgp neighbor 10.2.1.1
Peer: 10.2.1.1+179 AS 65002 Local: 10.1.1.1+52438 AS 65001
  Description: Confed-peer to Sub-AS 65002
  Group: CONFED-PEER            Routing-Instance: master
  Forwarding routing-instance: master
  Type: External    State: Established
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress ConfedExternal>  ## Note: ConfedExternal

## 2. Check AS path with confederation sequence
user@R1> show route 192.168.100.0/24 detail
192.168.100.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                AS path: (65002 65003) I  ## Confederation sequence in parentheses

## 3. View route as seen by external peer
user@ExtRouter> show route 192.168.100.0/24 detail
192.168.100.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                AS path: 65000 I          ## Only confederation AS visible
```

Common Troubleshooting Scenarios

Scenario 1: Routes Not Being Reflected

Symptom: Client1 doesn't see routes from Client2 through the RR

Diagnostic Commands:

```
user@Client1> show route 192.168.2.0/24
## No route found
```

```
user@RR1> show route 192.168.2.0/24
inet.0: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
192.168.2.0/24     *[BGP/170] 00:05:23, localpref 100
                      AS path: I, validation-state: unverified
                    >  to 10.0.0.2 via ge-0/0/0.0

user@RR1> show configuration protocols bgp group RR-CLIENTS
type internal;
local-address 1.1.1.1;
## Missing: cluster <cluster-id>
```

Cause: Cluster ID not configured - router is not acting as Route Reflector

Solution:

```
[edit protocols bgp group RR-CLIENTS]
user@RR1# set cluster 1.1.1.1
user@RR1# commit
```

Scenario 2: Route Reflection Loop Detected

Symptom: Routes are hidden due to cluster list loop

Diagnostic Commands:

```
user@RR1> show route hidden
inet.0: 20 destinations, 25 routes (18 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both
192.168.10.0/24    [BGP ] 00:02:15, localpref 100
                      AS path: I, validation-state: unverified
                      Unusable

user@RR1> show route 192.168.10.0/24 hidden detail
192.168.10.0/24 (1 entry, 0 announced)
        BGP     Preference: 170/-101
                Next hop type: Indirect
                Cluster ID: 1.1.1.1
                Source: 2.2.2.2
                Inactive reason: Cluster list loop   ## Loop detected!
                AS path: I
                Cluster list:  1.1.1.1 2.2.2.2 1.1.1.1  ## Our cluster ID appears twice
```

Cause: Route has looped back through multiple RRs with same cluster ID

Solution:

```
## Option 1: Use different cluster IDs for RRs that peer with each other
[edit protocols bgp group RR-CLIENTS]
user@RR2# set cluster 2.2.2.2  ## Different from RR1

## Option 2: Review RR hierarchy and peering design
## Ensure RRs in different levels use different cluster IDs
```

Scenario 3: Confederation AS Path Issues

Symptom: External peers see confederation sub-AS numbers

Diagnostic Commands:

```
user@ExtPeer> show route 192.168.1.0/24
192.168.1.0/24     *[BGP/170] 00:10:43, localpref 100
                      AS path: 65000 (65001 65002) I  ## Sub-AS visible!

user@ConfedRouter> show configuration routing-options
autonomous-system 65001;  ## Sub-AS configured as main AS
## Missing: confederation configuration
```

Cause: Confederation not properly configured in routing-options

Solution:

```
[edit routing-options]
user@ConfedRouter# delete autonomous-system 65001
user@ConfedRouter# set autonomous-system 65000  ## Confederation AS
user@ConfedRouter# set confederation 65000 members [ 65001 65002 65003 ]

[edit protocols bgp]
user@ConfedRouter# set local-as 65001  ## Sub-AS configured here instead
user@ConfedRouter# commit
```

Scenario 4: Suboptimal Path Selection with Route Reflection

Symptom: Traffic takes longer path due to Route Reflection

Diagnostic Commands:

```
user@Client1> show route 192.168.100.0/24
inet.0: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
192.168.100.0/24   *[BGP/170] 00:30:15, localpref 100, from 1.1.1.1
                      AS path: 100 I, validation-state: unverified
                    >  to 10.0.0.2 via ge-0/0/0.0  ## Next-hop is Client2

user@Client1> traceroute 192.168.100.1
traceroute to 192.168.100.1
 1  10.0.0.2    2.134 ms  ## Goes to Client2
 2  10.0.0.254  3.421 ms  ## Then to RR
 3  10.0.0.100  5.234 ms  ## Then to actual exit point
 4  192.168.100.1 7.532 ms

## But Client1 has direct connection to exit router!
user@Client1> show ospf neighbor
Address          Interface        State     ID               Pri  Dead
10.0.0.100       ge-0/0/1.0       Full      10.0.0.100       128    39  ## Direct path available
```

Cause: RR selected best path from its perspective, not client's perspective

Solution:

```
## Option 1: Add direct IBGP session for critical paths
[edit protocols bgp]
user@Client1# set group DIRECT type internal
user@Client1# set group DIRECT neighbor 10.0.0.100

## Option 2: Use BGP Add-Path to advertise multiple paths
[edit protocols bgp group RR-CLIENTS]
user@RR1# set family inet unicast add-path send path-count 2

## Option 3: Implement BGP-ORR (Optimal Route Reflection)
[edit protocols bgp group RR-CLIENTS]
user@RR1# set optimal-route-reflection IGP-METRIC
```

Additional Verification Commands Reference

```
## Monitor BGP update messages
user@router> monitor traffic interface ge-0/0/0 matching "port 179"

## Check BGP RIB-IN before policy
user@router> show route receive-protocol bgp 1.1.1.1 table inet.0

## Check BGP RIB-OUT after policy
user@router> show route advertising-protocol bgp 10.0.0.1 table inet.0

## View BGP path selection details
user@router> show route 192.168.1.0/24 detail active-path

## Display route reflection statistics
user@router> show bgp group RR-CLIENTS detail

## Check for duplicate Router IDs (causes issues)
user@router> show bgp neighbor | match "router-id|peer:"

## Verify confederation members
user@router> show bgp summary | match confederation
```

```
## Display all hidden routes with reasons
user@router> show route hidden extensive | match "reason|prefix"
```

This comprehensive module has equipped you with deep understanding of BGP Route Reflection and Confederations. You now understand why IBGP full mesh doesn't scale, how Route Reflection elegantly solves this with cluster hierarchy, and how Confederations provide an alternative through sub-AS division. You've learned the complete Junos configuration patterns for both technologies and gained practical troubleshooting skills through real-world scenarios. These BGP scaling techniques are fundamental to operating large service provider networks and are essential knowledge for the JNCIE-SP exam.

Module 14: BGP Route Damping - Complete Learning Guide

Part 1: The Conceptual Lecture (The Why)

The Fundamental Problem: Route Flapping

Imagine you're managing a postal delivery system where certain roads keep opening and closing unpredictably - maybe due to construction, accidents, or weather. Every time a road closes, all delivery trucks need to be notified to recalculate their routes. When it reopens minutes later, everyone recalculates again. If this happens repeatedly, your entire delivery fleet spends more time recalculating routes than actually delivering packages.

This is exactly what happens in BGP with "route flapping" - when a network path repeatedly becomes available and unavailable in quick succession. Each change triggers BGP updates that propagate across the entire Internet, consuming CPU cycles, memory, and bandwidth on thousands of routers.

Understanding Route Flapping Impact

```
Route Flapping Cascade Effect:

Provider Network
    |
   ISP1 --- Flapping Link (up/down/up/down every 30 seconds)
    |
   AS1 ---- AS2 ---- AS3 ---- AS4
    |        |        |        |
  1000     2000     3000     4000  <-- Number of routers affected

Each flap causes:
- BGP UPDATE messages to all peers
- Route recalculation in each AS
- FIB (Forwarding Information Base) updates
- Potential traffic blackholes during convergence
```

Route Damping: The Solution

BGP Route Damping is like implementing a "three strikes" policy for unreliable routes. Instead of immediately trusting a route that just came back up after failing, we assign it a "penalty" score. If the penalty exceeds certain thresholds, we suppress the route temporarily, giving it time to stabilize.

The Damping Algorithm

Route damping uses an exponential decay algorithm with these key concepts:

1. **Figure of Merit (Penalty Value)**: A numerical score assigned to each route
2. **Suppress Threshold**: When penalty exceeds this, route is suppressed
3. **Reuse Threshold**: When penalty decays below this, route is reused
4. **Half-Life**: Time for penalty to decay by 50%
5. **Max Suppress Time**: Maximum time a route can be suppressed

```
Damping Penalty Behavior:

Penalty
  ^
  |     Route Flap Events
  |     ↓    ↓    ↓
  |    /\   /\   /\
  |   /  \ /  \ /  \_____ Suppress Threshold (3000)
  |  /    X    X     \
  | /    / \  / \     \__ Exponential Decay
  |/    /   \/   \      \
  +----/------------\----\-- Reuse Threshold (750)
```

```
|   /             \    \
|  /               \    \_ Penalty decays over time
| /                 \
|/                   \_____ Route becomes active again
+------------------------> Time

Penalty increases: +1000 per flap
Penalty decays: 50% every half-life period
```

Mathematical Foundation of Damping

The penalty decay follows the formula:

$$P(t) = P_0 \times 2^{-\frac{t}{t_{half}}}$$

Where:

- $P(t)$ = Penalty at time t
- P_0 = Initial penalty value
- t = Time elapsed
- t_{half} = Half-life period

For example, with initial penalty of 2000 and half-life of 15 minutes:

- After 15 minutes: $2000 \times 2^{-1} = 1000$
- After 30 minutes: $2000 \times 2^{-2} = 500$
- After 45 minutes: $2000 \times 2^{-3} = 250$

Categories of Routes and Damping Behavior

Not all routes should be damped equally. Consider these categories:

1. **Customer Routes**: Usually single-homed, flapping indicates real problems
2. **Peer Routes**: Multi-homed, flapping might be intentional traffic engineering
3. **Critical Infrastructure**: Root DNS, crucial services - never damp
4. **Default Routes**: Damping could isolate your network

```
Damping Policy Hierarchy:

         Should We Damp?
               |
    +----------+----------+
    |                     |
Critical?             Normal?
    |                     |
   NO                Check Prefix
 (Never)                 |
              +----------+----------+
              |                     |
         Customer?             Provider?
              |                     |
        YES (Aggressive)      YES (Conservative)
        - Low thresholds      - High thresholds
        - Long suppress       - Short suppress
```

Part 2: The Junos CLI Masterclass (The How)

Understanding Damping Configuration Hierarchy

```
Configuration Hierarchy:

[edit protocols bgp]
  damping                           - Enable basic damping

[edit policy-options]
  damping <name>                    - Define named damping profiles
    half-life <minutes>
    suppress <value>
    reuse <value>
    max-suppress <minutes>
```

```
[edit policy-options policy-statement <name>]
  then damping <damping-profile>  - Apply damping profiles via policy
```

Step-by-Step Basic Damping Configuration

```
## Step 1: Enable basic BGP damping with default parameters
[edit protocols bgp]
user@router# set damping

## Default values applied:
## - Half-life: 15 minutes
## - Suppress: 3000
## - Reuse: 750
## - Max-suppress: 60 minutes

## Step 2: Verify damping is enabled
user@router# show protocols bgp damping
damping;
```

Configuring Custom Damping Profiles

```
## Step 1: Create an aggressive damping profile for unstable customers
[edit policy-options]
user@router# set damping aggressive-damping half-life 10
user@router# set damping aggressive-damping suppress 2000
user@router# set damping aggressive-damping reuse 500
user@router# set damping aggressive-damping max-suppress 45

## Step 2: Create a conservative profile for important peers
[edit policy-options]
user@router# set damping conservative-damping half-life 20
user@router# set damping conservative-damping suppress 5000
user@router# set damping conservative-damping reuse 2000
user@router# set damping conservative-damping max-suppress 30

## Step 3: Create a no-damping profile for critical routes
[edit policy-options]
user@router# set damping no-damp disable

## Show the complete damping profiles
user@router# show policy-options damping
damping aggressive-damping {
    half-life 10;
    reuse 500;
    suppress 2000;
    max-suppress 45;
}
damping conservative-damping {
    half-life 20;
    reuse 2000;
    suppress 5000;
    max-suppress 30;
}
damping no-damp {
    disable;
}
```

Applying Damping Profiles Through Policy

```
## Step 1: Create routing policy to apply different damping profiles
[edit policy-options]
user@router# set policy-statement DAMPING-POLICY term CRITICAL-ROUTES from route-filter 8.8.8.8/32 exact
user@router# set policy-statement DAMPING-POLICY term CRITICAL-ROUTES from route-filter 8.8.4.4/32 exact
user@router# set policy-statement DAMPING-POLICY term CRITICAL-ROUTES then damping no-damp
user@router# set policy-statement DAMPING-POLICY term CRITICAL-ROUTES then accept

## Step 2: Apply aggressive damping to customer routes
user@router# set policy-statement DAMPING-POLICY term CUSTOMER-ROUTES from community CUSTOMER-COMM
user@router# set policy-statement DAMPING-POLICY term CUSTOMER-ROUTES then damping aggressive-damping
user@router# set policy-statement DAMPING-POLICY term CUSTOMER-ROUTES then accept

## Step 3: Apply conservative damping to peer routes
user@router# set policy-statement DAMPING-POLICY term PEER-ROUTES from as-path PEER-AS
```

```
user@router# set policy-statement DAMPING-POLICY term PEER-ROUTES then damping conservative-damping
user@router# set policy-statement DAMPING-POLICY term PEER-ROUTES then accept

## Step 4: Default action for all other routes
user@router# set policy-statement DAMPING-POLICY then accept

## Step 5: Define the community and AS-path filters
user@router# set policy-options community CUSTOMER-COMM members 65000:100
user@router# set policy-options as-path PEER-AS "^(701|3356|174).*"

## Step 6: Apply the policy to BGP import
[edit protocols bgp group EBGP-PEERS]
user@router# set import DAMPING-POLICY
```

Complete Reference Configuration

```
## Complete BGP Damping Configuration Example

## Routing Options
routing-options {
    router-id 10.0.0.1;
    autonomous-system 65000;
}

## Policy Options with Damping Profiles
policy-options {
    ## Define different damping profiles
    damping aggressive-damping {
        half-life 10;
        reuse 500;
        suppress 2000;
        max-suppress 45;
    }
    damping conservative-damping {
        half-life 20;
        reuse 2000;
        suppress 5000;
        max-suppress 30;
    }
    damping no-damp {
        disable;
    }

    ## Define matching criteria
    prefix-list CRITICAL-PREFIXES {
        8.8.8.8/32;
        8.8.4.4/32;
        198.41.0.4/32;    ## Root DNS A
    }

    community CUSTOMER-COMM members 65000:100;
    community PEER-COMM members 65000:200;

    as-path CUSTOMER-AS ".* (65100|65101|65102)$";
    as-path PEER-AS "^(701|3356|174).*";

    ## Main damping policy
    policy-statement BGP-DAMPING-POLICY {
        term NO-DAMP-CRITICAL {
            from {
                prefix-list CRITICAL-PREFIXES;
            }
            then {
                damping no-damp;
                accept;
            }
        }
        term AGGRESSIVE-CUSTOMER {
            from {
                community CUSTOMER-COMM;
            }
            then {
                damping aggressive-damping;
                accept;
            }
```

```
        }
        term CONSERVATIVE-PEER {
            from {
                as-path PEER-AS;
            }
            then {
                damping conservative-damping;
                accept;
            }
        }
        term DEFAULT {
            then accept;  ## No damping for other routes
        }
    }
}

## BGP Configuration
protocols {
    bgp {
        damping;  ## Enable damping globally
        group EBGP-CUSTOMERS {
            type external;
            peer-as 65100;
            import BGP-DAMPING-POLICY;
            neighbor 192.168.1.1;
        }
        group EBGP-PEERS {
            type external;
            import BGP-DAMPING-POLICY;
            neighbor 192.168.2.1 {
                peer-as 701;
            }
            neighbor 192.168.3.1 {
                peer-as 3356;
            }
        }
    }
}
```

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

Essential Verification Commands

```
## 1. View damping configuration
user@router> show bgp summary
Groups: 2 Peers: 3 Down peers: 0
Damping: enabled    ## Confirms damping is active
Table          Tot Paths  Act Paths Suppressed
inet.0               1500       1450         50  ## 50 routes suppressed

## 2. Show all damped routes
user@router> show route damping suppressed
inet.0: 1500 destinations, 1550 routes (1450 active, 0 holddown, 50 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.0/24    [BGP ] 00:15:23, localpref 100
                     AS path: 65100 I
                     Damped, Reuse in: 00:12:15
203.0.113.0/24       [BGP ] 00:08:45, localpref 100
                     AS path: 65100 I
                     Damped, Reuse in: 00:05:30

## 3. View detailed damping information for a specific route
user@router> show route damping suppressed 192.168.100.0/24 detail
inet.0: 1500 destinations, 1550 routes (1450 active, 0 holddown, 50 hidden)

192.168.100.0/24 (1 entry, 0 announced)
        BGP     Preference: 170/-101
                Next hop type: Router
                Address: 0xb3d4d04
                Next-hop reference count: 25
                Source: 192.168.1.1
                Next hop: 192.168.1.1 via ge-0/0/0.0
                State: <Hidden Ext>
                Inactive reason: Damped
```

```
                Local AS: 65000 Peer AS: 65100
                Age: 15:23
                Validation State: unverified
                Merit (penalty): 3245    ## Current penalty value
                Damping parameters:
                  Half-life: 10:00
                  Reuse merit: 500
                  Suppress merit: 2000
                  Maximum suppress time: 45:00
                  Last update merit: 1000
                  Last update: 00:02:15 ago
                  Reuse in: 00:12:15      ## When route will be reused

## 4. View damping history for a prefix
user@router> show route damping history
inet.0: 1500 destinations, 1550 routes (1450 active, 0 holddown, 50 hidden)

192.168.100.0/24    Penalty: 3245 Last update: 00:02:15 Flaps: 4
                     Suppress threshold: 2000
                     Reuse in: 00:12:15
                     History: 00:00:00 Down
                            00:02:15 Up (Penalty 1000)
                            00:05:30 Down (Penalty 2000)
                            00:07:45 Up (Penalty 3000)
                            00:10:00 Down (Penalty 4000, Suppressed)

## 5. Clear damping information
user@router> clear bgp damping 192.168.100.0/24
Cleared damping state for 1 route

## 6. View damping statistics
user@router> show bgp damping statistics
Total routes monitored: 1500
Currently suppressed: 50
Routes reused: 245
Total flaps observed: 892
Damping profiles in use:
  aggressive-damping: 35 routes
  conservative-damping: 15 routes
  default: 0 routes
```

Common Troubleshooting Scenarios

Scenario 1: Routes Remain Suppressed Too Long

Symptom: Important route has been suppressed for hours

Diagnostic Commands:

```
user@router> show route damping suppressed 10.10.0.0/24 detail
10.10.0.0/24 (1 entry, 0 announced)
        State: <Hidden Ext>
        Inactive reason: Damped
        Merit (penalty): 15000      ## Very high penalty!
        Damping parameters:
          Maximum suppress time: 60:00
          Reuse in: 00:45:00        ## Still 45 minutes to go

user@router> show route damping history 10.10.0.0/24
10.10.0.0/24    Flaps: 25 in last 2 hours  ## Excessive flapping
```

Cause: Route flapped many times, accumulating massive penalty

Solution:

```
## Immediate fix: Clear damping for this specific route
user@router> clear bgp damping 10.10.0.0/24

## Long-term fix: Adjust damping parameters
[edit policy-options damping aggressive-damping]
user@router# set max-suppress 30    ## Reduce from 45 to 30 minutes
user@router# set suppress 3000      ## Increase suppress threshold
user@router# commit
```

Scenario 2: Critical Routes Being Damped

Symptom: DNS root server route gets suppressed

Diagnostic Commands:

```
user@router> show route 198.41.0.4/32
## Route not in routing table

user@router> show route damping suppressed 198.41.0.4/32
inet.0: 1500 destinations, 1551 routes (1450 active, 0 holddown, 51 hidden)
198.41.0.4/32       [BGP ] 00:05:00, localpref 100
                    Damped, Reuse in: 00:10:00  ## Critical route damped!
```

Cause: No exception configured for critical infrastructure

Solution:

```
[edit policy-options]
user@router# set prefix-list NEVER-DAMP 198.41.0.4/32
user@router# set prefix-list NEVER-DAMP 192.5.5.241/32  ## Add all root servers

[edit policy-options policy-statement BGP-DAMPING-POLICY]
user@router# set term NEVER-DAMP from prefix-list NEVER-DAMP
user@router# insert term NEVER-DAMP before term AGGRESSIVE-CUSTOMER
user@router# set term NEVER-DAMP then damping no-damp
user@router# set term NEVER-DAMP then accept
user@router# commit

## Clear existing damping state
user@router> clear bgp damping 198.41.0.4/32
```

Module 15: BGP Troubleshooting - Complete Learning Guide

Part 1: The Conceptual Lecture (The Why)

The BGP Troubleshooting Challenge

BGP is like conducting a symphony orchestra where each musician (router) must play in perfect harmony. When something goes wrong, you need to determine: Is it the sheet music (configuration)? Is a musician not following the conductor (policy)? Or is someone playing a different song entirely (mismatch)?

BGP troubleshooting is particularly challenging because:

1. **Distributed State**: Each router maintains its own BGP state
2. **Policy Complexity**: Multiple policies interact in non-obvious ways
3. **Silent Failures**: BGP can establish sessions but still not exchange routes
4. **Indirect Symptoms**: The problem's symptom often appears far from its cause

The BGP Troubleshooting Methodology

```
BGP Troubleshooting Flow:

     Is BGP Session Up?
           |
     No ---+--- Yes
     |           |
Check Layer 1-3  |
Check BGP Config |
     |           |
     Fix    Routes Received?
                 |
           No ---+--- Yes
           |           |
    Check Policies     |
    Check NLRI    Routes Installed?
           |           |
          Fix    No ---+--- Yes
                 |           |
          Check Policies     |
          Check Preference   Done
```

```
          Check Next-hop
                |
               Fix
```

Common IBGP Issues

Issue 1: The Next-Hop Reachability Problem

In IBGP, the protocol doesn't change the next-hop by default. This creates situations where routers learn routes they cannot use.

```
IBGP Next-Hop Problem:

    AS 100          AS 65000 (Your Network)         AS 200
       |                    |                          |
   [R-EXT1]----EBGP----[R-EDGE1]                       |
  10.1.1.1            172.16.1.1                       |
       |                    |                          |
                          IBGP                         |
                            |                          |
                     [R-CORE]----IBGP----[R-EDGE2]----EBGP----[R-EXT2]
                    172.16.2.1          172.16.3.1            10.2.2.1

Problem: R-CORE learns routes from R-EDGE1 with next-hop 10.1.1.1 (R-EXT1)
         But R-CORE has no route to 10.1.1.1!
```

Issue 2: The Route Reflection Hidden Route Problem

Route Reflectors can hide routes when the RR itself cannot resolve the next-hop.

```
Route Reflection Hidden Routes:

         [RR]
        /    \
       /      \
   [Client1] [Client2]----EBGP----[External]
                                  203.0.113.1

Client2 advertises route to RR with next-hop 203.0.113.1
If RR cannot resolve 203.0.113.1, it won't reflect to Client1
Client1 never learns the route!
```

Common EBGP Issues

Issue 1: The TTL Problem

EBGP uses TTL=1 by default, assuming direct connections. Multi-hop EBGP requires explicit configuration.

```
EBGP Multi-hop Scenario:

[Your Router]----L3 Switch----L3 Switch----[Peer Router]
  AS 65000         TTL=1       TTL=0         AS 65001
                    DROP!

Packet dropped because TTL expires
```

Issue 2: The Hidden AS-Path Loop

BGP's loop prevention can cause legitimate routes to be rejected.

```
AS-Path Loop Prevention:

AS Topology: 65000 --- 65001 --- 65002
                  \                /
                   \              /
                    --- 65003 ----

Your AS (65000) advertises routes to 65001
65001 advertises to 65002
65002 advertises to 65003
65003 tries to advertise back to 65000
You reject it! (Your AS in path)
```

```
But what if 65003 has a better path you want to use?
```

Part 2: The Junos CLI Masterclass (The How)

BGP Troubleshooting Command Hierarchy

```
## Session-level troubleshooting
show bgp summary                        ## Quick overview
show bgp neighbor <address>             ## Detailed neighbor state
show log messages | match BGP           ## BGP state changes

## Route-level troubleshooting
show route receive-protocol bgp <peer> ## Pre-policy routes
show route protocol bgp                 ## Post-policy routes
show route advertising-protocol bgp     ## Advertised routes

## Hidden routes investigation
show route hidden                       ## All hidden routes
show route hidden detail                ## Why routes are hidden

## BGP state and statistics
show bgp statistics                     ## Packet counters
monitor traffic interface <int> matching "port 179"  ## Live BGP packets
```

IBGP Troubleshooting Patterns

Pattern 1: Systematic IBGP Verification

```
## Step 1: Verify IBGP session establishment
user@router> show bgp summary
Groups: 1 Peers: 3 Down peers: 1    ## One peer is down!
Peer                AS      InPkt     OutPkt    State|#Active/Received/Accepted
172.16.1.1       65000        142        145    Establ 10/15/12
172.16.2.1       65000          0          0    Active      ## Problem peer
172.16.3.1       65000        150        148    Establ 5/8/8

## Step 2: Check specific peer details
user@router> show bgp neighbor 172.16.2.1
Peer: 172.16.2.1 AS 65000 Local: 172.16.0.1+51289 AS 65000
  Type: Internal    State: Active        ## Stuck in Active state
  Last State: Connect   Last Event: ConnectRetry
  Last Error: Hold Timer Expired Error
  Options: <Preference LocalAddress>
  Local Address: 172.16.0.1 Holdtime: 90 Preference: 170
  Number of flaps: 3
  Last flap event: HoldTime
  Error: 'Hold Timer Expired Error' Sent: 0 Recv: 3

## Step 3: Check connectivity to peer
user@router> ping 172.16.2.1 source 172.16.0.1
PING 172.16.2.1: 56 data bytes
^C
--- 172.16.2.1 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss

## Step 4: Check IGP for loopback reachability
user@router> show route 172.16.2.1
## No route to destination!

## Step 5: Check OSPF status
user@router> show ospf neighbor
## 172.16.2.1 not in OSPF neighbor list

## Solution: Fix IGP first, then BGP will establish
```

Pattern 2: IBGP Next-Hop Resolution

```
## Step 1: Check for hidden BGP routes
user@router> show route hidden protocol bgp
inet.0: 500 destinations, 520 routes (450 active, 0 holddown, 70 hidden)

203.0.113.0/24      [BGP ] 00:15:23, localpref 100
```

```
                    AS path: 100 I

## Step 2: Investigate why route is hidden
user@router> show route hidden 203.0.113.0/24 detail
203.0.113.0/24 (1 entry, 0 announced)
        BGP     Preference: 170/-101
                Next hop type: Unusable    ## Next-hop problem!
                Address: 0x0
                Next-hop reference count: 0
                State: <Hidden Int Ext>
                Inactive reason: Unusable path
                Protocol next hop: 10.1.1.1    ## External next-hop
                Indirect next hop: 0x0 -

## Step 3: Check next-hop reachability
user@router> show route 10.1.1.1
## No route found

## Step 4: Implement next-hop-self solution
[edit protocols bgp group IBGP]
user@router# set type internal
user@router# set export NEXT-HOP-SELF

[edit policy-options policy-statement NEXT-HOP-SELF]
user@router# set term 1 from protocol bgp
user@router# set term 1 then next-hop self
user@router# commit

## Alternative: Use IGP passive interface for external links
[edit protocols ospf area 0.0.0.0]
user@router# set interface ge-0/0/0.0 passive  ## External link
user@router# commit
```

EBGP Troubleshooting Patterns

Pattern 1: EBGP Multihop Configuration

```
## Problem: EBGP session not establishing through intermediate device

## Step 1: Check current configuration
user@router> show configuration protocols bgp group EBGP-PEER
type external;
peer-as 65001;
neighbor 203.0.113.1;

## Step 2: Test connectivity with TTL
user@router> ping 203.0.113.1 ttl 1
PING 203.0.113.1: 56 data bytes
Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 47f5   0 0000  01  01 d32c 192.168.1.1 203.0.113.1

## Step 3: Traceroute to see hops
user@router> traceroute 203.0.113.1
traceroute to 203.0.113.1, 30 hops max
 1  192.168.1.254  0.531 ms
 2  203.0.113.1    1.123 ms    ## 2 hops away!

## Step 4: Configure multihop
[edit protocols bgp group EBGP-PEER]
user@router# set multihop ttl 2
user@router# commit

## Step 5: Verify session comes up
user@router> show bgp neighbor 203.0.113.1 | match State
  Type: External    State: Established
```

Pattern 2: EBGP Authentication Mismatch

```
## Symptom: TCP connection resets immediately

## Step 1: Monitor BGP packets
user@router> monitor traffic interface ge-0/0/0 matching "port 179"
15:32:45.123456 Out IP 192.168.1.1.55234 > 203.0.113.1.179: Flags [S]
```

```
15:32:45.234567 In  IP 203.0.113.1.179 > 192.168.1.1.55234: Flags [R]
## TCP Reset received!

## Step 2: Check for MD5 authentication
user@router> show configuration protocols bgp group EBGP-PEER
type external;
peer-as 65001;
neighbor 203.0.113.1 {
    authentication-key "$9$H.5F/A0IEyr"; ## MD5 configured
}

## Step 3: Verify with peer (out of band)
## Peer says: "MD5 key is SecretKey123"

## Step 4: Reconfigure authentication
[edit protocols bgp group EBGP-PEER neighbor 203.0.113.1]
user@router# set authentication-key SecretKey123
user@router# commit

## Step 5: Verify session establishes
user@router> show bgp neighbor 203.0.113.1 | match -A2 Authentication
  Options: <Preference LocalAddress AuthenticationKey>
  Authentication key is configured
```

Complete Troubleshooting Command Reference

```
## BGP Session Troubleshooting Commands

## 1. Quick session status
show bgp summary

## 2. Detailed peer information
show bgp neighbor <address>

## 3. BGP group configuration
show bgp group <group-name>

## 4. Monitor state changes
monitor bgp neighbor <address>

## 5. Check BGP logs
show log messages | match "BGP|bgp"

## Route Advertisement Troubleshooting

## 6. Routes received from peer (pre-policy)
show route receive-protocol bgp <peer>

## 7. Routes after import policy
show route protocol bgp <peer>

## 8. Routes sent to peer (post-policy)
show route advertising-protocol bgp <peer>

## 9. Rejected routes
show route receive-protocol bgp <peer> hidden

## Hidden Route Analysis

## 10. All hidden routes
show route hidden

## 11. Hidden route details
show route hidden detail <prefix>

## 12. Resolution failures
show route resolution unresolved

## BGP Table Analysis

## 13. BGP routing table
show route table inet.0 protocol bgp

## 14. BGP path selection
show route <prefix> detail
```

```
## 15. AS path details
show route aspath-regex ".*65001.*"

## Statistics and Monitoring

## 16. BGP statistics
show bgp statistics

## 17. Monitor BGP packets
monitor traffic interface <interface> matching "port 179"

## 18. BGP send/receive counters
show bgp neighbor <address> statistics
```

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

Common BGP Troubleshooting Scenarios

Scenario 1: IBGP Routes Not Being Installed

Symptom: Routes received via IBGP but not in routing table

Diagnostic Commands:

```
user@router> show route receive-protocol bgp 172.16.1.1
inet.0: 1000 destinations, 1200 routes (1000 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 192.168.100.0/24        10.1.1.1                     100        I

user@router> show route 192.168.100.0/24
## No route found!

user@router> show route hidden 192.168.100.0/24 detail
192.168.100.0/24 (1 entry, 0 announced)
        BGP     Preference: 170/-101
                State: <Hidden Int Ext>
                Inactive reason: Unusable path
                Protocol next hop: 10.1.1.1
                Indirect next hop: 0x0 -     ## No recursive lookup possible
```

Cause: IBGP next-hop not reachable

Solution:

```
## Option 1: Configure next-hop self on edge router
[edit protocols bgp group IBGP]
user@edge-router# set export NHS-POLICY

[edit policy-options policy-statement NHS-POLICY]
user@edge-router# set term 1 from protocol bgp
user@edge-router# set term 1 from route-type external
user@edge-router# set term 1 then next-hop self
user@edge-router# commit

## Option 2: Redistribute connected interfaces into IGP
[edit protocols ospf]
user@edge-router# set export REDIS-CONNECTED

[edit policy-options policy-statement REDIS-CONNECTED]
user@edge-router# set term 1 from protocol direct
user@edge-router# set term 1 from interface ge-0/0/0.0
user@edge-router# set term 1 then accept
user@edge-router# commit
```

Scenario 2: BGP Session Flapping

Symptom: BGP session repeatedly goes up and down

Diagnostic Commands:

```
user@router> show bgp neighbor 203.0.113.1
Peer: 203.0.113.1+179 AS 65001
  Type: External    State: Active
```

```
  Last State: Established   Last Event: Stop
  Last Error: Cease
  Number of flaps: 15
  Last flap event: RecvNotify
  Peer ID: 203.0.113.1      Local ID: 192.168.1.1

user@router> show log messages | match 203.0.113.1 | last 10
Oct 20 10:15:23 BGP_PREFIX_LIMIT_EXCEEDED: Peer 203.0.113.1: Configured limit 1000 exceeded
Oct 20 10:15:23 bgp_send_notification: Peer 203.0.113.1: Cease
Oct 20 10:20:45 BGP_STATE_CHANGED: Peer 203.0.113.1: Established
Oct 20 10:25:18 BGP_PREFIX_LIMIT_EXCEEDED: Peer 203.0.113.1: Configured limit 1000 exceeded
```

Cause: Prefix limit being exceeded

Solution:

```
## Check current prefix limit
user@router> show configuration protocols bgp group EBGP neighbor 203.0.113.1
family inet {
    unicast {
        prefix-limit {
            maximum 1000;
        }
    }
}

## Check actual prefixes received
user@router> show bgp neighbor 203.0.113.1 | match prefix
  Prefixes received: 1245

## Increase prefix limit with warning threshold
[edit protocols bgp group EBGP neighbor 203.0.113.1 family inet unicast]
user@router# set prefix-limit maximum 2000
user@router# set prefix-limit teardown 90 idle-timeout 30
user@router# commit
```

Module 16: Policy Troubleshooting - Complete Learning Guide

Part 1: The Conceptual Lecture (The Why)

The Fundamental Challenge of Routing Policy

Imagine you're the traffic controller for a massive highway system. You don't just need to know where roads go - you need to enforce rules: commercial trucks use certain routes, local traffic gets priority on some roads, and hazardous materials must avoid tunnels. Routing policy is your traffic control system for network packets.

The challenge is that policies interact in complex ways:

- Multiple policies can be chained
- Order matters enormously
- One misplaced term can break everything
- Effects cascade through the network

Understanding Policy Processing Flow

```
Policy Processing Pipeline:

    Incoming       Import        Routing        Export        Outgoing
     Routes  -->  Policies  -->  Table   -->  Policies  -->  Routes
       |            |             |              |             |
    (Raw BGP)   (Filtering)   (Decision)    (Filtering)    (Advertised)
               (Modification)  Process     (Modification)

At Each Stage:
- Match conditions are evaluated
- Actions are applied
- Accept/Reject decisions are made
- Default actions apply if no match
```

Policy Structure Fundamentals

A Junos policy is like a legal document with clauses (terms). Each term has:

1. **Match Conditions** (from): What to look for
2. **Actions** (then): What to do when matched

```
Policy Logic Flow:

Policy EXAMPLE-POLICY
    |
    +-- Term 1
    |     |
    |     +-- Match: from protocol bgp
    |     +-- Action: then local-preference 200
    |     +-- Action: then accept  [STOP HERE if matched]
    |
    +-- Term 2
    |     |
    |     +-- Match: from route-filter 10.0.0.0/8 orlonger
    |     +-- Action: then reject  [STOP HERE if matched]
    |
    +-- Default Action
          |
          +-- Implicit: then reject (for import)
                    OR then accept (for export)
```

The Critical Role of Regular Expressions

Regular expressions (regex) in routing policy are like wildcards on steroids. They match patterns in AS paths, communities, and more.

```
Common Regex Patterns:

Pattern          Meaning                Example Matches
^65000$          Exactly AS 65000       "65000"
^65000           Starts with AS 65000   "65000 65100 65200"
65000$           Ends with AS 65000     "65100 65200 65000"
.*65000.*        Contains AS 65000      "65100 65000 65200"
^65000_65100$    AS 65000 then 65100    "65000 65100"
^(65000|65001)   Starts with either     "65000 ..." or "65001 ..."
^[0-9]+$         Any single AS          "65000" but not "65000 65100"
^([0-9]+)_\1$    Same AS repeated       "65000 65000"

Special Characters:
^     Start of string
$     End of string
.     Any character
*     Zero or more of preceding
+     One or more of preceding
?     Zero or one of preceding
_     AS path separator
|     OR operator
[]    Character class
()    Grouping
\     Escape character
```

Common Policy Pitfalls

1. **The Silent Drop**: Forgetting default actions
2. **The Order Trap**: Terms evaluated sequentially
3. **The Regex Typo**: Small regex errors with big impacts
4. **The Missing Accept**: Explicit accept needed
5. **The Chain Break**: Policy chains stop on first accept/reject

Part 2: The Junos CLI Masterclass (The How)

Policy Configuration Hierarchy

```
Configuration Structure:

[edit policy-options]
├── prefix-list <name>           # Named prefix lists
├── route-filter <prefix>        # Inline filters
├── as-path <name> <regex>       # AS path patterns
```

```
├── community <name> members     # Community definitions
├── policy-statement <name>      # The actual policy
│   ├── term <name>
│   │   ├── from                # Match conditions
│   │   └── then                # Actions
│   └── then                    # Default action
```

Building a Complete Policy Framework

```
## Step 1: Define reusable components
[edit policy-options]

## Prefix lists for organization
user@router# set prefix-list RFC1918 10.0.0.0/8
user@router# set prefix-list RFC1918 172.16.0.0/12
user@router# set prefix-list RFC1918 192.168.0.0/16

user@router# set prefix-list BOGONS 0.0.0.0/8
user@router# set prefix-list BOGONS 169.254.0.0/16
user@router# set prefix-list BOGONS 224.0.0.0/3

## AS-Path regular expressions
user@router# set as-path CUSTOMER-AS "^65100$"
user@router# set as-path PEER-AS "^(701|1239|3356).*"
user@router# set as-path TRANSIT-AS "^174.*"
user@router# set as-path CONTAINS-PRIVATE ".*(64512|65535).*"

## Community values
user@router# set community CUSTOMER-COMM members 65000:100
user@router# set community PEER-COMM members 65000:200
user@router# set community TRANSIT-COMM members 65000:300
user@router# set community NO-EXPORT members no-export
user@router# set community BLACKHOLE members 65000:666

## Step 2: Create comprehensive import policy
user@router# edit policy-statement BGP-IMPORT

## Term 1: Reject bogons and RFC1918
user@router# set term REJECT-BOGONS from prefix-list-filter BOGONS orlonger
user@router# set term REJECT-BOGONS then reject

user@router# set term REJECT-RFC1918 from prefix-list-filter RFC1918 orlonger
user@router# set term REJECT-RFC1918 then reject

## Term 2: Reject too-specific prefixes
user@router# set term REJECT-LONG-PREFIXES from route-filter 0.0.0.0/0 prefix-length-range /25-/32
user@router# set term REJECT-LONG-PREFIXES then reject

## Term 3: Customer routes - highest preference
user@router# set term CUSTOMER-ROUTES from as-path CUSTOMER-AS
user@router# set term CUSTOMER-ROUTES then local-preference 150
user@router# set term CUSTOMER-ROUTES then community add CUSTOMER-COMM
user@router# set term CUSTOMER-ROUTES then accept

## Term 4: Peer routes - medium preference
user@router# set term PEER-ROUTES from as-path PEER-AS
user@router# set term PEER-ROUTES then local-preference 100
user@router# set term PEER-ROUTES then community add PEER-COMM
user@router# set term PEER-ROUTES then accept

## Term 5: Transit routes - lowest preference
user@router# set term TRANSIT-ROUTES from as-path TRANSIT-AS
user@router# set term TRANSIT-ROUTES then local-preference 80
user@router# set term TRANSIT-ROUTES then community add TRANSIT-COMM
user@router# set term TRANSIT-ROUTES then accept

## Default action - reject everything else
user@router# set then reject

## Step 3: Create export policy
user@router# top
user@router# edit policy-options policy-statement BGP-EXPORT

## Term 1: Announce our prefixes
user@router# set term OUR-PREFIXES from protocol aggregate
```

```
user@router# set term OUR-PREFIXES from route-filter 203.0.113.0/24 exact
user@router# set term OUR-PREFIXES then accept

## Term 2: Announce customer prefixes to everyone
user@router# set term CUSTOMER-TO-ALL from community CUSTOMER-COMM
user@router# set term CUSTOMER-TO-ALL then accept

## Term 3: Don't announce peer/transit to other peers/transit
user@router# set term NO-PEER-TO-PEER from community [ PEER-COMM TRANSIT-COMM ]
user@router# set term NO-PEER-TO-PEER to as-path [ PEER-AS TRANSIT-AS ]
user@router# set term NO-PEER-TO-PEER then reject

## Default - reject
user@router# set then reject

## Step 4: Apply policies to BGP
[edit protocols bgp]
user@router# set group CUSTOMERS import BGP-IMPORT
user@router# set group CUSTOMERS export BGP-EXPORT
user@router# set group PEERS import BGP-IMPORT
user@router# set group PEERS export BGP-EXPORT
```

Advanced Policy Techniques

```
## Using policy chains
[edit policy-options]
user@router# set policy-statement BASIC-SANITY term 1 from route-filter 0.0.0.0/0 exact
user@router# set policy-statement BASIC-SANITY term 1 then reject
user@router# set policy-statement BASIC-SANITY term 2 from route-filter 0.0.0.0/0 prefix-length-range /25-/32
user@router# set policy-statement BASIC-SANITY term 2 then reject

user@router# set policy-statement CUSTOMER-SPECIFIC term 1 from neighbor 192.168.1.1
user@router# set policy-statement CUSTOMER-SPECIFIC term 1 then local-preference 200

## Chain policies together
[edit protocols bgp group CUSTOMERS]
user@router# set import [ BASIC-SANITY CUSTOMER-SPECIFIC BGP-IMPORT ]

## Using subroutines (policy calling policy)
[edit policy-options policy-statement MAIN-POLICY]
user@router# set term CALL-SUB from protocol bgp
user@router# set term CALL-SUB then policy SUB-POLICY

## Conditional advertisement
[edit policy-options]
user@router# set condition UPSTREAM-AVAILABLE if-route-exists 0.0.0.0/0
user@router# set condition UPSTREAM-AVAILABLE if-route-exists 0.0.0.0/0 table inet.0

user@router# set policy-statement CONDITIONAL-ADVERTISE term 1 from condition UPSTREAM-AVAILABLE
user@router# set policy-statement CONDITIONAL-ADVERTISE term 1 then accept
user@router# set policy-statement CONDITIONAL-ADVERTISE term 2 then reject
```

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

Essential Policy Troubleshooting Commands

```
## 1. Test policy without applying it
user@router> test policy BGP-IMPORT 203.0.113.0/24
Policy BGP-IMPORT: 1 prefix accepted, 0 prefix rejected

## 2. Show policy configuration
user@router> show configuration policy-options policy-statement BGP-IMPORT

## 3. Trace policy evaluation
user@router> show route 192.168.1.0/24 detail | match policy
     Policy: BGP-IMPORT
     Policy action: accept

## 4. Check which routes match a policy term
user@router> show route protocol bgp match-condition "policy BGP-IMPORT term CUSTOMER-ROUTES"

## 5. Display routes with specific communities
user@router> show route community 65000:100
```

```
## 6. Show routes matching AS-path regex
user@router> show route aspath-regex "^65100.*"

## 7. Policy statistics
user@router> show policy statistics
Policy: BGP-IMPORT
  Term REJECT-BOGONS: 15 matches
  Term REJECT-RFC1918: 3 matches
  Term CUSTOMER-ROUTES: 150 matches
  Term PEER-ROUTES: 500 matches
  Default action: 2 matches

## 8. Clear policy statistics
user@router> clear policy statistics
```

Common Policy Troubleshooting Scenarios

Scenario 1: Routes Not Being Accepted

Symptom: Expected routes not appearing in routing table

Diagnostic Commands:

```
user@router> show route receive-protocol bgp 192.168.1.1
inet.0: 1000 destinations, 1000 routes (950 active, 0 holddown, 50 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  10.10.10.0/24           192.168.1.1                             65100 I

user@router> show route 10.10.10.0/24
## No route!

user@router> show route receive-protocol bgp 192.168.1.1 hidden detail 10.10.10.0/24
inet.0: 1000 destinations, 1000 routes (950 active, 0 holddown, 50 hidden)
* 10.10.10.0/24 (1 entry)
     Import: BGP-IMPORT      ## Policy applied
     BGP    Preference: 170
            Next hop: 192.168.1.1
            State: <Hidden Ext>
            Inactive reason: Policy  ## Rejected by policy!

user@router> test policy BGP-IMPORT 10.10.10.0/24
inet.0: 10.10.10.0/24
  Policy BGP-IMPORT: rejected
  Term REJECT-RFC1918: matched
```

Cause: Route matches RFC1918 rejection term

Solution:

```
## Add exception for legitimate use of RFC1918
[edit policy-options policy-statement BGP-IMPORT]
user@router# insert term ALLOW-CUSTOMER-RFC1918 before term REJECT-RFC1918
user@router# set term ALLOW-CUSTOMER-RFC1918 from neighbor 192.168.1.1
user@router# set term ALLOW-CUSTOMER-RFC1918 from prefix-list-filter RFC1918 orlonger
user@router# set term ALLOW-CUSTOMER-RFC1918 then local-preference 150
user@router# set term ALLOW-CUSTOMER-RFC1918 then accept
user@router# commit
```

Scenario 2: Wrong Routes Being Advertised

Symptom: Internal routes leaked to peers

Diagnostic Commands:

```
user@router> show route advertising-protocol bgp 203.0.113.1
inet.0: 1000 destinations, 1000 routes (1000 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.0/24             Self                                 I
* 10.0.1.0/24             Self                                 I  ## Internal routes!

user@router> show route 10.0.0.0/24 detail
10.0.0.0/24 (1 entry, 1 announced)
        *OSPF   Preference: 10
                Next hop: via ge-0/0/1.0
```

```
                Protocol next hop: 172.16.1.2
                Communities: 65000:999     ## Internal community

user@router> show configuration policy-options policy-statement BGP-EXPORT
## Missing term to block internal routes!
```

Cause: Export policy missing filter for internal routes

Solution:

```
[edit policy-options policy-statement BGP-EXPORT]
user@router# insert term BLOCK-INTERNAL before term OUR-PREFIXES
user@router# set term BLOCK-INTERNAL from community INTERNAL-ONLY
user@router# set term BLOCK-INTERNAL then reject

## Or more specifically:
user@router# set term BLOCK-INTERNAL from protocol ospf
user@router# set term BLOCK-INTERNAL from route-filter 10.0.0.0/8 orlonger
user@router# set term BLOCK-INTERNAL then reject
user@router# commit
```

Scenario 3: AS-Path Regex Not Matching

Symptom: AS-path based policy not working

Diagnostic Commands:

```
user@router> show route aspath-regex "^65100$"
## No output

user@router> show route detail 192.168.100.0/24 | match path
                AS path: 65100 65200 I   ## Not just 65100!

user@router> show configuration policy-options as-path CUSTOMER-AS
"^65100$";    ## Exact match only

user@router> test policy BGP-IMPORT 192.168.100.0/24
Policy BGP-IMPORT: rejected
  No term matched
  Default action: reject
```

Cause: AS-path regex too restrictive

Solution:

```
[edit policy-options]
user@router# delete as-path CUSTOMER-AS
user@router# set as-path CUSTOMER-AS "^65100.*"  ## Match 65100 at start
## Or if customer can have multiple AS paths:
user@router# set as-path CUSTOMER-AS ".* 65100$"  ## Match 65100 at end
user@router# commit

## Verify fix
user@router> show route aspath-regex "^65100.*"
inet.0: 1000 destinations, 1000 routes (1000 active, 0 holddown, 0 hidden)
  192.168.100.0/24   *[BGP/170] 00:01:15
                       AS path: 65100 65200 I
```

Scenario 4: Policy Chain Breaking Unexpectedly

Symptom: Second policy in chain not being evaluated

Diagnostic Commands:

```
user@router> show configuration protocols bgp group CUSTOMERS import
[ BASIC-SANITY CUSTOMER-SPECIFIC BGP-IMPORT ];

user@router> test policy BASIC-SANITY 192.168.1.0/24
Policy BASIC-SANITY: 1 prefix accepted    ## Accepts!

user@router> test policy CUSTOMER-SPECIFIC 192.168.1.0/24
Policy CUSTOMER-SPECIFIC: No match, using default action (reject)

## But the route isn't in the table!
```

```
user@router> show route 192.168.1.0/24
## No route

user@router> show configuration policy-options policy-statement BASIC-SANITY
term DEFAULT {
    then accept;    ## Accepts everything - chain stops here!
}
```

Cause: First policy has explicit accept, stopping chain

Solution:

```
[edit policy-options policy-statement BASIC-SANITY]
user@router# delete term DEFAULT
user@router# set then next policy    ## Continue to next policy
## Or remove all "then accept/reject" from BASIC-SANITY
## Let it fall through to next policy
user@router# commit
```

Policy Troubleshooting Best Practices

```
## 1. Always test policies before applying
user@router> test policy NEW-POLICY 10.0.0.0/24

## 2. Use "show | compare" before committing
user@router# show | compare

## 3. Keep backup of working configuration
user@router> show configuration policy-options | save policy-backup.txt

## 4. Use commit confirmed for risky changes
user@router# commit confirmed 5

## 5. Log policy actions for debugging
[edit policy-options policy-statement BGP-IMPORT term REJECT-BOGONS]
user@router# set then log

## 6. Use trace options for detailed debugging
[edit protocols bgp]
user@router# set traceoptions file bgp-trace
user@router# set traceoptions flag policy

## 7. Monitor policy evaluation
user@router> monitor start bgp-trace
user@router> monitor stop bgp-trace
```

These three comprehensive modules have equipped you with deep expertise in BGP Route Damping, BGP Troubleshooting, and Policy Troubleshooting. You now understand how route damping prevents instability through penalty algorithms, how to systematically troubleshoot both IBGP and EBGP issues, and how to debug complex routing policies using regular expressions and policy chains. These skills are absolutely critical for operating service provider networks and passing the JNCIE-SP exam. Each module's practical configurations and real-world troubleshooting scenarios prepare you to handle the complex problems you'll face in production networks.