Ch.2 – OSPF
Single Area OSPF
An IGP

CCNA 3 version 3.0
Advantages of OSPF (1 of 2)

- **OSPF is link-state routing protocol**
  - RIP, IGRP and EIGRP are distance-vector (routing by rumor) routing protocols, susceptible to routing loops, split-horizon, and other issues.

- **OSPF has fast convergence**
  - RIP (30, 180, 180, 240) and IGRP hold-down timers can cause slow convergence.

- **OSPF supports VLSM and CIDR**
  - RIPv1 and IGRP do not
Advantages of OSPF (2 of 2)

• Cisco’s OSPF metric is based on bandwidth
  – RIP is based on hop count
  – IGRP/EIGRP bandwidth, delay, reliability, load
• OSPF only sends out changes when they occur.
  – RIP sends entire routing table every 30 seconds, IGRP every 90 seconds
  – Extra: With OSPF, a router does flood its own LSAs when its age reaches 30 minutes (later)
• OSPF also uses the concept of areas to implement hierarchical routing
• Two open-standard routing protocols to choose from:
  – RIP, simple but very limited, or
  – OSPF, robust but more sophisticated to implement.
• IGRP and EIGRP are Cisco proprietary
OSPFe’s Metric is Cost (Bandwidth)

RFC 2328, OSPF version 2, J. Moy

• “A cost is associated with the output side of each router interface. This cost is configurable by the system administrator. The lower the cost, the more likely the interface is to be used to forward data traffic.”
OSPF’s Metric is Cost (Bandwidth)

Cisco: Cost = Bandwidth

- Cisco uses a default cost of $10^8/$bandwidth
- Default bandwidth of the interface (bandwidth command)
- $10^8$ (100,000,000) as the reference bandwidth: This is used so that the faster links (higher bandwidth) have lower costs.
  - Routing metrics, lower the cost the better the route.
  - I.e. RIP: 3 hops is better than 10 hops
  - Extra: The reference bandwidth can be modified to accommodate networks with links faster than 100,000,000 bps (100 Mbps). See ospf auto-cost reference-bandwidth command.
- Cost of a route is the cumulative costs of the outgoing interfaces from this router to the network.
OSP F's Metric is Cost (Bandwidth)

Cisco default interface costs:

- 56-kbps serial link = 1785
- 64-kbps serial link = 1562  128-kbps serial link = 781
- T1 (1.544-Mbps serial link) = 64
- E1 (2.048-Mbps serial link) = 48
- 4-Mbps Token Ring = 25
- Ethernet = 10
- 16-Mbps Token Ring = 6
- Fast Ethernet = 1
- Problem: Gigabit Ethernet and faster = 1

Notes:

- Cisco routers default to T1 (1.544 Mbps) on all serial interfaces and require manual modification with the bandwidth command.
- \texttt{ospf auto-cost reference-bandwidth} \texttt{reference-bandwidth} can be used to modify the reference-bandwidth for higher speed interfaces
OSPФ’s Metric is Cost (Bandwidth)

Few final notes

• For serial links, if it is not a T1 line, use the bandwidth command to configure the interface to the right bandwidth
• Both sides of the link should have the same bandwidth value
• If you use the command `ospf auto-cost reference-bandwidth reference-bandwidth`, configure all of the routers to use the same value.
Terminology

- **Router ID** – Used to identify the routers in the OSPF network
  - IP address configured with the OSPF `router-id` command (extra)
  - **Highest loopback address** (configuration coming)
  - **Highest active IP address** (any IP address)

- Loopback address has the advantage of never going down, thus diminishing the possibility of having to re-establish adjacencies. (more in a moment)
CCNA 3.0 covers Single Area OSPF as opposed to Multi-Area OSPF

- All routers will be configured in a single area, the convention is to use area 0
- If OSPF has more than one area, it must have an area 0
- CCNP includes Multi-Area OSPF
- We will include a brief introduction to Multi-Area OSPF so you can see the real advantages to using OSPF
Terminology

- **Link**: Interface on a router
- **Link state**: Description of an interface and of its relationship to its neighboring routers, including:
  - IP address/mask of the interface,
  - The type of network it is connected to
  - The routers connected to that network
  - The metric (cost) of that link
- The collection of all the link-states would form a [link-state database](#).
1 – Flooding of link-state information

- The first thing that happens is that each router, on the network announces its own piece of link-state information to all other routers on the network. This includes who their neighboring routers are and the cost of the link between them.

- Example: “Hi, I’m RouterA, and I can reach RouterB via a T1 link and I can reach RouterC via an Ethernet link.”

- Each router sends these announcements to all of the routers in the network.
2. Building a Topological Database
• Each router collects all of this link-state information from other routers and puts it into a topological database.

3. Shortest-Path First (SPF), Dijkstra’s Algorithm
• Using this information, the routers can recreate a topology graph of the network.
• Believe it or not, this is actually a very simple algorithm and I highly suggest you look at it some time, or even better, take a class on algorithms. (Radia Perlman’s book, *Interconnections*, has a very nice example of how to build this graph – she is one of the contributors to the SPF and Spanning-Tree algorithms.)
4. Shortest Path First Tree
   • This algorithm creates an SPF tree, with the router making itself the root of the tree and the other routers and links to those routers, the various branches.

5. Routing Table
   • Using this information, the router creates a routing table.
Extra: Simplified Link State Example

- In order to keep it simple, we will take some liberties with the actual process and algorithm, but you will get the basic idea!
- You are **RouterA** and you have exchanged “Hellos” with:
  - RouterB on your network 11.0.0.0/8 with a cost of 15,
  - RouterC on your network 12.0.0.0/8 with a cost of 2
  - RouterD on your network 13.0.0.0/8 with a cost of 5
  - Have a “leaf” network 10.0.0.0/8 with a cost of 2
- This is your link-state information, which you will flood to all other routers.
- All other routers will also flood their link state information. (OSPF: only within the area)
We now get the following link-state information from **RouterB**:
- Connected to RouterA on network 11.0.0.0/8, cost of 15
- Connected to RouterE on network 15.0.0.0/8, cost of 2
- Have a “leaf” network 14.0.0.0/8, cost of 15

Now, **RouterA** attaches the two graphs...
Link State information from RouterC

We now get the following link-state information from **RouterC**:

- Connected to RouterA on network 12.0.0.0/8, cost of 2
- Connected to RouterD on network 16.0.0.0/8, cost of 2
- Have a “leaf” network 17.0.0.0/8, cost of 2

Now, **RouterA** attaches the two graphs…
We now get the following link-state information from **RouterD**:

- Connected to RouterA on network 13.0.0.0/8, cost of 5
- Connected to RouterC on network 16.0.0.0/8, cost of 2
- Connected to RouterE on network 18.0.0.0/8, cost of 2
- Have a “leaf” network 19.0.0.0/8, cost of 2

Now, **RouterA** attaches the two graphs...
Link State information from RouterE

We now get the following link-state information from RouterE:
- Connected to RouterB on network 15.0.0.0/8, cost of 2
- Connected to RouterD on network 18.0.0.0/8, cost of 10
- Have a “leaf” network 20.0.0.0/8, cost of 2

Now, RouterA attaches the two graphs…
Using the topological information we listed, RouterA has now built a complete topology of the network.

The next step is for the link-state algorithm to find the best path to each node and leaf network.
Extra: Simplified Link State Example

RouterA’s Topological Data Base (Link State Database)

All other routers flood their own link state information to all other routers.

RouterA gets all of this information and stores it in its LSD (Link State Database).

Using the link state information from each router, RouterA runs Dijkstra algorithm to create a SPT. (next)

RouterB:
- Connected to RouterA on network 11.0.0.0/8, cost of 15
- Connected to RouterE on network 15.0.0.0/8, cost of 2
- Has a “leaf” network 14.0.0.0/8, cost of 15

RouterC:
- Connected to RouterA on network 12.0.0.0/8, cost of 2
- Connected to RouterD on network 16.0.0.0/8, cost of 2
- Has a “leaf” network 17.0.0.0/8, cost of 2

RouterD:
- Connected to RouterA on network 13.0.0.0/8, cost of 5
- Connected to RouterC on network 16.0.0.0/8, cost of 2
- Connected to RouterE on network 18.0.0.0/8, cost of 2
- Has a “leaf” network 19.0.0.0/8, cost of 2

RouterE:
- Connected to RouterB on network 15.0.0.0/8, cost of 2
- Connected to RouterD on network 18.0.0.0/8, cost of 10
- Has a “leaf” network 20.0.0.0/8, cost of 2
Choosing the Best Path

- Using the link-state algorithm RouterA can now proceed to find the shortest path to each leaf network.
Choosing the Best Path

Now RouterA knows the best path to each network, creating an SPT (Shortest Path Tree).
SPT Results Get Put into the Routing Table

RouterA’s Routing Table

- 10.0.0.0/8  connected e0
- 11.0.0.0/8  connected s0
- 12.0.0.0/8  connected s1
- 13.0.0.0/8  connected s2
- 14.0.0.0/8  17 s0
- 15.0.0.0/8  17 s1
- 16.0.0.0/8   4 s1
- 17.0.0.0/8   4 s1
- 18.0.0.0/8   14 s1
- 19.0.0.0/8   6 s1
- 20.0.0.0/8   16 s1
Configuring Single Area OSPF

It’s easy!
Enabling OSPF

Rtr(config)# router ospf process-id

- **process-id: 1 - 65,535**
- Cisco feature, which allows you to run multiple, different OSPF routing processes on the same router. *(But don’t!)*
- **Process-id is locally significant,** and does **not** have to be the same number on other routers (they don’t care).
- This is different than the process-id used for IGRP and EIGRP which must be the same on all routers sharing routing information.
- **Extra:** FYI - Cisco IOS limits the number of dynamic routing processes to 30. This is because it limits the number of protocol descriptors to 32, using one for connected route sources, one for static route sources, and 30 for dynamic route sources.
Configuring the Network Command

Rtr(config)# `router ospf process-id`
Rtr(config-router)# `network address wildcard-mask area area-id`

- Tells OSPF which interfaces to enable OSPF on (send and receive updates), matching the address and wildcard mask.
- Also, tells OSPF to include this network in its routing updates
- Wildcard is necessary because OSPF supports CIDR and VLSM
- Most of the time you can just use an inverse-mask (like access-lists) as the network wildcard mask.

Rtr(config-if)# `ip add 10.5.1.1 255.255.255.0`

Rtr(config)# `router ospf 10`
Rtr(config-router)# `network 10.5.1.0 0.0.0.255 area 0`
Network Command and the Wildcard Mask

Merida
Merida(config)#router ospf 1
Merida(config-router)#network 192.168.1.0 0.0.0.255 area 0
Merida(config-router)#network 192.168.2.0 0.0.0.255 area 0
Merida(config-router)#network 192.168.20.0 0.0.0.3 area 0

Vargas
Vargas(config)#router ospf 10
Vargas(config-router)#network 192.168.20.0 0.0.0.3 area 0
Vargas(config-router)#network 192.168.30.0 0.0.0.255 area 0
Lab 3-1 Page 33

Configuring the OSPF Routing Process
Modifying the Cost

Rtr(config-if)# bandwidth 64 = Rtr(config-if)# ip ospf cost 1562

**bandwidth command**
Rtr(config-if)# bandwidth kilobits

(ex: 64 = 64,000bps)

- Changes the default bandwidth metric on a specific interface.
- Used in the $10^8$/bandwidth calculation for cumulating the cost of a route from the router to the network on the outgoing interfaces.
- Does **not** modify the actual speed of the link.

**ip ospf cost command**
RTB(config-if)# ip ospf cost value

(ex: 1562, same as bandwidth = 64kbps)

- Configures the cost metric for a specific interface
- Uses this value for the cost of this interface instead of the $10^8$/bandwidth calculation
- Common for multivendor environments.
Modifying OSPF Cost Metric
Configuring Simple Authentication

A router, by default, trusts that routing information received, has come from a router that should be sending it.

Rtr(config-if)# ip ospf authentication-key passwd
- Configured on an interface
- password = Clear text unless message-digest is used (next)
  - Easily captured using a packet sniffer
  - Passwords do not have to be the same throughout an area, but they must be same between neighbors.

After a password is configured, you enable authentication for the area on all participating area routers with:

Rtr(config-router)# area area area authentication
- Configured for an OSPF area, in ospf router mode.
Configuring Simple Authentication

**RouterA**

interface Serial1

  ip address 192.16.64.1 255.255.255.0

  ip ospf authentication-key secret

!

router ospf 10

  network 192.16.64.0 0.0.0.255 area 0

  network 70.0.0.0 0.255.255.255 area 0

area 0 authentication

**RouterB**

interface Serial2

  ip address 192.16.64.2 255.255.255.0

  ip ospf authentication-key secret

!

router ospf 10

  network 172.16.0.0 0.0.255.255 area 0

  network 192.16.64.0 0.0.0.255 area 0

area 0 authentication
Configuring MD5 Encrypted Authentication

Rtr(config-if)# ip ospf message-digest-key key-id md5 password

- **Key-id** = 1 to 255, must match on each router to authenticate.
- **md5** = Encryption-type
- **password** = encrypted
  - Passwords do not have to be the same throughout an area, but they must be same between neighbors.

After a password is configured, you enable authentication for the area on all participating area routers with:

Rtr(config-router)# area area area authentication [message-digest]

- message-digest option must be used if using message-digest-key
- If optional message-digest is used, a message digest, or hash, of the password is sent.
Configuring MD5 Encrypted Authentication

RouterA
interface Serial1
  ip address 192.16.64.1 255.255.255.0
  ip ospf message-digest-key 1 md5 secret
!
router ospf 10
  network 192.16.64.0 0.0.0.255 area 0
  network 70.0.0.0 0.255.255.255 area 0
  area 0 authentication message-digest

RouterB
interface Serial2
  ip address 192.16.64.2 255.255.255.0
  ip ospf message-digest-key 1 md5 secret
!
router ospf 10
  network 172.16.0.0 0.0.255.255 area 0
  network 192.16.64.0 0.0.0.255 area 0
  area 0 authentication message-digest

70.0.0.0/8

s1
192.16.64.1/24

s2
192.16.64.2/24

172.16.0.0/16
MD5 Encryption

- MD5 authentication, creates a message digest.
- This is scrambled data that is based on the password and the packet contents.
- The receiving router uses the shared password and the packet to re-calculate the digest.
- If the digests match, the router believes that the source of the packet and its contents have not been tampered with.
- In the case of message-digest authentication, the authentication data field contains the key-id and the length of the message digest that is appended to the packet.
- The Message Digest is like a watermark that can’t be faked.
Configuring OSPF Authentication
Configuring and Propagating a Default Route

Router(config)# ip route 0.0.0.0 0.0.0.0 serial0

Router(config)# router ospf 1

Router(config-router)# default-information originate [always]

- If the ASBR has a default route configured (ip route 0.0.0.0 0.0.0.0), the default-information originate command is necessary to advertise 0.0.0.0/0 to the other routers in the area.
- If the default-information originate command is not used, the default “quad-zero” route will not be propagated.
- **Important**: The default route and the default-information originate command are usually only be configured on your “Entrance” or “Gateway” router, the router that connects your network to the outside world.
  –This router is known as the ASBR (Autonomous System Boundary Router)
- **Extra**: The always option will propagate a default “quad-zero” route even if one is not configured on this router.
Default Route Example

Engineering and Marketing will have 0.0.0.0/0 default routes forwarding packets to the Entrance router.

```
Entrance(config)# ip route 0.0.0.0 0.0.0.0 serial 0
Entrance(config)# router ospf 1
Entrance(config-router)# network 10.0.0.0 0.0.0.255 area 0
Entrance(config-router)# network 11.0.0.0 0.0.0.255 area 0
Entrance(config-router)# default-information originate
```
Propagating Default Routes in an OSPF Domain
Network Types

show ip ospf interface

<table>
<thead>
<tr>
<th>Network Type</th>
<th>Table Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadcast multiaccess</td>
<td>Ethernet, Token Ring, or FDDI</td>
</tr>
<tr>
<td>Nonbroadcast multiaccess</td>
<td>Frame Relay, X.25, SMDS</td>
</tr>
<tr>
<td>Point-to-point</td>
<td>PPP, HDLC</td>
</tr>
<tr>
<td>Point-to-multipoint</td>
<td>Configured by an administrator</td>
</tr>
</tbody>
</table>

Unless you are configuring an NBMA network like Frame Relay, this won’t be an issue.

- Many administrators prefer to use **point-to-point** or **point-to-multipoint** for **NBMA** to avoid the DR/BDR and full-mesh issues.
Electing the DR and BDR

- On multi-access, broadcast links (Ethernet), a DR and BDR (if there is more than one router) need to be elected.

- **DR** - Designated Router
- **BDR** – Backup Designated Router
- DR’s serve as collection points for Link State Advertisements (LSAs) on multi-access networks
- A BDR backs up the DR.
- If the IP network is *multi-access*, the OSPF routers will elect one DR and one BDR

- Without a DR, the formation of an adjacency between every attached router would create many unnecessary LSA (Link State Advertisements), \( n(n-1)/2 \) adjacencies.
- Flooding on the network itself would be chaotic.
Electing the DR and BDR

- Router with the **highest Router ID** is elected the DR, next is BDR.
- But like other elections, this one can be rigged.
- The router’s priority field can be set to either ensure that it becomes the DR or prevent it from being the DR.
  
  \[ \text{Rtr(config-if)} \# \text{ip ospf priority } <0-255> \]

  - Higher priority becomes DR/BDR
  - Default = 1
  - 0 = Ineligible to become DR/BDR

- The router can be assigned a priority between 0 and 255, with 0 preventing this router from becoming the DR (or BDR) and 255 ensuring at least a tie. (The highest Router ID would break the tie.)
Electing the DR and BDR

- All other routers, “DROther”, establish adjacencies with only the DR and BDR.
- DROther routers multicast LSAs to only the DR and BDR
  - (224.0.0.6 - all DR routers)
- DR sends LSA to all DROthers
  - (224.0.0.5 - all OSPF routers)

Backup Designated Router - BDR

- Listens, but doesn’t act.
- If LSA is sent, BDR sets a timer.
- If timer expires before it sees the reply (Acknowledgement) from the DR, it becomes the DR and takes over the update process.
- The process for a new BDR begins.
Electing the DR and BDR

What happens when a new router enters the network?

- Once a DR is established, a new router that enters the network with a higher priority or Router ID it will **NOT** become the DR or BDR. Regardless of the priority or Router ID, that router will become a DROther.
- If DR fails, BDR takes over as DR and selection process for new BDR begins.
Configuring a Loopback Address

Rtr(config)# interface loopback 0
Rtr(config-if)# ip add 10.1.1.1 255.255.255.0

• Automatically are “up” and “up”
• Very useful in setting Router IDs as they never go down.
• RouterID is used to identify the routers in the OSPF network
  – IP address configured with the Router-ID command (extra)
  – Highest loopback address
  – Highest active IP address
• Important for DR/BDR elections unless you use the ip ospf priority command (next)
• Extra: Also, useful to configure “virtual” networks that you can ping and route as if they were attached networks.
DR/BDR Elections

Router with the highest Router ID is elected the DR, next is BDR. But like other elections, this one can be rigged.

Rtr(config)# interface fastethernet 0
Rtr(config-if)# ip ospf priority <0-255>

Higher priority becomes DR/BDR
Default = 1
Ineligible to become DR/BDR = 0
Clarifications

- **Hello packets** are still exchanged between all routers on a multi-access segment (DR, BDR, DROthers,....) to maintain neighbor adjacencies.

- **OSPF LSA packets** (coming) are packets which are sent from the BDR/DROthers to the DR, and then from the DR to the BDR/DROthers. (The reason for a DR/BDR.)

- Normal routing of **IP packets** still takes the lowest cost route, which might be between two DROthers.
Couple of notes on link state flooding…

- OSPF is a link state routing protocol and **does not send periodic updates** like RIP.
- OSPF only **floods** link state advertisements when there is a **change in topology** (this includes when routers are first booted).
- OSPF uses **hop-by-hop flooding** of LSAs; an LSA received on one interface are flooded out other OSPF enabled interfaces.
- If a link state entry in the LSDB (Link State DataBase) reaches an age of **60 minutes (MaxAge)** without being updated, it is removed and SPF is recalculated.
- Every **30 minutes (LSRefreshTime)**, OSPF routers flood only their link states to all other routers (in the area).
  - This is known as a “**paranoid update**”
  - These do not trigger SPF recalculations.
- Special note: When a **link goes down** and a router wants to send a LSA to tell other routers to remove this link state, it sends this link state with a **value of 60 minutes (MAXAGE)**.
Configuring OSPF with Loopback Address
Maintaining LSDB and Routing Table Information

Routers forward LSU out other interfaces

Routers rerun SPF to calculate a new routing table

Link-failure or neighbor unreachability detection
In OSPF, link failure can be determined by:
- Physical layer or data link layer – directly reporting a state change on a directly connected interface.
- The Hello subprotocol – The router’s interface has not received a Hello packet from an adjacent neighbor within the OSPF RouterDeadInterval time (40 seconds or 120 seconds on NBMA links).
Periodic updates

• Each LSA entry in the link-state database has its own age timer, with a default of **60 minutes** (3,600 seconds) – this is known as the **MaxAge** value of the LSA entry.

• When an LSA reaches MaxAge, it is flushed from the LSDB.

• Before this happens the LSA has a **Link State Refresh Time (LSRefreshTimer)**, **30 minutes** (1,800 seconds) and when this time expires **the router that originated the LSA** will floods a new LSA to all its neighbors, who will reset the age of the LSA in its LSDB.

• This is also known as the “**periodic update**.”

• These updates do **not** trigger recalculation of the routing table.
# OSPF Packet Types

<table>
<thead>
<tr>
<th>OSPF Packet Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1 - Hello</td>
<td>Establishes and maintains adjacency information with neighbors</td>
</tr>
<tr>
<td>Type 2 - Database description packet (DBD)</td>
<td>Describes the content of the link-state database on an OSPF router</td>
</tr>
<tr>
<td>Type 3 - Link-state request (LSR)</td>
<td>Requests specific pieces of a link-state database</td>
</tr>
<tr>
<td>Type 4 - Link-state update (LSU)</td>
<td>Transports link-state advertisements (LSAs) to neighbor routers</td>
</tr>
<tr>
<td>Type 5 - Link-state acknowledgement (LSAck)</td>
<td>Acknowledges receipt of a neighbor's LSA</td>
</tr>
</tbody>
</table>

## The OSPF Packet Header

<table>
<thead>
<tr>
<th>Version</th>
<th>Type</th>
<th>Packet Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checksum</td>
<td>Authentication Type</td>
<td></td>
</tr>
<tr>
<td>Authentication Data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Hello subprotocol is intended to perform the following tasks within OSPF:

- Dynamic neighbor discovery
- Detect unreachable neighbors
- Ensure two-way communications between neighbors
- Ensure correctness of basic interface parameters between neighbors
- Provide necessary information for the election of the Designated and Backup Designated routers on a LAN segment (coming)
OSPF Hello Protocol

- OSPF routers send **Hellos** on OSPF enabled interfaces:
  - Default **every 10 seconds** on multi-access and point-to-point segments
  - Default **every 30 seconds** on NBMA segments (Frame Relay, X.25, ATM)
  - Most cases OSPF Hello packets are sent as multicast to ALLSPFRouters (224.0.0.5)

- **HelloInterval** - Cisco default = 10 seconds or 30 seconds and can be changed with the command `ip ospf hello-interval`.

- **RouterDeadInterval** - The period in seconds that the router will wait to hear a Hello from a neighbor before declaring the neighbor down.
  - Cisco uses a default of **four-times the HelloInterval** (4 x 10 sec. = 40 seconds, 120 seconds for NBMA) and can be changed with the command `ip ospf dead-interval`.

- **Note**: For routers to become adjacent, the **Hello**, **DeadInterval** and **network types** must be identical between routers or Hello packets get dropped!
Configuring OSPF Timers

Rtr(config-if)# ip ospf hello-interval seconds
Rtr(config-if)# ip ospf dead-interval seconds

- Configured on an interface
- For OSPF routers to be able to exchange information, they must have the same hello intervals and dead intervals.
- By default, the dead interval is 4 times the hello interval, so a router has four chances to send a hello packet before being declared dead. (not required)
- In multi-vendor networks, hello timers may need to be adjusted.
- Do not modify defaults unless you have a compelling need to do so.

Defaults
- On broadcast networks hello interval = 10 seconds, dead interval 40 seconds.
- On non-broadcast networks hello interval = 30 seconds, dead interval 120 seconds.
- **Note**: On some IOS’s, the dead interval automatically changes when the hello interval is modified.
show ip route

```
Router# show ip route
   172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
O  IA  172.16.51.1/32  [110/783] via 172.16.1.2, 00:11:44, FastEthernet0
O       172.16.20.0/24  [110/782] via 172.16.10.6, 00:12:29, Serial0
C       172.16.10.4/30 is directly connected, Serial0
C       172.16.1.0/24 is directly connected, FastEthernet0
O  E2  11.0.0.0/8  [110/20] via 172.16.1.1, 00:11:44, FastEthernet0
O  E1  12.0.0.0/8  [110/782] via 172.16.1.1, 00:11:44, FastEthernet0
```

- **O** = OSPF routes within the same area (intra-area routes)
- **110/number** = Administrative Distance/metric (cumulative $10^8$/bandwidth)
- **E2** = Routes outside of the OSPF routing domain, redistributed into OSPF.
  - Default is E2 with a cost of 20 and does **not** get modified within the OSPF
- **O IA** = OSPF routes from another area (inter-area routes)
- **E1** = Routes outside of the OSPF routing domain and get additional cumulative costs added on by each router, just like other OSPF routes.
show ip ospf

Router#show ip ospf

Routing Process "ospf 1" with ID 192.168.3.1
Supports only single TOS(TOS0) routes
It is an area border router
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 3. Checksum Sum 0x97E3
Number of DCbitless external LSA 0
Number of DoNotAge external LSA 0
Number of areas in this router is 2. 2 normal 0 stub 0 nssa
External flood list length 0

Area BACKBONE(0)
   Number of interfaces in this area is 1
   Area has no authentication
   SPF algorithm executed 8 times
<text omitted>

Area 1
<text omitted>
Router# show ip ospf interface

Ethernet0 is up, line protocol is up
Internet Address 206.202.2.1/24, Area 1
Process ID 1, Router ID 1.2.202.206, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, State BDR, Priority 1
Designated Router (ID) 2.2.202.206, Interface address 206.202.2.2
Backup Designated router (ID) 1.2.202.206, Interface address 206.202.2.1
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:00
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 2.2.202.206 (Designated Router)
Suppress hello for 0 neighbor(s)

Serial0 is up, line protocol is up
Internet Address 206.202.1.2/24, Area 1
Process ID 1, Router ID 1.2.202.206, Network Type POINT_TO_POINT, Cost: 64
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:04
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 2.0.202.206
Suppress hello for 0 neighbor(s)
show ip ospf neighbor

RouterB#show ip ospf neighbor

<table>
<thead>
<tr>
<th>Neighbor ID</th>
<th>Pri</th>
<th>State</th>
<th>Dead Time</th>
<th>Address</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5.202.206</td>
<td>1</td>
<td>FULL/DROther</td>
<td>00:00:33</td>
<td>206.202.0.3</td>
<td>Ethernet0</td>
</tr>
<tr>
<td>1.10.202.206</td>
<td>1</td>
<td>FULL/BDR</td>
<td>00:00:32</td>
<td>206.202.0.4</td>
<td>Ethernet0</td>
</tr>
<tr>
<td>1.0.202.206</td>
<td>1</td>
<td>2WAY/DROther</td>
<td>00:00:30</td>
<td>206.202.0.1</td>
<td>Ethernet0</td>
</tr>
<tr>
<td>1.2.202.206</td>
<td>1</td>
<td>FULL/ -</td>
<td>00:00:32</td>
<td>206.202.1.2</td>
<td>Serial0</td>
</tr>
</tbody>
</table>

- In this example, we are the DR
- DROther may be in FULL or 2 WAY state, both cases are normal.
- Usually if there are multiple DROtherS, they will be in either FULL or 2WAY state but not both.
debug ip ospf adj (adjacency)

Router# debug ip ospf adj
04:19:46: OSPF: Rcv hello from 201.0.0.1 area 0 from FastEthernet0 192.168.20.1
04:19:46: OSPF: 2 Way Communication to 201.0.0.1 on FastEthernet0, state 2WAY
04:19:46: OSPF: End of hello processing
<br>
04:20:22: OSPF: end of Wait on interface FastEthernet0
04:20:22: OSPF: DR/BDR election on FastEthernet0
04:20:22: OSPF: Elect BDR 200.0.0.1
04:20:22: OSPF: Elect DR 200.0.0.1
04:20:22: OSPF: Elect BDR 201.0.0.1
04:20:22: OSPF: Elect DR 200.0.0.1
04:20:22: [DR: 201.0.0.1 (Id)    BDR: 200.0.0.1 (Id)]
04:20:23: OSPF: Rcv DBD from 201.0.0.1 on FastEthernet0 seq 0x2657 opt 0x2 flag 0x7 len 32 mtu 1500 state EXSTART
04:20:23: OSPF: NBR Negotiation Done. We are the SLAVE
04:20:23: OSPF: Send DBD to 201.0.0.1 on FastEthernet0 seq 0x2657 opt 0x2 flag 0 x2 len 92
04:20:23: OSPF: Rcv DBD from 201.0.0.1 on FastEthernet0 seq 0x2658 opt 0x2 flag 0x3 len 72 mtu 1500 state EXCHANGE
<br>
04:20:23: OSPF: Synchronized with 201.0.0.1 on FastEthernet0, state FULL

• Displays adjacency information including Hello processing, DR/BDR election, authentication, and the “Steps to OSPF Operation.”
debug ip ospf events

Router# debug ip ospf events
08:00:56: OSPF: Rcv hello from 201.0.0.1 area 0 from FastEthernet0 192.168.20.1
08:00:56: OSPF: Mismatched hello parameters from 192.168.20.1
08:00:56: Dead R 40 C 20, Hello R 10 C 5 Mask R 255.255.255.252 C 255.255.255.2
52

- Shows much of the same information as debug ip ospf adj in the previous slide including, adjacencies, flooding information, designated router selection, and shortest path first (SPF) calculation.
- This information is also displayed with debug ip ospf events.
- R = Received
- C = Current (?)
Later: show ip ospf database  
(summary of link state database)

Internal#show ip ospf data

OSPF Router with ID (192.168.4.1) (Process ID 1)

Link states within this area, this is what the SPF uses.
Router Link States (Area 0)

<table>
<thead>
<tr>
<th>Link ID</th>
<th>ADV Router</th>
<th>Age</th>
<th>Seq#</th>
<th>Checksum</th>
<th>Link count</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.3.1</td>
<td>192.168.3.1</td>
<td>898</td>
<td>0x80000003</td>
<td>0xCE56</td>
<td>2</td>
</tr>
<tr>
<td>192.168.4.1</td>
<td>192.168.4.1</td>
<td>937</td>
<td>0x80000003</td>
<td>0xFD44</td>
<td>3</td>
</tr>
</tbody>
</table>

Link states of any DRs in this area.
Summary Net Link States (Area 0)

<table>
<thead>
<tr>
<th>Link ID</th>
<th>ADV Router</th>
<th>Age</th>
<th>Seq#</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.16.1.0</td>
<td>192.168.3.1</td>
<td>848</td>
<td>0x80000005</td>
<td>0xD339</td>
</tr>
<tr>
<td>172.16.51.1</td>
<td>192.168.3.1</td>
<td>843</td>
<td>0x80000001</td>
<td>0x8B329</td>
</tr>
</tbody>
</table>

Link states summaries of links outside this area.  (No SPF)
Summary ASB Link States (Area 0)

<table>
<thead>
<tr>
<th>Link ID</th>
<th>ADV Router</th>
<th>Age</th>
<th>Seq#</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.1</td>
<td>192.168.3.1</td>
<td>912</td>
<td>0x80000003</td>
<td>0x93CC</td>
</tr>
</tbody>
</table>

Link states summaries of links external routes.  (No SPF)
Type-5 AS External Link States

<table>
<thead>
<tr>
<th>Link ID</th>
<th>ADV Router</th>
<th>Age</th>
<th>Seq#</th>
<th>Checksum</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.0.0.0</td>
<td>192.168.1.1</td>
<td>1302</td>
<td>0x80000001</td>
<td>0x3FEA</td>
<td>0</td>
</tr>
<tr>
<td>12.0.0.0</td>
<td>192.168.1.1</td>
<td>1303</td>
<td>0x80000001</td>
<td>0x32F6</td>
<td>0</td>
</tr>
</tbody>
</table>
OSPF Configuration Commands - Review

**Required Commands:**

Rtr(config)# router ospf process-id
Rtr(config-router)# network address wildcard-mask area area-id

**Optional Commands:**

Rtr(config-router)# default-information originate (Send default)
Rtr(config-router)# area area authentication (Plain authen.)
Rtr(config-router)# area area authentication message-digest (md5 authen.)

Rtr(config)# interface loopback number (Configure lo as RtrID)
Rtr(config)# interface type slot/port
Rtr(config-if)# ip ospf priority <0-255> (DR/BDR election)
Rtr(config-if)# bandwidth kbps (Modify default bandwidth)
RTB(config-if)# ip ospf cost cost (Modify inter. cost)
Rtr(config-if)# ip ospf hello-interval seconds (Modify Hello)
Rtr(config-if)# ip ospf dead-interval seconds (Modify Dead)
Rtr(config-if)# ip ospf authentication-key-key passwd (Plain/md5authen)
Rtr(config-if)# ip ospf message-digest-key key-id md5 password
For more information on OSPF, link-state routing protocol, Dijkstra’s algorithm and routing in general, check out these sources.

- I highly recommend Jeff Doyle’s book for those new to OSPF.