

OSPF Troubleshooting

OSPF Data Structures

Data Structure	Description
OSPF interface table	All the router interfaces that have been configured to participate in an OSPF routing process are listed in this table.
OSPF neighbor table	OSPF neighbors learned via Hello packets are present in this table. A neighbor is removed from this table if Hellos have not been heard from the neighbor within the dead time interval. Additionally, a neighbor is removed from this table if the interface associated with the neighbor goes down.
OSPF link-state database	This data structure contains topology information for all areas in which a router participates, in addition to information about how to route traffic to networks residing in other areas or autonomous systems.
OSPF Routing Information Base	The OSPF Routing Information Base (RIB) stores the results of the OSPF shortest path first (SPF) calculations.

OSPF Link-state database

- Stores full topology info for a specific OSPF area.
- A router in multiple areas keeps one database per area.
- All routers in an area must hold identical link-state databases.



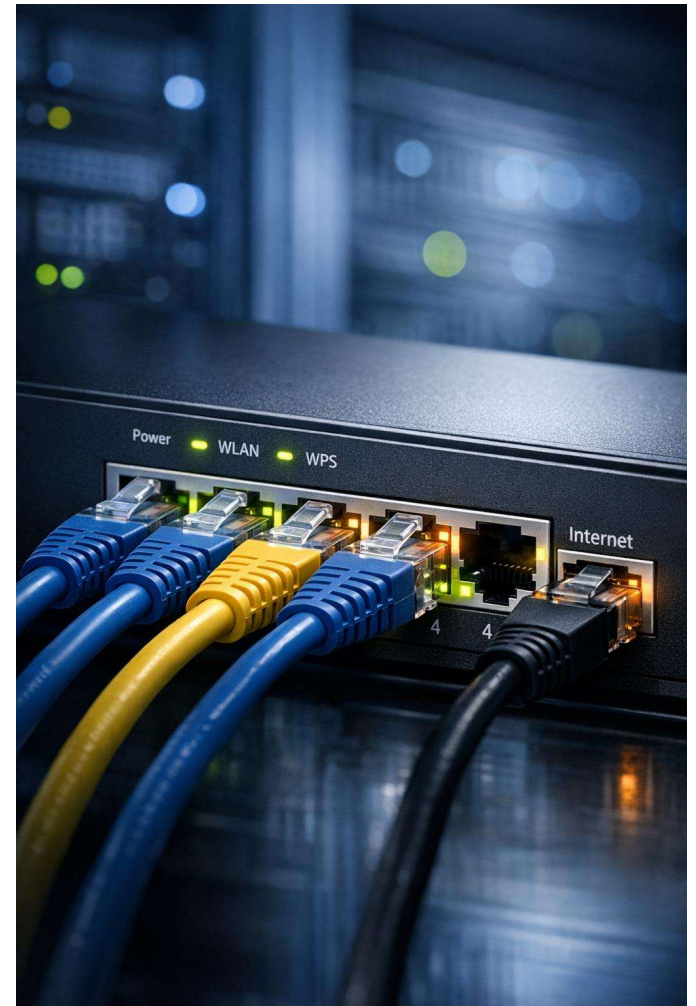
OSPF Operation

- OSPF is a link-state protocol receiving LSAs from adjacent routers.
- Dijkstra SPF algorithm computes shortest paths from LSA data.



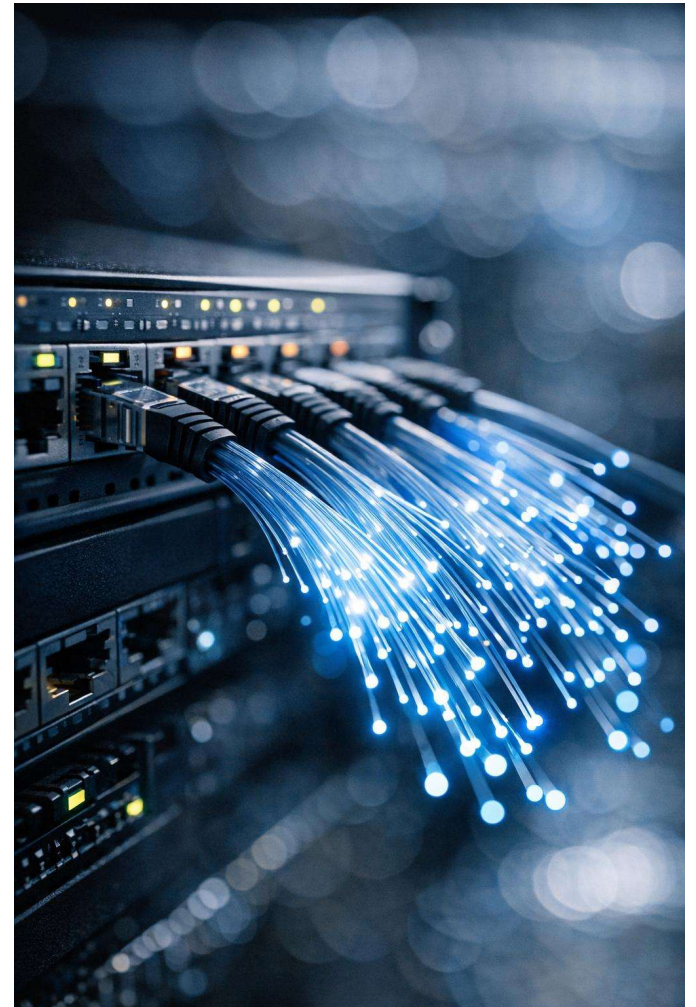
OSPF Operation

- Large OSPF networks are split into multiple areas.
- A backbone area 0 must exist; all areas connect to it.
- Use a virtual link if an area is not adjacent to area 0.



OSPF Metric

- OSPF metric = cost, based on bandwidth.
- Formula: $\text{cost} = 100,000,000 / \text{bandwidth (kbps)}$.



Designated Router

- Multiaccess networks can host many routers on one segment.
- Instead of full mesh, a Designated Router (DR) is elected.
- Other routers form adjacency only with the DR.



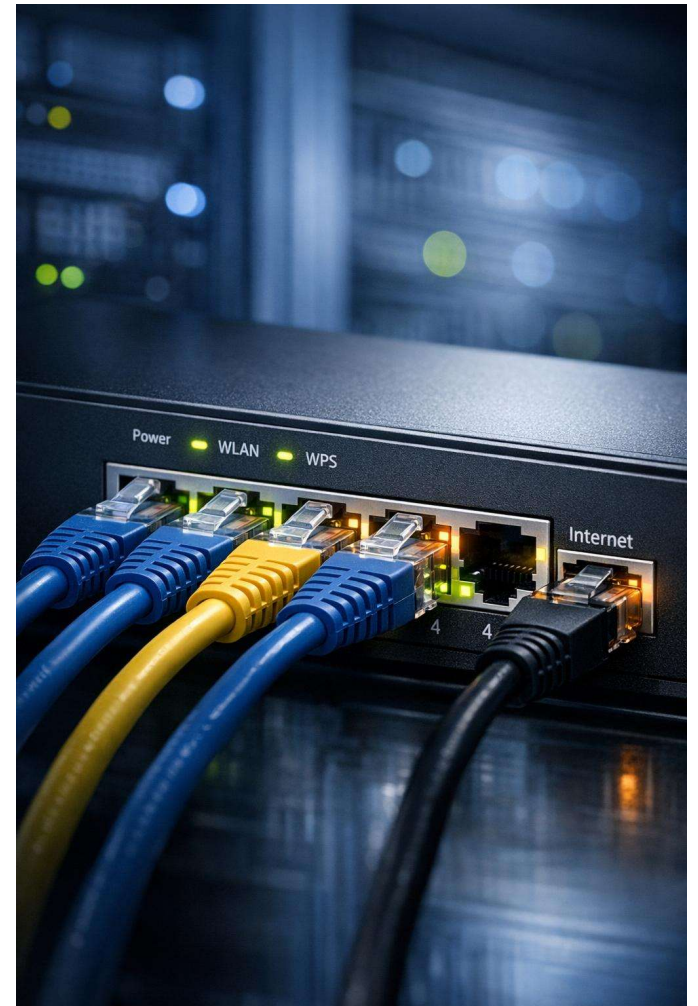
Designated Router

- DR elected by highest router priority.
- Tie-break: highest OSPF router ID.
- Router ID = loopback IP, highest active IP, or static config.

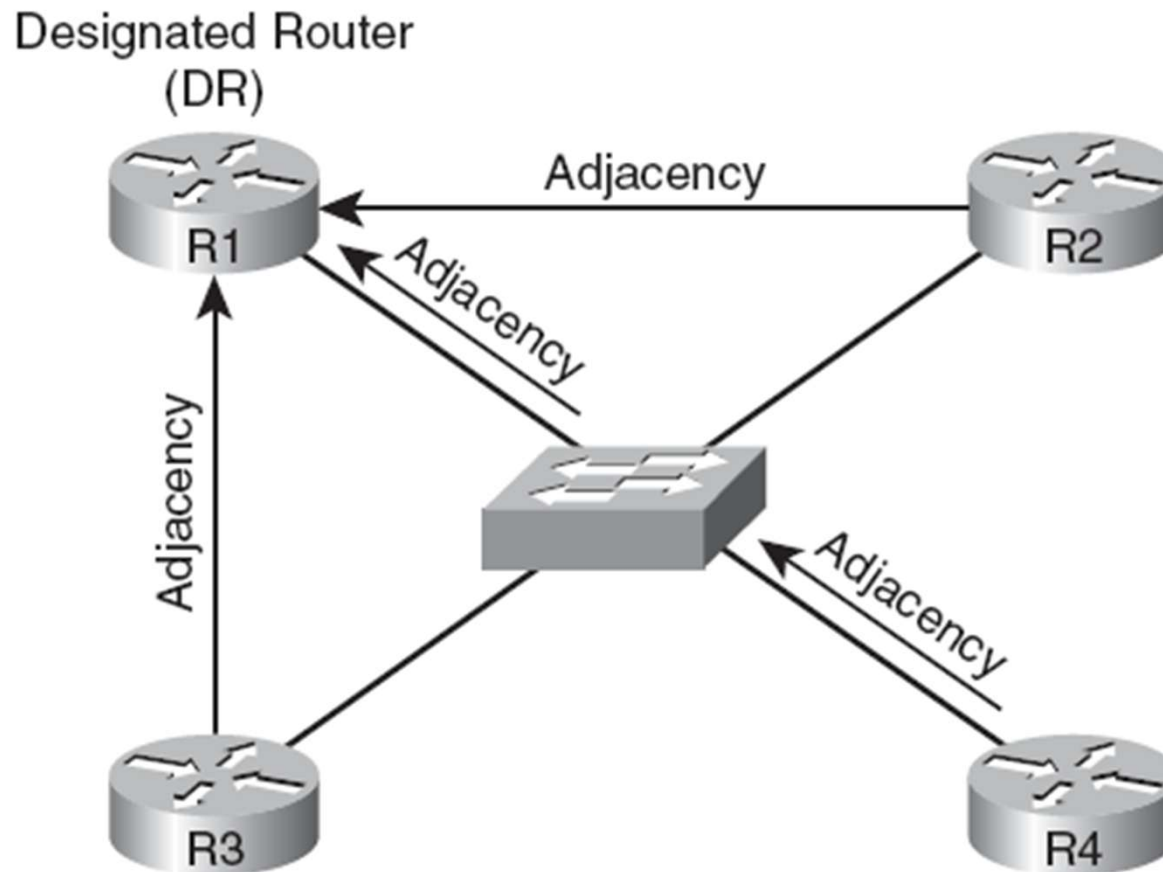


Backup Designated Router

- A Backup DR (BDR) is also elected.
- Routers form adjacency with the BDR for failover when DR fails.



Designated Router Adjacencies



OSPF Router Types

- Internal router: all connected networks in the same area.
- ABR: connects multiple areas, holds a database per area.
- Backbone router: at least one interface in area 0.
- ASBR: connects OSPF to another autonomous system.



OSPF LSA Types

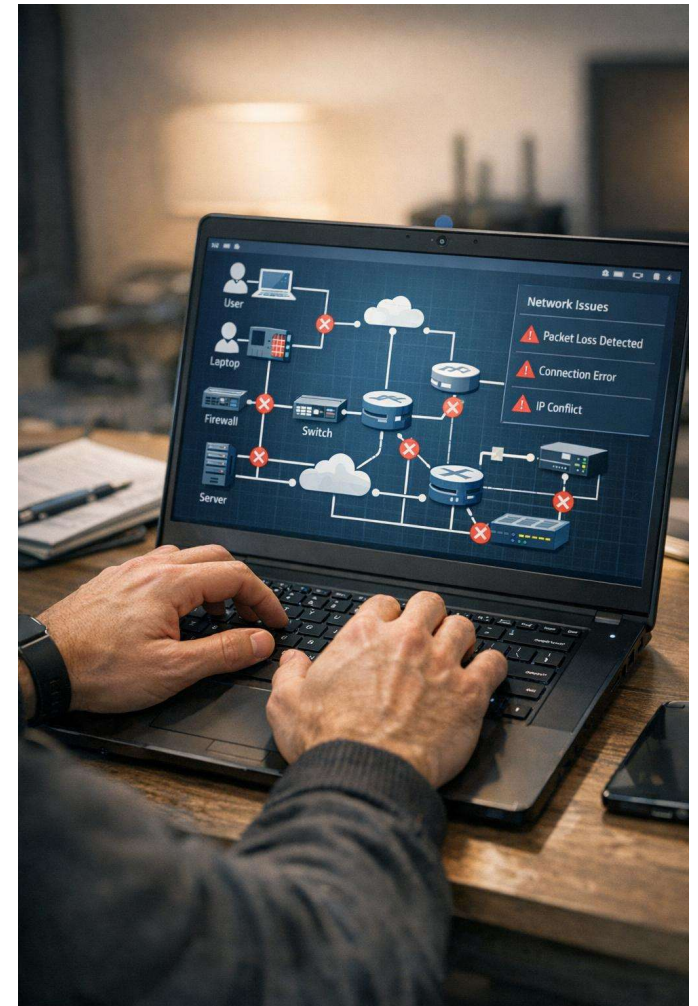
LSA Type	Description
1	All OSPF routers source Type 1 LSAs. These advertisements list information about directly connected subnets, the OSPF connection types of a router, and the known OSPF adjacencies of a router. A Type 1 LSA is not sent out of its local area.
2	The designated router on a multiaccess network sends a Type 2 LSA for that network if the network contains at least two routers. A Type 2 LSA contains a listing of routers connected to the multiaccess network and, like a Type 1 LSA, is constrained to its local area.
3	A Type 3 LSA is sourced by an ABR. Each Type 3 LSA sent into an area contains information about a network reachable in a different area. Note that network information is exchanged only between the backbone area and a nonbackbone area, as opposed to being exchanged between two nonbackbone areas.
4	Similar to a Type 3 LSA, a Type 4 LSA is sourced by an ABR. However, instead of containing information about OSPF networks, a Type 4 LSA contains information stating how to reach an ASBR.
5	A Type 5 LSA is sourced by an ASBR and contains information about networks reachable outside the ospf domain. A Type 5 LSA is sent to all OSPF areas, except for stub areas. Note: The ABR for a stub area sends default route information into the stub area, rather than the network-specific Type 5 LSAs.
7	A Type 7 LSA is sourced from a router within a not-so-stubby-area (NSSA). Whereas a stub area does not connect to an external autonomous system, an NSSA can. Those external routes are announced by an ABR of the NSSA using Type 7 LSAs. Like a stub area, however, external routes known to another OSPF area are not forwarded into an NSSA.

OSPF Network Types

	Broadcast	Nonbroadcast	Point-to-Point	Point-to-Multipoint
Characteristics	Default OSPF network type on LAN interfaces	Default OSPF network type on Frame Relay serial interfaces	Default OSPF network type on non-Frame Relay serial interfaces	Can be configured on any interface
	Neighbors automatically discovered	Neighbors statically configured	Routers at each end of a link form adjacencies	Neighbors automatically determined
	All routers on same subnet	All routers on same subnet	Each point-to-point link on a separate subnet	All routers on same subnet
	Has a designated router	Has a designated router	Does not have a designated router	Does not have a designated router

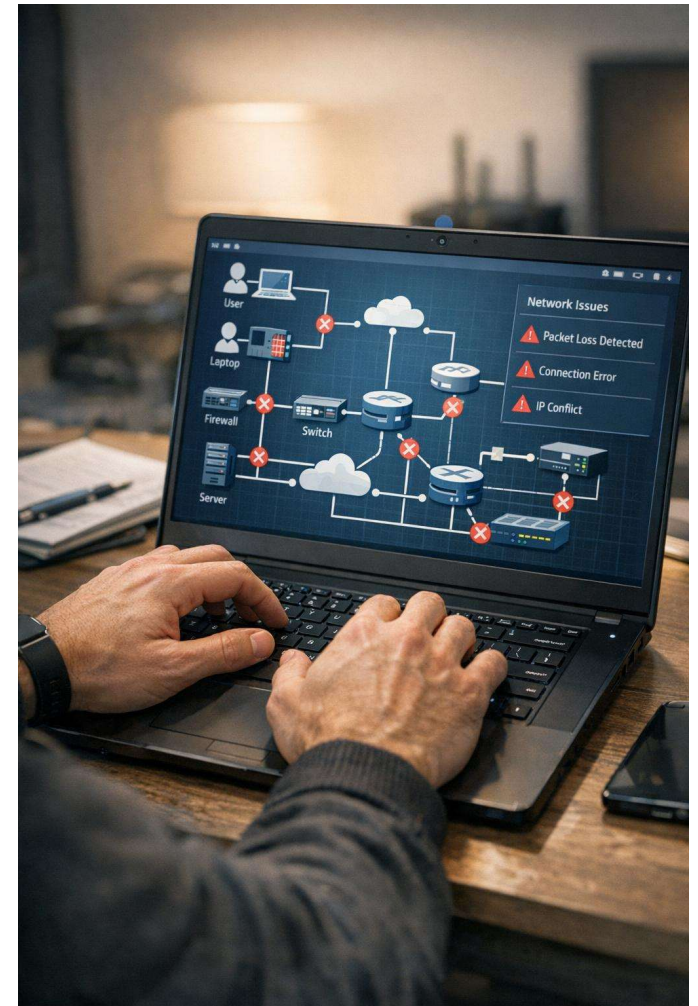
Forming an OSPF Adjacency

- Routers form adjacency via Hello packets — key timers must match.
- Hello timer: 10s (broadcast/p2p), 30s (NBMA/p2mp).
- Dead timer: 40s (broadcast/p2p), 120s (NBMA/p2mp).



Forming an OSPF Adjacency

- Interface down or not running OSPF
- Mismatched timers, area, or network type
- Different subnets or duplicate router IDs
- Passive interface or MTU mismatch
- Authentication mismatch



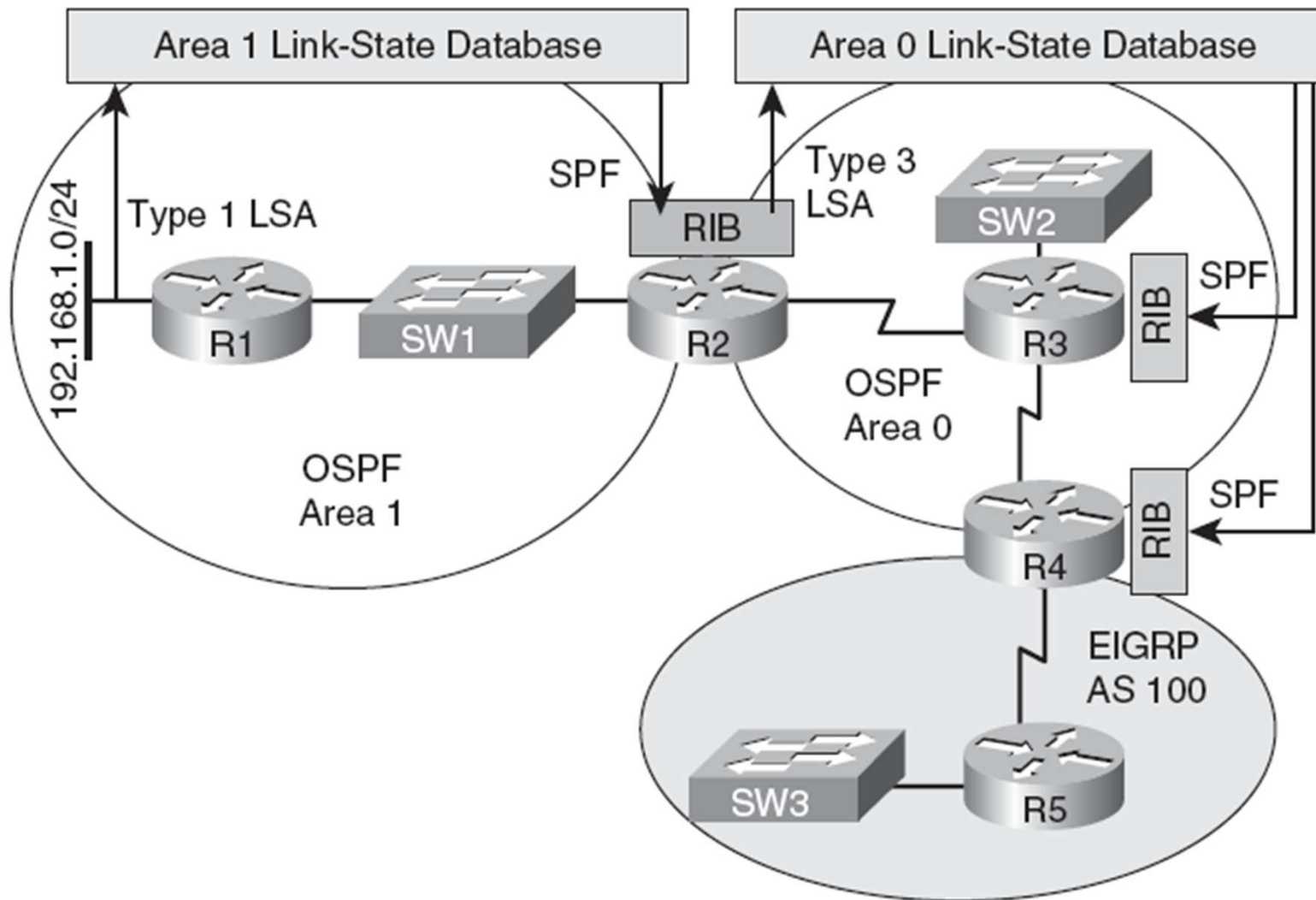
Adjacency States

State	Description
Down	This state indicates that no Hellos have been received from a neighbor.
Attempt	This state occurs after a router sends a unicast Hello (as opposed to a multicast Hello) to a configured neighbor and has not yet received a Hello from that neighbor.
Init	This state occurs on a router that has received a Hello message from its neighbor; however, the OSPF router ID of the receiving router was not contained in the Hello message. If a router remains in this state for a long period, something is probably preventing that router from correctly receiving Hello packets from the neighboring router.
2-Way	This state occurs when two OSPF routers have received Hello messages from each other, and each router saw its own OSPF router ID in the Hello message it received. The 2-Way state could be a final state for a multiaccess network if the network has already elected a designated router and a backup designated router.

Adjacency States

ExStart	This state occurs when the designated and nondesignated routers of a multiaccess network begin to exchange information with other routers in the multiaccess network. If a router remains in this state for a long period, a maximum transmission unit (MTU) mismatch could exist between the neighboring routers, or a duplicate OSPF router ID might exist.
Exchange	This state occurs when the two routers forming an adjacency send one another database descriptor (DBD) packets containing information about a router's link-state database. Each router compares the DBD packets received from the other router to identify missing entries in its own link-state database. Like the ExStart state, if a router remains in the Exchange state for a long period, the issue could be an MTU mismatch or a duplicate OSPF router ID.
Loading	Based on the missing link-state database entries identified in the Exchange state, the Loading state occurs when each neighboring router requests the other router to send those missing entries. If a router remains in this state for a long period, a packet might have been corrupted, or a router might have a memory corruption issue. Alternatively, it is possible that such a condition could result from the neighboring routers having an MTU mismatch.
Full	This state indicates that the neighboring OSPF routers have successfully exchanged their link-state information with one another, and an adjacency has been formed.

Tracking OSPF Advertisements Through a Network



OSPF Troubleshooting Commands

Command	Routing Component or Data Structure	Description
<code>show ip ospf interface [brief]</code>	OSPF interface table	This command displays all of a router's interfaces configured to participate in an OSPF routing process. The <code>brief</code> option provides a more concise view of OSPF interface information.
<code>show ip ospf neighbor</code>	OSPF neighbor table	This command displays the state of OSPF neighbors learned off a router's active OSPF interfaces.
<code>show ip ospf database</code>	OSPF link-state database	This command displays the LSA headers contained in a router's OSPF link-state database.
<code>show ip ospf statistics</code>	OSPF RIB	This command provides information about how frequently a router is executing the SPF algorithm. Additionally, this command shows when the SPF algorithm last ran.

OSPF Troubleshooting Commands

<code>debug ip ospf monitor</code>	OSPF RIB	This command provides real-time updates showing when a router's SPF algorithm is scheduled to run.
<code>debug ip routing</code>	IP routing table	This command displays updates that occur in a router's IP routing table. Therefore, this command is not specific to OSPF.
<code>show ip route ospf</code>	IP routing table	This command shows routes known to a router's IP routing table that were learned via OSPF.
<code>debug ip ospf packet</code>	Exchanging OSPF information with neighbors	This command shows the transmission and reception of OSPF packets in real time. This command is useful for monitoring Hello messages.
<code>debug ip ospf adj</code>	Exchanging OSPF information with neighbors	This command provides real-time updates about the formation of an OSPF adjacency.

OSPF Troubleshooting Commands

Command	Routing Component or Data Structure	Description
<code>debug ip ospf events</code>	Exchanging OSPF information with neighbors	This command shows real-time information about OSPF events, including the transmission and reception of Hello messages and LSAs. This command might be useful on a router that appears to be ignoring Hello messages received from a neighboring router.
<code>show ip ospf virtual-links</code>	OSPF interface table	This command provides information about the status of OSPF virtual links that are required for areas not physically adjacent to the backbone area (that is, area 0).

OSPF Troubleshooting Commands

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

Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs F/C
Lo0	1	0	10.2.2.2/32	1	LOOP	0/0
Se1/0.2	1	0	172.16.2.1/30	64	P2P	1/1
Se1/0.1	1	0	172.16.1.2/30	64	P2P	1/1
Fa0/0	1	0	192.168.0.22/24	10	DR	1/1

```
R2# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.4.4.4	0	FULL/ -	00:00:38	172.16.2.2	Serial1/0.2
10.3.3.3	0	FULL/ -	00:00:38	172.16.1.1	Serial1/0.1
10.1.1.1	1	FULL/BDR	00:00:36	192.168.0.11	FastEthernet0/0

OSPF Troubleshooting Commands

```
R2# show ip ospf database
```

```
      OSPF Router with ID (10.2.2.2) (Process ID 1)
```

```
      Router Link States (Area 0)
```

Link ID	ADV Router	Age	Seq#	Checksum	Link count
10.1.1.1	10.1.1.1	969	0x8000000C	0x0092ED	3
10.2.2.2	10.2.2.2	968	0x80000013	0x00FD7D	6
10.3.3.3	10.3.3.3	1598	0x80000007	0x00DAEA	6
10.4.4.4	10.4.4.4	1597	0x80000007	0x0009B1	6

```
      Net Link States (Area 0)
```

Link ID	ADV Router	Age	Seq#	Checksum
10.1.2.2	10.4.4.4	1622	0x80000001	0x00CE1D
192.168.0.22	10.2.2.2	968	0x8000000B	0x008AF8

OSPF Troubleshooting Commands

```
R2# show ip ospf statistics
```

```
OSPF Router with ID (10.2.2.2) (Process ID 1)
```

```
Area 0: SPF algorithm executed 23 times
```

```
Summary OSPF SPF statistic
```

```
SPF calculation time
```

Delta T	Intra	D-Intra	Summ	D-Summ	Ext	D-Ext	Total	Reason
00:21:00	12	0	0	0	0	0	16	R, N,
00:20:50	16	0	0	0	0	0	20	R,
00:19:40	12	4	0	0	0	0	20	R, N,
00:19:30	12	0	0	0	0	0	16	R, N,
00:19:20	16	0	0	0	4	0	24	R,

OSPF Troubleshooting Commands

```
R2# debug ip routing
IP routing debugging is on
*Mar  1 00:29:58.163: RT: del 10.1.1.1/32 via 192.168.0.11, ospf metric [110/11]
*Mar  1 00:29:58.163: RT: delete subnet route to 10.1.1.1/32
*Mar  1 00:29:58.163: RT: NET-RED 10.1.1.1/32
*Mar  1 00:30:08.175: RT: SET_LAST_RDB for 10.1.1.1/32
      NEW rdb: via 192.168.0.11
*Mar  1 00:30:08.179: RT: add 10.1.1.1/32 via 192.168.0.11, ospf metric [110/11]
*Mar  1 00:30:08.183: RT: NET-RED 10.1.1.1/32
```

```
R2# show ip route ospf
      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O       10.1.3.0/30 [110/128] via 172.16.2.2, 00:00:20, Serial1/0.2
          [110/128] via 172.16.1.1, 00:00:20, Serial1/0.1
O       10.3.3.3/32 [110/65] via 172.16.1.1, 00:00:20, Serial1/0.1
O       10.1.2.0/24 [110/74] via 172.16.2.2, 00:00:20, Serial1/0.2
          [110/74] via 172.16.1.1, 00:00:20, Serial1/0.1
O       10.1.1.1/32 [110/11] via 192.168.0.11, 00:00:20, FastEthernet0/0
O       10.4.4.4/32 [110/65] via 172.16.2.2, 00:00:20, Serial1/0.2
```

OSPF Troubleshooting Commands

```
R2# show ip route ospf
  10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O       10.1.3.0/30 [110/128] via 172.16.2.2, 00:00:20, Serial1/0.2
        [110/128] via 172.16.1.1, 00:00:20, Serial1/0.1
O       10.3.3.3/32 [110/65] via 172.16.1.1, 00:00:20, Serial1/0.1
O       10.1.2.0/24 [110/74] via 172.16.2.2, 00:00:20, Serial1/0.2
        [110/74] via 172.16.1.1, 00:00:20, Serial1/0.1
O       10.1.1.1/32 [110/11] via 192.168.0.11, 00:00:20, FastEthernet0/0
O       10.4.4.4/32 [110/65] via 172.16.2.2, 00:00:20, Serial1/0.2
```

OSPF Troubleshooting Commands

```
R2# debug ip ospf packet
```

```
OSPF packet debugging is on
```

```
*Mar 1 00:30:42.995: OSPF: rcv. v:2 t:1 l:48 rid:10.1.1.1  
aid:0.0.0.0 chk:5422 aut:0 auk: from FastEthernet0/0  
*Mar 1 00:30:45.039: OSPF: rcv. v:2 t:4 l:76 rid:10.1.1.1  
aid:0.0.0.0 chk:E9B4 aut:0 auk: from FastEthernet0/0  
*Mar 1 00:30:45.179: OSPF: rcv. v:2 t:1 l:48 rid:10.3.3.3  
aid:0.0.0.0 chk:D294 aut:0 auk: from Serial1/0.1  
*Mar 1 00:30:45.363: OSPF: rcv. v:2 t:1 l:48 rid:10.4.4.4  
aid:0.0.0.0 chk:D192 aut:0 auk: from Serial1/0.2  
*Mar 1 00:30:47.727: OSPF: rcv. v:2 t:5 l:44 rid:10.3.3.3  
aid:0.0.0.0 chk:2EF0 aut:0 auk: from Serial1/0.1
```

OSPF Troubleshooting Commands

```
R2# debug ip ospf adj
OSPF adjacency events debugging is on
*Mar 1 00:31:44.719: OSPF: Rcv LS UPD from 10.3.3.3 on Serial1/0.1 length 100
  LSA count 1
*Mar 1 00:31:57.475: OSPF: Cannot see ourself in hello from 10.4.4.4 on
  Serial1/0.2, state INIT
*Mar 1 00:31:57.807: OSPF: 2 Way Communication to 10.4.4.4 on Serial1/0.2,
  state 2WAY
*Mar 1 00:31:57.811: OSPF: Send DBD to 10.4.4.4 on Serial1/0.2 seq 0x7FD opt
  0x52 flag 0x7 len 32
*Mar 1 00:31:57.815: OSPF: Rcv DBD from 10.4.4.4 on Serial1/0.2 seq 0xDF9 opt
  0x52 flag 0x7 len 32 mtu 1500 state EXSTART
*Mar 1 00:31:57.819: OSPF: NBR Negotiation Done. We are the SLAVE
*Mar 1 00:31:57.819: OSPF: Send DBD to 10.4.4.4 on Serial1/0.2 seq 0xDF9 opt
  0x52 flag 0x2 len 152
*Mar 1 00:31:57.983: OSPF: Build router LSA for area 0, router ID 10.2.2.2, seq
  0x80000014
*Mar 1 00:31:58.191: OSPF: Rcv DBD from 10.4.4.4 on Serial1/0.2 seq 0xDFA opt
  0x52 flag 0x3 len 152 mtu 1500 state EXCHANGE
*Mar 1 00:31:58.195: OSPF: Send DBD to 10.4.4.4 on Serial1/0.2 seq 0xDFA opt
  0x52 flag 0x0 len 32
*Mar 1 00:31:58.199: OSPF: Rcv LS UPD from 10.3.3.3 on Serial1/0.1 length 112
  LSA count 1
*Mar 1 00:31:58.455: OSPF: Rcv DBD from 10.4.4.4 on Serial1/0.2 seq 0xDFB opt
  0x52 flag 0x1 len 32 mtu 1500 state EXCHANGE
*Mar 1 00:31:58.459: OSPF: Exchange Done with 10.4.4.4 on Serial1/0.2
*Mar 1 00:31:58.463: OSPF: Synchronized with 10.4.4.4 on Serial1/0.2, state FULL
*Mar 1 00:31:58.463: %OSPF-5-ADJCHG: Process 1, Nbr 10.4.4.4 on Serial1/0.2 from
  LOADING to FULL, Loading Done
```

OSPF Troubleshooting Commands

```
R2# debug ip ospf events
OSPF events debugging is on
*Mar  1 00:32:23.007: OSPF: Rcv hello from 10.1.1.1 area 0 from FastEthernet0/0
192.168.0.11
*Mar  1 00:32:23.011: OSPF: End of hello processing
*Mar  1 00:32:25.195: OSPF: Rcv hello from 10.3.3.3 area 0 from Serial1/0.1
172.16.1.1
*Mar  1 00:32:25.199: OSPF: End of hello processing
*Mar  1 00:32:27.507: OSPF: Rcv hello from 10.4.4.4 area 0 from Serial1/0.2
172.16.2.2
*Mar  1 00:32:27.511: OSPF: End of hello processing
*Mar  1 00:32:28.595: OSPF: Send hello to 224.0.0.5 area 0 on Serial1/0.1 from
172.16.1.2
*Mar  1 00:32:28.687: OSPF: Send hello to 224.0.0.5 area 0 on Serial1/0.2 from
172.16.2.1
*Mar  1 00:32:29.163: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet0/0
from 192.168.0.22
*Mar  1 00:32:32.139: OSPF: Rcv LS UPD from 10.3.3.3 on Serial1/0.1 length 100
LSA count 1
*Mar  1 00:32:33.007: OSPF: Rcv hello from 10.1.1.1 area 0 from FastEthernet0/0
192.168.0.11
*Mar  1 00:32:33.011: OSPF: End of hello processing
*Mar  1 00:32:35.087: OSPF: Rcv hello from 10.3.3.3 area 0 from Serial1/0.1
172.16.1.1
*Mar  1 00:32:35.091: OSPF: End of hello processing
```

OSPF Troubleshooting Commands

```
R1# show ip ospf virtual-links
```

```
Virtual Link OSPF_VL5 to router 10.2.2.2 is up
```

```
Run as demand circuit
```

```
DoNotAge LSA allowed.
```

```
Transit area 1, via interface FastEthernet0/1, Cost of using 1
```

```
Transmit Delay is 1 sec, State POINT_TO_POINT,
```

```
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
```

```
Hello due in 00:00:00
```

```
Adjacency State FULL (Hello suppressed)
```

```
Index 1/2, retransmission queue length 0, number of retransmission 0
```



Route Redistribution Troubleshooting

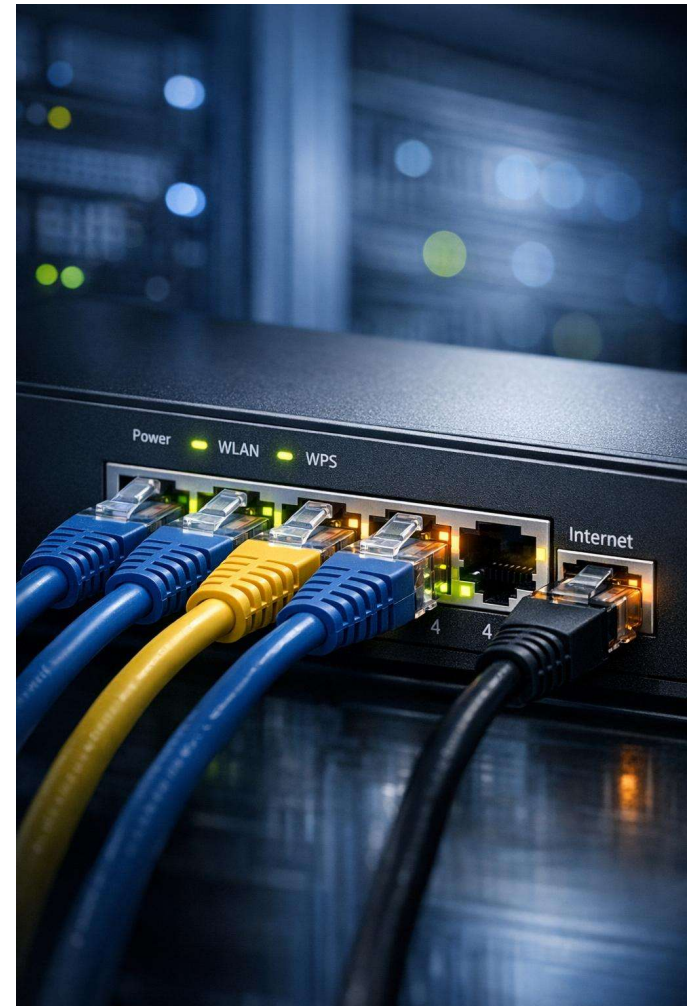
Route Redistribution Overview

- Redistribution injects routes learned one way (static, connected, or via a protocol) into another routing protocol.

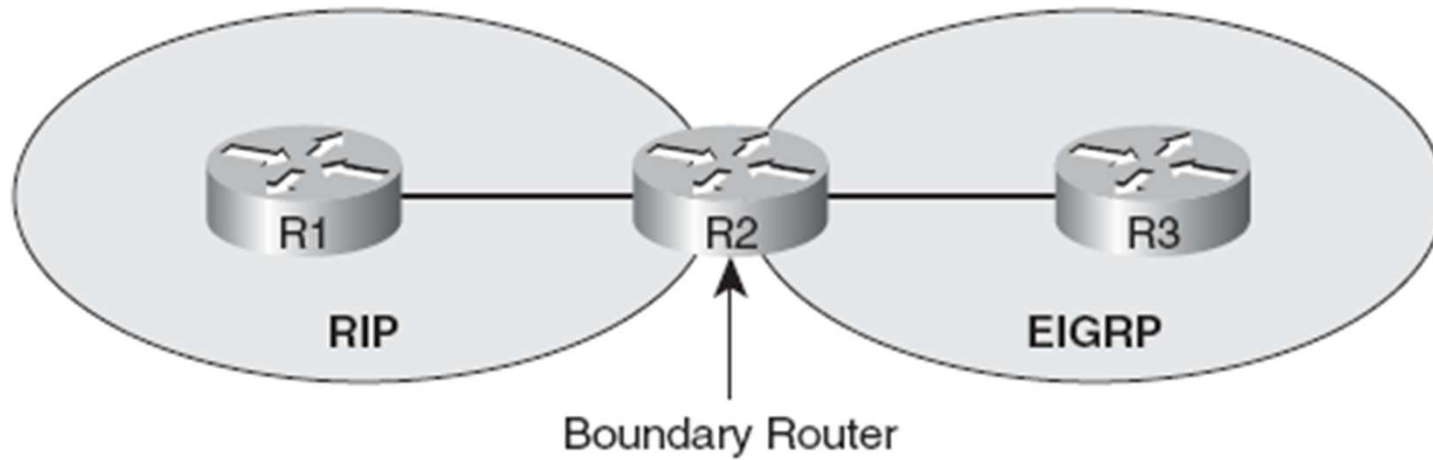


Route Redistribution Overview

- A boundary router sits between routing domains.
- It can redistribute static, connected, or protocol-learned routes into another protocol.

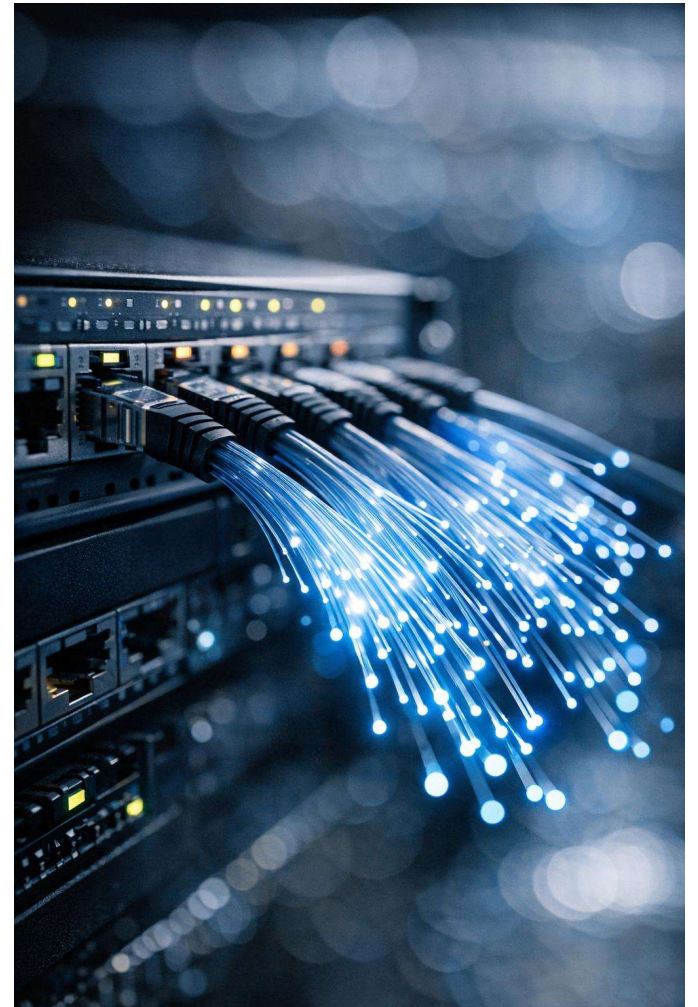


Boundary Router

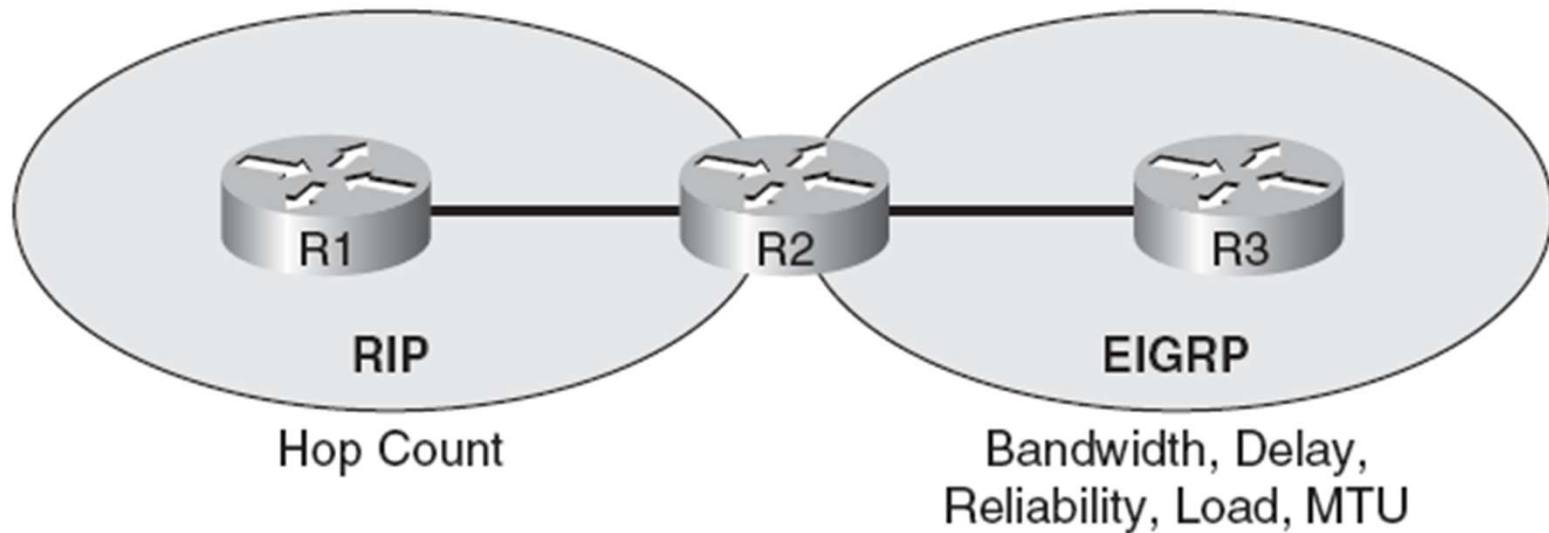


Metric

- Each routing protocol uses a different metric type.
- When injecting a route, attach a metric the destination protocol understands.

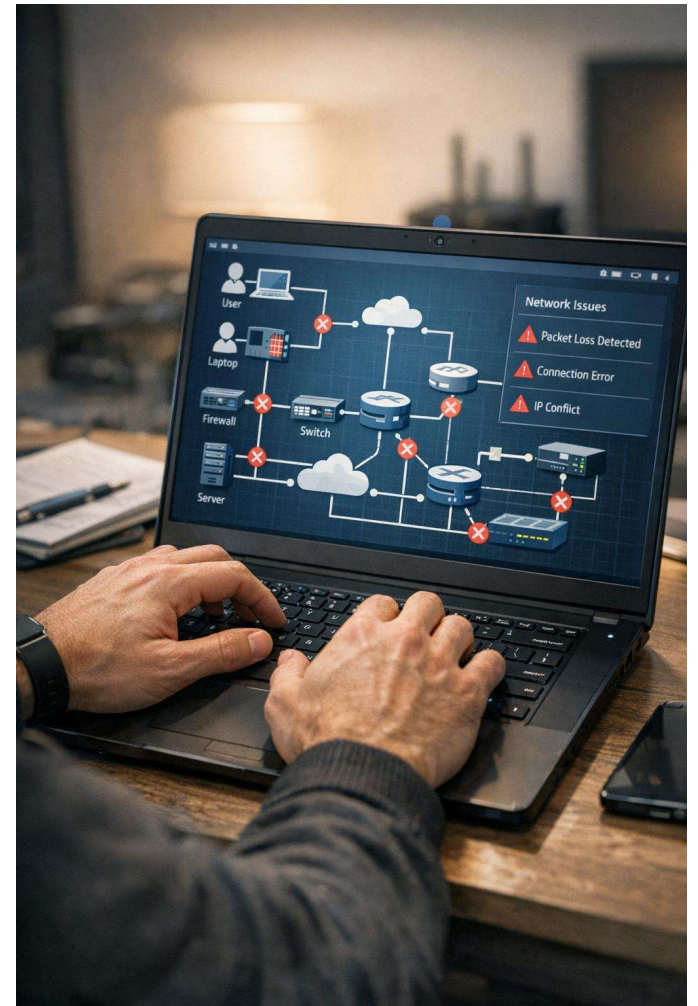


Differing Metric Parameters Between Routing Protocols



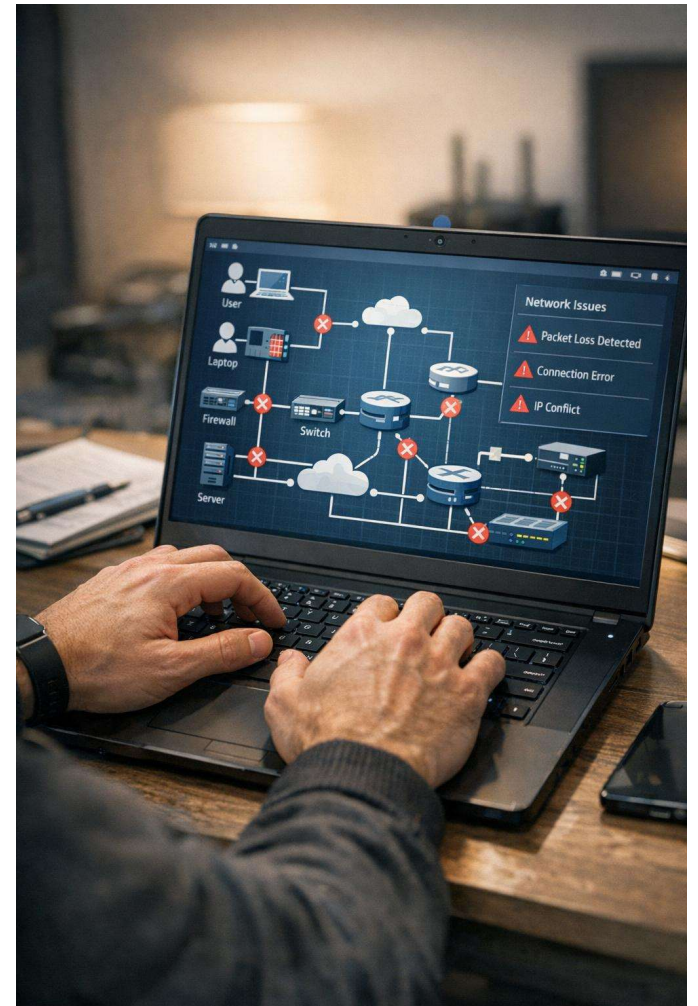
Seed Metric

- Seed metric: the value assigned to a route entering another protocol.
- It expresses relative reachability between dissimilar protocols.
- Set it three ways: default-metric, redistribute metric, or route map.



Seed Metric

- Without a seed metric, defaults apply.
- RIP and EIGRP default metrics are unreachable.
- Always set a seed metric when redistributing into RIP or EIGRP.



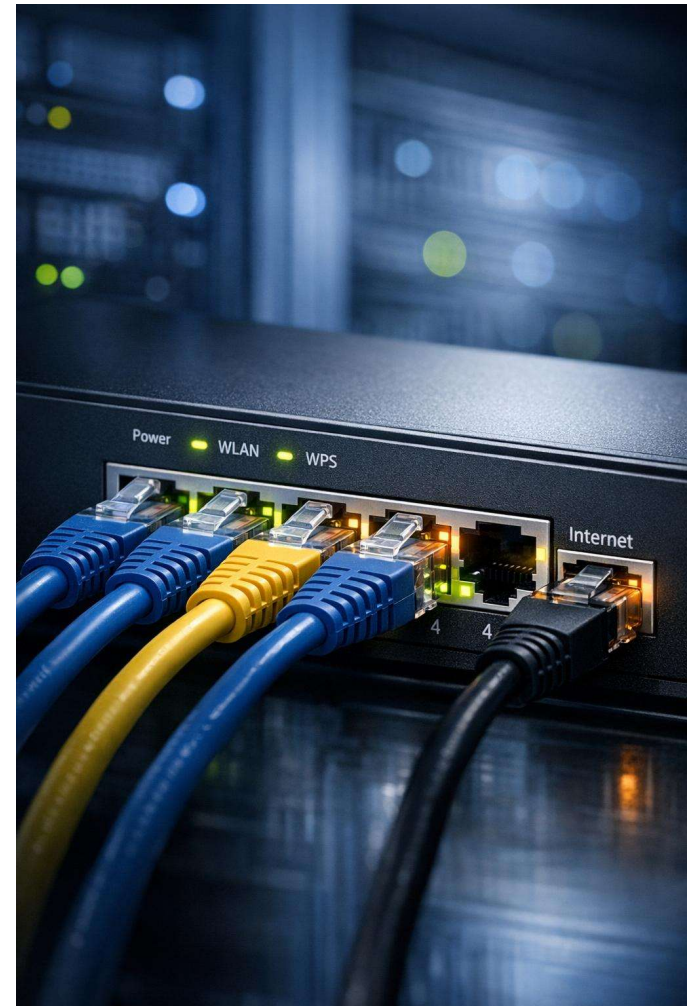
Route Redistribution Overview

- Protocols like EIGRP, OSPF, IS-IS tag routes as internal or external.
- Internal routes get priority, preventing redistribution loops.

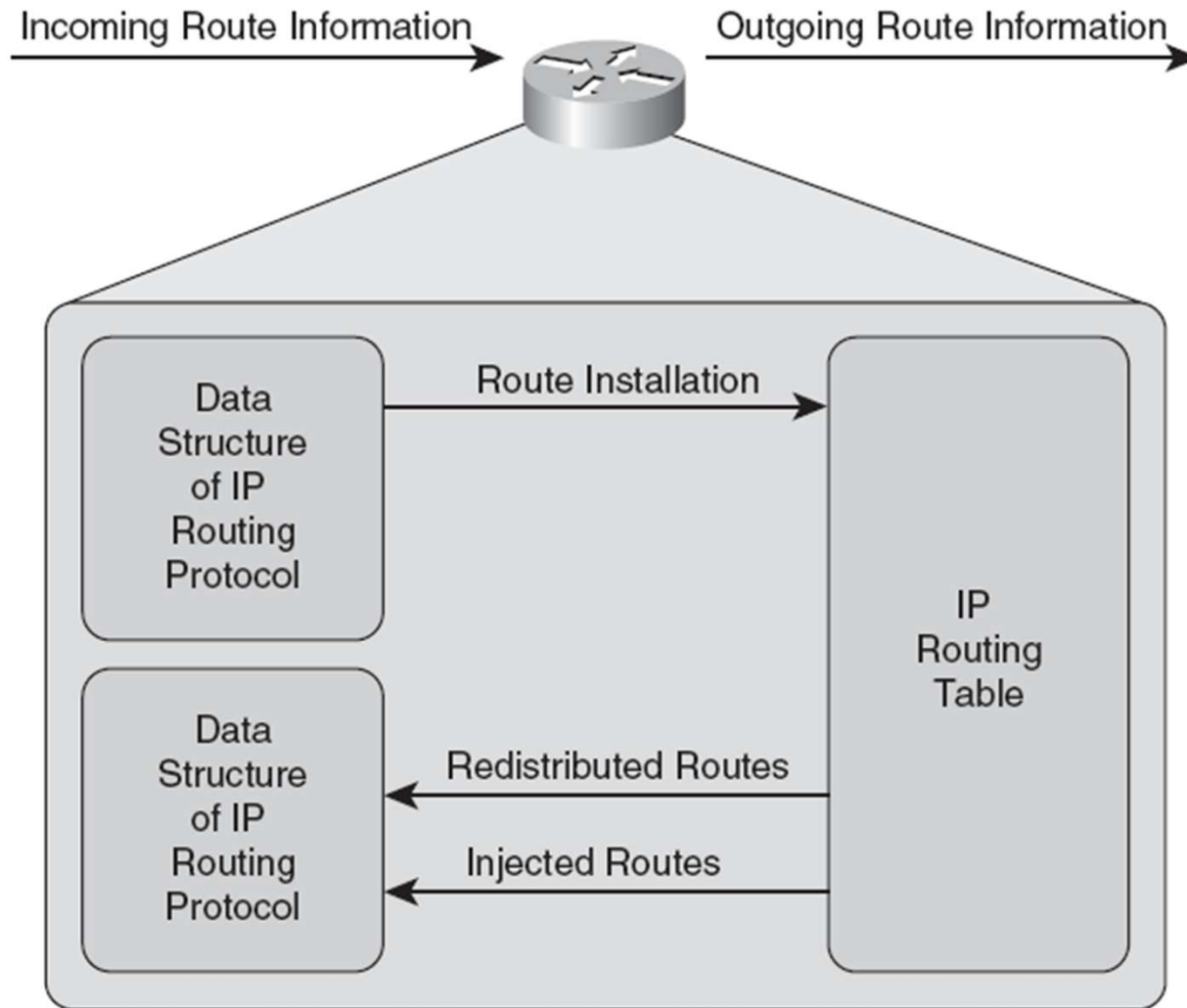


Route Redistribution Overview

- Two prerequisites for redistribution:
- The route must be in the router's IP routing table.
- The destination protocol needs a reachable metric for the route.



Route Redistribution Between Two IP Routing Protocols



Troubleshooting Targets for Route Redistribution

Troubleshooting Target	Troubleshooting Recommendation
Source routing protocol	Verify that a route to be redistributed from a routing protocol has been learned by that routing protocol. Therefore, you can issue appropriate <code>show</code> commands for the data structures of the source routing protocol to ensure that the source routing protocol has learned the route in question.
Route selection	Because a route must be in a router's IP routing table to be redistributed, you should ensure that the routes of the source routing protocol are indeed being injected into the router's IP routing table.

Troubleshooting Targets for Route Redistribution

Redistribution configuration

If a route has been injected into a router's IP routing table from a source routing protocol but not redistributed into the destination routing protocol, you should check the redistribution configuration. This involves checking the metric applied to routes as they are redistributed into the destination routing protocol, checking for any route filtering that might be preventing redistribution, and checking the redistribution syntax to confirm the correct routing process ID or autonomous system number is specified.

Destination routing protocol

If a route has been successfully redistributed into a destination routing protocol but the route has not been successfully learned by neighboring routers, you should investigate the destination routing protocol. You could use traditional methods of troubleshooting a destination routing protocol; however, keep in mind that the redistributed route might be marked as an external route. Therefore, check the characteristics of the destination routing protocol to determine if it treats external routes differently from internal ones.
