OSPF Deep Dive

This is a self-study, lab based tutorial using **Juniper Networks** routers. Although this was developed on some old J2300 routers, any Junos based router should work for purposes of this tutorial. If Junos based routers are unavailable, the **Junosphere** virtual environment can definately be used. This assumes that the reader has some working knowledge of Junos operation and configuration.

Topics

Labs

Configurations

Topologies

OSPF Overview

- OSPF is an interior gateway protocol that routes Internet Protocol (IP) packets solely within a single routing domain (autonomous system). (From Wikipedia) Each router gathers link state information and builds a complete network topology for the entire routing domain.
- Runs on top of IP and has it's own protocol number
  - OSPFv2 routes IPv4
  - OSPFv3 routes IPv6
- One of the first protocols developed completely within the IETF

Precursors and Rationality

- In the early Internet routing was done with static routes, or dynamically with RIP within an Autonomous System
  - Static routes are a lot of maintenance, not dynamic
  - RIP had slow convergence, prone to loops, noisy, updates can consume a lot of bandwidth
- Search started for a new routing protocol for IP and the Internet
  - Open Interior Gateway Protocol Working Group forms at IETF in 1987
  - Huge arguments and differences of opinion caused formation of two groups, one that though a link-state protocol was the future and the other that wanted to tune distance vector protocols
  - Distance Vector vs. Link State
    - Distance Vector
      - Operation
        - Nodes advertise their connectivity neighbors
        - Based on received advertisements, a node can calculate the distance to a destination and the direction
        - Node does not have full knowledge of the path to a destination
      - Simple to implement
      - Computationally less intensive
      - Viewed as slow to converge
    - Link State
      - Operation
        - Every node in a routing domain actively forwards packets constructs a complete picture of the network topology of the domain
        - Each node advertises information about itself and it's connectivity
          - This information is flooded throughout the routing domain so every node has a complete picture connectivity
        - Each node can use this topology information to calculate a loop free path to every other node on the network
        - Any network changes are flooded throughout the entire routing domain so every node can maintain their own topology
        - Can be CPU intensive
        - *Critical that every node has the same view of the network topology*
      - Viewed as having faster convergence and less prone to transient loops
      - More difficult to implement especially in multi-vendor environment
      - More computational intensive
  - Link State Camp Searches for it's RIP replacement
    - Link State Routing was pioneered by BBN for use on the ARPANET
      - Assumed everything was a point to point link
      - Had some holes in the flooding alorythim
      - Suffered from some general inefficencies
      - Only worked on BBN gear
    - IS-IS was being developed for ISO
      - Offshoot of DECnet Phase IV
      - Viewed as having too many issues to be adapted for IP
  - Decision was made it would be easier to develop a new protocol rather than adapt or modify an existing one
  - Link-state camp splits off into the Open Shortest Path First Interior Gateway Protocol Working Group (OSPFIGP), and then later becomes the OSPF Working Group

OSPF History

- Development started in 1987, first complete specification published in 1991 (OSPFv2)in **RFC 2328**
  - Developed completely within the IETF
  - Numerous revisions and extensions
    - NSSA Support in **RFC 3101**
    - MPLS-TE in **RFC 3630**
    - Graceful Restart in **RFC 3623**
  - Age of the protocol explains some of it's behavior today
    - Routers had limited memory and CPU cycles
      - Some of the initial routers were UNIX workstations
    - Abundance of large broadcast networks
    - Slow and expensive WAN links
  - Initial OSPF Design Goals -- make up for RIPs shortcomings
    - Fast Convergence
    - Hierarchical Routing
    - More descriptive and flexible metrics
    - Distinction between internal and external routes
    - Support for classless routing
    - Ability to secure the routing protocols
    - Support for Type of Service routing: normal, cost, reliability, throughput and delay
  - OSPFv1 was deemed a failure
    - Too little detail on what constitutes the "best route"
    - OSPFv2 succeeded OSPFv1 and does not inter-operate with it's predecessor

OSPF Operation

- Runs directly on top of IP using protocol 89 for IPv4
- Makes use of two well known link-local scoped multicast 224.0.0.0/24 to limit to OSPF messages only interested parties
  - Uses 224.0.0.5 AllSPFRouters
  - Uses 224.0.0.6 AllDRRouters for broadcast links
- Takes advantage of IP's ability to run over almost any link-layer protocols

- OSPF routers discover each other with a hello mechanism
  - Hellos are periodically sent to the AllSPFRouters address
  - Once OSPF routers see hellos from another router, they will compare configuration parameters, and if they agree will form an adjacency and become neighbors
- OSPF routers advertise information about themselves and their connectivity in Link State Advertisements (LSAs) to their OSPF neighbors
- OSPF routers will send any LSAs they receive to all of their neighbors (except the one that they received the advertisement from) -- flooding
  - OSPFs uses a reliable flooding mechanism
    - All LSAs sent much be acknowledged by all routers the LSA was sent to
      - Explicitly through an LSA Acknowledgment
      - Implicitly by seeing the LSAs ID that was sent in another update from neighbor the LSA was sent to
    - Acknowledgments can be send direct or delayed
      - Direct acknowledgement are unicast to a nieghbor
      - Delayed acknowledgement allow some time to receive multiple LSAs and acknowledge them all with a single acknowledgement
    - Link State Advertisements are aged out over time to prevent stale information from building up
      - LSAs must be periodically refreshed by the originating router
    - Routers use LSAs to construct the Link State Database (LSDB)
      - Contains topology information for the entire routing domain
    - Each router uses it's own LSDB to calculate paths to every other node using Dijkstra's Shortest Path Algorithm
    - Can subdivide the entire OSPF routing domain into smaller regions known as areas
      - Can summarize routing information between areas to cut down on the size of the LSDB and the routing tables
        - Each area has it's own LSDB
    - Makes use of a two level hierarchy creating a simple hub and spoke topology
    - Top level of the hierarchy is known as the backbone area (0.0.0.0)
    - All areas must be connected directly to the backbone
      - Mechanisms to allow tunneling of routing information to get around this restriction

The Link State Database (LSDB)

- Routers collect Link State Advertisements and add the information into a database known as the Link-State Database or LSDB
  - LSDB is a model of the topology of the network
  - Contains tuples of the node's router id, neighbors router id, and a cost to that neighbor
  - Router runs Dijkstra's Shortest Path First Algorithm on the LSDB every the LSDB changes
    - Builds a shortest path tree to every other node in the network
    - The more nodes, more links, more LSA types, and more network changes the more SPF runs will have to be made
- SPF algorithm
  - Junos OSPF implements Dijkstra's Shortest Path First Algorithm
    - For any given node, the find the shortest path to every other node
    - Efficiently constructs a shortest path tree through an iterative process
  - OSPF Uses three databases during each SPF run
    - LSDB - Complete picture of all of the routing information in an area
    - Candidate Database - Matrix of all potential paths to every node in the network
    - Tree Database - Matrix of all of the shortest paths to every node in the network
      - The tree database is what is used to populate the routing table

Viewing the LSDB on Junos

- Can see a summary of the LSDB with the operational command `show ospf database summary`

```
admin@J2300-1> show ospf database summary
Area 0.0.0.0:
   7 Router LSAs
   8 Network LSAs
Externals:
Interface fe-0/0/1.12:
Area 0.0.0.0:
Interface fe-0/0/1.13:
Area 0.0.0.0:
Interface lo0.0:
Area 0.0.0.0:
```

- The LSDB can be viewed with the operational command`show ospf database`
  - Shows an overview of all of the LSAs in the LSDB for each area
    - Same as using the `brief` flag

```
admin@J2300-1> show ospf database

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.0.0.1         10.0.0.1         0x8000000a   801  0x22 0x2e48  60
Router   10.0.0.2         10.0.0.2         0x80000005   768  0x22 0x84da  60
Router   10.0.0.3         10.0.0.3         0x80000007   506  0x22 0x7a4d  72
Router   10.0.0.4         10.0.0.4         0x80000009   372  0x22 0x4b44  72
Router   10.0.0.5         10.0.0.5         0x80000005   337  0x22 0x1fbf  60
Router   10.0.0.6         10.0.0.6         0x80000004   338  0x22 0x6c55  60
Router   10.0.12.2        10.0.12.2        0x80000008  1115  0x22 0x6af7  48
Network *10.0.12.1        10.0.0.1         0x80000003   801  0x22 0x5f9   32
Network  10.0.13.3        10.0.0.3         0x80000002  1296  0x22 0xe118  32
Network  10.0.24.2        10.0.12.2        0x80000001  1115  0x22 0xb225  32
Network  10.0.24.4        10.0.0.4         0x80000002   808  0x22 0x707a  32
Network  10.0.34.3        10.0.0.3         0x80000001  1079  0x22 0x26bc  32
Network  10.0.35.3        10.0.0.3         0x80000001   506  0x22 0x29b7  32
Network  10.0.46.6        10.0.0.6         0x80000002   378  0x22 0x8d3f  32
Network  10.0.56.5        10.0.0.5         0x80000001   337  0x22 0x4381  32
```

- Using the `detail` flag shows information about the contents of each LSA

```
admin@J2300-1> show ospf database detail

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.0.0.1         10.0.0.1         0x8000000a  1091  0x22 0x2e48  60
  bits 0x0, link count 3
  id 10.0.12.1, data 10.0.12.1, Type Transit (2)
    Topology count: 0, Default metric: 10
  id 10.0.13.3, data 10.0.13.1, Type Transit (2)
    Topology count: 0, Default metric: 10
  id 10.0.0.1, data 255.255.255.255, Type Stub (3)
    Topology count: 0, Default metric: 0
```

```
 Topology default (ID 0)
   Type: Transit, Node ID: 10.0.13.3
     Metric: 10, Bidirectional
   Type: Transit, Node ID: 10.0.12.1
     Metric: 10, Bidirectional
Router   10.0.0.2         10.0.0.2         0x80000005  1058  0x22 0x84da  60
 bits 0x0, link count 3
 id 10.0.12.1, data 10.0.12.2, Type Transit (2)
   Topology count: 0, Default metric: 10
 id 10.0.24.4, data 10.0.24.2, Type Transit (2)
   Topology count: 0, Default metric: 10
 id 10.0.0.2, data 255.255.255.255, Type Stub (3)
.
.
```

- Using the `extensive` flag shows all of the information contained in each LSA

```
admin@J2300-1> show ospf database extensive

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.0.0.1         10.0.0.1         0x8000000a  1182  0x22 0x2e48  60
 bits 0x0, link count 3
 id 10.0.12.1, data 10.0.12.1, Type Transit (2)
   Topology count: 0, Default metric: 10
 id 10.0.13.3, data 10.0.13.1, Type Transit (2)
   Topology count: 0, Default metric: 10
 id 10.0.0.1, data 255.255.255.255, Type Stub (3)
   Topology count: 0, Default metric: 0
 Topology default (ID 0)
   Type: Transit, Node ID: 10.0.13.3
     Metric: 10, Bidirectional
   Type: Transit, Node ID: 10.0.12.1
     Metric: 10, Bidirectional
 Gen timer 00:30:17
 Aging timer 00:40:17
 Installed 00:19:42 ago, expires in 00:40:18, sent 00:19:42 ago
 Last changed 00:19:42 ago, Change count: 6, Ours
Router   10.0.0.2         10.0.0.2         0x80000005  1149  0x22 0x84da  60
 bits 0x0, link count 3
 id 10.0.12.1, data 10.0.12.2, Type Transit (2)
.
.
```

- As the LSDB can grow very large, can use several flags to the `show ospf database` command to narrow down the results:
  - `lsa-id` to view a specific LSA
  - `advertising-router (address | self)` to view LSAs originated from specific routers
  - `area (area id)` to view LSAs only in the LSDB for the specified area
  - `asbrsummary|external|network|netsummary|nssa|opaque-area|router|opaque-area` to view just certain LSA types

Checking which Interfaces are contributing to the LSDB

- Operational mode command `show ospf interface` shows the status of each interface that is configured to participate in the OSPF process
  - Can use the `detail` and `extensive` flags to see more information
  - Displays each interface that will contribute to the LSDB in some form, and which LSDB (area) they will populate
  - Also shows designated router status and number of neighbors

```
admin@J2300-1> show ospf interface brief
Interface           State   Area            DR ID           BDR ID          Nbrs
fe-0/0/1.12             DR      0.0.0.0         10.0.0.1        10.0.0.2           1
fe-0/0/1.13             BDR     0.0.0.0         10.0.0.3        10.0.0.1           1
lo0.0               DR      0.0.0.0         10.0.0.1        0.0.0.0           0
```

Clearing the LSDB

- The contents of the LSDB can be discarded with the operational mode command `clear ospf database`
  - Forces a complete LSDB resynchronization
  - Deletes all of the LSAs in the LSDB
  - Regenerates all of the LSAs the router originates
  - Destroys all adjacencies
  - Can be very disruptive!!!
- Can use the `purge` flag to discard all LSAs, and age out all of the system generated LSAS
  - Sets the maximum age on all the system's generated LSAs and refloods them
- Can use the same flags that are supported in `show ospf database` to selectively discard and purge certain LSAs rather than do them all enmass

Viewing the effects of the LSDB on the Routing Table

- Operational mode command `show ospf route` displays the OSPF routing table
  - A separate memory structure that contains all of the best OSPF routes
    - These may not be the best routes to a certain destination, but is only the OSPF routes
  - Can use the `detail` and `extensive` flags to provide more detail
  - Can narrow the routes displayed down with `abr|asbr|extern|inter|intra|network|router` flags

```
admin@J2300-1> show ospf route
Topology default Route Table:

Prefix             Path  Route      NH       Metric NextHop       Nexthop
                   Type  Type       Type            Interface     Address/LSP
10.0.0.2           Intra Router     IP           10 fe-0/0/1.12   10.0.12.2
10.0.0.3           Intra Router     IP           10 fe-0/0/1.13   10.0.13.3
10.0.0.4           Intra Router     IP           20 fe-0/0/1.12   10.0.12.2
                                                    fe-0/0/1.13   10.0.13.3
10.0.0.5           Intra Router     IP           20 fe-0/0/1.13   10.0.13.3
10.0.0.6           Intra Router     IP           30 fe-0/0/1.12   10.0.12.2
                                                    fe-0/0/1.13   10.0.13.3
10.0.0.1/32        Intra Network    IP            0 lo0.0
10.0.0.2/32        Intra Network    IP           10 fe-0/0/1.12   10.0.12.2
10.0.0.3/32        Intra Network    IP           10 fe-0/0/1.13   10.0.13.3
10.0.0.4/32        Intra Network    IP           20 fe-0/0/1.12   10.0.12.2
                                                    fe-0/0/1.13   10.0.13.3
10.0.0.5/32        Intra Network    IP           20 fe-0/0/1.13   10.0.13.3
10.0.0.6/32        Intra Network    IP           30 fe-0/0/1.12   10.0.12.2
```

```
                                       fe-0/0/1.13   10.0.13.3
10.0.12.0/24       Intra Network    IP           10 fe-0/0/1.12
10.0.13.0/24       Intra Network    IP           10 fe-0/0/1.13
10.0.24.0/24       Intra Network    IP           20 fe-0/0/1.12   10.0.12.2
10.0.34.0/24       Intra Network    IP           20 fe-0/0/1.13   10.0.13.3
10.0.35.0/24       Intra Network    IP           20 fe-0/0/1.13   10.0.13.3
10.0.46.0/24       Intra Network    IP           30 fe-0/0/1.12   10.0.12.2
                                       fe-0/0/1.13   10.0.13.3
10.0.56.0/24       Intra Network    IP           30 fe0/0/1.13    10.0.13.3

admin@J2300-1>
```

- Can view the OSPF routes that have been installed in the routing table with `show route protocol ospf`
  - Can use the `terse,brief, detail and extensive flags to vary the level of information displayed on the output`
  - Can use the multitude of other flags in the `show route command` to scope the output

```
admin@J2300-1> show route protocol ospf

inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.2/32        *[OSPF/10] 00:50:13, metric 10
                    > to 10.0.12.2 via fe-0/0/1.12
10.0.0.3/32        *[OSPF/10] 00:58:22, metric 10
                    > to 10.0.13.3 via fe-0/0/1.13
10.0.0.4/32        *[OSPF/10] 00:50:13, metric 20
                      to 10.0.12.2 via fe-0/0/1.12
                    > to 10.0.13.3 via fe-0/0/1.13
10.0.0.5/32        *[OSPF/10] 00:45:17, metric 20
                    > to 10.0.13.3 via fe-0/0/1.13
10.0.0.6/32        *[OSPF/10] 00:43:01, metric 30
                      to 10.0.12.2 via fe-0/0/1.12
                    > to 10.0.13.3 via fe-0/0/1.13
10.0.24.0/24       *[OSPF/10] 00:50:13, metric 20
                    > to 10.0.12.2 via fe-0/0/1.12
10.0.34.0/24       *[OSPF/10] 00:58:22, metric 20
                    > to 10.0.13.3 via fe-0/0/1.13
10.0.35.0/24       *[OSPF/10] 00:58:22, metric 20
                    > to 10.0.13.3 via fe-0/0/1.13
10.0.46.0/24       *[OSPF/10] 00:50:13, metric 30
                      to 10.0.12.2 via fe-0/0/1.12
                    > to 10.0.13.3 via fe-0/0/1.13
10.0.56.0/24       *[OSPF/10] 00:45:17, metric 30
                    > to 10.0.13.3 via fe-0/0/1.13
224.0.0.5/32       *[OSPF/10] 01:03:15, metric 1
                      MultiRecv

admin@J2300-1>
```

Basic OSPF Configuration on Junos

- All OSPFv2 specific configuration is done under the `procotols ospf` level of the Junos heirarchy
  - Must define at least one area with `edit protocols ospf area`
    - Area Id is displayed in dotted quad notation
    - Junos will translate any Area IDs into dotted quad notation
      - `edit protocols ospf area 0` becomes `area 0.0.0.0`
      - `edit protocols ospf area 1` becomes `area 0.0.0.1`
      - `edit protocols ospf area 256` becomes `area 0.0.1.0`
  - Define what interfaces belong to which area by adding the interface under the appropriate area level of the hierarchy with`set protocols ospf area interface`
    - Be sure to include the logical unit number on the interface definition
    - Advertises all configured networks on the specified interface
    - Can also configure an interface to participate in OSPF by using the IP address instead of the interface name with `set protocols ospf area IP`
      - Router will only advertise the subnet that matches the configured IP address
      - No additional networks that are configured on the interface will be advertised by OSPF
    - Cannot configure an interface to participate in OSPF by specifying an interface name and an IP address
      - Creates a commit error
      - Junos assumes logical unit 0 if omitted
  - Multiple area configuration
  -
    - Simply define another area and add interfaces into it
    - An interface can normally belong only to one area
    - **RFC 5185** OSPF Multi-Area Adjacency, defined a method for an interface to belong to more than one area
      - Allows a router to establish multiple adjacencies over a single link
        - Solves some routing inefficiencies
      - To configure an interface in Junos to belong to more than one area, simply add the `secondary` flag after the interface in the secondary area
      - Secondary interfaces must be point-to-point connections
      - Can't configure a secondary by IP address only

Example:*Adding interface fe-0/0/1.0 to the backbone area, and interface ge-0/0/3.300 to area 1.2.3.4*

```
user@Router> edit
Entering configuration mode

[edit]
user@Router# edit protocols ospf

[edit protocols ospf]
user@Router# set area 0 interface fe-0/0/1.0

[edit protocols ospf]
user@Router# set area 1.2.3.4 interface ge-0/0/3.300

[edit protocols ospf]
user@Router# show
area 0.0.0.0 {
    interface fe-0/0/1.0;
}
area 1.2.3.4 {
    interface ge-0/0/3.300;
}

[edit protocols ospf]
user@Router#
```

- Viewing the OSPF process
  - Can get an overview of OSPF with the operational mode command `show ospf overview`
    - Displays all OSPF processes, Router ID, Router type, general parameters, areas configured, authentication, neighbors
    - Can use the `extensive` flag to get more information
  - Can view general OSPF stats with operational mode command `show ospf statistics`
    - Shows statistics on numbers and types of packets sent and received, LSAs, flooding, acknowledgments, retransmissions and errors

Exercise:*Build the backbone*

SPF Tuning and Troubleshooting

- Default SPF timers normally work well and there is no need to change them
- Can tune them if needed
  - Interoperability with other vendors
  - Decreasing convergence time
  - Save CPU cycles
- Configured under `protocols ospf spf-options`
  - Can modify the delay in the time between the detection of a topology change and when the SPF algorithm actually runs with `delay (milliseconds)`
    - Default is 200 ms
    - Can change from 50 to 8000 ms
  - Can change maximum number of times that the SPF algorithm can run in succession before the hold-down timer begins
    - Configure with `holddown (milliseconds)`
    -
    - Default is 5000 ms
    - Can change from 2000 to 20,000 ms
  - Can configure the hold down, or wait, before running another SPF calculation after the SPF algorithm has run in succession the configured maximum number of times
    - Configure with `rapid-runs (number)`
    - Default is 3
    - Can range from 1 to 5
  - Viewing SPF Runs
    - A log of the SPF runs can be seen with the operational mode command `show ospf log`
      - Gives a log of what caused a SPF run, and how long it took
  - General stats as related to SPF runs can be seen with the operational mode command `show ospf io-statistics`

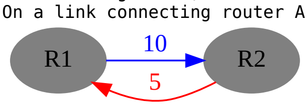
Example:*The SPF log*

```
admin@J2300-1> show ospf log
Topology default SPF log:

  Last instance of each event type
When            Type            Elapsed
00:09:42        SPF             0.000433
00:09:42        Stub            0.000024
00:09:42        Interarea       0.000007
00:09:42        External        0.000002
00:09:42        NSSA            0.000001
00:09:42        Cleanup         0.000026

  Maximum length of each event type
When            Type            Elapsed
02:02:57        SPF             0.001650
02:02:57        Stub            0.000852
00:09:42        Interarea       0.000007
02:03:02        External        0.000020
02:03:02        NSSA            0.000018
02:03:02        Cleanup         0.003413

  Last 100 events
When            Type            Elapsed

02:02:05          Total         0.001847
02:01:23        SPF             0.000032
02:01:23        Stub            0.000006
02:01:23        Interarea       0.000001
02:01:23        External        0.000001
02:01:23        NSSA            0.000001
02:01:23        Cleanup         0.000012
02:01:23          Total         0.000077
02:01:18        SPF             0.000043
02:01:18        Stub            0.000011
02:01:18        Interarea       0.000002
02:01:18        External        0.000001
.
.
```

  - Detailed information on individual SPF runs can be logged and viewed using Junos traceoptions for OSPF
    - Set with `set protocols ospf traceoptions spf`
      - Can use the `detail` flag for more in depth information

OSPF Cost

- Cost is a measure of how desirable (or undesirable) it is for a link
  - Cost is advertised as a 16 bit integer - 0 to 65535
    - Cost of 0 is reserved only for connected networks
  - Max cost of a route from end-point to end-point is practically limited to a 24 bit integer - 16,777,215
    - Summary LSAs, External LSAs use 24 bit field for metric
- Cost is a vector
  - Has both magnitude, and direction
  - On a link connecting router A and B, cost from A to B can differ from the cost from B to A



    - Can cause Asymmetric routing
      - Not necessarily bad, however:

        - Can make troubleshooting more difficult
        - Stateful devices (Firewalls, Stateful Packet Filters) may block return traffic
        - Many security devices (IDS, IPS, etc.) can't fully analyze a network flow
            - May even assume some sort of an attack is in progress
- OSPF spec specifies a method for automatically calculating metrics based on interface bandwidth
    - `cost = (reference bandwidth) / (physical interface bandwidth)`
    - Protocol definition uses 100 Mbps as a default reference bandwidth
    - Any value < 1 is rounded up to 1
    - Physical interface bandwidth is automatically calculated based on hard coded values for physical interface speeds
        - Can override the physical interface bandwidth by specifying the interface bandwidth with a `bandwidth` (bandwidth in bps) statement on the logical interface
    - Spec is fairly old -- Fast Ethernet was seen as REALLY Fast
        - Lose granularity on modern networks with Gigabit Ethernet, 10GbE, 100GbE, High Speed SONET/SDH as the automatically calculated cost comes out as 1
        - No way to discern Fast Ethernet from 100 GbE
    - Overcome this limitation by changing the reference bandwidth with `set protocols ospf reference-bandwidth (bandwidth in bps)`
        - Range from 9600 bps, to 1000000000000 bps (1 Terabit/sec)
        - Can use k, m, and g suffixes in Junos instead of typing out vast amounts of zeros
- Manually specifying an interface bandwidth
    - Set on a per interface basis under `set protocols ospf area interface metric (cost)`
        - Cost ranges from 1 to 65535
        - Setting a cost manually overrides any other calculated metrics

Example:*Setting the reference bandwidth to 10 Gbps*

```
[edit]
admin@J2300-1# edit protocols ospf

[edit protocols ospf]
admin@J2300-1# set reference-bandwidth 10g

[edit protocols ospf]
admin@J2300-1#
```

Example:*Setting the interface bandwidth to 192 Kbps*

```
[edit]
admin@J2300-1# edit interfaces fe-0/0/1 unit 12

[edit interfaces fe-0/0/1 unit 12]
admin@J2300-1# set bandwidth 192k

[edit interfaces fe-0/0/1 unit 12]
admin@J2300-1#
```

Example:*Manually specifying the OSPF interface cost to 1234*

```
[edit]
admin@J2300-1# edit protocols ospf area 0 interface fe-0/0/1.12

[edit protocols ospf area 0.0.0.0 interface fe-0/0/1.12]
admin@J2300-1# set metric 1234

[edit protocols ospf area 0.0.0.0 interface fe-0/0/1.12]
admin@J2300-1#
```

- Overload
    - Concept borrowed from IS-IS
    - An "overloaded" router will participate in OSPF, but will not be desirable for transit traffic
        - Examples of an overloaded router:
            - Router with high CPU usage
            - Router undergoing maintenance
            - Router that is restarting
            - Router that has congested links
        - Traffic destined to the router will still continue
    - Unlike IS-IS, there is no way for a router to actually tell everyone else not to consider it as part of the transit topology
        - OSPF fakes it, by setting the metric on all transit links in the Router LSA to the maximum metric (65535)
            - *Note: This method can also be used by IS-IS*
        - Configured with the `set protocols ospf overload command>`
            - Can set this permanently, or have it timeout after a period of time
                - To use a timeout `set protocols ospf overload timeout`
                - Time ranges from 60 to 1800 seconds after the OSPF instance started

Example:*Setting the overload bit to timeout after 60 seconds*

```
admin@J2300-1> edit
Entering configuration mode

[edit]
admin@J2300-1# set protocols ospf overload timeout 60

[edit]
admin@J2300-1#
```

- Examining link costs
    - Originated by the router
        - Easiest way is to use the `detail` flag with the operational mode command `show ospf interface`
        - Can also specify a specific interface

Example:*Examining link cost for a specific interface*

```
admin@J2300-1> show ospf interface fe-0/0/1.12 detail
Interface           State   Area            DR ID           BDR ID          Nbrs
fe-0/0/1.12             BDR     0.0.0.0         10.0.0.2        10.0.0.1           1
  Type: LAN, Address: 10.0.12.1, Mask: 255.255.255.0, MTU: 1496, Cost: 1234
  DR addr: 10.0.12.2, BDR addr: 10.0.12.1, Priority: 128
  Adj count: 1
  Hello: 10, Dead: 40, ReXmit: 5, Not Stub
  Auth type: None
```

```
 Protection type: None
 Topology default (ID 0) -> Cost: 1234

admin@J2300-1>
```

- Can also examine the contents of the LSDB
  - Can see the link costs of any router per configured area
- Note that `show route` and `show ospf route` commands show the overall cost to a destination -- not link costs

Exercise:OSPF Metrics

OSPF Packets

This section examines the OSPF packet structure. It's boring, but necessary.

- OSPF runs directly on the network layer (Layer 3) with it's own IP protocol number 89
  - Most OSPF packets only travel a single hop so the TTL is normally set to 1
    - Virtual-links are the exception
- Destination IP address is usually set to the AllSPFRouters multicast address of 224.0.0.5, AllDRRouters address of 225.0.0.6 or to the neighbors IP address
- OSPF can use normal IP fragmentation and reassembly mechanisms if needed
  - Most OSPF implementations try to avoid this by fragmenting large OSPF messages into several smaller individual messages
-
- Most implementations, including Junos, will use the IP precedence (DSCP) bits to allow for prioritization of OSPF messages over other traffic
  - Junos sets all OSPF packets to IP Precedence 110 (Internetwork control, or CS6 for DSCP) by default
- OSPF Packets Have a common 24 byte header

Common OSPF Packet Header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |     Type      |         Packet length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             AuType            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Fields are defined as follows:

Version
Version of the protocol

Type of Packet
Currently 5 Types of OSPF Packets Defined:

1 Hello
Used to build and maintain adjacencies between routers

2 Database Description
Used during adjacency formation so neighbors can initially synchronize their Link State Databases. Contains header information for all of the LSAs in the routers LSDB.

3 Link State Request
Used by routers to request a specific or updated LSA from a neighbor

4 Link State Update
Used by a router to advertise an LSA

5 Link State Acknowledgement
Used to acknowledge receipt of LSAs

Router ID
The Router ID of the source of the packet. A 32 bit number most often represented in dotted quad notation.

Area ID
The OSPF area for which the packet belongs, also a 32 bit number most often represented in dotted quad notation. A packet can belong only to a single area.

Checksum
A 16 bit CRC on the OSPF packet excluding the authentication fields.

AuType
Allows the recieving router to identfy what kind, if any, authtication is being used

0 Null
Authentication field is ignored by receiving routers

1 Simple
Authentication field contains a simple clear text string

2 Cryptographic
The entire OSPF packet is hashed together with a secret key using the MD5 algorithm

The Hello Packet

Note: This includes the common OSPF header.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |       1       |         Packet length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             AuType            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|                          Authentication                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Authentication                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Network Mask                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         HelloInterval         |    Options    |    Rtr Pri    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     RouterDeadInterval                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Designated Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Backup Designated Router                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Neighbor                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
```

- These packets are sent periodically on all interfaces (including virtual links) in order to establish and maintain adjacencies
- Multicast on networks with multicast or broadcast capability
  - Allows for dynamic discovery of neighbors

The Fields are defined as follows:

Network mask
: 32 bit network mask used for subnetting

HelloInterval
: Time in seconds between periodic hello messages

RtrPriority
: Used in DR elections on multi-access links to decide which router will be responsible for flooding LSAs on the link and which router will serve as a backup

DeadInterval
Time in seconds before a silent neighboring router is declared inaccessible over the network link

Designated Router
: The Router ID of the router on a multiaccess segment that is repsonsible for advertising the status of the link. This is set to all zeros (0.0.0.0) if there is no DR.

Backup Designated Router
: The Router ID of the backup router on a multiaccess segment. This is set to all zeros (0.0.0.0) if there is no BDR.

- In order for to neighboring routers to form a relationship several of the fields in the hello packet have to match
  - Network mask
  - HelloInterval
  - RouterDeadInterval
- Based on the contents of the options field, a router may reject forming an adjacency

Options Field

```
+--------------------------------------+
| DN | O | DC | EA | N/P | MC | E | MT |
+--------------------------------------+
```

The Options Field Bits are defined as follows:

DN-bit
: Used to prevent looping in L3 MPLS VPNs

O-bit
: Set if the router supports Opaque LSAs

DC-bit
: Set if the router supports demand circuits

EA-bit
: Set the router supports External-Attributes-LSAs

N/P-bit
: Set if the router supports Type-7 LSAs for NSSA support

MC-bit
: Describes whether IP multicast datagrams are forwarded

E-bit
: Describes the way AS-external-LSAs are flooded

MT-bit
: Describes a router's multi-topology capability

The Database Description Packet

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |       2       |         Packet length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             AuType            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Interface MTU         |    Options    |0|0|0|0|0|I|M|MS
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     DD sequence number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                                                             -+
|                                                               |
+-                      An LSA Header                          -+
|                                                               |
+-                                                             -+
|                                                               |
```

```
+-                                                               -+
|                                                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                                |
```

- Used when an adjacency is being initialized between two neighboring routers. They describe the contents of the link-state database.
    - Multiple packets may be used

Fields are defined as follows:

Interface MTU
: MTU of the interface on the network link

Options
: Same field as the options field used in the hello packet. Routers can use the options at this stage to determine if they need to forward certain LSAs due to the functionality of the neighboring router.

I-bit
: Initial bit
When set this is the first packet in the sequence of Database Description Packets

M-bit
: More bit.
When set it indicates that more Database Description Packets are to follow.

MS-bit
: The Master/Slave bit.
When set it indicates that the router will be the master during the Database exchange process.

DD sequence number
: Used to sequence all of the Database Description packets.

LSA Header(s)
: A list (possibly partial) of all of the router's link-state database headers.

- If the Interface MTU does not match during the database exchange, the exchange will not continue

The Link State Request Packet

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |       3       |         Packet length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             AuType            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          LS type                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Link State ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
```

- Sent when a router discovers that parts of it's link state database are missing or out of date<
    - Used to request pieces of a neighbors database
        - Specific LSAs are requested and identified using the LS type, Link Sate ID and the Advertising Router's Router Id to uniquely identify the LSA
    - May use multiple packets

The Link State Update Packet

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |       4       |         Packet length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             AuType            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            # LSAs                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                                                            +-+
|                             LSAs                              |
+-                                                            +-+
|                              ...                              |
```

- Implements the flooding of LSAs.
- Multiple LSAs may be included in a packet
- Multicast over capable networks
- LSAs must be acknowledged
    - Ensures that the flooding operation is reliable
    - Explicit acknowlegements occur by recieving a LSA Ack Packet back from the neighbor
    - LSAs can be implicitly acknowledged if the LSA is seen from the neighbor in another LSA Update packet
    - LSAs will be restransmitted if necessary
        - Retransmitted LSAs are unicast directly to the neighbor

Fields are defined as follows:

 # LSAs
The number of LSAs contained in the update
LSAs
A complete LSA

The Link State Acknowledgment packet

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |       5       |         Packet length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             AuType            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                                                             -+
|                                                               |
+-                         An LSA Header                       -+
|                                                               |
+-                                                             -+
|                                                               |
+-                                                             -+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
```

- Explicit acknowledgement that a router received a neighbors LSAs that were flooded
  - Multiple LSAs can be acknowledged in a single LSA Acknowledgement Packet
- LSA Acknowlegements are normally multicast to the AllSPFRouters or AllDRRouters addresses
  - Acks for a restransmitted update that was sent to the routers unicast address is sent back as a unicast as well
- Format is very similar to the Database Description packet
  - List of LSA headers

Adjacency Formation

This section examines the how OSPF neighbors form adjacencies.

- OSPF creates adjacency for the purpose of exchanging routing information
- Neighbors discover each other by periodically sending Hello packets out all of it's interfaces
  - Hello protocol ensures that each router can send and receive packets from each other -- bidirectional communication
    - OSPF will not allow packets to be forwarded over a unidirectional link
  - Ensures that neighboring routers agree in intervals for Hellos and declaring each other unreachable
  - Checks that there are no duplicate Router IDs
  - Used to detect and negotiate certain types of extensions
  - Hellos work differently on different types of networks
    - On broadcast networks routers periodicly mutlicast Hello Packets to AllSPFRouters
      - Contains the routers view of the current Designated Router, Backup Designated Router and the list of routers whose Hello packets have been seen
      - The Router Priority field is used for electing the DR and BDR
      - Hellos are sent every 10 seconds by default on Junos
    - On non-broadcast multiple-access networks (NBMA) it may be necessary to unicast hello's to all neighbors IP addresses
      - No dynamic discovery of neighbors possible, must configure all neighbors manually
      - The Router Priority field is used for electing the DR and BDR
      - Junos sends hellos every 120 seconds by default until active neigbhors are discovered
    - Routers will typically ignore or discard OSPF hellos that belong to a different area or the authentication does not match
    - Point-to-Point networks do not elect a DR. Hellos are multicast to the router on the other side of the link
      - Hellos are sent every 10 seconds by default on Junos
- Once two OSPF speakers have seen each other on a network they will attempt to synchronize their link state databases
  - Routers form a master/slave relationship for the initial synchronization
    - Master is the router with the higher Router ID
    - Master controls the database synchronization process
  - The "quickest" router to respond sends and empty DD packet with the first DD packet and sets the initial sequence number
    - The receiving router examines the Router ID in the empty DD packet and compares it to it's own
      - If the receiving router's Router ID is lower than the sending router's RID, it knows that it is the slave for the Database synchronization and responds with a DD packet listing all of it's LSAs
      - If the receiving router Router ID is higher that the send router's RID, it knows it is the master asserts itself by sending back and empty DD with it's own initial sequence number
    - Master manages the entire DD exchange

Finite State Machine for OSPF Adjacencies

OSPF Moves Through Several States before two neighbors become fully adjacent

DOWN
Router has not seen hellos from the neighbor since the last Dead interval.
ATTEMPT
Only on NBMA networks where neighbors have been manually configured. Indicates that the router is attempting to get a response from a neighbor.
INIT
Router has seen a hello packet from a neighbor but has not yet seen it's own Router ID in the neighbor list

2-WAY
: Router has seen a hello packet from a neighbor (or DR/BDR for broadcast networks) that has it's own Router ID in the neighbor field of the hello. On broadcast networks, routers will remain in this state for adjacencies other than with the DR and BDR. A router knows it has bidirectional communication with a neighbor at this point.

EXSTART
: Router is initializing initial sequence numbers for database exchange an maintenance. For broadcast networks without a DR or BDR, the election takes place at this state.

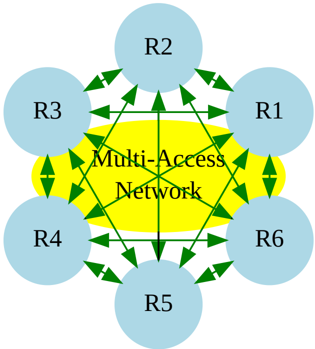
EXCHANGE
: Router's exchange Database Descriptors (DBDs) that list each router's LSAs. Rotuers figure out which LSAs they are missing, or have an outdated copy.

LOADING
: Routers are actively exchanging LSAs that were identified during the EXCHANGE phase.

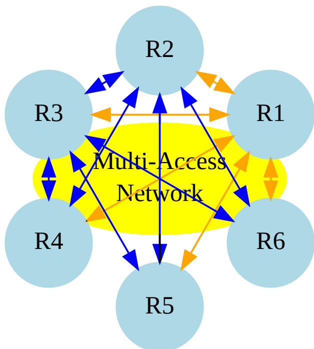
FULL
: Routers are fully adjacent

Designated Routers

- DR concept is a way to solve the n*(n-1)/2 adjacency problem on multi-access networks
  - Every router needs n-1 adjacencies:
    - Total number of adjacencies needed would be n*(n-1)/2. Adjacencies are shown in green across the multiaccess network in yellow.



  - Huge number of link state updates and acknowlegements sent over the network as every router keeps in sync with every other router on the subnet
  - Lots of duplicate routing information
    - Electing a router to act on behalf of the network cuts down on the number of adjacencies each router must maintain - The Designated Router
      - Also eliminates duplicate routing information
    - Also elect a backup Designated Router to provide for faster convergence should the Designated Router fail
  - All routers on the network only need to form adjacencies with the two DRs on the network
    - Cuts down on the total number of adjacencies
      - Can really cut down on the numbers of LSAs that need to be generated
      - Only the DR generates a LSA for the shared network segment
        - R1 is elected the DR, and R2 is elected the BDR. Adjacencies are shown in orange to the DR, and in blue to the BDR



      - Caveot: This can be counterproductive on broadcast networks with small numbers of neighbors in numbers of adjacencies
    - Every multi-access network has a designated router.
    - DR originates all advertisements for the network (Type 2 LSA - Network LSA)
      - Contains a list of all routers attached to the network
      - Becomes adjacent to all other routers on the network
      - Plays the central role in the database synchronization process with all routers on the network
    - DR is elected by the Hello Protocol
      - Configurable on a per interface basis
      - 8 bit field with 0 being the lowest priority, and 255 being the highst
        - A DR with a priority of 0 is ineligible to become a DR
        - Junos default is 128
    - DR election is non-deterministic
      - DR election commences if no DR or BDR exist on a network (DR and BDR fields are 0.0.0.0)
      - Router with the highest Router Priority in the hello packet assumes the DR role
      - Router with the second highest Router Priority assumes the BDR role
      - In the case of equal priorities, the router with the higher Router ID wins for both DR and BDR elections
      - No automatic DR premption
      - A DR is a DR until it dies (drops of the network)
    - DR maintains adjacencies to every other router on the multi-access network
    - DR multicasts Link State Updates to the AllSPFRouters address rather than sending separate packets over every adjacency
    - DR has more state to keep, so should be one of the more stable and powerful routers on the network
    - BDR is elected to make the transition to a new DR smoother on the network
    - BDR is also adjacent to all other routers on the network
    - BDR doesn't do much until the DR dies, at which time it assumes it's new role as the DR for the networks

- Since the BDR already has adjacencies to all the other routers on the network, the transition time is reduced from having to have a whole new election to find another DR
- Once a BDR assumes the DR role, a new election is held to determine a new DR

OSPF Authentication

- OSPF packet header has an authentication type field and a 64 bit data field for authentication data
    - Both fields are configurable on a per interface basis
- Authentication Types
    - Null Authentication
        - Authentication type is set to 0
        - Authentication field is not checked
            - Can contain any value
        - Default on Junos
    - Simple Authentication
        - Authentication type is set to 1
        - Authentication field contains a clear text password
            - Limited to 8 characters (64 bits)
        - Guards against routers inadvertently forming an adjacency
        - Configure with `set protocols ospf area interface authentication simple-password`
            - Junos will obscure the password in the config file, but it will still be transmitted as plain text over the network

Example: *Configuring a simple password for OSPF authentication*

```
admin@J2300-1> edit
Entering configuration mode

[edit]
admin@J2300-1# edit protocols ospf area 0 interface fe-0/0/1.12

[edit protocols ospf area 0.0.0.0 interface fe-0/0/1.12]
admin@J2300-1# set authentication simple-password OSPFpass

[edit protocols ospf area 0.0.0.0 interface fe-0/0/1.12]
admin@J2300-1# show
authentication {
    simple-password "$9$9WJOpBRSrKv8xwYmTzFAtWLxdYo"; ## SECRET-DATA
}

[edit protocols ospf area 0.0.0.0 interface fe-0/0/1.12]
admin@J2300-1#
```

Example: *Packet capture of OSPF packet with simple authentication*

```
admin@J2300-1> monitor interface fe-0/0/1 no-resolve detail
Address resolution is OFF.
Listening on fe-0/0/1, capture size 1514 bytes
08:12:44.695436 Out IP (tos 0xc0, ttl   1, id 6212, offset 0, flags [none], proto: OSPF (89), length: 64) 10.0.12.1 > 2
        Router-ID 10.0.0.1, Backbone Area, Authentication Type: simple (1)
        Simple text password: OSPFpass
        Options [External]
          Hello Timer 10s, Dead Timer 40s, Mask 255.255.255.0, Priority 128
          Designated Router 10.0.12.1
```

- Cyrptographic authentication
    - Requires a shared secret to be configured on all routers on a network link (subnet)
        - A digest is computed using the shared secret and the contents of the OSPF protocol packet
        - Digest is appended to the end of the packet
    - Authentication type is set to 2
    - Authentication field is redefined into the following fields:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              0                |    Key ID     | Auth Data Len |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 Cryptographic sequence number                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fields are defined as follows:

Key ID
Used to identify the algorithm and the key used
Auth Data Len
Length of the message digest in bytes that has been appended to the OSPF packet
Cryptographic Sequence Number
An incrementing sequence number used to protect against replay attacks

- Cyrptographic Authentication does **not** provide confidentiality on the OSPF packet contents!!!

Example: *Packet capture of OSPF packet with cryptographic authentication*

```
admin@J2300-1# run monitor traffic interface fe-0/0/1 detail no-resolve
Address resolution is OFF.
Listening on fe-0/0/1, capture size 1514 bytes

08:21:48.855167 Out IP (tos 0xc0, ttl   1, id 6339, offset 0, flags [none], proto: OSPF (89), length: 80) 10.0.12.1 > 2
        Router-ID 10.0.0.1, Backbone Area, Authentication Type: MD5 (2)
        Key-ID: 0, Auth-Length: 16, Crypto Sequence Number: 0x4e57579d
        Options [External]
          Hello Timer 10s, Dead Timer 40s, Mask 255.255.255.0, Priority 128
          Designated Router 10.0.12.1
```

- Keys can be given a lifetime
    - More than one key can be active at a time to smooth key expiration
    - Important that OSPF routers clocks are synchronized
- Junos only supports MD5 hashes on OSPFv2 ATT
    - **RFC 5709** adds support for HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512
- Configure with `set protocols ospf area interface authentication md5 key`
    - MD5 key can be from 1 to 16 characters in length
    - By default in Junos, keys have an infinite lifetime and are always transmitted
    - Can add a time to start using a key at a certain time by using the `start-time` option for a particular key

- Start time takes the format: YYYY-MM-DD.HH:MM
- Junos transmits cryptographically hashed OSPF packets with key-id's in the following priority
  1. Key with the latest start-time
  2. Key with the greater key ID

  Example: *Configuring OSPF for MD5 authentication with a key that is always valid, and a key to start using at the start of 2012*

```
[edit protocols ospf area 0.0.0.0 interface fe-0/0/1.12]
admin@J2300-1# set authentication md5 0 key AlwaysGoodKEY

[edit protocols ospf area 0.0.0.0 interface fe-0/0/1.12]
admin@J2300-1# set authentication md5 12 key NextYearKEY start-time 2012-01-01.00:00

[edit protocols ospf area 0.0.0.0 interface fe-0/0/1.12]
admin@J2300-1# show
metric 5;
authentication {
    md5 0 key "$9$cv1yrKbwgDHmM8kPQzAtREclMXJZj.mTp0ESrKXx"; ## SECRET-DATA
    md5 12 key "$9$Rqbclvx7Vb2aNdTFn/0O1RhyvL2gajiqEc" start-time "2012-1-1.00:00:00 +0000"; ## SECRET-DATA
}

[edit protocols ospf area 0.0.0.0 interface fe-0/0/1.12]
admin@J2300-1#
```

- Troubleshooting Authentication Issues

  - Enable traceoptions under the OSPF protocol
  - Use the `error` flag
    - Packets that do not pass the authentication check through mismatch type authentication type will generate the error:

```
OSPF packet ignored: authentication type mismatch (2) from 10.0.12.1
```

    - Packets where the key-id does not match will generate two errors:

```
OSPF packet ignored: authentication failure (missing key-id).
OSPF packet ignored: authentication failure from 10.0.12.1
```

    - Packets where the key does not match will also generate two errors:

```
OSPF packet ignored: authentication failure (bad cksum).
OSPF packet ignored: authentication failure from 10.0.12.1
```

    Example: *Configuring traceoptions to troubleshoot an authentication mismatch problem*

```
admin@J2300-2> show ospf interface
Interface           State   Area            DR ID           BDR ID          Nbrs
fe-0/0/1.12         DR      0.0.0.0         10.0.0.2        0.0.0.0            0
fe-0/0/1.24         DR      0.0.0.0         10.0.0.2        10.0.0.4           1
lo0.0               DR      0.0.0.0         10.0.0.2        0.0.0.0            0

admin@J2300-2> edit
Entering configuration mode

[edit]
admin@J2300-2# edit protocols ospf traceoptions

[edit protocols ospf traceoptions]
admin@J2300-2# set file ospf.log size 1m

[edit protocols ospf traceoptions]
admin@J2300-2# set flag error detail

[edit protocols ospf traceoptions]
admin@J2300-2# commit
commit complete

[edit protocols ospf traceoptions]
admin@J2300-2# run monitor start ospf.log

[edit protocols ospf traceoptions]
admin@J2300-2#
*** ospf.log ***
Aug 26 09:30:05.016062 OSPF packet ignored: authentication type mismatch (2) from 10.0.12.1
Aug 26 09:30:13.672798 OSPF packet ignored: authentication type mismatch (2) from 10.0.12.1
Aug 26 09:30:21.478337 OSPF packet ignored: authentication type mismatch (2) from 10.0.12.1
Aug 26 09:30:29.048099 OSPF packet ignored: authentication type mismatch (2) from 10.0.12.1

[edit protocols ospf traceoptions]
admin@J2300-2# run monitor stop
```

- Junos also supports IPSEC authentication for OSPF

  - Only transport mode is supported
  - Only bidirectional manual SAs are supported
  - Can use IPSEC to both authenticate and encrypt the entire OSPF packet
  - Can use IPSEC together with simple or MD5 authentication
  - Configure with `set protocols ospf area interface ipsec-sa`

Exercise:OSPF Authentication

Tuning Interfaces

This section examines changing parameters on interfaces running OSPF to change their behavior.

- All interface "tuning" is done on a per area basis in Junos under the `set protocols ospf area interface` level of the heirarchy
- Can change the hello interval with `set protocols ospf area interface hello-interval`
  - Must be the same for all devices on the network for adjacency formation
  - Ranges from 1 to 255 seconds
  - Default is 10 seconds for broadcast capable networks
  - Default is 120 seconds for nonbroadcast networks until an active neighbor comes up
    - Change this interval with `set protocols ospf area interface poll-interval`

        - Ranges from 1 to 65535 seconds
        - Once an adjacency is established, the hello interval is used
- Can change the dead interval with `set protocols ospf area interface dead-interval`
    - Can set range from 1 to 65535 seconds
    - By default this is 40 seconds on Junos (4 times the default hello interval)
    - Must be the same for all devices on the network for adjacency formation
    - Does not make sense to have dead interval less than the hello interval
- Configuring DR/BDR priorities
    - Set interface priority under the interface section per area `set protocols ospf area interface priority`
    - Default is 128
    - Can set from 0 to 255
    - Value of 0 signifies the router will never become a DR or BDR
- Controlling LSA Retransmission Interval
    - All LSAs that a router sends must be acknowledged -- reliable flooding
    - Junos starts a timer once a LSA is sent -- if not acknowledged by the time the timer expires the router will resend the LSA
        - Continues to restransmit LSAs until receiving an acknowledgment or the neighbor disappears
    - Can set the timer with `set protocols ospf area interface retransmit-interval`
        - Default is 5 seconds
        - Can specify from 1 to 65535 seconds
            - Should never specify below 3 seconds as Junos delays acknowledgement by up to 2 seconds
            - Allows for consolidations of acknowledgements
            - Avoids global network LSA/ACK floods
- Configuring the transmission delay
    - Before a link-state update packet is transmitted out an interface, the router increases the age of the packet by a default of 1 second
    - Aging link-state updates protects against the router receiving an update packet back that is younger than the original copy
    - On very slow links (low bandwidth, satellite shots) 1 second may not be long enough
    - Can configure the aging of the packet as it is transmitted with `set protocols ospf area interface retransmit-interval`
    - Transit delay can vary between 1 and 65535 seconds
- Overriding the default interface type
    - By default Junos chooses the interface type based upon the type of physical interface
        - Serial links are Point-to-Point
        - Ethernet links are broadcast
        - Nothing is defined as NBMA by default in Junos
    - Can override the default selection with `set protocols ospf area interface interface-type (nbma|p2mp|p2p)`
    - For NBMA networks specify neigbors explicitly by IP address with the `set protocols ospf area interface neighbor` statement
        - Hellos are unicast to configured neighbors
- Unnumbered Interfaces
    - Junos supports running OSPF only on point-to-point interfaces
        - Simply configure the interface to support the `inet` address family but don't assign any IP address
            - Junos will borrow the default address for the system, normally on lo0, for anything that needs an IP for identification over the unnumbered interface
            - Can use Ethernet interfaces that have been manually set to point-to-point for OSPF adjacencies
                - Instead of a assigning an address, use `set interface unit unnumbered` where the donor interface is an interface that the IP will be borrowed from
                    - Donor interface becomes the source address that all IP packets are generated from
- Demand circuits
    - Hello packets and LSAs are suppressed as soon as synchronization is achieved
    - Hello packets and LSAs resume once there is a change in topology
    - Only valid on point-to-point and point-to-multipoint links
    - Both sides of the link must support demand circuits
        - Negotiated during adjacency formation
    - Specify the circuit is a demand circuit with `set protocols ospf area interface demand-circuit`
- Flood Reduction
    - Normally LSAs that are generated by a router age out over time and are reflooded every 30 minutes
    - Can stop this behavior by forcing the router to set the DoNot Age bit in all LSAs that are self generated with `set protocols ospf area interface flood-reduction`
    - LSAs are only reflooded when the contents change
    - Reduces OSPF overhead in stable topologies

Troubleshooting Adjacencies

This section examines the how to troubleshoot OSPF adjacencies.

- View adjacencies with operational mode command `show ospf neighbor`
    - Can use the `area,interface` or `neighbor` to narrow results
    - Can use the detail and extensiveflags to vary the amount of information

Example: *Viewing neighbors on an interface*

```
admin@J2300-1> show ospf neighbor extensive interface fe-0/0/1.12
Address          Interface              State     ID               Pri  Dead
10.0.12.2        fe-0/0/1.12            Full      10.0.0.2         128    96
  Area 0.0.0.0, opt 0x42, DR 10.0.12.1, BDR 10.0.12.2
  Up 02:07:16, adjacent 02:07:16
  Topology default (ID 0) -> Bidirectional

admin@J2300-1>
```

- Can view number of adjacencies on each interface with operational command `show ospf interface`
    - Can use `area` or `interface` to narrow down results
    - Can use the `detail` and `extensive` flags to vary the amount of information
        - Shows information about timers, priorites, MTU, metrics as well

Example: *Examening an interface*

```
admin@J2300-1> show ospf interface fe-0/0/1.12 extensive
Interface           State   Area            DR ID           BDR ID          Nbrs
fe-0/0/1.12             DR      0.0.0.0         10.0.0.1        10.0.0.2            1
  Type: NBMA, Address: 10.0.12.1, Mask: 255.255.255.0, MTU: 1496, Cost: 5
  DR addr: 10.0.12.1, BDR addr: 10.0.12.2, Priority: 128
  Adj count: 1
  Hello: 30, Poll: 90, Dead: 120, ReXmit: 8, Not Stub
  Auth type: Password
  Protection type: None
  Topology default (ID 0) -> Cost: 5
```

```
admin@J2300-1>
```

- Clearing an adjacency can be accomplished with the operational mode command `clear ospf neighbor`
  - Can use the `area,interface` or `neighbor` clear specific neihbors
  - Be careful with `clear ospf neighbor` without any arguments as it will clear all the OSPF adjacencies on the router

Example: *Clearing all the neighbors!*

```
admin@J2300-1> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
10.0.12.2        fe-0/0/1.12            Full      10.0.0.2         128   111
10.0.13.3        fe-0/0/1.13            Full      10.0.0.3         128    39
10.101.0.10      fe-0/0/1.101           Full      10.10.10.10      128    31
10.102.0.10      fe-0/0/1.102           Full      10.10.10.10      128    39
10.1.111.101     fe-0/0/1.1001          Full      10.1.0.1         128    10

admin@J2300-1> clear ospf neighbor

admin@J2300-1> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
10.0.12.2        fe-0/0/1.12            Full      10.0.0.2         128   119
10.0.13.3        fe-0/0/1.13            Full      10.0.0.3         128    39
10.101.0.10      fe-0/0/1.101           Full      10.10.10.10      128    39
10.102.0.10      fe-0/0/1.102           Exchange  10.10.10.10      128    39
10.1.111.101     fe-0/0/1.1001          Full      10.1.0.1         128    10

admin@J2300-1>
```

- Hellos are essentially ignored if any of the following conditions do not match:
  - Authentication Type
  - Authentication keys
    - Authentication key ID if using cartographic hashes
  - Area ID
  - Hello Interval
  - Dead Interval
  - Interface Type (Point-to-Point, NBMA, etc)
  - No common IP subnets
    - Unless using an unnumbered interface
  - No Duplicate router ID
  - An adjacency does not even begin to form under any of these conditions
  - Can catch these problems with Junos by enabling OSPF traceoptions and looking at error conditions
    - Configured under `protocols ospf traceoptions`
    - Set a filename to log to with `file`
      - Can tweak the maximum file size, archiving behavior and permissions of the log file
      - File is stored in `/var/log` by default
      - Can view with operational commands:
        - show log >
        - file show
        - Or can drop to the shell and view using standard UN*X commands like cat, more, tail and vi
    - Can monitor in realtime with operational command `monitor start`
      - Stop with `monitor stop`
    - Tracing does not start until "committed"
      - Tracing will continue until the traceoptions are deleted
      - Deleting the traceoptions in the configuration does not delete any of the log files

  Example: *Configuring traceoptions to look for OSPF error conditions*

```
admin@J2300-1> edit
Entering configuration mode

[edit]
admin@J2300-1# edit protocols ospf traceoptions

[edit protocols ospf traceoptions]
admin@J2300-1# set file ospf.log

[edit protocols ospf traceoptions]
admin@J2300-1# set flag error detail

[edit protocols ospf traceoptions]
admin@J2300-1# commit
commit complete

[edit protocols ospf traceoptions]
admin@J2300-1#
```

  Example: *Monitoring the ospf.log file in realtime*

```
admin@J2300-1> monitor start ospf.log

admin@J2300-1>
*** ospf.log ***
Aug 30 08:45:48.098450 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:45:49.052569 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:45:50.002191 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:45:50.950214 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:45:51.886726 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0

monitor stop

admin@J2300-1>
```

  Example: *Examening the ospf.log file*

```
admin@J2300-1> show log ospf.log
Aug 30 08:37:06 trace_on: Tracing to "/var/log/ospf.log" started
Aug 30 08:37:06.590718 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:37:07.533255 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:37:08.522669 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:37:09.356771 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:37:10.170148 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:37:11.016943 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:37:11.825845 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:37:12.589728 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
```

```
Aug 30 08:37:13.543683 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:37:14.388735 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0
Aug 30 08:37:15.160456 OSPF packet ignored: configuration mismatch from 10.1.111.101 on intf fxp1.1001 area 0.0.0

admin@J2300-1>
```

- If an adjacency gets stuck in the ExStart phase, there is most likely a MTU mismatch
    - Database Exchange checks the MTU size to see what size packet the neighbors can send between themselves
- Fix by setting the Layer 2 MTU on the interface with `set interface mtu`
- Or, fix by setting the Layer 3 MTU for the IPv4 address family with `set interface unit family inet mtu`
- Can view the interface MTUs with the `show interface` operational mode command

Example: *ExStart Debugging*

```
admin@srx100-1> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
1.0.0.2          fe-0/0/0.0             ExStart   2.2.2.2          128    37

admin@srx100-1> edit
Entering configuration mode

[edit]
admin@srx100-1# top
warning: already at top of configuration; use 'exit' to exit

[edit]
admin@srx100-1# edit protocols ospf traceoptions

[edit protocols ospf traceoptions]
admin@srx100-1# set file ospf.log

[edit protocols ospf traceoptions]
admin@srx100-1# set flag error detail

[edit protocols ospf traceoptions]
admin@srx100-1# commit
commit complete

[edit protocols ospf traceoptions]
admin@srx100-1# run monitor start ospf.log

[edit protocols ospf traceoptions]
admin@srx100-1#
*** ospf.log ***
Aug 30 11:03:40.323386 OSPF packet ignored: MTU mismatch from 1.0.0.2 on intf fe-0/0/0.0 area 0.0.0.0
Aug 30 11:03:42.741430 OSPF packet ignored: no matching interface from 10.0.100.76, IFL 73
Aug 30 11:03:44.690793 OSPF packet ignored: MTU mismatch from 1.0.0.2 on intf fe-0/0/0.0 area 0.0.0.0
run monitor stop Aug 30 11:03:49.673924 OSPF packet ignored: MTU mismatch from 1.0.0.2 on intf fe-0/0/0.0 area 0.0.0.0
Aug 30 11:03:52.741840 OSPF packet ignored: no matching interface from 10.0.100.76, IFL 73
ospf.log

[edit protocols ospf traceoptions]
admin@srx100-1#
```

- Keep in mind, on a broadcast network, routers will only form a Full neighbor relationship with the DR and BDR
    - Adjacencies with Non-DR routers will remain in the 2way state

Exercise:Interface Tuning and Troubleshooting

Router ID

- A 32-bit number that uniquely identifies a router -- Normally written in dotted quad notation - Used as a tie breaker in several routing protocol and routing decisions -- OSPF uses it for a tie breaker in DR/BDR elections - Can't have any duplicates -- OSPF uses it as part of the LSA identifier - Changing the router-id in OSPF causes the router to:
    1. Flush all of it's LSAs
    2. Drop all of it's adjacencies
- Junos automatically uses the IP address of the first interface that comes up as the router-id
    - Normally will be a loopback interface if one is defined.
    - Junos will use the IP defined or identified as the primary address of the interface
    - Best practice is to hardcode the router-id to avoid any inconsistencies or unexpected behavior
        - Set for the entir router with `set router-id` under the `routing-options` level of the Junos hierarchy

Example: *Setting the Router ID*

```
user@Router> edit
Entering configuration mode

[edit]
user@Router# set routing-options router-id 1.2.3.4

[edit]
user@Router# show routing-options
router-id 1.2.3.4;

[edit]
```

Exercise:DR Battle

Link State Advertisements (LSAs)

- LSAs make up the "guts" of LS Update Packets, and the headers are used in LSA Requests and Acknowlegements
- Contain all the real data used to build and maintain a topology
- All LSAs have a common 20 byte header
    - Has enough info to uniquely identify the LSA
    - Has enough info to identify which is the most recent LSA

Common LSA Header

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
 |            LS age             |    Options    |    LS type    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Link State ID                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                     Advertising Router                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                     LS sequence number                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         LS checksum           |             length            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Fields are defined as follows:

LS age
Time in seconds since the LSA was originated
Options
Same options as in the hello packet to describe any optional capabilities
LS type
Type of LSA, each LSA type has a separate format
Link State ID
Identifies the portion of the topology that is being described
Will differ based upon the LSA type
Advertising Router
The Router ID of the router that originated the LSA
LS sequence number
Used to detect old or duplicate LSAs
LS checksum
Checksum of the whole LSA including the header, but not the LS age
Length
Length in bytes of the LSA including the header

- LSA Aging
  - LSAs are aged in increments of seconds
    - 16 bit integer
  - Age is set to zero when the LSA is originated
  - Incremented by the Transit Delay time of the interface every time the LSA is flooded
    - Default of 1 second
  - LSAs also age in the LSDB of every router
  - Once a LSA is 3600 seconds old (1 hour) the LSA is deleted from the LSDB
    - LSAs need to be refreshed periodically to keep them from aging out
      - Junos refreshes LSAs every 50 minutes
    - Routers can prematurely age LSAs they originate to flush them from the OSPF domain
      - Done when the sequence number is about to wrap
      - Done for LSAs that carry external routing information where the destination is no longer reachable
    - Can configure flood reduction to keep LSAs from aging out -- good for stable networks
  - Can view LSA refresh times with operational mode command `show ospf overview`
  - Can view LSA ages by examining the LSDB

Type 1 LSA - The Router LSA

Flooding scope:*Area*

- Each OSPF router originates a router LSA for each area it belongs to
  - Describes the state and cost of all of the routers links into the area
  - All must be contained in a single router LSA
- Flooded throughout a single area only

Type 1 - Router LSA *Including the common LSA header*

```
     0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |            LS age             |     Options   |       1       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Link State ID                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                     Advertising Router                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                     LS sequence number                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         LS checksum           |             length            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  0  Nt|W|V|E|B|        0      |            # links            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                          Link ID                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                         Link Data                             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |     # TOS     |            metric             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                              ...                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |      TOS      |        0      |          TOS  metric          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                          Link ID                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                         Link Data                             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                              ...                              |
```

The Fields are defined and populated as follows:

Link State ID of the common LSA header is set to the Router ID of the originating router.

Unamed Bitfield
Directly following the common LSA header is an unamed bitfield which is used to describe some of the capabilities of a router

B bit
Signifies that the router is an Area Border Router (ABR)
Set whenever a router belongs to two or more areas
Bit position 1, 0x1

E bit
Signifies that the router is an Autonomous System Boundary Router (ASBR)
E signifies External
Set whenever a router will inject external routing information into OSPF (redistribution)
Bit position 2, 0x2

V bit
When set signifies the router is an endpoint of a virtual link
Bit position 3, 0x4

W bit
Signifies the router is a wild-card multicast receiver
Bit position 4, 0x8

Nt bit
Signifies that the router will translate Type7 LSAs into Type5 LSAs (redistribution)
Bit position 5, 0x10

 # links
total number of links described in the LSA

Link ID
Identifies the object that this router link connects to
Value depends on the link Type
When two routers connect to the same link the Link ID provides a key for finding the neighbors LSA

Type
The type of link connecting to the router
Four types of connections are defined
1. Point-to-point
2. Transit
3. Stub
4. Virtual link

Link Data
The information about the link such as addressing, neigbhors, etc.
Contents varies depending on the Link Type

 # TOS
Number of different Type of Service metrics for TOS routing
Not implemented in Junos

Metric
A value from 1 to 65535
Higher value indicates a more costly route

TOS
Type of Service type
Not implemented in Junos

TOS Metric
Type of Service metric
Not implemented in Junos

Link Types

The link types in the Router LSA are defined as follows:

1 - Point-to-Point
Only two endpoints: serial interfaces, T1, E1, SONET/SDH
May or may not have broadcast capability
Link ID is set to the neighboring routers router ID
- Link Data contains the interfaces IP address
  - For unnumbered interfaces the Link Data contains the interface's ifIndex value
- Each router on the link will also advertise a Type 3 link (stub network) containing the IP network value and subnet mask in the Link Data

2 - Transit
A network that may contain more than two endpoints, Ethernet, Frame Relay, FDDI
May or may not have broadcast capability
Will have a DR and BDR election
If the router on a transit network has a full adjacency with a DR (or if it is the DR) the Link ID is set to the DR routers interface IP address
- The link data contains the routers own IP address on the transit network

3 - Stub
A stub network that can be treated as if it terminates on the router originating the LSA
Link ID is set to the IP network number
Link Data is set to the IP subnet mask

4 - Virtual Link
Used to tunnel routing information across areas
Link ID is set to the Router ID of a virtual neighbor
Link Data is set to the IP interface address of the interface associted with the virtual link

Examining Router LSAs in Junos

- Use the operational mode command `show ospf database router`
  - Shows all router LSAs in all areas by default
  - Can use `area` , `advertising-router` , `lsa-id` to narrow down the output results
  - Can use the `detail` and `extensive` flags to tailor output level of detail
  - Can use the `summary` flag for an overview of the LSAs

Example: *Viewing all of the information for the router LSA for the 10.0.0.6 router*

```
admin@J2300-1> show ospf database router lsa-id 10.0.0.6 extensive

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.0.0.6         10.0.0.6         0x80000017  1392  0x22 0x4e4d  72
  bits 0x1, link count 4
  id 10.0.46.4, data 10.0.46.6, Type Transit (2)
    Topology count: 0, Default metric: 5
  id 10.0.0.5, data 10.0.56.6, Type PointToPoint (1)
    Topology count: 0, Default metric: 5
  id 10.0.56.0, data 255.255.255.0, Type Stub (3)
    Topology count: 0, Default metric: 5
  id 10.0.0.6, data 255.255.255.255, Type Stub (3)
    Topology count: 0, Default metric: 0
  Topology default (ID 0)
```

```
    Type: PointToPoint, Node ID: 10.0.0.5
      Metric: 5, Bidirectional
    Type: Transit, Node ID: 10.0.46.4
      Metric: 5, Bidirectional
  Aging timer 00:36:48
  Installed 00:23:07 ago, expires in 00:36:48, sent 00:23:07 ago
  Last changed 01:23:20 ago, Change count: 1

admin@J2300-1>
```

Decoding the bits in the Router LSA

- Bit field is unnamed, but defined in **RFC 3101**
- Junos displays this as a hex value in the form 0x?
  - Corresponds with the bit positions mentioned above
- Can use this to determine if a router is an ABR, ASBR, NSSA ABR, has a virtual link (or any combination)

Exercise:Router LSAs

Type 2 LSA - The Network LSA

Flooding scope:*Area*

- A network-LSA is originated for each transit network
  - Broadcast and NBMA networks which supports two or more routers
- The DR is the sole originator of the Network LSA
  - DR only originates a network LSA if it has at least one full adjacency with another router
  - Advertises IP addressing info for the transit network
  - Lists all of the routers attached to the network segment
    - Including the DR and BDR
- Note that there is no metric field
  - Distance from the network to all attached routers is zero
  - Costs for each router are advertised in each routers Router LSA as a cost to a transit link
- Flooded within the area that the link belongs

Type 2 - Network LSA *Including the common LSA header*

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LS age             |      Options  |      2        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Link State ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     LS sequence number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         LS checksum           |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Network Mask                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Attached Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
```

The Fields are defined and populated as follows:

- Link State ID of the common LSA header is set to the interface IP address of the DR on the network segment.
- The Advertising Router in the common LSA header is set to the Router ID of the DR

Network Mask
: Subnet mask for the network

Attached Router
: Router IDs of all routers on the attached network that are fully adjacent to the DR
  Includes the DR itself

Examining Network LSAs in Junos

- Use the operational mode command `show ospf database network`
  - Shows all Network LSAs in all areas by default
  - Can use `area` , `advertising-router` , `lsa-id` to narrow down the output results
  - Can use the `detail` and `extensive` flags to tailor output level of detail
  - Can use the `summary` flag for an overview of the LSAs

Example: *Viewing all of the information for the network LSAs in area 0.0.0.0*

```
admin@J2300-1> show ospf database network detail area 0

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Network  10.0.12.2        10.0.0.2         0x80000012  2552  0x22 0xd21b  32
  mask 255.255.255.0
  attached router 10.0.0.2
  attached router 10.0.0.1
  Topology default (ID 0)
    Type: Transit, Node ID: 10.0.0.1
      Metric: 0, Bidirectional
    Type: Transit, Node ID: 10.0.0.2
      Metric: 0, Bidirectional
Network  10.0.34.4        10.0.0.4         0x80000008  1027  0x22 0x4d5   32
  mask 255.255.255.0
  attached router 10.0.0.4
  attached router 10.0.0.3
  Topology default (ID 0)
    Type: Transit, Node ID: 10.0.0.3
      Metric: 0, Bidirectional
    Type: Transit, Node ID: 10.0.0.4
```

```
      Metric: 0, Bidirectional
Network  10.0.35.5       10.0.0.5         0x80000008   636  0x22 0xf2e2  32
  mask 255.255.255.0
  attached router 10.0.0.5
  attached router 10.0.0.3
  Topology default (ID 0)
    Type: Transit, Node ID: 10.0.0.3
      Metric: 0, Bidirectional
    Type: Transit, Node ID: 10.0.0.5
      Metric: 0, Bidirectional
Network  10.0.46.4       10.0.0.4         0x80000009  1185  0x22 0xa722  32
  mask 255.255.255.0
  attached router 10.0.0.4
  attached router 10.0.0.6
  Topology default (ID 0)
    Type: Transit, Node ID: 10.0.0.6
      Metric: 0, Bidirectional
    Type: Transit, Node ID: 10.0.0.4
      Metric: 0, Bidirectional

admin@J2300-1>
```

- On point-to-point Ethernet links, Network LSAs can be eliminated by defining the interface type as Point-to-Point rather than leaving it to the defaults
  - Network LSAs are eliminated for the network segment
  - Each attached router sends all of the information to describe the link in the Router LSAs

Exercise:Network LSAs

Type 3 LSA - The Network Summary LSA

Flooding scope:*Area*

- A Type 3 Network Summary LSAs are originated by Area Border Routers (ABR)
- Describe Inter-Area destinations
- ABR is a router that has an interface in more than one area

Type 2 - Network LSA *Including the common LSA header*

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |            LS age             |     Options   |       3       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                        Link State ID                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     Advertising Router                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     LS sequence number                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         LS checksum           |             length            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Network Mask                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |      0        |                  metric                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |     TOS       |                TOS  metric                    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                              ...                              |
```

The Fields are defined and populated as follows:

- Link State ID of the common LSA header is set to the interface IP network number of the subnet being described.
- The Advertising Router in the common LSA header is set to the Router ID of the ABR advertising the Summary LSA

Network Mask
Subnet mask for the network

- A Default Route can be represented in OSPF by a Type 3 Network Summary LSA where both the Link ID and the Subnet mask are all zeros

Examining Network Summary LSAs in Junos

- Use the operational mode command `show ospf database netsummary`
  - Shows all Network Summary LSAs in all areas by default
  - Can use `area` , `advertising-router` , `lsa-id` to narrow down the output results
  - Can use the `detail` and `extensive` flags to tailor output level of detail
  - Can use the `summary` flag for an overview of the LSAs

Example: *Viewing network summary LSAs of the information for the network LSAs in area 0.0.0.0*

```
admin@J2300-1> show ospf database netsummary area 0

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Summary *10.1.14.0       10.0.0.1         0x80000005   619  0x22 0x5ddf  28
Summary  10.1.14.0       10.0.0.3         0x80000006   426  0x22 0x7b6e  28
Summary *10.1.23.0       10.0.0.1         0x80000005   562  0x22 0xc25d  28
Summary  10.1.23.0       10.0.0.3         0x80000006   343  0x22 0x4fa5  28
Summary *10.1.34.0       10.0.0.1         0x80000005   505  0x22 0xe43a  28
Summary  10.1.34.0       10.0.0.3         0x80000006   261  0x22 0x3aa5  28
Summary *10.1.111.0      10.0.0.1         0x80000007   390  0x22 0xc51e  28
Summary  10.1.111.0      10.0.0.3         0x80000006   178  0x22 0xb0cd  28
Summary *10.1.123.0      10.0.0.1         0x80000005   448  0x22 0xd6da  28
Summary  10.1.123.0      10.0.0.3         0x80000009  1252  0x22 0x9403  28
Summary  10.2.57.0       10.0.0.2         0x80000006   752  0x22 0x3692  28
Summary  10.2.57.0       10.0.0.4         0x80000006    98  0x22 0x2a9c  28
Summary  10.2.58.0       10.0.0.2         0x80000006   670  0x22 0xc60b  28
Summary  10.2.58.0       10.0.0.4         0x80000005  1255  0x22 0x8537  28
```

```
Summary  10.2.67.0        10.0.0.2         0x80000006   588  0x22 0x2c88  28
Summary  10.2.67.0        10.0.0.4         0x80000005  1173  0x22 0x596e  28
Summary  10.2.128.0       10.0.0.2         0x8000000b   425  0x22 0x533d  28
Summary  10.2.128.0       10.0.0.4         0x80000005  1090  0x22 0xe487  28
Summary  10.2.146.0       10.0.0.2         0x80000006   507  0x22 0x2833  28
Summary  10.2.146.0       10.0.0.4         0x80000009  1008  0x22 0x84f9  28
Summary  10.3.91.0        10.0.0.5         0x80000006   700  0x22 0x3c70  28
Summary  10.3.91.0        10.0.0.6         0x80000004  1117  0x22 0x396   28
Summary  10.3.120.0       10.0.0.5         0x80000006   620  0x22 0xc4b6  28
Summary  10.3.120.0       10.0.0.6         0x80000004  1038  0x22 0xf996  28
Summary  10.3.121.0       10.0.0.5         0x80000006   540  0x22 0x552f  28
Summary  10.3.121.0       10.0.0.6         0x80000004   960  0x22 0x5332  28
Summary  10.3.159.0       10.0.0.5         0x8000000a   380  0x22 0xe08d  28
Summary  10.3.159.0       10.0.0.6         0x80000004   881  0x22 0x78d2  28
Summary  10.3.161.0       10.0.0.5         0x80000006   460  0x22 0x64e3  28
Summary  10.3.161.0       10.0.0.6         0x80000009   802  0x22 0xc6a5  28
Summary *10.30.0.0        10.0.0.1         0x80000005   733  0x22 0xb8fe  28
Summary  10.30.0.0        10.0.0.2         0x80000006   855  0x22 0x1465  28
Summary  10.30.0.0        10.0.0.3         0x80000006   818  0x22 0xe6a   28
Summary  10.30.0.0        10.0.0.4         0x80000006   532  0x22 0x86f   28
Summary  10.30.0.0        10.0.0.5         0x80000006   860  0x22 0x274   28
Summary  10.30.0.0        10.0.0.6         0x80000006   487  0x22 0xfb79  28
Summary *10.101.0.0       10.0.0.1         0x80000008   276  0x22 0xd815  28
Summary  10.101.0.0       10.0.0.2         0x80000009   343  0x22 0x347b  28
Summary  10.101.0.0       10.0.0.3         0x80000008  1086  0x22 0x307f  28
Summary  10.101.0.0       10.0.0.4         0x80000008   925  0x22 0x2a84  28
Summary  10.101.0.0       10.0.0.5         0x80000009   300  0x22 0x228a  28
Summary  10.101.0.0       10.0.0.6         0x80000008   723  0x22 0x1e8e  28
Summary *10.102.0.0       10.0.0.1         0x80000007   333  0x22 0xce1f  28
Summary  10.102.0.0       10.0.0.2         0x80000009   261  0x22 0x2886  28
Summary  10.102.0.0       10.0.0.3         0x80000009   674  0x22 0x228b  28
Summary  10.102.0.0       10.0.0.4         0x80000008   842  0x22 0x1e8f  28
Summary  10.102.0.0       10.0.0.5         0x80000009   780  0x22 0x1695  28
Summary  10.102.0.0       10.0.0.6         0x80000008   645  0x22 0x1299  28

admin@J2300-1>
```

Example: *Viewing network summary LSAs for the 10.102.0.0 network in the backbone area*

```
admin@J2300-1> show ospf database netsummary area 0 lsa-id 10.102.0.0 detail

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Summary *10.102.0.0       10.0.0.1         0x80000007   502  0x22 0xce1f  28
  mask 255.255.255.0
  Topology default (ID 0) -> Metric: 2500
Summary  10.102.0.0       10.0.0.2         0x80000009   430  0x22 0x2886  28
  mask 255.255.255.0
  Topology default (ID 0) -> Metric: 10
Summary  10.102.0.0       10.0.0.3         0x80000009   843  0x22 0x228b  28
  mask 255.255.255.0
  Topology default (ID 0) -> Metric: 10
Summary  10.102.0.0       10.0.0.4         0x80000008  1011  0x22 0x1e8f  28
  mask 255.255.255.0
  Topology default (ID 0) -> Metric: 10
Summary  10.102.0.0       10.0.0.5         0x80000009   949  0x22 0x1695  28
  mask 255.255.255.0
  Topology default (ID 0) -> Metric: 10
Summary  10.102.0.0       10.0.0.6         0x80000008   814  0x22 0x1299  28
  mask 255.255.255.0
  Topology default (ID 0) -> Metric: 10

admin@J2300-1>
```

Generating Network Summary LSAs

- Network Summary LSAs are originated by ABRs
  - ABRs advertise only intra-area routes into the backbone
  - ABRs advertise both intra-area and inter-area routes into all other areas
  - ABRs generate a Type 3 LSA for:
    - Stub links in Router LSAs from within an area
    - Subnets described in Network LSAs from within an area
  - ABRs will not advertise Network Summary LSAs for:
    - Routes into the area from which the route originated
      - Split Horizon Logic
        - Protects against potential loops
        - Eliminates unnessary LSAs that duplicate other routing information
    - Routes where the cost will exceed the maximum metric allowed
- If a network that is being described in a Network Summary LSA becomes unreachable
  - The Network Summary LSA is prematurely aged so it can be flushed from the rest of the routing domain

Consolidating Summary LSAs

- By default every IP subnet listed in every Router LSA or Network LSA will be translated into a separate Network Summary LSA
- Can summarize groups or ranges of contiguous addresses into a single advertisement
  - Careful IP address allocation can make this very efficient
- Configured in Junos with the `set protocols ospf area area-range`
  - Controls what routes are advertised out of an area
  - The metric associated with the LSA will inherit the highest metric of the contributing routes that are being summarized
    - Can override this metric with the `override-metric` flag
  - Can block networks from having Network Summary LSAs created by using the `restrict` flag
    - Type 3 LSAs will not be generated by the ABR for any networks that match
- Be wary that when routes are summarized, some reachability information is being lost, so routing inefficiencies can be introduced
- Junos will create an OSPF route with a next hop of discard, and the maximum metric for OSPF on the ABR for any area ranges that are being summarized
  - OSPF aggregate route
- Keep in mind, ABRs can only summarize routes that are contained in Type 1 and Type 2 LSAs
  - Type 3 LSAs cannot be summarized for inter-area routes
  - Summarization needs to be done on the ABR from the originating area

  Example: *Configure an ABR to coalesce all of the addresses in the 10.0.0.0/8 subnet into a single Network Summary LSA from area 0.0.0.1*

```
admin@J2300-1> show ospf database advertising-router 10.0.0.1 netsummary area 0

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Summary *10.1.14.0        10.0.0.1         0x80000001    29  0x22 0x65db  28
Summary *10.1.23.0        10.0.0.1         0x80000001    29  0x22 0xca59  28
Summary *10.1.34.0        10.0.0.1         0x80000001    29  0x22 0xec36  28
Summary *10.1.111.0       10.0.0.1         0x80000001    29  0x22 0xd118  28
Summary *10.1.123.0       10.0.0.1         0x80000001    29  0x22 0xded6  28
Summary *10.30.0.0        10.0.0.1         0x80000006   951  0x22 0xb6ff  28
Summary *10.101.0.0       10.0.0.1         0x80000009   458  0x22 0xd616  28
Summary *10.102.0.0       10.0.0.1         0x80000008   519  0x22 0xcc20  28

admin@J2300-1> edit
Entering configuration mode

[edit]
admin@J2300-1# edit protocols ospf area 1

[edit protocols ospf area 0.0.0.1]
admin@J2300-1# set area-range 10/8

[edit protocols ospf area 0.0.0.1]
admin@J2300-1# commit and-quit
commit complete
Exiting configuration mode

admin@J2300-1> show ospf database advertising-router 10.0.0.1 netsummary area 0

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Summary *10.0.0.0         10.0.0.1         0x80000001    10  0x22 0x39f8  28
Summary *10.30.0.0        10.0.0.1         0x80000006  1051  0x22 0xb6ff  28
Summary *10.101.0.0       10.0.0.1         0x80000009   558  0x22 0xd616  28
Summary *10.102.0.0       10.0.0.1         0x80000008   619  0x22 0xcc20  28

admin@J2300-1>
```

Limiting Network Summaries

- By default, and by OSPF spec, ABRs flood network summaries to all areas based upon Router LSAs and Network LSAs and floods all Network Summary LSAs across area boundaries
  - Can control how Network Summary LSAs are distributed and generated
    - LSA Manipulation - can be dangerous!
- Can configure an export policy to specify which routes to create a Network Summary LSAs for with `network-summary-export` configured under the area
  - Default is not to create any "extra" Network Summary LSAs
  - Can configure an import policy to specify which routes from an area are used to generate Network Summary LSAs into other areas with `network-summary-import` configured under the area
  - Default is to accept all OSPF routes
- Can really screw up the LSDB if care is not taken!

Exercise:Network Summary LSAs

Type 4 LSA - The Autonomous System Border Router Summary LSARouter

Flooding scope:*Area*

- Type 4 ASBR LSAs are originated by Area Border Routers (ABR)

  - Describe the location of a router that is injecting external routes (redistribution)
  - ASBR is a router that has is redistributing routes from one routing protocol into OSPF
    - Marked as external to describe loss of routing information
    - Identification for loop protection purposes
  - Has the same LSA format as a Type 3 LSA
  - Used to help in metric calculations for external routes

Type 2 - ASBR Summary LSA*Including the common LSA header*

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |            LS age             |     Options   |       4       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                        Link State ID                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     Advertising Router                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     LS sequence number                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         LS checksum           |             length            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Network Mask                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |      0        |                  metric                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |     TOS       |                TOS  metric                    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                              ...                              |
```

The Fields are defined and populated as follows:

- Link State ID of the common LSA header is set to the Router ID of the ASBR.
- The Advertising Router in the common LSA header is set to the Router ID of the ABR advertising the ASBR Summary LSA

Network Mask
Set to all zeros
Metric

Cost to reach the ASBR

Examining ASBR Summary LSAs in Junos

- Use the operational mode command `show ospf database asbrsummary`
  - Shows all Network Summary LSAs in all areas by default
  - Can use `area` , `advertising-router` , `lsa-id` to narrow down the output results
  - Can use the `detail` and `extensive` flags to tailor output level of detail
  - Can use the `summary` flag for an overview of the LSAs

Example: *Viewing a specfiic ASBR summary LSAs in detail*

```
admin@J2300-1> show ospf database asbrsummary area 0 lsa-id 10.10.10.10 advertising-router 10.0.0.1 detail

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
ASBRSum *10.10.10.10      10.0.0.1         0x80000011  1115  0x22 0x2efc  28
  mask 0.0.0.0
  Topology default (ID 0) -> Metric: 2500

admin@J2300-1>
[edit]
```

Type 5 LSA - The AS-external LSA

Flooding scope:*OSPF Domain*

- Type 5 ASBR LSAs are originated by ASBRs
  - Describe the destinations external to the AS (OSPF domain)
  - Routes being redistributed from other protocols (RIP, static, Connected, BGP, IS-IS, other OSPF instances, etc)

Type 2 - AS-external LSA*Including the common LSA header*

```
      0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            LS age             |     Options   |      5        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Link State ID                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Advertising Router                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     LS sequence number                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         LS checksum           |             length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Network Mask                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |E|     0       |                  metric                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Forwarding address                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      External Route Tag                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |E|    TOS      |                TOS  metric                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Forwarding address                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      External Route Tag                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              ...                              |
```

The Fields are defined and populated as follows:

- Link State ID of the common LSA header is set to IP address of the network being advertisedthe .
- The Advertising Router in the common LSA header is set to the Router ID of the ASBR

Network Mask
: Set to subnet mask of the advertising network

E bit
: Specifies the type of external metric
- Type 1
  - E bit is not set
  - Indicates that the metric indicated in the metric field is the same units as the metrics used in the LSDB
  - The total cost of the route should be equal to the metric contained in the metric field, plus the cost to reach the ASBR from the router calculating the total cost of the path.
    - Need a ASBR Summary LSA to complete this calculation
- Type 2
  - E bit is set
  - Indicates that the metric is not compatible with the LSDB link costs and should be used as the sole metric for the external route.

Metric
: Cost of the route
: Interpretation depends on how the E bit is set

Forwarding address
: Data traffic for the advertised destination will be forwarded to the address set here
: If the address is set to 0.0.0.0, traffic is forwarded to the originator of the LSA (ASBR)

External Route Tag
: A 32 bit field used for administrative purposes.
- Use is analogous to BGP communities
- Administrative tag that has meaning only within the administrative domain
- Can be used for policy actions, mark of redistribution, etc.

Examining External LSAs in Junos

- Use the operational mode command `show ospf database external`

- Shows all Network Summary LSAs in all areas by default
- Can use the `detail` and `extensive` flags to tailor output level of detail
- Can use the `summary` flag for an overview of the LSAs

Example: *Viewing a specfiic external LSA in detail*

```
    admin@J2300-1> ...atabase external advertising-router 10.1.0.1 extensive
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   10.1.0.1         10.1.0.1         0x80000008  1933  0x22 0xf1a1  36
  mask 255.255.255.255
  Topology default (ID 0)
    Type: 2, Metric: 0, Fwd addr: 0.0.0.0, Tag: 0.0.0.0
  Aging timer 00:27:47
  Installed 00:32:10 ago, expires in 00:27:47, sent 00:32:08 ago
  Last changed 05:48:52 ago, Change count: 1
```

- By default in Junos, no External LSAs are generated
  - Must use an `export` policy to have a router generate them
    - Configure a routing policy under `policy-options policy-statement`
    - Several policies can be chained together to create more flexible export policies
      - Policies are evaluated in order against the routing table until a match is found, or the policy evaluation ends -- at which point the default policy action is applied
      - Default Junos action is not to redistribute any routes into OSPF
  - Then actions that make sense with OSPF policies:
    - `metric` - Set the metric value
    - `external` - Export as an external route
      - Default in Junos is to export routes into OSPF as Type 2 externals
      - Can specify the type with thhe `type (1|2)` to specifiy the route should be a Type 1 or Type 2 external route
  - tag - set the value of the Tag field
    - Can also add and subtract from the value already present in the tag field with the `add` or `subtract`

Example: *Policy to export RFC-1918 static routes as Type 2 Externals and tag them with a value of 100, and to export all other static routes as Type 1 externals with a metric of 50000 and a tag of 333*

```
         admin@J2300-1# show
policy-statement EXPORT-STATIC {
    term RFC-1918 {
        from {
            protocol static;
            route-filter 10.0.0.0/8 orlonger;
            route-filter 192.168.0.0/16 orlonger;
            route-filter 172.16.0.0/12 orlonger;
        }
        then {
            tag 100;
            external {
                type 2;
            }
        }
    }
    term OTHER-STATICS {
        from protocol static;
        then {
            metric 50000;
            tag 333;
            external {
                type 1;
            }
            accept;
        }
    }
}

[edit policy-options]
admin@J2300-1#
```

Example: *Policy to export RIP routes with a tag according to the metric of the RIP route*

```
         admin@J2300-1# show
policy-statement EXPORT-RIP {
    term RIP-Metric-1 {
        from {
            protocol rip;
            metric 1;
        }
        then {
            metric 1000;
            tag 1;
            accept;
        }
    }
    term RIP-Metric-2 {
        from {
            protocol rip;
            metric 2;
        }
        then {
            metric 1000;
            tag 2;
            accept;
        }
    }
    term RIP-Metric-3 {
        from {
            protocol rip;
            metric 3;
        }
        then {
            metric 1000;
            tag 3;
            accept;
        }
    }
    term RIP-Too-Many-Hops {
```

```
        from protocol rip;
        then {
            metric 1000;
            tag 16;
            accept;
        }
    }
}
```

Example: *Applying the two policies above as export policies for OSPF*

```
[edit]
admin@J2300-1# edit protocols ospf

[edit protocols ospf]
admin@J2300-1# set export EXPORT-STATIC

[edit protocols ospf]
admin@J2300-1# set export EXPORT-RIP

[edit protocols ospf]
admin@J2300-1# show
export [ EXPORT-STATIC EXPORT-RIP ];
```

Summarizing Type 5 External LSAs

- Needs to be done on the ASBR when the route is redistributed
  - Create an aggregate route, and redistribute the aggregate

Controlling Type 5 External LSAs

- Junos can apply import policies to external routes received via External LSAs to block routes from entering a local routers routing table
  - Can use any applicable policy action in the policy framework
  - Import policies do not block or modify the LSA - only the route as it enters the routing table
  - Routers still flood External LSAs to the routing domain regardless of the import policy
  - Configure with `set protocols ospf import`
- Can limit the number of external prefixes an ASBR will generate with `set protocols ospf prefix-export-limit` where the number ranges from 0 to 4294967295
  - Used to protect a router from flooding the network with External LSAs in case of a disaster

Exercise:External LSAs

OSPF Route Preference

This section discusses the preference of routes within the OSPF itself.

- Junos is compatible with the path selection alogritim in **RFC 1583** by default

1. OSPF intra-area paths
   - OSPF routes that originate from Type 1 and Type 2 LSAs
2. OSPF inter-area paths through
   - Routes that originate from Type 3 LSAs
3. OSPF external paths
   - OSPF External Type 1 paths are prefered over Type 2 paths
     - **RFC 2328** added that intra-area paths through non-backbone areas to an ASBR are preferred over paths though the backbone area.
     - Can enable this functionality with the knob `set protocols ospf no-rfc-1583`

Equal Cost Multipath (ECMP)

- OSPF supports ECMP by design
  - A router has several potential next hops towards a destination that all have the same cost
- Junos selects one potential path based on a hashing algorithm to install in the forwarding table
  - To enable load balancing (based upon flows), must apply a export policy to the forwarding table

Example: *Route with more than one next-hop, and it's forwarding entry*

```
admin@J2300-1> show route 7.7.7.0

inet.0: 49 destinations, 49 routes (49 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

7.7.7.0/24         *[OSPF/150] 15:56:46, metric 1000, tag 7
                      to 10.0.12.2 via fe-0/0/1.12
                    > to 10.0.13.3 via fe-0/0/1.13

admin@J2300-1> show route forwarding-table destination 7.7.7.0
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
7.7.7.0/24         user     0 10.0.13.3          ucst   574    13 fe-0/0/1.13

Routing table: __master.anon__.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct   525     1

admin@J2300-1>
```

Example: *Enabling flow based load balancing in Junos with an export policy*

```
admin@J2300-1> edit
Entering configuration mode
The configuration has been changed but not committed

[edit]
admin@J2300-1# edit policy-options policy-statement LOAD-BALANCE
```

```
[edit policy-options policy-statement LOAD-BALANCE]
admin@J2300-1# set then load-balance per-packet

[edit policy-options policy-statement LOAD-BALANCE]
admin@J2300-1# top

[edit]
admin@J2300-1# set routing-options forwarding-table export LOAD-BALANCE

[edit]
admin@J2300-1# commit and-quit
commit complete
Exiting configuration mode
```

Example: *ECMP route with micro-flow based load balancing applied. Note two potential next hops appear in the forwarding table*

```
admin@J2300-1> show route forwarding-table destination 7.7.7.0
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
7.7.7.0/24         user     0                    ulst 131070     6
                              10.0.12.2          ucst   575      8 fe-0/0/1.12
                              10.0.13.3          ucst   574     11 fe-0/0/1.13

Routing table: __master.anon__.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct   525      1

admin@J2300-1>
```

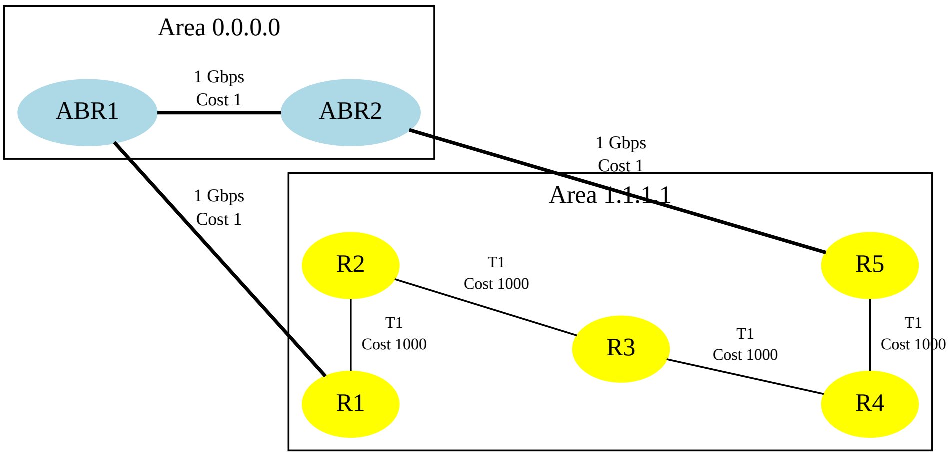
Junos Route Preference

- Junos assigns different preferences to Internal and External Routes
  - In Junos the lowest preference is considered more desireable
  - 10 for Internal OSPF routes
    - Can modify this by setting globally for OSPF `set protocols ospf preference` where the preference is from 0 to 4294967295
  - 150 for External routes
  - Can modify this globally for OSPF with `set protocols ospf external-preference` where the preference is from 0 to 4294967295
  - Can modify External route preferences individually with an import policy

Multi-area Adjacencies

Since intra-area paths are always considered better than inter-area paths, this can introduce some routing inefficiencies.

Example: *For the network below, the best path from R1 to R5 is over the slow T1 links through Area 1.1.1.1 even though some nice fast Gigabit links exist*



- To eliminate some routing inefficiencies, it is permissible to have a link in more than one area
  - Defined in **RFC 5185**
  - ABRs can establish multiple adjacencies belonging to different areas over the same logical interface
    - Announced as a point-to-point unnumbered link (Type 1) in the router LSA
    - Only valid for Point-to-Point links
  - A stub route (Type 3) is only attached to the Router LSA for the area where the interface is primary
- Configure an interface to participate in multiple areas with `set protocols ospf area interface secondary`

Passive Interfaces

- Eliminate External LSAs for connected interfaces by marking them passive
  - Possible to advertise an interface in OSPF as part of the Router LSA without actively running OSPF on the interface by marking the interface with the `passive` flag
  - Interface and any IP networks are announced as part of the Router LSA
  - No hellos are sent out the interface, so no adjacencies can form

Exercise:OSPF Route Preference

Stub Areas

Stub area definition

- Exit point taken from the area is not dependent on any external destinations
    - Not important that routers in the area have explicit information on how to reach destinations outside the AS

Stub area operation

- Disable support of Type 5 LSAs (External LSAs)
    - As Type 5 LSAs are no longer supported in the area, there is no need for Type 4 LSAs either, ASBR Summary LSAs
- External LSAs are not flooded into the area (by the ABRs)
    - No ASBR Summary LSAs are advertised into the area as they are not needed
- External LSAs are not flooded out of the area either by the ABRs
- Internal routers in a stub area do not support Type 5 LSAs
    - Routers in a stub area agree that the area is a stub
        - E bit in the options field of the OSPF header is cleared to inidcate the router does not support External LSAs
        - In order to form an adjacency, the E bit setting must match on all interfaces attempting to form an adjacency

- A default route can be flooded into a stub area from the ABRs instead (if needed) as a Type 3 LSA, Network Summary LSA

Configuring Stub Areas in Junos

- To define an area as a stub area, add the `stub` keyword underneath the area definition with `set protocols area stub`
    - To flood a default route into a stub area include the `default-metric` after the stub keyword where the metric is from 1 to 16777215

Troubleshooting Stub Areas

- An interface configured for inclusion in a stub area will be visible with `show ospf interface` with the `detail` flag set

Example: *Interface participating in a stub area*

```
admin@J2300-1> show ospf interface fe-0/0/1.1001 detail
Interface           State   Area            DR ID           BDR ID          Nbrs
fe-0/0/1.1001       DRother 0.0.0.1         0.0.0.0         0.0.0.0            0
  Type: LAN, Address: 10.1.111.1, Mask: 255.255.255.0, MTU: 1496, Cost: 2500
  Priority: 0
  Adj count: 0
  Hello: 1, Dead: 11, ReXmit: 3, Stub
  Auth type: None
  Protection type: None
  Topology default (ID 0) -> Cost: 2500

admin@J2300-1>
```

- An area configured as a stub will be displayed in the Router LSA as well

Example: *Router LSA for a stub area*

```
admin@J2300-1> show ospf database area 1 detail

    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.0.0.1         10.0.0.1         0x80000001   314  0x20 0xc510  36
  bits 0x1, link count 1
  id 10.1.111.0, data 255.255.255.0, Type Stub (3)
    Topology count: 0, Default metric: 2500
Summary *10.0.0.1         10.0.0.1         0x80000001   313  0x20 0xba6e  28
  mask 255.255.255.255
  Topology default (ID 0) -> Metric: 0
```

- An adjacency will not form if routers on both ends do not agree that the link is a stub area
    - Can be troubleshot by enabling traceroptions and looking for errors reporting "stubness" mismatches

```
Sep  1 13:54:46.684118 OSPF packet ignored: area stubness mismatch from 10.1.111.101 on intf fe-0/0/1.1001 ar
```

- If a default route is being injected into an area, it will show up as a Type 3 Network Summary LSA with a Link ID of 0.0.0.0

Example: *Viewing a default route injected into a stub area by an ABR*

```
admin@J2300-1> show ospf database lsa-id 0.0.0.0 area 1 detail

    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Summary *0.0.0.0          10.0.0.1         0x80000001   111  0x20 0xa123  28
  mask 0.0.0.0
  Topology default (ID 0) -> Metric: 111
```

Totally Stubby Areas

- In some cases it may not even be necessary to flood all of the Type 3 LSAs into a stub area
    - Single exit point
- ABR can be instructed not to inject any Type 3 LSAs
    - ABR must inject a default route (as a Type 3 LSA)
- As this functionality is dependent solely on the behavior of the ABRs it only needs to be configured there
    - Other routers still need to be configured as stubs

Configuring a Totally Stubby Area

- Simply add the `no-summaries` flag to the stub directive: `set protocols ospf area stub no-summaries`
  - Only needs to be configured on the ABRs
  - A default route will need to be injected by the ABRs
    - Configured the same as for a stub area
  - Troubleshooting is the same as for a stub area

Scaling with Stub Areas

- Stub areas are a great way to scale OSPF!
  - Shrinks the LSDB for the area
    - Eliminates all External LSAs
    - Eliminates all ASBR Summary LSAs
- Good to protect older routers with limited memory resources
- Beware of the limitations
  - Cannot inject any external routing information into a stub area -- no redistribution into OSPF
  - Lack of routing information can introduce suboptimal routing
  - Cannot support a virtual link through a stub area

Exercise:Stub Areas

Type 7 LSAs - Not So Stubby Areas (NSSA)

Flooding scope:*Only within the NSSA*

- Not So Stubby Area is a stub area with the ability to support external routes
  - First defined in **RFC 1587**, and updated in **RFC 3101**
- Prime motivation: Remote areas separated from the backbone by low-speed links
  - Minimize LSDB
  - Need to support external routes
  - Need to limit advertisements across the links
- Operates in much the same way as a stub area
  - Type 3 Network Summary LSAs are flooded into the area by the ABRs
    - Can disable this behavior as with a Totally Stubby Area (Stub Area with No Summaries)
  - No Type 5 External LSAs allowed in the area
    - May need the ABRs to flood a default LSA into the area to make up for a lack of routing information
- As No Type 5 External LSAs are allowed in a NSSA, need to make up a new LSA - Type 7 NSSA LSA
  - Basically the same as a Type 5 External LSA
    - Three differences exist:
      1. LSA Type is set to 7
      2. Must have the N/P bit set
      3. Forwarding address behavior is different

Type-7 LSA Options Field

```
            +--------------------------------------+
            | DN | O | DC | EA | N/P | MC | E | MT |
            +--------------------------------------+
```

- New bit defined in the options field of the standard OSPF packet header - N/P bit
  - Referenced as the N bit in a hello packet
    - N bit functions much the same way as the E bit
      - Can't set the N bit and E bit at the same time
      - Proposed neighbors must agree on the N bit settings
      - Keeps configurations consistent within an area
  - Referenced as the Propagate (P) bit in a LSA
    - Identifies if a Type 7 LSA should be translated into a Type 5 LSA by a NSSA ABR
      - If set, the ABR for the NSSA will translate the Type 7 LSA into a Type 5 External LSA and flood it throughout the rest of the OSPF domain as a Type 5 LSA

Type 7 - NSSA LSA*Including the common LSA header*

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |            LS age             |     Options   |       7       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                        Link State ID                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     Advertising Router                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     LS sequence number                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         LS checksum           |             length            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Network Mask                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |E|     0       |                  metric                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                      Forwarding address                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                      External Route Tag                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |E|    TOS      |                TOS  metric                    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                      Forwarding address                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                      External Route Tag                       |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
```

The Fields are defined and populated as follows: Most everything is the same as a Type 5 External LSA

- Link State ID of the common LSA header is set to IP address of the network being advertisedthe .
- The Advertising Router in the common LSA header is set to the Router ID of the ASBR

Options Field
: Sets the P bit as described above

Forwarding address
: Traffic for the advertised destination will be forwarded to

    - If the P bit is set, the address is set an address on the NSSA router injecting the External route
        - Prefers to use an internal loopback address, but will use an active physical address in the NSSA if no loopback is available
    - If set to 0.0.0.0, traffic is forwarded to the NSSA router injecting the External route

Examining NSSA LSAs in Junos

- Use the operational mode command `show ospf database nssa`
    - Shows all NSSA LSAs in all areas by default
    - Can use `advertising-router` , `lsa-id` to narrow down the output results
    - Can use the `detail` and `extensive` flags to tailor output level of detail
    - Can use the `summary` flag for an overview of the LSAs

Example: *Viewing a specific NSSA LSAs in detail*

```
admin@J2300-1> show ospf database nssa lsa-id 5.1.3.0 detail
    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
NSSA     5.1.3.0          10.0.0.3         0x80000002  2243  0x20 0xc1d7  36
  mask 255.255.255.128
  Topology default (ID 0)
    Type: 1, Metric: 1, Fwd addr: 0.0.0.0, Tag: 0.0.0.1
```

- By default in Junos, no NSSA LSAs are generated
    - Same rules apply as for Junos's treatment of External LSAs

Translating Type 7 LSAs to Type 5 LSAs

- NSSA ABRs can translate a Type 7 LSA from an NSSA to a Type 5 LSA and flood it to the rest of the OSPF domain as a Type 7 LSA
    - Type 7 LSAs that have the P bit set are propogated to the rest of the domain
    - A NSSA ABR can unconditionally translate all LSAs
- Junos by default translates all Type 7 LSAs to Type 5 LSAs
    - Each NSSA ABR generates a Type 4 ASBR Summary LSA for each NSSA ASBR that needs one
        - Supports multiple translator ABRs
        - Increases the LSDB size
            - Can disable this behavior if desired by summarizing or restricting Type 7 translations into Type 5 Routes

Changes with the Router LSA

- Changes are in the options field
    - All NSSA Border Routers set the E bit in the Router LSA
    - All NSSA ABRs that translate Type 7 LSAs into Type 5 LSAs set the E bit in the Router LSA
    - A NSSA ABR that is translating all Type 7 LSAs into Type 5 LSAs will set the Nt bit

Not-so-Totally-Stubby Areas

- Instruct the NSSA ABRs not to flood Type 3 LSAs into the area
    - Ironically, RFC calls for a Type 3 default route to be flooded in place of all other Type 3 LSAs

Configuring NSSAs in Junos

- To define an area as a NSSA, add the `nssa` keyword underneath the area definition with `set protocols area nssa` >
- To disable flooding Type-3 LSAs into the NSSA by the ABR, include the `no-summaries` keyword
    - To flood a default route into a NSSA from an ABR, include the `default-lsa default-metric` after the `nssa` keyword where the metric is from 1 to 16777215
        - By default, this is flooded as a Type 7 NSSA LSA into the NSSA
            - Flooded as a Type 3 Network Summary LSA if no-summaries keyword is configured for the NSSA to disable sending Type 3 Network Summary LSAs into the area
                - Can flood as a Type-7 LSA instead with the `type-7` keyword
        - Can set the external metric type to 1 or 2 `metric-type (1|2)`
- Can summarize or block Type-7 to Type-5 LSA translation by the ABR with the `area-range` command for the NSSA
    - Operates in the same manner as Type-3 Network Summary LSA summarization by operates only on Type 7 LSAs
    - Metric of an aggregated translation is the highest metric of the contributing routes
        - Can override with the `override-metric` keyword where metric is from 1 to 16777215
    - Can match exact prefixes with the `exact` flag
    - Can block translations in the matched range with the `restrict` flag
- On an NSSA ABR with muliple NSSAs attached, if an ABR is originating any external routes it will send a separate NSSA LSA into each NSSA it serves by default
    - Can override this behavior with the `no-nssa-abr` flag at the protocols ospf level

Troubleshooting NSSAs

- An interface configured for inclusion in a NSSA will be visible with `show ospf interface` with the `detail` flag set

```
admin@J2300-1> show ospf interface fe-0/0/1.1001 detail
Interface           State   Area            DR ID           BDR ID          Nbrs
fe-0/0/1.1001       DRother 0.0.0.1         10.1.0.1        0.0.0.0            1
  Type: LAN, Address: 10.1.111.1, Mask: 255.255.255.0, MTU: 1496, Cost: 2500
  DR addr: 10.1.111.101, Priority: 0
  Adj count: 1
  Hello: 1, Dead: 11, ReXmit: 3, Stub NSSA
  Auth type: None
  Protection type: None
  Topology default (ID 0) -> Cost: 2500

admin@J2300-1>
```

- An adjacency will not form if routers on both ends do not agree that the link is a NSSA area
  - Can be troubleshot by enabling traceroptions and looking for errors reporting "stubness" mismatches or "nssaness" mismatches
    - Stubness mismatches occur between interfaces configured for non-stub areas and ones configured NSSAs
    - Nssaness mismatches occur between interfaces configured for NSSA areas and ones configured for stub areas

```
Sep  1 13:54:46.684118 OSPF packet ignored: area stubness mismatch from 10.1.111.101 on intf fe-0/0/1.1001 ar

Sep  5 10:16:31.272796 OSPF packet ignored: area nssaness mismatch from 10.1.111.101 on intf fe-0/0/1.1001 ar
```

- If a default route is being injected into an area, it will show up as a Type 3 Network Summary LSA or a Type 7 NSSA LSA with a Link ID of 0.0.0.0
  - Type 3 LSA if `no-summaries` is configured

Example: *Viewing a default route injected into a stub area by an ABR as a Type 7 LSA*

```
admin@J2300-1> show ospf database lsa-id 0.0.0.0 area 1 detail

    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
NSSA    *0.0.0.0          10.0.0.1         0x80000001    11  0x20 0x53e8  36
  mask 0.0.0.0
  Topology default (ID 0)
    Type: 1, Metric: 1000, Fwd addr: 0.0.0.0, Tag: 0.0.0.0

admin@J2300-1>
```

- LSAs that are translated will show up having a NSSA LSA in the originating area, and a Type 5 LSA in all other areas (except for stub areas)
  - Forwarding address will be set to the originating router

Example: *Type-7 LSA and it's Tranlation to an External LSA*

```
admin@J2300-1> show ospf database lsa-id 3.3.3.0 detail

    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
NSSA     3.3.3.0          10.1.0.3         0x80000003  1046  0x28 0xe8b3  36
  mask 255.255.255.0
  Topology default (ID 0)
    Type: 1, Metric: 100, Fwd addr: 10.1.0.3, Tag: 0.0.0.3
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   3.3.3.0          10.0.0.3         0x80000002  1034  0x22 0x693d  36
  mask 255.255.255.0
  Topology default (ID 0)
    Type: 1, Metric: 100, Fwd addr: 10.1.0.3, Tag: 0.0.0.3

admin@J2300-1>
```

- Junos NSSA ABRs will create a Type 4 ASBR Summary LSA for every NSSA ASBR that it performs a NSSA to External Translation for

Example: *ASBR Summary LSA created because of a NSSA to External LSA Translation*

```
admin@J2300-1> show ospf database asbrsummary lsa-id 10.1.0.3 detail

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
ASBRSum *10.1.0.3         10.0.0.1         0x80000004   378  0x22 0xcdac  28
  mask 0.0.0.0
  Topology default (ID 0) -> Metric: 5010
ASBRSum  10.1.0.3         10.0.0.3         0x80000005   181  0x22 0xbf85  28
  mask 0.0.0.0
  Topology default (ID 0) -> Metric: 2510
...
..
.
```

- If an NSSA ABR creates is the originator of a "unique" External LSA (one that is summarized or created on the ABR) the forwarding address will be set to 0.0.0.0

Example: *NSSA LSA and matching External LSA created by an NSSA ABR*

```
admin@J2300-1> show ospf database lsa-id 5.1.1.0 detail

    OSPF database, Area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
NSSA    *5.1.1.0          10.0.0.1         0x80000004  2684  0x20 0xdfbb  36
  mask 255.255.255.128
  Topology default (ID 0)
    Type: 1, Metric: 1, Fwd addr: 0.0.0.0, Tag: 0.0.0.1
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
```

```
Extern  *5.1.1.0         10.0.0.1        0x80000008   113  0x22 0xd5c1  36
  mask 255.255.255.128
  Topology default (ID 0)
    Type: 1, Metric: 1, Fwd addr: 0.0.0.0, Tag: 0.0.0.1

admin@J2300-1>
```

Exercise: NSSA Areas

Opaque LSAs

- Original OSPF developers thought it would be best to hard code all of the types, options, etc for the OSPF protocol
  - Viewed as more efficient CPU and bandwidth-wise, and a way to keep implementations honest
- With a few extentions, OSPF started to quickly run out of space for new options and new types
- Opaque LSAs were envisioned to be a generalized type of advertisement
  - Designed to be extensible with future expansion in mind
  - Use a paradigm borrowed from BGP (and IS-IS) - the Type Length Variable
  - The OSPF state machine does not necessarily need to know about the contents of the opaque fields
    - Left up to the application to determine what the contents signify
  - Some applications are:
    Router Information, Graceful Restart, Traffic Engineering Extensions
- First defined in **RFC 2370**
  - Later updated in **RFC 5250**
  - Lots of RFCs on the individual extensions - the TLV definitions contained in the LSAs
- Three Types of Opaque LSAs defined -- each with a defined flooding scope
  - Type 9 denotes link-local scope
    - LSA is not to be flooded past the local subnet on which the interface is attached
  - Type 10 denotes area scope
    - LSA is not to be flooded beyond the area the LSA was orginated in
    - Type 11 deontes domain scope
      - LSA is to be flooded throught the entire OSPF routing domain
      - Has the same flooding scope as Type 5 External LSAs
      - Cannot violate any of the area restrictions for stub or NSSA networks
- Adjacency Formation
  - Potential neighbors that support Opaque LSAs will set the O bit in the Options field in Hello packets
  - Indicates that the neighbor can support and can forward Opaque LSAs
  - Mismatched O bit settings between neighbors don't mean that an adjacency will not form

Options Field

```
+--------------------------------------+
| DN | O | DC | EA | N/P | MC | E | MT |
+--------------------------------------+
```

- Opaque LSAs vary in size, but are aligned to 32 bit boundaries
  - Length and values vary depending on the application
- Link-state ID has redefined
  - Divided into an Opaque Type field and an Opaque ID field

Opaque LSAs*Including the common LSA header*

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LS age             |     Options   |  9, 10, or 11 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Opaque Type  |               Opaque ID                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Advertising Router                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      LS Sequence Number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         LS checksum           |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                      Opaque Information                       |
+                                                               +
|                              ...                              |
```

The Fields are defined and populated as follows:

Options Field
    Sets the O bit as described above, along with other options as necessary
Type
    Set to 9, 10 or 11 depending on the flooding scope
Opaque Type
    8 bit field
Opaque ID
    a 24 bit ID field

Examining Opaque LSAs in Junos

- Most of the information in any Opaque LSAs is going to be application specific
  - Need to troubleshoot within the application context
- Use the operational mode command `show ospf database opaque-area`

- Shows all Opaque LSAs with area-flooding scope in all areas by default
- Can use `advertising-router` , `lsa-id` to narrow down the output results
- Can use the `detail` and `extensive` flags to tailor output level of detail
- Can use the `summary` flag for an overview of the LSAs

Troubleshooting

- Can trace opaque LSA operations like any other LSA in the context of OSPF
- Need to troubleshoot within the context of the application that is using opaque LSAs to convey information
- Junos does not support any applications that use Type 11 - Domain Scope Opaque LSAs

Type 9 LSAs - Link Scope Opaque LSA

Flooding scope:*Local Link*

Applications:*OSPF Graceful Restart*

Graceful Restart in a Nutshell

- Many routers have a forwarding plane that operates separately from the control plane
- M, T and MX series routers from Juniper have total separation of the control plane and forwarding plane
  - Routing Engine serves all control plane functions
    - Runs the OSPF process
    - Programs the forwarding table into the forwarding plane from the calculated best routes
    - *Note: Some of the hardware of the forwarding plane can offload boring repetive things like generating and receiving Hello packets*
  - Forwarding plane runs on the Packet Forwarding Engines (PFEs) on the FPCs, DPCs, etc (depending on hardware architecture)
    - Forwarding plane will keep forwarding packets even if the control plane goes away
      - No changes in the forwarding state can be made
      - Eventually neighbors running dynamic routing protocols will learn of the death of the routers control plane, and take their own actions
        - Declare neighbor dead and update routing protocols
        - Take a backup path
- When the control plane restarts after a service impacting event:
  - The control plane will need to reestablish all of it's dynamic adjacencies (OSPF, IS-IS, BGP, PIM, RIP, etc.)
  - Relearn network topologies
  - Recalculate best routes
  - Update it's forwarding table
  - Can cause a network wide noticeable event while the network routes around, and then establishes a restarting router back in the network topology - a big ripple
  - A lot of this is unnecessary when the forwarding plane of the restarting router can continue to forward packets without the control plane being fully operational

• Graceful Restart is a concept that allows a lot of these ripples to be avoided if the following conditions are met:

1. A router is capable of forwarding packets without a fully operational control plane
2. The forwarding plane operation isn't interrupted
3. The interfaces and path between adjacent routers remains operational
4. No other topology changes occur on the network anywhere else

• For Graceful Restart to take place routers must agree between them that they help each other out in case of a catastrophe

  - Best if all neighbors support it to minimize any topology changes

Graceful Restart Operation

- Three modes of operation for Graceful Restart
  1. ***Possible Helper***
     A router that is capable of helping a neighbor reestablish itself in the network
  2. ***Helper***
     A router that is assisting a restarting router
     - By hiding the occurance from neighbors
     - By helping the restarting router rebuild it's routing topologies
  3. ***Restart Candidate***
     A router that is about to restart, and has informed it's neighbors, or one that is undergoing a restart event
- During a restart event:
  - Neighbors around the restarting router wait a period of time, the restart duration, before declaring the neighbor down and informing other neighbors of any topology changes
  - A restarting router can restart due to a planned or unplanned event
    - In the event of a planned restart, the restarting router, informs it's neighbors that it is about to restart
      - Sends a Grace LSA politely informing it's neighbors of the impending event
    - In the event of an unplanned restart, the restarting router sends a Grace LSA once it's control plane has recovered
  - Neighbors hide the failure of the restarting router from the rest of the network
  - When the restarting routers control plane starts to come back up after a restart, it's neighbors that assist it (helper routers) dump as much topology information as possible to help the router rebuild it's state
    - For OSPF, this is essentially a copy of the LSDB
  - If all goes well, other than the restarting routers neighbors, no other router in the network was aware of the failure
  - If anything doesn't go well or the restart timer expires the graceful restart is aborted

Graceful Restart for OSPF

- Defined in **RFC 3623**

Grace LSA

- Uses a Type 9 link-local Opaque LSA
  - Opaque Type 3
  - Opaque ID 0
- Grace LSA requests that the router's neighbors aid it's restart
  - Request to advertise the restarting router as fully adjacent during a grace perioid
  - LS age is set to 0 when the request is first advertised
    - Neighbors use the age of the Grace LSA to keep track of how long the restarting router made the request
  - Body of the LSA is TLV encoded
    - Length of the requested grace period
    - Reason
    - IP address of the interface of restarting router (if on a link that has a DR election take place)

Grace LSA Header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LS age             |     Options   |       9       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       3       |                    0                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     LS sequence number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         LS checksum           |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                            TLVs                             -+
|                             ...                               |
```

TLV Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Value...                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The following TLVs have been defined for the Grace LSA

Grace Period (Type 1, Length = 4)
Number of seconds neighbors should advertise the restarting router as adjacent
Mandatory in every Grace LSA
Graceful Restart Reason (Type 2, length = 1)
0 - unknown
1 - software restart
3 - switch to redundant control processor
Mandatory in every Grace LSA
IP interface address (Type 3, Length 4)
Interface IP address on multiaccess networks
Used to identify the restarting router

Configuring OSPF Graceful Restart in Junos

- Graceful restart is a global option in Junos
  - Disabled by default
    - Once enabled, support for all protocols that are supported is enabled
      - Can disable and tweak each protocol independently for Graceful Restart
- Enable by setting `set routing-options graceful-restart`
  - Can set the restart duration expected with `set routing-options graceful-restart restart-duration`
  - Good to know an approximate value of how long it takes a router to undergo a restart
- OSPF specific items for graceful restart are in `protocols ospf graceful-restart`
  - Can disable with `disable` flag
  - Can disable helper mode with `helper-disable`
  - Can set the graceful restart duration for OSPF with `restart-duration` where seconds is from 1 to 3600
    - Default of 180 seconds
    - Amount of time the restarting router requests for restarts
  - Can set the amount of time before the restarting router notifies helper OSPF routers that it has completed graceful restart with notify-duration where seconds is 1 to 3600
    - Default is the restart-duration + 30 seconds
  - Can disable the constraint about no other network topology changes by setting the `no-strict-lsa-checking flag`
    - Helper router won't abort the graceful restart process

Troubleshooting

- Can tell if the router supports graceful restart for the OSPF process with opertional `command show ospf overveiw`
  - Restart support and timer values will be displayed

Example: *OSPF router with Graceful Restart enabled for OSPF*

```
admin@J2300-1> show ospf overview
Instance: master
  Router ID: 10.0.0.1
  Route table index: 0
  Area border router, AS boundary router, NSSA router
  LSA refresh time: 50 minutes
```

```
  Restart: Enabled
    Restart duration: 180 sec
    Restart grace period: 210 sec
    Helper mode: Enabled
  Area: 0.0.0.0
    Stub type: Not Stub
    Authentication Type: None
    Area border routers: 5, AS boundary routers: 5
    Neighbors
      Up (in full state): 2
  Area: 0.0.0.1
    Stub type: Stub NSSA, Stub cost: 1000
    Authentication Type: None
    Area border routers: 1, AS boundary routers: 4
    Neighbors
      Up (in full state): 2
  Area: 30.30.30.30
    Stub type: Not Stub
    Authentication Type: None
    Area border routers: 5, AS boundary routers: 5
    Neighbors
      Up (in full state): 2
  Area: 101.101.101.101
    Stub type: Not Stub
    Authentication Type: None
    Area border routers: 6, AS boundary routers: 6
    Neighbors
      Up (in full state): 2
  Area: 102.102.102.102
    Stub type: Not Stub
    Authentication Type: None
    Area border routers: 6, AS boundary routers: 6
    Neighbors
      Up (in full state): 2
  Topology: default (ID 0)
    Prefix export count: 2
    Full SPF runs: 13
    SPF delay: 0.200000 sec, SPF holddown: 5 sec, SPF rapid runs: 3
    Backup SPF: Not Needed

admin@J2300-1>
```

- Can track graceful restart OSPF events with the `graceful-restart` flag under the traceoptions for the protocol

Exercise: Graceful Restart

Type 10 LSAs - Area Scope LSA

Flooding scope:*Areak*

Applications:*Traffic Engineering*

Traffic Engineering in a Nutshell

Traffic Engineering is basically controlling and regulating the path that packets take through the network. This can be done a number of ways, by tweaking link costs, and policy routing for example. OSPF can help construct a special database that can be used for calculating the paths of MPLS Label Switched Paths (LSPs) on which to map network onto. When these LSPs are initiated, they can consult the database built by OSPF (or IS-IS) to help them determine the paths through the network based on bandwidth, priority, usage, cost, link type and class. To help construct the Traffic Engineering Database (TED), a special LSA was added to OSPF.

- Traffic Engineering LSA defined in **RFC 3630**
  - Updated by **RFC 4203** to add support for GMPLS, and **RFC 5786** on how a router should advertise it's own addresses for TE extensions
- Uses Type 10 Opaque LSAs with a type of 1
  - Instance ID is set to an arbitrary value from 0 to 16777216 for multiple Traffic Engineering LSAs
  - LSA ID has no topological significance
- Designed to carry extra information about TE routers and their links
  - Intended to be used for traffic engineering, monitoring extended link attributes, constraint based routing
  - TE data and topology is independent of the regular routed topology and data
  - Traffic Engineering LSAs may be flooded whenever the contents change
  - Upon receipt of new TE LSAs, a router should update it's Traffic Engineering Database
    - No need to run a new SPF
    - Can be throttled back only to flood when certain thresholds are met
    - Can be a bit noisy!
- Reccomended for point-to-point links
  - Can also work on multi-access links, but reservations are a bit hard to concretely make

Traffic Engineering LSA

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            LS age             |    Options    |      10       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |       1       |                   Instance                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Advertising Router                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      LS sequence number                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         LS checksum           |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TLV Header

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Value...                           |
.                                                               .
.                                                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Three TLV Types Defined

Router Address TLV (Type 1, Length = 4)
Specifies a stable address for the router that is always reachable. Address must still be reachable if any physical interface goes down. Typically a loopback interface which is assigned the Router ID for the system.
Also be used to correlate IS-IS TE data

Link TLV (Type 2, Length = variable)
Describes a single link
Only one Link TLV per LSA is allowed
Uses several sub-TLVs for description of link properties

- sub-TLVs use 32 bit IEEE floating point numbers
  1. Link type (1 octet)
     - Mandatory subTLV
     - Describes the type of link, 1 = point-to-point, 2 = multiaccess
  2. Link ID (4 octets)
     - Mandatory subTLV
     - Identifies the other end of the link
       - Router ID for point-to-point links
       - Interface IP address for multiaccess links
  3. Local interface IP address (4 octets)
     - Identifies IP address(es) for the local router on the link
     - Length is 4*n octets, where n is the number of addresses on the link
  4. Remote interface IP address (4 octets)
     - IP address(es) of the neighbors interface
     - Length is also 4*n octets, where n is the number of addresses on the link
  5. Traffic engineering metric (4 octets)
     - A separate 24 bit metric for traffic engineering purposes
     - May be different than the OSPF cost for the link
  6. Maximum bandwidth (4 octets)
     - Maximum bandwidth that can be used on the link in the direction from the system originating the LSA towards it's neighbor
     - Link capacity in bytes per second
  7. Maximum reservable bandwidth (4 octets)
     - Maximum bandwidth that may be reserved on the link
     - May be greater than the maximum bandwidth (oversubscription)
     - Units are bytes per second
  8. Unreserved bandwidth (32 octets)
     - Bandwidth that can be reserved with a setup priority of 0 - 7
     - Units are bytes per second
  9. Administrative group (4 octets)
     - Also called Resource Class, Color, Affinity
     - 32 bit mask assigned by the network administrator
     - Each bit corresponds to a separate administrative group
     - A link may belong to multiple groups or none at all

Node Attributes TLV (Type 5, Length = variable)
Carries information about multiple addresses that are assigned to a router
Used for specifying different next-hops for different tolpologies or protocols

Configuring Traffic Engineering for OSPF on Junos

- Enable generation of Traffic Engineering LSAs by setting `set protocols ospf traffic-engineering`
  - Use the `advertise-unnumbered-interfaces` flag to use unnumbered interfaces for TE
  - Set `credibility-protocol-preference` to instruct the router that OSPF is the more preferred protocol to contribute to the Traffic Engineering Database (TED)
    - By default, Junos prefers IS-IS to for TED population
  - Set `no-topology` to disable TE topology information
    - No TE LSAs are orignated by the router, no TED is built
  - Can set `shortcuts` directive to instruct OSPF to use MPLS LSPs as next hops
    - Copies OSPF routes that can use configured LSPs into the inet.3 table
      - Usually want to avoid this -- better ways to do this....
    - Can configure OSPF to advertise the LSPs metric, rather than the link (or sum of link) metrics into summary LSAs with the `set protocols ospf traffic-engineering shortcuts lsp-metric-into-summary` directive
  - Can ignore LSP metrics when doing shortcuts with the `ignore-lsp-metrics` flag

- Can set an independent metric on a link for TE with `set protocols ospf area interface te-metric` where the metric is from 1 to 4294967295

Viewing a Traffic Engineering LSA

- No specific command to view a TE LSA on Junos
  - Can view all Type 10 Opaque Area-Scope LSAs with operational command `show ospf database opaque-area`
  - TE is the only application that uses Type 10 LSAs at this time
  - Can use the `advertising-router`, `area` to limit output
  - Can use the `lsa-id` to limit it to certain LSAS
    - Remember, with the TE LSA the LSA ID was redefined to an 8 bit type field, and a 24 bit Opaque ID field
    - LSA Id's don't have any topological significance
  - Can use the `detail`, `extensive` and `summary` flags to vary the amount of information shown
    - TLV Type, and subTLV data is visible with detail and extensive flags

Example: *Viewing a TE LSA with a Router Address TLV Note: the TLV Type, Length and value is shown in the LSA guts*

```
admin@J2300-1> show ospf database opaque-area area 0 advertising-router 10.0.0.1 lsa-id 1.0.0.1 detail

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
OpaqArea*1.0.0.1         10.0.0.1         0x80000011  1377  0x22 0xfb0d  28
  Area-opaque TE LSA
  RtrAddr (1), length 4: 10.0.0.1

admin@J2300-1>
```

Example: *Viewing a TE LSA with a Link TLV and it subTLVs*

```
admin@J2300-1> show ospf database opaque-area area 0 advertising-router 10.0.0.1 lsa-id 1.0.0.3 detail

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
OpaqArea*1.0.0.3         10.0.0.1         0x80000012   952  0x22 0xcba9 136
  Area-opaque TE LSA
  Link (2), length 112:
    Linktype (1), length 1:
      1
    LinkID (2), length 4:
      10.0.0.2
    LocIfAdr (3), length 4:
      10.0.12.1
    RemIfAdr (4), length 4:
      10.0.12.2
    TEMetric (5), length 4:
      5
    MaxBW (6), length 4:
      10Gbps
    MaxRsvBW (7), length 4:
      10Gbps
    UnRsvBW (8), length 32:
        Priority 0, 10Gbps
        Priority 1, 10Gbps
        Priority 2, 10Gbps
        Priority 3, 10Gbps
        Priority 4, 10Gbps
        Priority 5, 10Gbps
        Priority 6, 10Gbps
        Priority 7, 10Gbps
    LinkLocalRemoteIdentifier (11), length 8:
      Local 68, Remote 0
    Color (9), length 4:
      0

admin@J2300-1>
```

Troubleshooting Traffic Engineering

- Junos will only create a TE LSA with a Link Attributes TLV if an interface is configured to process RSVP and MPLS
- Junos will create a TE LSA with a Router Address TLV as long as it is configured to run TE extensions to OSPF
- A TE LSA with a Link TLV will pull it's values from the following places for each subTLV:

  1. Link type - OSPF interface type
  2. Link ID - OSPF adjacency
  3. Local interface IP address - Local interface
  4. Remote interface IP address - OSPF adjacency
  5. Traffic Engineering Metric - Defaults to the OSPF cost, or the configured TE metric if one exists
  6. Maximum bandwidth - RSVP
  7. Maximum reservable bandwidth - RSVP
  8. Unreserved bandwidth - RSVP
  9. Administrative group - MPLS interface configuration

- Can view the protocols that are populating the TED with `show ted protocol`

Example: *TED Protocol Contributions*

```
admin@J2300-1> show ted protocol
Protocol name        Credibility  Self node
OSPF(0)              502          10.0.0.1
OSPF(1)              502          10.0.0.1
OSPF(505290270)      502          10.0.0.1
OSPF(1701143909)     502          10.0.0.1
OSPF(1717986918)     502          10.0.0.1

admin@J2300-1>
```

- Can view the TED with operational command `show ted database`
  - Can use the `detail` and `extensive` flags to vary the amount of information shown
  - Can specify the System ID (From the Router Address TLV) to view TED on just node with `show ted database`

Example: *Viewing the TED for a particular system*

```
admin@J2300-1> show ted database 10.0.0.2 detail
TED database: 0 ISIS nodes 15 INET nodes
NodeID: 10.0.0.2
  Type: Rtr, Age: 13902 secs, LinkIn: 4, LinkOut: 1
  Protocol: OSPF(0.0.0.0)
    To: 10.0.0.1, Local: 10.0.12.2, Remote: 10.0.12.1
      Local interface index: 68, Remote interface index: 0
  Protocol: OSPF(30.30.30.30)
  Protocol: OSPF(101.101.101.101)
  Protocol: OSPF(102.102.102.102)

admin@J2300-1>
```

- Can view all of the link information with operational command `show ted link`
  - Can use the `detail` flag for more information

Example: *Viewing the links in the TED*

```
admin@J2300-1> show ted link
ID                        ->ID                       LocalPath LocalBW
10.0.0.1                   10.0.0.2                          1 0bps
10.0.0.2                   10.0.0.1                          0 0bps
10.0.0.3                   10.0.0.1                          0 0bps
10.101.0.10-1              10.0.0.1                          0 0bps
10.101.0.10-1              10.0.0.6                          0 0bps
10.101.0.10-1              10.0.0.5                          0 0bps
10.101.0.10-1              10.0.0.4                          0 0bps
10.101.0.10-1              10.0.0.3                          0 0bps
10.101.0.10-1              10.0.0.2                          0 0bps
10.101.0.10-1              10.10.10.10                       0 0bps
10.102.0.10-1              10.0.0.1                          0 0bps
10.102.0.10-1              10.0.0.6                          0 0bps
10.102.0.10-1              10.0.0.5                          0 0bps
10.102.0.10-1              10.0.0.4                          0 0bps
10.102.0.10-1              10.0.0.3                          0 0bps
10.102.0.10-1              10.0.0.2                          0 0bps
10.102.0.10-1              10.10.10.10                       0 0bps
10.30.30.30-1              10.0.0.1                          0 0bps
10.30.30.30-1              99.99.99.99                       0 0bps
10.30.30.30-1              10.0.0.6                          0 0bps
10.30.30.30-1              10.0.0.5                          0 0bps
10.30.30.30-1              10.0.0.4                          0 0bps
10.30.30.30-1              10.0.0.3                          0 0bps
10.30.30.30-1              10.0.0.2                          0 0bps
10.1.111.101-1             10.0.0.1                          0 0bps
10.1.111.101-1             10.1.0.1                          0 0bps
10.1.123.102-1             10.0.0.3                          0 0bps
10.1.123.102-1             10.1.0.2                          0 0bps

admin@J2300-1>
```

Editorial Note: *Since an LSA with area flooding scope is used to build the TED, you wind up with a separate TED for each area. Due to the nature of OSPF, it isn't guaranteed that a router, especially a non-backbone router, will have complete information of the entire domain topology. This is certain if any kind of stub areas, summarizing addresses at ABRs. Thus, a router trying to precompute the path for a LSP won't necessarily have all of the needed information if the LSP terminates outside it's own area. So if you're planning on doing any TE, do your best to keep your OSPF design to a single area. To do TE in multiple areas you need to arrange for meeting points of LSPs in each area, and stitch them together. There are a lot of expired RFCs and things in the works. So stay tuned for a good working implementation, but don't hold your breath.*

Exercise: OSPF Traffic Engineering Database

MPLS and OSPF

LDP synchronization

- LDP distributes labels based on the best path determined by the IGP
  - Situation can arise where the IGP is operational on a link, but LDP is not
    - Can result in L3VPN, L2VPN and other applications that depend on MPLS for forwarding being black holed
    - Can arise from misconfigurations, restarting routers, protocol problems
- Can enable LDP synchronization with OSPF
  - Causes OSPF to advertise the maximum metric over a link until LDP is operational across the link
- Configure as an option to the interface desired to syncrhonize with the LDP process with `set protocols ospf interface ldp-synchronization`
  - Can also configure a timer to advertise the maximum link metric with `set protocols ospf interface ldp-synchronization hold-time` where the time is from 1 to 65536 seconds

Advertising LSPs with OSPF

- Can configure OSPF to advertise a LSP as a point-to-point link
  - Need a LSP in reverse in order to use it for SPF calculations
- Configure with `set protocols ospf area label-switched-path`
  - Can set a metric for the LSP for OSPF like it is any other interface
    - Will use the metric assigned under `protocols mpls label-switched-path metric` if one is not configured under OSPF
    - If no metric is configured on a LSP anywhere, it will be advertised with a metric of 1
  - LSP will show up as an OSPF interface
  - LSP will be added to the advertising routers Router LSA as if it was an unnumbered point-to-point interface
- OSPF will advertise LSPs with the metric

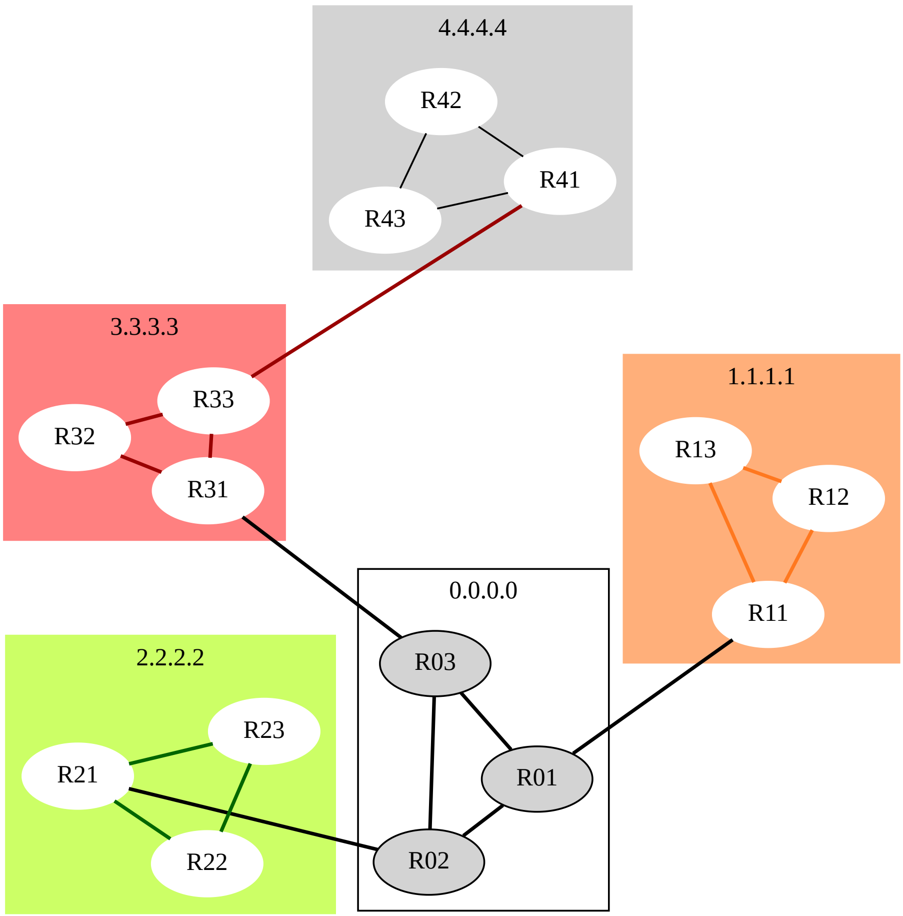
Loop Free Alternative Routes

- Provides a MPLS Fast Reroute like capability for OSPF
  - Works for IP traffic and helps with MPLS traffic using LDP signaled paths for forwarding during link or node outages
- Allows a pre-computed backup path to be installed in the Packet Forwarding Engine for OSPF routes
  - Traffic can be shunted down the backup path by the local router until routing converges globally
  - Care must be taken that the backup path does not wind up looping traffic back to the router that is making the repair
    - Junos runs the SPF calculation from the perspective of each neighbor that is one hop away to ensure a loop free path
- Link Protection
  - Used when only a single link may be come unavailable but the neighboring node would still be reachable on another interface
  - Configured by setting `set protocols ospf area interface link-protection` for the interface to be protected
- Node-Link Protection
  - Establishes an alternate path through another router
  - Configured by setting `set protocols ospf area interface node-link-protection` for the interface to be protected
- Can use RSVP signaled LSPs as a backup path used in the link-node schemes by setting the `backup` flag in the LSPs configuration
  - Configure with `set protocols mpls label-switched-path` to be used for Backup
- When calculating an alternative path, Junos by default will use any available interface as a potential backup path for a protected interface
  - Can exclude an interface from being used by setting the set protocols ospf area interface no-eligible-backup for the interface to be avoided
- Need to configure load balancing allow the PFE to install all the potential next hops for all of the destinations that will wind up being protected with either link or node-link protection

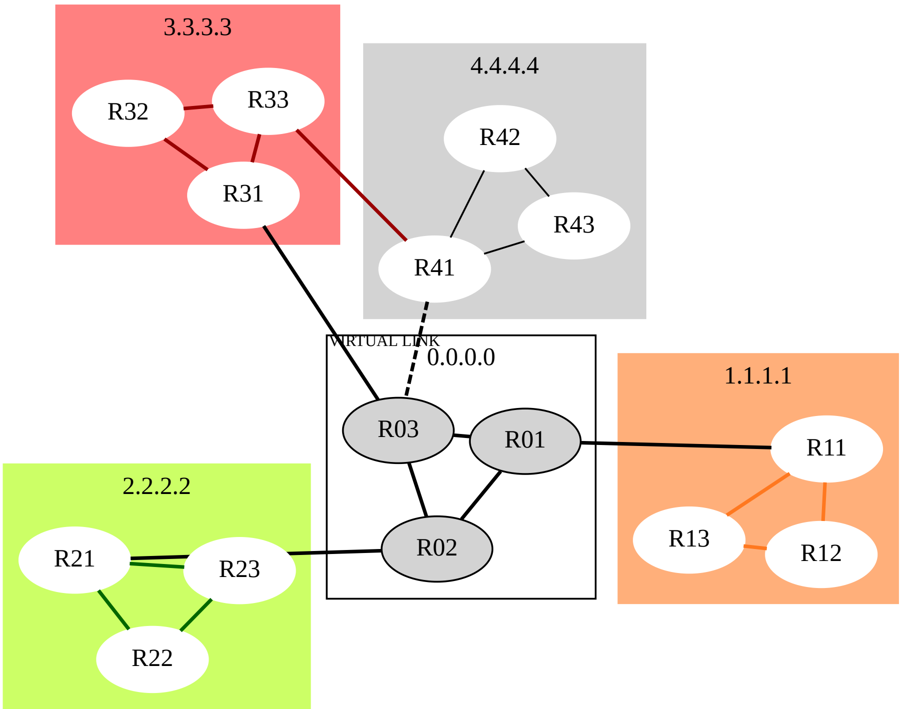
Virtual Links

- OSPF has a two level hierarchy
    - Area 0.0.0.0, the backbone area, is at the top of the hierarchy
        - Backbone area must be contiguous
        - All areas must connect to the backbone area to transit traffic to another area
    - OSPF loop protection mechanism will not allow ABRs connected to the backbone to install routes from Network Summary LSAs that did not come from the backbone
        - LSAs are still flooded per the flooding scope of their LSA type, but they may be ignored by the SPF
- To overcome this feature/limitation two ABRs can install a virtual link through a non-backbone area
    - Can be used to join two disparate backbone areas together in the event of an outage or to support a network merger
    - Can be used to join a separated non-backbone area to the backbone through another non-backbone area
    - Configured between ABRs
- Virtual link is treated as an unnumbered point-to-point circuit that is part of the backbone area
    - Configured between ABRS through another non-backbone area
        - The non-backbone area that is being traversed by the virtual link is referred to as a transit area
        - Appears as a Type 4 link in the Router LSA of the endpoints
    - A "virtual adjacency" will establish between the routers on the ends of the virtual link
        - OSPF packets that belong to the backbone area flow across the virtual link
            - Type 5 Exernal LSAs are not flooded across virtual links as their domain wide flooding scope would cause duplicates
- Cannot configure virutal links through stub areas
    - Stub areas won't allow ASBR Summary or External LSAs
- Cost of a virtual link is dynamically calculated

Example: *Area 4.4.4.4 is severed from the backbone area, but R41 has a connection to R33 in area 3.3.3.3*



A virtual link is put up from R41 (ABR) to R03 (ABR) to connnect area 4.4.4.4 virtually to the backbone using area 3.3.3.3 as the Transit area.



Configuring Virtual Links

- Under area 0.0.0.0, use the `virtual-link`
    - Need to specify the Transit area with `transit-area`
    - Need to specify the Router ID of the other side of the virtual link with `neighbor-id`

Example: *Configuring a virtual link*

```
[edit]
admin@J2300-1# edit protocols ospf area 0

[edit protocols ospf area 0.0.0.0]
admin@J2300-1# set virtual-link rou
                                ^
syntax error, expecting  or .
admin@J2300-1# set virtual-link neighbor-id 13.13.13.13 transit-area 101.101.101.10

[edit protocols ospf area 0.0.0.0]
admin@J2300-1#
```

- The virtual link will have to be configured on both sides
- Note that the neighbor is a Router ID which is not necessarily an interface IP address
    - Need to be able to see the Router ID in a Type 1 LSA for the virtual link to come up

Troubleshooting Virtual Links

- Once configured, a virtual link will show up as an interface for OSPF to use, thus the `show ospf interface` command will return results
    - In Junos virtual links will be displayed with the interface name `vl-` of remote side>

Example: *Virtual link is displayed as interface vl-13.13.13.13*

```
admin@J2300-1> show ospf interface
Interface           State   Area            DR ID           BDR ID          Nbrs
fxp1.12             PtToPt  0.0.0.0         0.0.0.0         0.0.0.0            1
fxp1.13             PtToPt  0.0.0.0         0.0.0.0         0.0.0.0            1
lo0.0               DRother 0.0.0.0         0.0.0.0         0.0.0.0            0
vl-13.13.13.13      PtToPt  0.0.0.0         0.0.0.0         0.0.0.0            1
fxp1.1001           DRother 0.0.0.1         10.1.0.1        0.0.0.0            1
fxp1.13             PtToPt  0.0.0.1         0.0.0.0         0.0.0.0            1
fxp1.100            DRother 30.30.30.30     99.99.99.99     10.0.0.4           7
fxp1.101            DRother 101.101.101.101 10.10.10.10     10.0.0.4           6
fxp1.102            DRother 102.102.102.102 10.10.10.10     10.0.0.4           6
```

- Inspecting the virtual link interface in detail will give more information on what the virtual link parameters are

Example: *Virtual link is displayed in detail*

```
admin@J2300-1> show ospf interface vl-13.13.13.13 detail
Interface           State   Area            DR ID           BDR ID          Nbrs
vl-13.13.13.13      PtToPt  0.0.0.0         0.0.0.0         0.0.0.0            1
  Type: Virtual, Address: 10.101.0.1, Mask: 0.0.0.0, MTU: 0, Cost: 2510
  Transit Area: 101.101.101.101, Destination: 13.0.0.0
  Adj count: 1
  Hello: 10, Dead: 40, ReXmit: 5, Not Stub
  Auth type: None
  Protection type: None, No eligible backup
  Topology default (ID 0) -> Cost: 2510

admin@J2300-1>
```

- Virtual link will show a state of `Down` if parameters are not correct

Example: *Virtual link in the Down state*

```
admin@J2300-1> show ospf interface vl-14.14.14.14 detail
Interface           State   Area            DR ID           BDR ID          Nbrs
```

```
vl-14.14.14.14      Down    0.0.0.0         0.0.0.0         0.0.0.0            0
  Type: Virtual, Address: 0.0.0.0, Mask: 0.0.0.0, MTU: 0, Cost: 1
  Transit Area: 14.14.14.14
  Adj count: 0
  Hello: 10, Dead: 40, ReXmit: 5, Not Stub
  Auth type: None
  Protection type: None, No eligible backup
  Topology default (ID 0) -> Down, Cost: 65535
```

- Need to have a Router LSA present in the Transit area with an LSA ID of the target of the other side of the virutal link
- A virtual link will show up in the neighbor list once it is operational

Example: *Adjacency over a virtual link in Junos*

```
admin@J2300-1> show ospf neighbor | match vl
13.0.0.0         vl-13.13.13.13         Full      13.13.13.13        0    39
```

- A router with an operational virtual link will have it displayed in its Router LSA that is sent into the backbone area
  - A Type 4 link will be displayed for each operational virtual link

Example: *A Virtual Link Displayed in the Router LSA*

```
admin@J2300-1> show ospf database router lsa-id 10.0.0.1 area 0 detail

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.0.0.1         10.0.0.1         0x80000026   281  0x22 0xdfb6  96
  bits 0x3, link count 6
  id 10.0.0.2, data 10.0.12.1, Type PointToPoint (1)
    Topology count: 0, Default metric: 5
  id 10.0.12.0, data 255.255.255.0, Type Stub (3)
    Topology count: 0, Default metric: 5
  id 10.0.0.3, data 10.0.13.1, Type PointToPoint (1)
    Topology count: 0, Default metric: 5
  id 10.0.13.0, data 255.255.255.0, Type Stub (3)
    Topology count: 0, Default metric: 5
  id 10.0.0.1, data 255.255.255.255, Type Stub (3)
    Topology count: 0, Default metric: 0
  id 13.13.13.13, data 10.101.0.1, Type Virtual (4)
    Topology count: 0, Default metric: 2510
  Topology default (ID 0)
    Type: Virtual, Node ID: 13.13.13.13
      Metric: 2510, Bidirectional
    Type: PointToPoint, Node ID: 10.0.0.3
      Metric: 5, Bidirectional
    Type: PointToPoint, Node ID: 10.0.0.2
      Metric: 5, Bidirectional
```

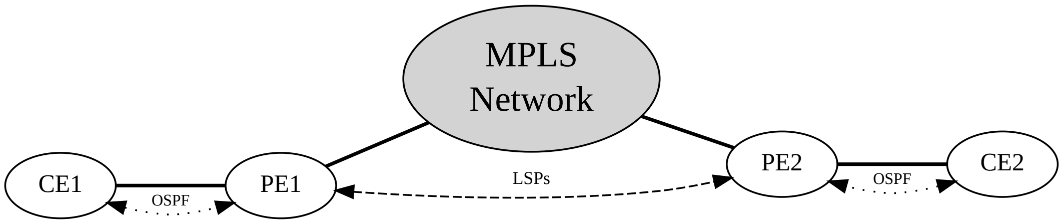
Editorial Note: *Most network books seem to paint a really awesome picture of a network that has an area severed from the backbone, and a heroic network engineer steps in and saves the day with a zero-cost virutal link! However, much like their ugly cousin the GRE tunnel, try to avoid using these wherever possible. They are like putting a band-aid on a gunshot wound. Sure it stops the immediate bleeding, but there are bigger problems a lot deeper that the virtual link band aide is just covering up. A network that needs a virtual link actually needs a real architectural overhaul. If you're using these, there are probably a lot better ways to "fix" your problem.*

Exercise: Virtual Links

OSPF and Layer 3 MPLS VPNS

- OSPF can be used as a routing protocol between a CE and PE device for route distribution in a MPLS Layer 3 VPN
- OSPF attributes from LSAs are encoded into BGP extended attributes so the OSPF advertisement can be recreated at remote VPN sites
  - OSPF cost
  - Area
  - Route type
  - Router ID
  - Route tags

Diagram: *OSPF used as the routing protocol between the CE and PE in a MPLS L3VPN*



- There are a few extensions and features in OSPF that enhance it's use as a routing protocol in L3VPNs

Down Bit

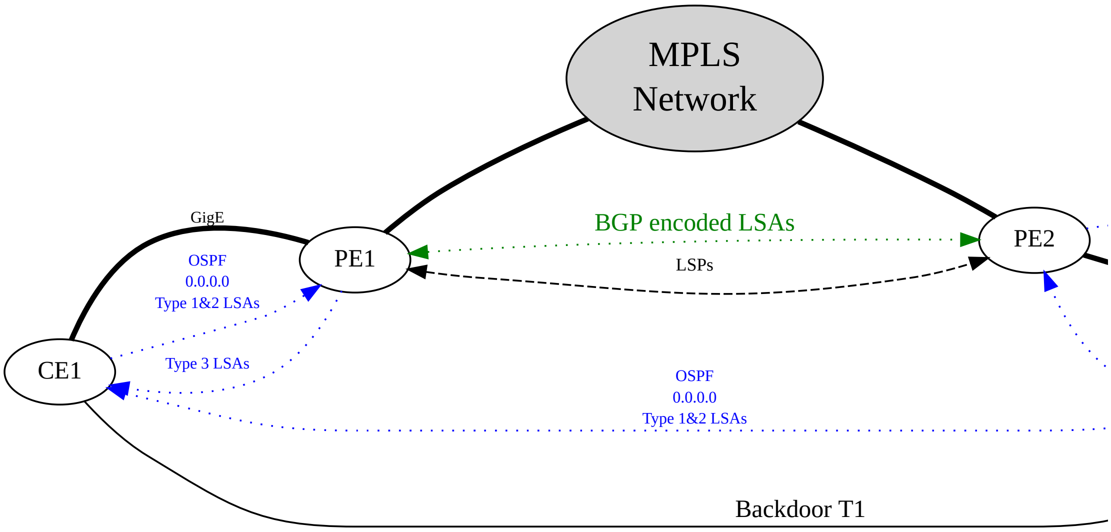
- L3VPNs usually require mutual redistribution between BGP, for the service providers signalling between PE devices, and OSPF for dynamic routing between the PE devices and the CE devices
  - Can result in looping -- thus the definition of the Dn bit in the OSPF options field
  - Down bit is set in the options field of a recreated OSPF LSA to indicate the route has been sent down by a PE device
    - A PE device should not import (redistribute) any LSA with the down bit set to remote sites

Sham Links

- Normally when a PE device recreates an LSA, it is advertised by the PE device as a Network Summary LSA
  - This is true even when both CE devices are in area 0.0.0.0
  - Not normally a problem when all customer sites are only connected through the providers network

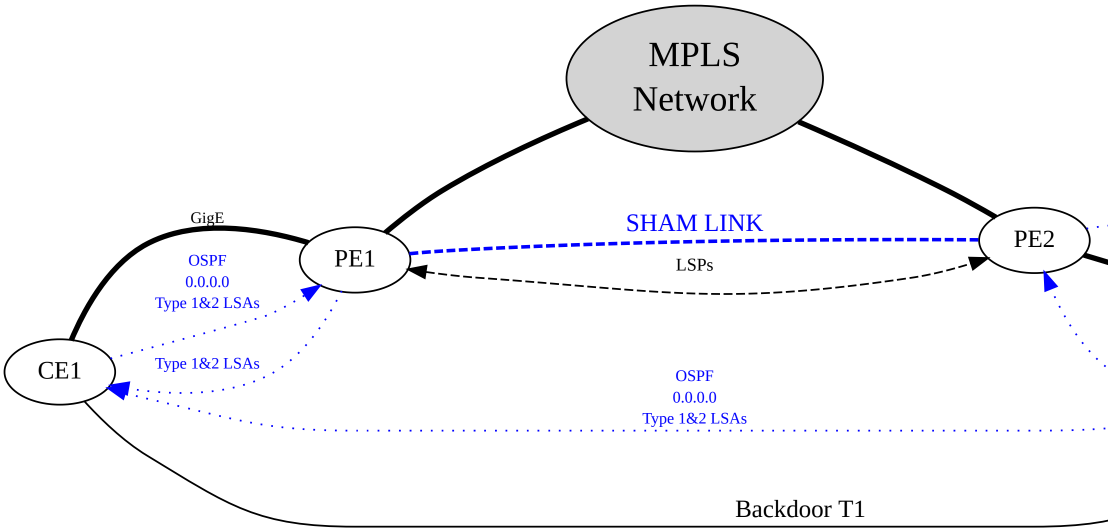
- If a direct link exists between customer sites, these two sites will always see the direct link as a better path due to OSPF route preference
  - Intra-area routes (learned from Type 1 and Type 2 LSAs) are preferred over Inter-Area routes (Type 3 Network Summary LSAs)
  - This is known as a backdoor link
  - Can lead to inefficiencies or cost
    - For example, if Gigabit Etherent circuits are purchased from a provider, and the direct link is a T1
- Sham Links were conceived to solve this problem
  - Sham links are much like virtual links that are setup between PE devices
  - Allows Type 1 and Type 2 LSAs to cross a sham link

Diagram: *Backdoor Problem with L3VPNs running OSPF between Customer Sites*



The Type 1 and 2 LSAs in area 0.0.0.0 at site 1 are advertised into BGP by PE1. The OSPF values are encoded into extended BGP communities and advertised to PE2 via BGP. The LSAs are reconstructed as Type 3 LSAs at the other side by PE2 and flooded into the OSPF area at site 2. However, due to the fact that Type 1 and 2 LSAs can flow over the T1 link between Site 1 and Site 2, these will always be preferred.

Diagram: *Backdoor Problem solved by configuring a sham link*



The Type 1 and 2 LSAs can now flow over the sham link. Thus, preference on whether or not to send traffic between the sites over the T1 as opposed to the L3VPN becomes a matter of the cost of the links.

- There are some other aspects of OSPF in a L3VPN that come into play on the BGP side of the router when the PE devices are advertising the BGP encoded LSAs back and forth to each other in the MPLS network
  - The origin BGP extended community is used to identify where a route originated
    - Used to prevent an OSPF route from being advertised back into the site from which it originated
  - The domain-id BGP extended community identifies the OSPF area from where a route originated
    - If domain-id's match, a route is flooded as a Type 3 LSA as it is assumed the remote CE sites belong to the same area
    - If domain-id's are different, a routed is flooded as a Type 5 LSA

Editorial Note: *As much fun it is to set up OSPF as the routing protocol between the CE and PE in a MPLS L3VPN, it is really quite nasty and should not be attempted by mortal network engineers (really , it is fun). This involves some of the most advanced level routing concepts you'll ever run into between BGP and OSPF. This is also plagued by some strangeish behavior - the Type 1 & 2 LSA converstion to a Type 3 LSA which can be really daunting and misleading. It also suffers from some messy hacks -- the sham link, which sleeps in the bed next to the virtual link and GRE tunnel. It's really good to understand how all this works for one simple reason -- talking people out of using it! There are far better protocols for route distribution between the CE and PE - RIP, BGP and even static routes. As scared as some people are about using BGP, it is for the most part straight forward and predictable -- use it instead of OSPF.*

Unused LSAs

- Several LSAs are presently unused in Junos, but exist in standards documents
- Type 6 LSAs - Group Membership LSA
  - Was defined for use with MOSPF (Multicast Extensions for OSPF)

  - Designed to carry source, group information
  - Nobody ever really implemented MOSPF and was never really used
  - Has been since deprecated
- Type 8 LSAs - External-Attributes-LSA
  - Was intended to allow BGP attributes to be mapped into OSPF LSAs
    - A Type 8 LSA would match a Type 5 LSA and carry the BGP attributes for the prefix in the Type 5 LSA
  - Supposed to be a replacement for BGP inside autonomous systems
  - Nobody ever really implemented it
    - Can OSPF scale to 500,000 routes?
- Type 11 LSAs - Domain Scope LSA
  - Opaque LSA with flooding domain wide aside from stub areas of all types
  - No real applications at this time

Bidirectional Forwarding Detection (BFD)

BFD Overview

- Described in **RFC 5880**
- BFD is a simple hello mechanism that detects network failures
  - Hello packets are sent at regular intervals
  - A BFD neighbor failure is declared after a device stops recieving a reply
    - Timers can also be adaptive
  - Can be set up for faster intervals that most dynamic routing protocols allow
  - Can detect failures between devices connected on any kind of path
    - MPLS LSPs
    - Direct phycsical links
    - Tunnels
    - Multihop routed paths
  - Detects unidirectional paths
- Operates on any data protocol bewteen two systems
  - Network layer
  - Link layer
  - Tunnels

BFD Operation

- Always runs as a point-to-point connection
- Implements a three way handshake for BFD session establishment and teardown
  - Ensures that both systems are aware of change of state
  - Uses an identifying number known as a discriminator to uniquely identify separate BFD sessions between neighbors
  - Allows for negotiation of session parameters
    - Transmit interval
    - Receive interval
    - Operating mode
- Operates in two modes
  - Asynchronous mode
    - Systems periodically send control packets to each other
      - If a number of packets in a row are not received the session is declared down
  - Demand mode
    - Control packets are only send when systems want to verify connectivity
- Both modes have an echo function
  - BFD packets are looped back from one system to the other
- Can authenticate BFD sessions
  - Currently simple passwords, MD5 and SHA1 are supported
- System takes an active or passive role during a BFD session
  - Active system must send BFD control packets
  - Passive system may not send BFD control packets until it has recieved one
  - Both systems in a BFD session may take an active role
- BFD session begins with slow, periodic transmission of control packets
  - When bidirectional communication is obtained the BFD session moves to the Up state
    - BFD session neighbors negotiate if they want to use echo mode, demand mode, and the rate at which packets will be sent
  - If a session goes down, the session resumes slow transmission of packets
    - Once a session is down, it can not come back up until both parties acknowledged to each other that the previous session was down

Configuring BFD for OSPF on Junos

  - BFD sessions can be setup for each interface that is setup to form an adjacency
    - Configuration done under `edit protcols ospf area interface bfd-liveness-detection`
      - Specify the interval that BFD packets are sent, and the minimum the system will accept recieving them with the `minimum-interval` directive where the time is from 1 to 255000 milliseconds
        - Can specify the transmit interval separately with the `transmit-interval minimum-interval` directive
        - Can specify the receive interval separately with the `minimum-receive-interval` directive
        - Not recommended to use timers less than 300 ms, as it can lead to instablilites
        - If the hardware is capable, the BFD session will be run by the forwarding hardware as much as possible
      - Adaptive timers are used by default, but can be disabled with the `no-adaptation` flag
  - The default time for declaring a BFD session dead is 3x the transmission interval
    - Can change the multiplier from 1 to 255 with the `multiplier` command
  - For OSPF, BFD sessions can be configured to start only for OSPF neighbors that are in the Full state by adding the `full-neighbors-only` flag

Example: *Setting up a BFD session on a neighbor session using a 1 second packet interval*

```
admin@J2300-1> edit
Entering configuration mode

[edit]
admin@J2300-1# edit protocols ospf area 0 interface fxp1.13

[edit protocols ospf area 0.0.0.0 interface fxp1.13]
admin@J2300-1# edit bfd-liveness-detection

[edit protocols ospf area 0.0.0.0 interface fxp1.13 bfd-liveness-detection]
admin@J2300-1# set minimum-interval 1000
```

```
[edit protocols ospf area 0.0.0.0 interface fxp1.13 bfd-liveness-detection]
admin@J2300-1# commit
commit complete
```

Troubleshooting BFD Sessions for OSPF

- Operational command `show bfd session` shows BFD sessions and status
  - Can specify `detail` and `extensive` flags to vary amount of detail
  - Can specify `address`, `discriminator` and `prefix` to narrow down results that are returned
  - Can specify summary for a quick overview of BFD sessions

Example: *BFD session details*

```
admin@J2300-1> show bfd session detail
                                              Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
10.0.12.2                Up        fe-0/0/1.12    4.500     1.500       3
 Client OSPF realm ospf-v2 Area 0.0.0.0, TX interval 1.000, RX interval 1.000
 Session up time 00:27:33
 Local diagnostic NbrSignal, remote diagnostic None
 Remote state Up, version 1
                                              Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
10.0.13.3                Down      fe-0/1/1.13    0.000     1.000       3
 Client OSPF realm ospf-v2 Area 0.0.0.0, TX interval 1.000, RX interval 1.000
 Local diagnostic None, remote diagnostic None
 Remote state AdminDown, version 1

2 sessions, 2 clients
Cumulative transmit rate 1.7 pps, cumulative receive rate 0.7 pps
```

- BFD sessions can be in one of the following states

  Init
  : Session is in the process of being setup, three way handshake is underway

  Down
  Session is down

  Up
  Session is up

  AdminDown
  : Session has been administratively disabled, or not configured

- Can restart a BFD session with the operational mode `clear bfd session`
  - Will clear all BFD sessions unless a specific session is identified with a combination of `address` *and `discriminator`*
- Can reset adaptive packet intervals for an active session with `clear bfd adaptation`
  - Will reset adaptation on all BFD sessions unless a specific session is identified with a combination of `address`
    and `discriminator`
- Debugging and tracing on the actual BFD protocol can be done by setting `traceoptions` under `edit protocols bfd`

OSPFv3 in Very Brief

- Separate OSPF protocol for IPv6
  - Can also carry routing information for IPv4
  - First defined in **RFC 2470** and updated in **RFC 5340**
    - Also has numerous other RFC extensions
- Operates in the same basic manner as OSPFv2
  - **However, there are some differences**
    1. OSPFv3 concerns itself with forming adjacencies over common layer2 "links" and is not concerned about the addressing information on each system
    2. Addressing semantics have been removed, more concerned about building topologies
       - IPv6 addresses are only present in the payloads of LSAs
       - Type 1 Router and Type 2 Network LSAs no longer contain addresses, just topology information
       - Neighboring routers are always referred to by Router ID
         - Were previously referred to by interface IP address on multi-access networks
    3. Flooding scopes have been more generalized (like with the Opaque LSAs)
       - Link-local - LSA is only flooded on the local link
       - Area - Flooded throughout a single area
       - AS - Flooded through the entire routing domain
    4. Supports running multiple OSPF instances over a single link
       - Needed to hack the authentication fields to do this with OSPFv2
    5. Use of IPv6 link-local addresess for most neighbor related functions
       - Discovery, autoconfiguration
    6. Removal of authentication from the OSPFv3 protocol
       - Relies on IPSEC to do the authentication work for it
    7. Packet format changes
       - Removal of all addressing symantics
    8. LSA changes
       - Removal of the options field from the LSA header
         - Expanded to 24 bits (from 8) and moved to the body of router-LSAs, network-LSAs, inter-area-router-LSAs, and link-LSAs
       - LSA Type field expanded to 16 bits
         - Upper three bits indicate the flooding scope and how to handle unknown LSA types
       - Addresses expresed in prefix/length insetad of address, subnet mask
         - Default route is now just zeros with a lenght of 0
       - Router LSAs and Network LSAs no longer have address information
       - Routers can now use multiple Router LSAs instead of just one
       - New LSA - the Link LSA
         - Local flooding scope
           - Provide a routers IPv6 link local address to all other routers
           - Advertise all of the IPv6 addresss on a link
       - Type-3 Network Summary LSAs renamed "inter-area-prefix-LSAs"
       - Type-4 ASBR Summary LSAs renamed "inter-area-router-LSAs"
       - Link State ID lost it's addressing symantics and now just serves to identify an LSA
       - New LSA called the intra-area-prefix-LSA
         - Carries all of the addressing information that was in Router LSAs and Network LSAs
       - AS External LSAs now optionally have forwarding addresses and route tags
         - Can now reference other LSAs as well for additional information

9. Unknown LSAs can now be flooded or stored instead of just discarded
10. Stub and NSSA areas were redefined to allow the new LSA Types
11. Neighboring routers are always associated by Router ID

- **Peculiar Similarities**
  - Router IDs, Area IDs, LSA Link State IDs remain 32 bits in size (like an IPv4 address)
    - Cannot be assigned as IPv6 addresses

Configuring OSPFv3 Under Junos

- Done under `edit protocols ospf3`
  - Very similar configuration as OSPFv2

Editorial Note: *I wish we could divorce ourself of IPv4 addresses in routing protocols designed for IPv6*

OSPF Configuration Overview

OSPF Configuration Options for OSPFv2 for Junos 10.0

```
protocols {
   ospf {
   disable;
   export [ policy-names ];
   external-preference preference;
   graceful-restart {
      disable;
      helper-disable;
      notify-duration seconds;
      restart-duration seconds;
   }
   import [ policy-names ];
   no-nssa-abr;
   no-rfc-1583;
   overload {
      timeout seconds;
   }
   preference preference;
   prefix-export-limit;
   rib-group group-name;
   reference-bandwidth reference-bandwidth;
   sham-link {
      local address;
   }
   spf-options {
      delay milliseconds;
      rapid-runs number;
      holddown milliseconds;
   }
   traffic-engineering {
      advertise-unnumbered-interfaces;
      multicast-rpf-routes;
      no-topology;
      shortcuts {
         ignore-lsp-metrics;
         lsp-metric-into-summary;
      }
   }
   traceoptions {
      file filename   ;
      flag flag  ;
   }
   area area-id {
      area-range network/mask-length   ;
      interface interface-name {
         disable;
         authentication {
            md5 key-id {
               key [ key-values ];
               start-time time;
            }
            simple-password key;
         }
         bfd-liveness-detection {
            authentication {
               algorithm algorithm-name;
               key-chain key-chain-name;
               loose-check;
            }
            detection-time {
            threshold milliseconds;
            }
            full-neighbors-only;
            minimum-interval milliseconds;
            minimum-receive-interval milliseconds;
            multiplier number;
            no-adaptation;
            transmit-interval {
               threshold milliseconds;
               minimum-interval milliseconds;
            }
            version (1 | automatic);
         }
         dead-interval seconds;
         demand-circuit;
         flood-reduction;
         hello-interval seconds;
         interface-type type;
         ipsec-sa name;
         ldp-synchronization {
            disable;
            hold-time seconds;
         }
         metric metric;
         neighbor address ;
         passive {
            traffic-engineering {
               remote-node-id address;
            }
```

```
        }
        poll-interval seconds;
        priority number;
        retransmit-interval seconds;
        secondary;
        te-metric metric;
        topology (ipv4-multicast | name) {
           metric metric;
        }
        transit-delay seconds;
     }
     label-switched-path name metric metric;
     network-summary-export [ policy-names ];
     network-summary-import [policy-names ];
     nssa {
        area-range network/mask-length   ;
        default-lsa {
           default-metric metric;
           metric-type type;
           type-7;
        }
        (summaries | no-summaries);
     }
     peer-interface interface-name {
        disable;
        dead-interval seconds;
        demand-circuit;
        flood-reduction;
        hello-interval seconds;
        retransmit-interval seconds;
        transit-delay seconds;
     }
     sham-link-remote address {
        demand-circuit;
        flood-reduction;
        ipsec-sa name;
        metric metric;
        }
     }
        stub  ;
        virtual-link neighbor-id router-id transit-area area-id {
        disable;
        authentication {
           md5 key-id {
              key [ key-values ];
           }
           simple-password key;
        }
        dead-interval seconds;
        demand-circuit;
        flood-reduction;
        hello-interval seconds;
        ipsec-sa name;
        retransmit-interval seconds;
        topology (ipv4-multicast | name) disable;
        transit-delay seconds;
     }
  }
}
```

Final Comments

Whew!

After all of this is said and done, I actually really prefer IS-IS as my IGP.

If I ever find some time, I may expand this section a bit in the following ways:

- Do a in depth section on IPSEC authentication for OSPF
- Come up with a BFD lab
- Go through OSPFv3 in the same amount of detail -- and labs of course

References

RFC 1583 OSPF Version 2

RFC 1587 The OSPF NSSA Option

RFC 1918 Address Allocation for Private Internets

RFC 2328 OSPF Version 2

RFC 2370 The OSPF Opaque LSA Option

RFC 2470 Transmission of IPv6 Packets over Token Ring Networks

RFC 3101 The OSPF Not-So-Stubby Area (NSSA) Option

RFC 3623 Graceful OSPF Restart

RFC 3630 Traffic Engineering (TE) Extensions to OSPF Version 2

RFC 4203 OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)

RFC 5185 OSPF Multi-Area Adjacency

RFC 5250 The OSPF Opaque LSA Option

RFC 5340 OSPF for IPv6

RFC 5709 OSPFv2 HMAC-SHA Cryptographic Authentication

RFC 5786 Advertising a Router's Local Addresses in OSPF Traffic Engineering (TE) Extensions

RFC 5880 Bidirectional Forwarding Detection (BFD)

John T. Moy OSPF: Anatomy of an Internet Routing Protcol, ISBN: 9780201634723

Juniper Networks Junos OS Documentation, Release 10.0

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version 0   |       C       |           Plenty              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Router ID - www.blackhole-networks.com               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Area ID - OSPF Deep Dive                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Checksum  OK          |         Construction          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                                                             -+
|                         PAGE STILL                            |
+-                          UNDER                              -+
|                        CONSTRUCTION                           |
+-                                                             -+
|                   ROUGH AROUND THE EDGES                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Last page update 16 December 2012

Concerning comments and/or errors, please contact **OSPF**

With some occaisonal help from Bluefish.