Copyright Information

Disclaimer

The following publication, CCIE R&S Lab Workbook Volume I Version 5.0, is designed to assist candidates in the preparation for Cisco Systems' CCIE Routing & Switching Lab Exam. While every effort has been made to ensure that all material is as complete and accurate as possible, the enclosed material is presented on an "as is" basis. Neither the authors nor Internetwork Expert, Inc. assume any liability or responsibility to any person or entity with respect to loss or damages incurred from the information contained in this workbook.

This workbook was developed by Internetwork Expert, Inc. and is an original work of the aforementioned authors. Any similarities between material presented in this workbook and actual CCIE lab material is completely coincidental.

Table of Contents

IPv6

> ✐ **Note**
>
> Load the *IPv6* initial configurations, which provide basic IPv4 connectivity across the topology with RIP as the IGP.

9.1 IPv6 Link-Local Addressing

- Configure link-local IPv6 addresses on the Frame-Relay clould between R1, R2, R3, R4, and R5.
- Use the IPv6 addressing format FE80::Y, where Y is the router number.

9.2 IPv6 Unique Local Addressing

- Enable IPv6 processing on the links between R6 & SW1 and R3 & SW1.
- Use the ULA IPv6 prefixes FC00:X:0:67::Y/64 and FC00:X:0:37::Y/64 for the links between R1 & R6 and R3 & SW1 respectively.
- Here X is your rack number in decimal notation, (e.g. 10 for Rack10, not the hex 0A) and Y is your router number (7 for SW1).

9.3 IPv6 Global Aggregatable Addressing

- Configure globally aggregatable IPv6 addresses in format 2001:X:0:1234::Y/64 on the Frame-Relay interfaces of R1, R2, R3, R4, and R5.
- Here X is your rack number in decimal notation, (e.g. 10 for Rack10, not the hex 0A) and Y is your router number (7 for SW1).

9.4 IPv6 EUI-64 Addressing

- Configure globally aggregatable IPv6 addresses in format 2001:X:0:146::Y/64 on the VLAN146 interfaces of R1, R4, and R6.
- Here X is your rack number in decimal notation, (e.g. 10 for Rack10, not the hex 0A) and Y should be automatically derived by the router based on the interface MAC address.

9.5 IPv6 Auto-Configuration

- Use the unique local address FC00:X:0:58::5/64 for R5's VLAN58 interface, where X is your rack number in decimal notation.
- Configure an additional IPv6 address FC00:X:0:85::5/64 on the same interface of R5.
- Configure R5 to broadcast both prefixes using ICMPv6 ND RAs, but only allow the hosts to use the second prefix for auto-configuration.
- The valid and preferred lifetime for both prefixes should be set to 4 hours.
- R5 should advertise itself as the default router on the segment every 40 seconds with the lifetime interval of 60 seconds.
- SW2 should learn its IPv6 address on VLAN58 automatically and use R5 as its default gateway.

9.6 RIPng

- Enable RIPng on the VLAN146 segment connecting R1, R4, and R6 using the process name "RIPNG".
- Create new Loopback100 interfaces on R1, R4, and R6 with the IPv6 addresses FC00:X:0:Y::Y/64, where X is your rack number in decimal notation and Y is your router number.
- Advertise these Loopback interfaces into RIPng.

9.7 RIPng over NBMA

- Enable RIPng on the Frame-Relay interfaces of R1, R4, and R5 using the process name RIPNG.
- Ensure that R1 and R4 can reach each other Loopback100 subnets even if any of their VLAN146 interfaces is down.

9.8 RIPng Summarization

- Create a new Loopback100 interface on R5 with the IPv6 address FC00:X:0:5::5/64 where X is you rack number in decimal notation and advertise this interface into RIPng.
- Ensure that R1 and R4 advertise only the summary prefix for the Loopback100 interfaces of R1, R4, and R5 to R6 across the VLAN146 segment.

9.9 RIPng Prefix Filtering

- Configure R5 to block the IPv6 prefix corresponding to R6's Loopback100 interface from entering the local routing table.
- Permit any other IPv6 prefixes

9.10 RIPng Metric Manipulation

- Enable IPv6 and RIPng on the Serial link connecting R4 and R5.
- Ensure that R4 prefers the Serial link to reach the Loopback100 interface of R5 and uses the Frame-Relay link as a backup only.
- Do not use summarization to accomplish this task.

9.11 RIPng Default Routing

- Configure R6 to advertise a default route into the RIPng routing domain with the initial metric of 5.

9.12 OSPFv3

- Configure OSPFv3 as the routing protocol on the links connecting R3, SW1, and R6.
- Manually configure the router identifiers for all OSPFv3 processes to match the IPv4 Loopback0 addresses.
- Use the Area 0 between SW1 and R6 and use the Area 37 between R3 and SW1.
- Change the network type to point-to-point on the link between R3 and SW1.
- Make the default hello/dead intervals on the segment connecting R6 and SW1 10 times less than the default.

9.13 OSPFv3 over NBMA

- Configure OSPFv3 area 0 on the Frame-Relay links of R5, R2, and R3.
- Use the network type that does not elect a DR, and ensure OSPF does not send broadcast hello packets.

9.14 OSPFv3 Virtual Links

- Restore the discontinuous OSPFv3 Area0 by provisioning a virtual link between R3 and SW1.

9.15 OSPFv3 Summarization

- Configure OSPFv3 Area 58 on the link between SW2 and R5.
- Create two new interfaces Loopback100 and Loopback101 on SW2 with IPv6 addresses FC00:X:0:8::8/64 and FC00:X:0:88::88/64 respectively.
- Advertise both Loopbacks into OSPF Area 58 on SW2.
- Configure R5 to summarize the two subnets to an optimal prefix.

9.16 IPv6 Redistribution

- Redistribute between OSPFv3 and RIPng on R5.
- Ensure each protocol prefers to reach native prefixes using internal paths.
- R6 should be the preferred exit point for each IPv6 routing domain.

9.17 IPv6 Filtering

- Configure R3’s Frame-Relay interface so that only FTP and HTTP traffic from the users on VLAN67 is permitted to flow access the interface.
- Allow DNS queries and responses from the same subnet, and ensure IPv6 routing is not affected.

9.18 IPv6 NAT-PT

- Configure R6 so that SW1 can access the IPv4 address 150.X.4.4 as 2000::9601:404.
- SW1 should source IPv6 packets off its VLAN67 interface and R6 should translate it to the IPv4 address 155.X.146.7.

9.19 IPv6 MP-BGP

- Configure R1 and R5 in BGP autonomous systems 100 and 500 respectively, and peer them over the Frame-Relay cloud.
- Create new Loopback interfaces on R1 with the IPv6 addresses 2003:X:0:1::1/64 and 2003:X:0:11::11/61, and advertise them into BGP.
- Ensure R5 sees only an optimal summary route encompassing both IPv6 prefixes.

Note

Clear all router configurations and load the *IPv6* initial configurations.

9.20 IPv6 Tunneling

- Create new Loopback interfaces on R2, R5, and R6 with the IP addresses 2001:X:0:Y::Y/64 where Y is the router number and X is your rack number in decimal notation (e.g. 10 for rack 10, 11 for rack 11 etc.)
- Create tunnels between R5 & R6 and R2 & R6. Use the IPv6 addresses 2001:X:0:56::Y/64 for the tunnel between R5 and R6 and the IPv6 address 2001:X:0:26::Y/64 for the tunnel between R2 and R6.
- The tunnel connecting R2 and R6 should use encapsulation suitable for transporting different Layer 3 protocols.
- The tunnel connecting R5 and R6 should use the encapsulated specifically designed for transporting IPv6 over IPv4.
- Use static routing to obtain connectivity.

9.21 Automatic 6to4 Tunnels

- Using the IPv4 Loopback0 interfaces create automatic 6to4 tunnels connecting R3, R4, and R5.
- Create additional Loopback interfaces on every router with the subnet number zero and the prefix length of 64 under the respective 6to4 /48 prefix.
- Use static routing to obtain connectivity between the newly allocated subnets.

9.22 ISATAP Tunnels

- Remove the 6to4 tunnels connecting R3, R4, and R5.
- Using the Loopback0 IPv4 addresses connect R1, SW1, and SW2 via ISATAP tunnels.
- Use the /64 IPv6 prefix 2001:1:0:345::/64 to allocated IPv6 addresses to the tunnel endpoints.
- Create additional IPv6 Loopback interfaces on all three routers with the IPv6 addresses 2001:1:0:Y::Y/64, and use static routing to obtain full connectivity.

IPv6 Solutions

9.1 IPv6 Link-Local Addressing

- Configure link-local IPv6 addresses on the Frame-Relay clould between R1, R2, R3, R4, and R5.
- Use the IPv6 addressing format FE80::Y, where Y is the router number.

Configuration

```
R1:
interface Serial 0/0
 ipv6 address fe80::1 link-local
 frame-relay map ipv6 fe80::5 105 broadcast
 frame-relay map ipv6 fe80::2 105
 frame-relay map ipv6 fe80::3 105
 frame-relay map ipv6 fe80::4 105

R2:
interface Serial 0/0
 ipv6 address fe80::2 link-local
 frame-relay map ipv6 fe80::5 205 broadcast
 frame-relay map ipv6 fe80::1 205
 frame-relay map ipv6 fe80::3 205
 frame-relay map ipv6 fe80::4 205

R3:
interface Serial 1/0
 ipv6 address fe80::3 link-local
 frame-relay map ipv6 fe80::5 305 broadcast
 frame-relay map ipv6 fe80::1 305
 frame-relay map ipv6 fe80::2 305
 frame-relay map ipv6 fe80::4 305

R4:
interface Serial 0/0
 ipv6 address fe80::4 link-local
 frame-relay map ipv6 fe80::5 405 broadcast
 frame-relay map ipv6 fe80::1 405
 frame-relay map ipv6 fe80::2 405
 frame-relay map ipv6 fe80::3 405

R5:
interface Serial 0/0
 ipv6 address fe80::5 link-local
 frame-relay map ipv6 fe80::1 501 broadcast
 frame-relay map ipv6 fe80::2 502 broadcast
 frame-relay map ipv6 fe80::3 503 broadcast
 frame-relay map ipv6 fe80::4 504 broadcast
```

Verification

> **Note**
>
> Link-local IPv6 addresses are significant only within the context of a single link. This means that packets with link-local addresses cannot be routed between interfaces, and link-local addresses may overlap as long as they exist on different interfaces. The address format for IPv6 link-local addresses are FE80::/10, and are synonymous with the range 169.254.0.0/16 in IPv4.
>
> Packets with link-local sources or destinations are mostly used by the router's control plane protocols, such as IPv6 routing protocols. For broadcast segments, such as Ethernet, link-local reachability is implicit due to automatic resolution through ICMP Neighbor Discovery (ICMPND). However, since multipoint NBMA interfaces do not support Inverse ICMPND, it is important to ensure link-local IPv6 address reachability over these links by configuring static mappings. Without these mappings routing protocols may not being able to establish adjacencies or properly encapsulate packets.

```
Rack1R5#show frame-relay map
Serial0/0 (up): ipv6 FE80::2 dlci 502(0x1F6,0x7C60), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ipv6 FE80::3 dlci 503(0x1F7,0x7C70), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ipv6 FE80::4 dlci 504(0x1F8,0x7C80), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 155.1.0.1 dlci 501(0x1F5,0x7C50), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 155.1.0.2 dlci 502(0x1F6,0x7C60), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 155.1.0.3 dlci 503(0x1F7,0x7C70), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 155.1.0.4 dlci 504(0x1F8,0x7C80), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ipv6 FE80::1 dlci 501(0x1F5,0x7C50), static,
              broadcast,
              CISCO, status defined, active
```


Since the same link-local address can exist on multiple interfaces, the router must be manually told which interface to send traffic out when pinging link-local addresses.

```
Rack1R5#ping ipv6 fe80::1
Output Interface: Serial0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::1, timeout is 2 seconds:
Packet sent with a source address of FE80::5
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/61/80 ms

Rack1R5#ping ipv6 fe80::2
Output Interface: Serial0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::2, timeout is 2 seconds:
Packet sent with a source address of FE80::5
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

Rack1R5#ping ipv6 fe80::3
Output Interface: Serial0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::3, timeout is 2 seconds:
Packet sent with a source address of FE80::5
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/60/61 ms

Rack1R5#ping ipv6 fe80::4
Output Interface: Serial0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::4, timeout is 2 seconds:
Packet sent with a source address of FE80::5
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```


9.2 IPv6 Unique Local Addressing

- Enable IPv6 processing on the links between R6 & SW1 and R3 & SW1.
- Use the ULA IPv6 prefixes FC00:X:0:67::Y/64 and FC00:X:0:37::Y/64 for the links between R1 & R6 and R3 & SW1 respectively.
- Here X is your rack number in decimal notation, (e.g. 10 for Rack10, not the hex 0A) and Y is your router number (7 for SW1).

Configuration

> **Note**
>
> You need to change the SDM (Switch Database Manager) template for the 3560 switches in order to configure IPv6 addressing. Changing SDM template requires the switch to be reloaded.

```
R3:
interface FastEthernet 0/0
 ipv6 address FC00:1:0:37::3/64

R6:
interface FastEthernet 0/0.67
 ipv6 address FC00:1:0:67::6/64

SW1:
sdm prefer dual-ipv4-and-ipv6 routing
!
interface Vlan 67
 ipv6 address FC00:1:0:67::7/64
!
interface FastEthernet 0/3
 ipv6 address FC00:1:0:37::7/64
```

Verification

Note

Unique Local IPv6 Unicast Addressing (ULA), defined in RFC 4193, deprecates the previously used Site-Local (FEC0::/10) addressing. ULA addresses in IPv6 are synonymous with the RFC 1918 private addresses in IPv4, i.e. the 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16 prefixes. ULA addresses are considered private, and are not publicly routable prefixes on the Internet.

The format of the ULA is

FC00 (7 bits) + Unique ID (41 bits) + Link ID (16 bits) + Interface ID (64 bits).

The Unique ID is supposed to be randomly generated, in order to avoid address collisions. Other than the addressing format, ULA addressing exhibits no other unique behavior when compared to normal publicly routable IPv6 addresses.

```
Rack1SW1#ping fc00:1:0:37::3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FC00:1:0:37::3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/1/9 ms

Rack1SW1#ping fc00:1:0:67::6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FC00:1:0:67::6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/1/9 ms
```

9.3 IPv6 Global Aggregatable Addressing

- Configure globally aggregatable IPv6 addresses in format 2001:X:0:1234::Y/64 on the Frame-Relay interfaces of R1, R2, R3, R4, and R5.

  Here X is your rack number in decimal notation, (e.g. 10 for Rack10, not the hex 0A) and Y is your router number (7 for SW1).

Configuration

```
R1:
interface Serial 0/0
 ipv6 address 2001:1:0:1234::1/64
 frame-relay map ipv6 2001:1:0:1234::5 105
 frame-relay map ipv6 2001:1:0:1234::2 105
 frame-relay map ipv6 2001:1:0:1234::3 105
 frame-relay map ipv6 2001:1:0:1234::4 105

R2:
interface Serial 0/0
 ipv6 address 2001:1:0:1234::2/64
 frame-relay map ipv6 2001:1:0:1234::5 205
 frame-relay map ipv6 2001:1:0:1234::1 205
 frame-relay map ipv6 2001:1:0:1234::3 205
 frame-relay map ipv6 2001:1:0:1234::4 205

R3:
interface Serial 1/0
 ipv6 address 2001:1:0:1234::3/64
 frame-relay map ipv6 2001:1:0:1234::5 305
 frame-relay map ipv6 2001:1:0:1234::1 305
 frame-relay map ipv6 2001:1:0:1234::2 305
 frame-relay map ipv6 2001:1:0:1234::4 305

R4:
interface Serial 0/0
 ipv6 address 2001:1:0:1234::4/64
 frame-relay map ipv6 2001:1:0:1234::5 405
 frame-relay map ipv6 2001:1:0:1234::1 405
 frame-relay map ipv6 2001:1:0:1234::2 405
 frame-relay map ipv6 2001:1:0:1234::3 405

R5:
interface Serial 0/0
 ipv6 address 2001:1:0:1234::5/64
 frame-relay map ipv6 2001:1:0:1234::1 501
 frame-relay map ipv6 2001:1:0:1234::2 502
 frame-relay map ipv6 2001:1:0:1234::3 503
 frame-relay map ipv6 2001:1:0:1234::4 504
```

Verification

Note

Globally Aggregatable IPv6 addresses, AKA global unicast addresses, are publicly allocated and routable addresses in the Internet. Global unicast addresses start with the binary prefix 001 (2000::/3), and therefore encompass the range 2000:: - 3FFF::. Allocation of these addresses through Internet registries such as ARIN and APNIC is designed to be hierarchical, unlike IPv4 allocation, and is defined in RIPE-450, *IPv6 Address Allocation and Assignment Policy*.

Although this range is extremely large – 1/8th of the total IPv6 address space – generally only the prefix 2001::/16 is currently used for allocation. Other ranges, such as the 6to4 Tunnel range of 2002::/16, are generally used for special purposes.

Once assigned the behavior of a global unicast address is identical to a ULA address, but has global routing significance.

```
Rack1R5#ping 2001:1:0:1234::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1:0:1234::1, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/61 ms

Rack1R5#ping 2001:1:0:1234::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1:0:1234::2, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms

Rack1R5#ping 2001:1:0:1234::3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1:0:1234::3, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
```

```
Rack1R5#ping 2001:1:0:1234::4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1:0:1234::4, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

Rack1R5#show frame-relay map
Serial0/0 (up): ipv6 FE80::2 dlci 502(0x1F6,0x7C60), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ipv6 FE80::3 dlci 503(0x1F7,0x7C70), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ipv6 FE80::4 dlci 504(0x1F8,0x7C80), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 155.1.0.1 dlci 501(0x1F5,0x7C50), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 155.1.0.2 dlci 502(0x1F6,0x7C60), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 155.1.0.3 dlci 503(0x1F7,0x7C70), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 155.1.0.4 dlci 504(0x1F8,0x7C80), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ipv6 2001:1:0:1234::1 dlci 501(0x1F5,0x7C50), static,
              CISCO, status defined, active
Serial0/0 (up): ipv6 2001:1:0:1234::2 dlci 502(0x1F6,0x7C60), static,
              CISCO, status defined, active
Serial0/0 (up): ipv6 2001:1:0:1234::3 dlci 503(0x1F7,0x7C70), static,
              CISCO, status defined, active
Serial0/0 (up): ipv6 2001:1:0:1234::4 dlci 504(0x1F8,0x7C80), static,
              CISCO, status defined, active
Serial0/0 (up): ipv6 FE80::1 dlci 501(0x1F5,0x7C50), static,
              broadcast,
              CISCO, status defined, active
```

9.4 IPv6 EUI-64 Addressing

- Configure globally aggregatable IPv6 addresses in format 2001:X:0:146::Y/64 on the VLAN146 interfaces of R1, R4, and R6.
- Here X is your rack number in decimal notation, (e.g. 10 for Rack10, not the hex 0A) and Y should be automatically derived by the router based on the interface MAC address.

Configuration

```
R1:
interface FastEthernet 0/0
 ipv6 address 2001:1:0:146::/64 eui-64

R4:
interface FastEthernet 0/1
 ipv6 address 2001:1:0:146::/64 eui-64

R6:
interface FastEthernet 0/0.146
 ipv6 address 2001:1:0:146::/64 eui-64
```

Verification

Note

EUI stands for Extended Unique Identifier, and is a unique 64-bit value that is used to give an identifier to a physical interface. In many respects, the EUI-64 resembles the IEEE MAC address, but designed to be used universally, not just with the hardware addresses.

With IPv6, EUI-64 is commonly used to automatically construct a unique host address on a shared Ethernet segment. When you use the **`eui-64`** keyword, the IOS uses the 48 bit hardware address of the Ethernet interface to automatically construct the unique 64-bit Interface identifier of the IPv6 address.

Specifically this is accomplished by swapping the 7th most significant bit of the Ethernet MAC address, called the Universal/Local (U/L) bit, and then adding 16 bits of padding in the format FFFE in the middle.


```
Rack1R1#show ipv6 interface fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20D:65FF:FE84:6560
  Global unicast address(es):
    2001:1:0:146:20D:65FF:FE84:6560, subnet is 2001:1:0:146::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF84:6560
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
```

From the above output we can see that the EUI-64 derived host address of R1's Fa0/0 interface is 20D:65FF:FE84:6560. This means that the MAC address of R1's Fa0/0 interface is 000D:6584:6560. Note that the link-local address of the interface automatically uses the format FE80::[EUI-64] unless manually modified.

9.5 IPv6 Auto-Configuration

- Use the unique local address FC00:X:0:58::5/64 for R5's VLAN58 interface, where X is your rack number in decimal notation.
- Configure an additional IPv6 address FC00:X:0:85::5/64 on the same interface of R5.
- Configure R5 to broadcast both prefixes using ICMPv6 ND RAs, but only allow the hosts to use the second prefix for auto-configuration.
- The valid and preferred lifetime for both prefixes should be set to 4 hours.
- R5 should advertise itself as the default router on the segment every 40 seconds with the lifetime interval of 60 seconds.
- SW2 should learn its IPv6 address on VLAN58 automatically and use R5 as its default gateway.

Configuration

```
R5:
ipv6 unicast-routing
!
interface FastEthernet 0/0
 ipv6 address fc00:1:0:58::5/64
 ipv6 address fc00:1:0:85::5/64
 ipv6 nd prefix fc00:1:0:58::/64 14400 14400 no-autoconfig
 ipv6 nd prefix fc00:1:0:85::/64 14400 14400
 ipv6 nd ra-interval 40
 ipv6 nd ra-lifetime 60
 no ipv6 nd suppress-ra

SW2:
sdm prefer dual-ipv4-and-ipv6 routing
!
interface Vlan 58
 ipv6 address autoconfig default
```

Verification

Note

One of the special IPv6 features is auto-configuration. It replaces many functions served by DHCP in IPv4 networks (yet there is a special DHCPv6 edition of the DHCP protocol). With IPv6 auto-configuration, an IPv6 host may automatically learn the IPv6 prefixes assigned to the local segment, as well as determine the default routers on the same segment. To accomplish this, a special type of link-local IPv6 addressing is used along with the ICMPv6 ND (Neighbor Discovery) protocol.

You may configure a Cisco router with the interval to send the Router Advertisements or completely suppress this feature (which is enabled by default on Ethernet interfaces). Note that you must enable IPv6 unicast routing in order to send the router advertisements. The following commands control the IPv6 RA announcements:

`ipv6 nd ra-interval` – specifies the periodic interval to send RAs.
`ipv6 nd ra-lifetime` – specifies the validity interval of the router's IPv6 address
`ipv6 nd prefix` – manipulates the IPv6 network prefixes included into RA. By default all prefixes are included.

You may adjust the interval the prefix is valid and preferred with every command. Additionally, you may instruct the hosts not to use this prefix for auto-configuration, by using the **`no-autoconfig`** keyword.

```
Rack1SW2#show ipv6 interface vlan 58
Vlan58 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::21B:D4FF:FED4:4D45
  Global unicast address(es):
    FC00:1:0:85:21B:D4FF:FED4:4D45, subnet is FC00:1:0:85::/64 [PRE]
      valid lifetime 14374 preferred lifetime 14374
  Joined group address(es):
    FF02::1
    FF02::1:FFD4:4D45
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  Default router is FE80::212:FF:FE1C:DAA0 on Vlan58

Rack1R5#debug ipv6 nd
ICMP Neighbor Discovery events debugging is on
Rack1R5#
ICMPv6-ND: Sending RA to FF02::1 on FastEthernet0/0
ICMPv6-ND:     MTU = 1500
ICMPv6-ND:     prefix = FC00:1:0:58::/64 onlink
ICMPv6-ND:           14400/14400 (valid/preferred)
ICMPv6-ND:     prefix = FC00:1:0:85::/64 onlink autoconfig
ICMPv6-ND:           14400/14400 (valid/preferred)

Rack1R5#show ipv6 interface fastEthernet 0/0 prefix
IPv6 Prefix Advertisements FastEthernet0/0
Codes: A - Address, P - Prefix-Advertisement, O - Pool
       U - Per-user prefix, D - Default
       N - Not advertised, C - Calendar

     default [LA] Valid lifetime 2592000, preferred lifetime 604800
AP   FC00:1:0:58::/64 [L] Valid lifetime 14400, preferred lifetime
14400
AP   FC00:1:0:85::/64 [LA] Valid lifetime 14400, preferred lifetime
14400
```

9.6 RIPng

- Enable RIPng on the VLAN146 segment connecting R1, R4, and R6 using the process name “RIPNG”.
- Create new Loopback100 interfaces on R1, R4, and R6 with the IPv6 addresses FC00:X:0:Y::Y/64, where X is your rack number in decimal notation and Y is your router number.
- Advertise these Loopback interfaces into RIPng.

Configuration

```
R1:
ipv6 unicast-routing
!
interface FastEthernet 0/0
 ipv6 rip RIPNG enable
!
interface Loopback100
 ipv6 address fc00:1:0:1::1/64
 ipv6 rip RIPNG enable

R4:
ipv6 unicast-routing
!
interface FastEthernet 0/1
 ipv6 rip RIPNG enable
!
interface Loopback100
 ipv6 address fc00:1:0:4::4/64
 ipv6 rip RIPNG enable

R6:
ipv6 unicast-routing
!
interface FastEthernet 0/0.146
 ipv6 rip RIPNG enable
!
interface Loopback100
 ipv6 address fc00:1:0:6::6/64
 ipv6 rip RIPNG enable
```

Verification

> **Note**
>
> RIPng is configured by simply enabling it on the respective interfaces. Global RIPng parameters can be configured in the routing process subconfiguration mode by issuing the **`ipv6 rip [pid]`** command in global configuration.
>
> Many of the routing verification commands are similar between IPv4 and IPv6, with the exception of the **`ip`** keyword being replaced by the **`ipv6`** keyword.

```
Rack1R6#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip RIPNG"
  Interfaces:
    Loopback100
    FastEthernet0/0.146
  Redistribution:
    None

Rack1R6#show ipv6 rip RIPNG
RIP process "RIPNG", port 521, multicast-group FF02::9, pid 207
     Administrative distance is 120. Maximum paths is 16
     Updates every 30 seconds, expire after 180
     Holddown lasts 0 seconds, garbage collect after 120
     Split horizon is on; poison reverse is off
     Default routes are not generated
     Periodic updates 5, trigger updates 1
  Interfaces:
    Loopback100
    FastEthernet0/0.146
  Redistribution:
    None
```

```
Rack1R6#show ipv6 route rip
IPv6 Routing Table - 10 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   FC00:1:0:1::/64 [120/2]
     via FE80::20D:65FF:FE84:6560, FastEthernet0/0.146
R   FC00:1:0:4::/64 [120/2]
     via FE80::20F:24FF:FE11:4961, FastEthernet0/0.146


Rack1R6#debug ipv6 rip
RIP Routing Protocol debugging is on
Rack1R6#
RIPng: response received from FE80::20D:65FF:FE84:6560 on
FastEthernet0/0.146 for RIPNG
       src=FE80::20D:65FF:FE84:6560 (FastEthernet0/0.146)
       dst=FF02::9
       sport=521, dport=521, length=52
       command=2, version=1, mbz=0, #rte=2
       tag=0, metric=1, prefix=2001:1:0:146::/64
       tag=0, metric=1, prefix=FC00:1:0:1::/64
Rack1R6#
RIPng: response received from FE80::20F:24FF:FE11:4961 on
FastEthernet0/0.146 for RIPNG
       src=FE80::20F:24FF:FE11:4961 (FastEthernet0/0.146)
       dst=FF02::9
       sport=521, dport=521, length=52
       command=2, version=1, mbz=0, #rte=2
       tag=0, metric=1, prefix=2001:1:0:146::/64
       tag=0, metric=1, prefix=FC00:1:0:4::/64
Rack1R6#
RIPng: Sending multicast update on Loopback100 for RIPNG
       src=FE80::20C:85FF:FEC1:FC60
       dst=FF02::9 (Loopback100)
       sport=521, dport=521, length=92
       command=2, version=1, mbz=0, #rte=4
       tag=0, metric=1, prefix=2001:1:0:146::/64
       tag=0, metric=1, prefix=FC00:1:0:6::/64
       tag=0, metric=2, prefix=FC00:1:0:4::/64
       tag=0, metric=2, prefix=FC00:1:0:1::/64
RIPng: Sending multicast update on FastEthernet0/0.146 for RIPNG
       src=FE80::20C:85FF:FEC1:FC60
       dst=FF02::9 (FastEthernet0/0.146)
       sport=521, dport=521, length=52
       command=2, version=1, mbz=0, #rte=2
       tag=0, metric=1, prefix=2001:1:0:146::/64
Rack1R6#
       tag=0, metric=1, prefix=FC00:1:0:6::/64
RIPng: Process RIPNG received own response on Loopback100
```

9.7 RIPng over NBMA

- Enable RIPng on the Frame-Relay interfaces of R1, R4, and R5 using the process name RIPNG.
- Ensure that R1 and R4 can reach each others Loopback100 subnets even if any of their VLAN146 interfaces is down.

Configuration

```
R1:
interface Serial 0/0
 ipv6 rip RIPNG enable

R4:
interface Serial0/0
 ipv6 rip RIPNG enable

R5:
ipv6 unicast-routing
!
ipv6 router rip RIPNG
 no split-horizon
!
interface Serial 0/0
 ipv6 rip RIPNG enable
```

Verification

Note

NBMA networks pose two potential issues to IPv6 routing. First, unmapped link-local IPv6 addresses might prevent routing updates from being processed correctly, as the next-hop is always a link-local address. Do not forget to use the **`broadcast`** keyword with the IPv6 mapping statements on DLCIs that should switch routing updates, as otherwise the router will not send any multicast packets.

Secondly, partial mesh connectivity might prevent some nodes from receiving routing updates. Commonly, this problem arises in Hub-and-Spoke topologies, where spokes do not receive routing information from other spokes. In order to deal with this issue, disable the split horizon function in RIPng. Note that split-horizon is only disabled globally, under the routing process, not per interface.

Alternatively, you may want to use default routing or summarization at the hub router to prevent the loss of reachability. In our case, simply disable split-horizon on R5.

```
Rack1R5#show ipv6 route rip
IPv6 Routing Table - 12 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   2001:1:0:146::/64 [120/2]
     via FE80::4, Serial0/0
     via FE80::1, Serial0/0
R   FC00:1:0:1::/64 [120/2]
     via FE80::1, Serial0/0
R   FC00:1:0:4::/64 [120/2]
     via FE80::4, Serial0/0
R   FC00:1:0:6::/64 [120/3]
     via FE80::4, Serial0/0
     via FE80::1, Serial0/0
```

```
Rack1R5#debug ipv6 rip
RIPng: Sending multicast update on Serial0/0 for RIPNG
       src=FE80::5
       dst=FF02::9 (Serial0/0)
       sport=521, dport=521, length=112
       command=2, version=1, mbz=0, #rte=5
       tag=0, metric=1, prefix=2001:1:0:1234::/64
       tag=0, metric=2, prefix=2001:1:0:146::/64
       tag=0, metric=2, prefix=FC00:1:0:4::/64
       tag=0, metric=2, prefix=FC00:1:0:1::/64
       tag=0, metric=3, prefix=FC00:1:0:6::/64
Rack1R5#
```

9.8 RIPng Summarization

- Create a new Loopback100 interface on R5 with the IPv6 address FC00:X:0:5::5/64 where X is you rack number in decimal notation and advertise this interface into RIPng.
- Ensure that R1 and R4 advertise only the summary prefix for the Loopback100 interfaces of R1, R4, and R5 to R6 across the VLAN146 segment.

Configuration

```
R1:
interface FastEthernet 0/0
 ipv6 rip RIPNG summary fc00:1::/61

R4:
interface FastEthernet 0/1
 ipv6 rip RIPNG summary fc00:1::/61

R5:
interface Loopback 100
 ipv6 address fc00:1:0:5::5/64
 ipv6 rip RIPNG enable
```

Verification

Note

In order to summarize RIPng routes, use the interface level command **`ipv6 rip [pid] summary`**. Use the standard rules of summarization, converting IPv6 prefixes from hex format into binary format. In our case we have three prefixes:

FC00:X:0:5::/64
FC00:X:0:1::/64
FC00:X:0:4::/64

Prefixes differ in the 4th position (remember each block of the IPv6 address is 16 bits):

0000 0000 0000 0**101**
0000 0000 0000 0**001**
0000 0000 0000 0**100**

In order to summarize, move the prefix length 3 bit positions to the left. The resulting prefix is:

FC00:X::/61

```
Rack1R6#show ipv6 route rip
IPv6 Routing Table - 10 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   2001:1:0:1234::/64 [120/2]
     via FE80::20D:65FF:FE84:6560, FastEthernet0/0.146
     via FE80::20F:24FF:FE11:4961, FastEthernet0/0.146
R   FC00:1::/61 [120/2]
     via FE80::20F:24FF:FE11:4961, FastEthernet0/0.146
     via FE80::20D:65FF:FE84:6560, FastEthernet0/0.146

Rack1R4#show ipv6 route rip
IPv6 Routing Table - 9 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   FC00:1::/61 [120/2]
     via FE80::20D:65FF:FE84:6560, FastEthernet0/1
R   FC00:1:0:5::/64 [120/2]
     via FE80::5, Serial0/0
R   FC00:1:0:6::/64 [120/2]
     via FE80::20C:85FF:FEC1:FC60, FastEthernet0/1

Rack1R1#show ipv6 route rip
IPv6 Routing Table - 11 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   FC00:1::/61 [120/3]
     via FE80::20F:24FF:FE11:4961, FastEthernet0/0
R   FC00:1:0:5::/64 [120/2]
     via FE80::5, Serial0/0
R   FC00:1:0:6::/64 [120/2]
     via FE80::20C:85FF:FEC1:FC60, FastEthernet0/0
```

9.9 RIPng Prefix Filtering

- Configure R5 to block the IPv6 prefix corresponding to R6's Loopback100 interface from entering the local routing table.
- Permit any other IPv6 prefixes

Configuration

```
R5:
ipv6 prefix-list FILTER_R6_LO100 deny fc00:1:0:6::/64
ipv6 prefix-list FILTER_R6_LO100 permit ::/0 le 128
!
ipv6 router rip RIPNG
 distribute-list prefix-list FILTER_R6_LO100 in
```

Verification

Note

RIPng uses IPv6 prefix lists to filter routing updates. The lists are applied either inbound or outbound under the RIPng process configuration mode. You may choose to associate an interface with the distribute-list, or apply it to all interfaces simultaneously by not specifying and interface.

Before the filter has been applied:

```
Rack1R5#show ipv6 route rip
IPv6 Routing Table - 13 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS
summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF
ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   2001:1:0:146::/64 [120/2]
     via FE80::4, Serial0/0
     via FE80::1, Serial0/0
R   FC00:1::/61 [120/3]
     via FE80::4, Serial0/0
     via FE80::1, Serial0/0
R   FC00:1:0:1::/64 [120/2]
     via FE80::1, Serial0/0
R   FC00:1:0:4::/64 [120/2]
     via FE80::4, Serial0/0
R   FC00:1:0:6::/64 [120/3]
     via FE80::4, Serial0/0
     via FE80::1, Serial0/0
```

It may take some time for the routes to get flushed once the filter is applied. This process can be sped up by clearing the rip process with the **`clear ipv6 rip [pid]`** command. Once the table ages out the old prefix, the route is no longer found:

```
Rack1R5#show ipv6 route fc00:1:0:6::/64
% Route not found
```

9.10 RIPng Metric Manipulation

- Enable IPv6 and RIPng on the Serial link connecting R4 and R5.
- Ensure that R4 prefers the Serial link to reach the Loopback100 interface of R5 and uses the Frame-Relay link as a backup only.
- Do not use summarization to accomplish this task.

Configuration

```
R4:
interface Serial 0/1
 ipv6 rip RIPNG enable
 ipv6 enable
!
interface Serial 0/0
 ipv6 rip RIPNG metric-offset 2

R5:
interface Serial 0/1
 ipv6 rip RIPNG enable
 ipv6 enable
```

Verification

Note

RIPng does not have granular metric manipulation mechanisms. You can only apply metric offsets to all routes received via a particular interface, thus defining outgoing interface preference. Note that you can also filter routes using this feature, by specifying a metric offset that results in total metric of 16 or higher.

This task also demonstrates that you only need link-local IPv6 addresses on a link to activate IPv6 routing. By issuing the **`ipv6 enable`** command you configure the link for automatic link-local IPv6 address allocation.

Before the metric-offset:

```
Rack1R4#show ipv6 route fc00:1:0:5::/64
IPv6 Routing Table - 9 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   FC00:1:0:5::/64 [120/2]
     via FE80::5, Serial0/0
     via FE80::212:FF:FE1C:DAA0, Serial0/1
```

After metric offset:

```
Rack1R4#show ipv6 route fc00:1:0:5::/64
IPv6 Routing Table - 9 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   FC00:1:0:5::/64 [120/2]
     via FE80::212:FF:FE1C:DAA0, Serial0/1
```

9.11 RIPng Default Routing

- Configure R6 to advertise a default route into the RIPng routing domain with the initial metric of 5.

Configuration

```
R6:
interface FastEthernet 0/0.146
 ipv6 rip RIPNG default-information originate metric 5
```

Verification

> **Note**
>
> RIPng allows sending a default route out of any interface of the router. Additionally, you may specify the default route metric. Using the keyword **only** you may configure the router to send just the default route exclusively out of the particular interface, and filter all other subnet advertisements.

```
Rack1R4#show ipv6 route rip
IPv6 Routing Table - 10 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS
summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF
ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   ::/0 [120/6]
     via FE80::20C:85FF:FEC1:FC60, FastEthernet0/1
R   FC00:1::/61 [120/2]
     via FE80::20D:65FF:FE84:6560, FastEthernet0/1
R   FC00:1:0:5::/64 [120/2]
     via FE80::212:FF:FE1C:DAA0, Serial0/1
R   FC00:1:0:6::/64 [120/2]
     via FE80::20C:85FF:FEC1:FC60, FastEthernet0/1
```

9.12 OSPFv3

- Configure OSPFv3 as the routing protocol on the links connecting R3, SW1, and R6.
- Manually configure the router identifiers for all OSPFv3 processes to match the IPv4 Loopback0 addresses.
- Use the Area 0 between SW1 and R6 and use the Area 37 between R3 and SW1.
- Change the network type to point-to-point on the link between R3 and SW1.
- Make the default hello/dead intervals on the segment connecting R6 and SW1 10 times less than the default.

Configuration

```
R3:
ipv6 unicast-routing
!
ipv6 router ospf 1
 router-id 150.1.3.3
!
interface FastEthernet 0/0
 ipv6 ospf 1 area 37
 ipv6 ospf network point-to-point

R6:
ipv6 router ospf 1
 router-id 150.1.6.6
!
interface FastEthernet 0/0.67
 ipv6 ospf hello-interval 1
 ipv6 ospf 1 area 0

SW1:
ipv6 unicast-routing
!
ipv6 router ospf 1
 router-id 150.1.7.7
!
interface Vlan 67
 ipv6 ospf hello-interval 1
 ipv6 ospf 1 area 0
!
interface FastEthernet 0/3
 ipv6 ospf 1 area 37
 ipv6 ospf network point-to-point
```

Verification

> **Note**
>
> OSPFv3 for IPv6 retains many concepts of OSPFv2 for IPv4, practically the same set of LSA types plus the concept of areas, ABRs, area types and so on. Even the router ID for OSPFv3 is the same 32 bit value, derived from the highest Loopback IPv4 address.
>
> Most OSPFv3 interface-related settings are now configured at the interface level only. Instead of using the **`network`** command, you simply assign interfaces to the OSPFv3 process.
>
> Like RIPng, many of the OSPFv3 verification commands use the same format as the OSPFv2 counterparts, with the exception of the **`ipv6`** keyword in place of the **`ip`** keyword.

```
Rack1SW1#show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID  Interface
150.1.6.6         1   FULL/DR         00:00:03    16            Vlan67
150.1.3.3         1   FULL/  -        00:00:38    4             FastEthernet0/3

Rack1SW1#show ipv6 ospf 1
 Routing Process "ospfv3 1" with ID 150.1.7.7
 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
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of areas in this router is 2. 2 normal 0 stub 0 nssa
    Area BACKBONE(0)
        Number of interfaces in this area is 1
        SPF algorithm executed 3 times
        Number of LSA 7. Checksum Sum 0x02FE9B
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
    Area 37
        Number of interfaces in this area is 1
        SPF algorithm executed 3 times
        Number of LSA 7. Checksum Sum 0x0401D4
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

```
Rack1SW1#show ipv6 ospf interface fastEthernet 0/3
FastEthernet0/3 is up, line protocol is up (connected)
  Link Local Address FE80::212:1FF:FE31:41, Interface ID 1021
  Area 37, Process ID 1, Instance ID 0, Router ID 150.1.7.7
  Network Type POINT_TO_POINT, Cost: 1
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:01
  Index 1/1/2, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 150.1.3.3
  Suppress hello for 0 neighbor(s)

Rack1SW1#show ipv6 ospf interface Vlan 67
Vlan67 is up, line protocol is up
  Link Local Address FE80::212:1FF:FE31:43, Interface ID 2135
  Area 0, Process ID 1, Instance ID 0, Router ID 150.1.7.7
  Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 150.1.6.6, local address
FE80::20C:85FF:FEC1:FC60
  Backup Designated router (ID) 150.1.7.7, local address
FE80::212:1FF:FE31:43
  Timer intervals configured, Hello 1, Dead 4, Wait 4, Retransmit 5
    Hello due in 00:00:00
  Index 1/1/1, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 150.1.6.6  (Designated Router)
  Suppress hello for 0 neighbor(s)
```

9.13 OSPFv3 over NBMA

- Configure OSPFv3 area 0 on the Frame-Relay links of R5, R2, and R3.
- Use the network type that does not elect a DR, and ensure OSPF does not send broadcast hello packets.

Configuration

```
R2:
ipv6 unicast-routing
!
ipv6 router ospf 1
 router-id 150.1.2.2
!
interface Serial 0/0
 ipv6 ospf 1 area 0
 ipv6 ospf network point-to-multipoint non-broadcast
 ipv6 ospf neighbor fe80::5

R3:
interface Serial 1/0
 ipv6 ospf 1 area 0
 ipv6 ospf network point-to-multipoint non-broadcast
 ipv6 ospf neighbor fe80::5

R5:
ipv6 router ospf 1
 router-id 150.1.5.5
!
interface Serial 0/0
 ipv6 ospf 1 area 0
 ipv6 ospf network point-to-multipoint non-broadcast
 ipv6 ospf neighbor fe80::3
 ipv6 ospf neighbor fe80::2
```

Verification

Note

The task wording prompts to use the multipoint non-broadcast network type. Just like in OSPFv2, this network type supports multiple neighbors on the link, sends it updates as unicast, and does not perform the DR/BDR election. Note that in OSPFv3 you configure the neighbors at the interface level. As usual, the link-local IPv6 addresses should be properly mapped to exchange the OSPF packets. Note the use of link-local IPv6 addresses with the **`neighbor`** command.

With the multipoint network-type you can also specify the neighbor cost, like with OSPFv2.

```
Rack1R5#show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
150.1.2.2         1   FULL/  -        00:01:55    5               Serial0/0
150.1.3.3         1   FULL/  -        00:01:45    6               Serial0/0

Rack1R5#show ipv6 ospf int se 0/0
Serial0/0 is up, line protocol is up
  Link Local Address FE80::5, Interface ID 6
  Area 0, Process ID 1, Instance ID 0, Router ID 150.1.5.5
  Network Type POINT_TO_MULTIPOINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_MULTIPOINT,
  Timer intervals configured, Hello 30, Dead 120, Wait 120,Retransmit 5
    Hello due in 00:00:24
  Index 1/1/1, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 3
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 150.1.2.2
    Adjacent with neighbor 150.1.3.3
  Suppress hello for 0 neighbor(s)
```

9.14 OSPFv3 Virtual Links

- Restore the discontiguous OSPFv3 Area0 by provisioning a virtual link between R3 and SW1.

Configuration

```
R3:
ipv6 router ospf 1
 area 37 virtual-link 150.1.7.7

SW1:
ipv6 router ospf 1
 area 37 virtual-link 150.1.3.3
```

Verification

Note

The concept of the virtual-link remains the same in OSPFv3 as it does in OSPFv2. Using it you may repair the discontiguous backbone, attach a non-zero area to the backbone across a transit area, or perform traffic engineering.

Virtual-links are configured under the global OSPFv3 process configuration mode. Check the virtual-link health and verify that you can see Area 0 link prefixes on R6.

```
Rack1R3#show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
150.1.7.7         1   FULL/  -        00:00:36    2033            OSPFv3_VL0
150.1.5.5         1   FULL/  -        00:01:59    6               Serial1/0
150.1.7.7         1   FULL/  -        00:00:38    1021            FastEthernet0/0

Rack1R3#show ipv6 ospf virtual-links
Virtual Link OSPFv3_VL0 to router 150.1.7.7 is up
  Interface ID 17, IPv6 address FC00:1:0:37::7
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 37, via interface FastEthernet0/0, Cost of using 1
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Adjacency State FULL (Hello suppressed)
    Index 1/2/3, retransmission queue length 0, number of
retransmission 0
    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

```
Rack1R6#show ipv6 route ospf
IPv6 Routing Table - 16 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
O   2001:1:0:1234::2/128 [110/847]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
O   2001:1:0:1234::3/128 [110/2]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
O   2001:1:0:1234::5/128 [110/783]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
OI  FC00:1:0:37::/64 [110/2]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
OI  FC00:1:0:37::1/128 [110/2]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
OI  FC00:1:0:37::7/128 [110/1]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
```

9.15 OSPFv3 Summarization

- Configure OSPFv3 Area 58 on the link between SW2 and R5.
- Create two new interfaces Loopback100 and Loopback101 on SW2 with IPv6 addresses FC00:X:0:8::8/64 and FC00:X:0:88::88/64 respectively.
- Advertise both Loopbacks into OSPF Area 58 on SW2.
- Configure R5 to summarize the two subnets to an optimal prefix.

Configuration

```
R5:
ipv6 router ospf 1
 area 58 range fc00:1::/56
!
interface FastEthernet 0/0
 ipv6 ospf 1 area 58

SW2:
ipv6 unicast-routing
!
interface Vlan 58
 ipv6 ospf 1 area 58
!
interface Loopback100
 ipv6 address fc00:1:0:8::8/64
 ipv6 ospf 1 area 58
!
interface Loopback101
 ipv6 address fc00:1:0:88::88/64
 ipv6 ospf 1 area 58
```

Verification

> **Note**
>
> Summarization concepts are the same for OSPFv3 as for OSPFv2. Even the commands remain the same, you just need to apply them under the OSPFv3 process context.
>
> When summarizing prefixes fc00:1:0:8::/64 and fc001:0:88::/64 remember that 8 and 88 are in hex. Convert them to binary format using the Windows calculator application.
>
> 88 = 0000 0000 **1000 1000**
> 8 = 0000 0000 **0000 1000**
>
> Therefore, to summarize, shrink the prefix length by 8, to remove the mismatched parts. The resulting prefix is: fc00:1:0::/56.

```
Rack1R5#show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID  Interface
150.1.2.2         1   FULL/  -        00:01:37    5             Serial0/0
150.1.3.3         1   FULL/  -        00:01:50    6             Serial0/0
150.1.8.8         1   FULL/DR         00:00:35    2126          FastEthernet0/0

Rack1R5#show ipv6 route ospf
IPv6 Routing Table - 20 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
O   2001:1:0:1234::2/128 [110/64]
     via FE80::2, Serial0/0
O   2001:1:0:1234::3/128 [110/64]
     via FE80::3, Serial0/0
O   FC00:1::/56 [110/0]
     via ::, Null0
O   FC00:1:0:8::8/128 [110/1]
     via FE80::21B:D4FF:FED4:4D45, FastEthernet0/0
OI  FC00:1:0:37::/64 [110/65]
     via FE80::3, Serial0/0
OI  FC00:1:0:37::1/128 [110/64]
     via FE80::3, Serial0/0
OI  FC00:1:0:37::7/128 [110/65]
     via FE80::3, Serial0/0
O   FC00:1:0:67::/64 [110/66]
     via FE80::3, Serial0/0
O   FC00:1:0:88::88/128 [110/1]
     via FE80::21B:D4FF:FED4:4D45, FastEthernet0/0
```

```
Rack1R3#show ipv6 route ospf
IPv6 Routing Table - 12 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
O   2001:1:0:1234::2/128 [110/845]
     via FE80::5, Serial1/0
O   2001:1:0:1234::5/128 [110/781]
     via FE80::5, Serial1/0
OI  FC00:1::/56 [110/782]
     via FE80::5, Serial1/0
O   FC00:1:0:37::7/128 [110/1]
     via FE80::212:1FF:FE31:41, FastEthernet0/0
O   FC00:1:0:67::/64 [110/2]
     via FE80::212:1FF:FE31:41, FastEthernet0/0

Rack1SW1#show ipv6 route ospf
IPv6 Routing Table - 11 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
O   2001:1:0:1234::2/128 [110/846]
     via FE80::20C:CEFF:FE4B:A2A0, FastEthernet0/3
O   2001:1:0:1234::3/128 [110/1]
     via FE80::20C:CEFF:FE4B:A2A0, FastEthernet0/3
O   2001:1:0:1234::5/128 [110/782]
     via FE80::20C:CEFF:FE4B:A2A0, FastEthernet0/3
OI  FC00:1::/56 [110/783]
     via FE80::20C:CEFF:FE4B:A2A0, FastEthernet0/3
O   FC00:1:0:37::1/128 [110/1]
     via FE80::20C:CEFF:FE4B:A2A0, FastEthernet0/3

Rack1R6#show ipv6 route ospf
IPv6 Routing Table - 17 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
O   2001:1:0:1234::2/128 [110/847]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
O   2001:1:0:1234::3/128 [110/2]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
O   2001:1:0:1234::5/128 [110/783]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
OI  FC00:1::/56 [110/784]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
OI  FC00:1:0:37::/64 [110/2]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
OI  FC00:1:0:37::1/128 [110/2]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.67
OI  FC00:1:0:37::7/128 [110/1]
     via FE80::212:1FF:FE31:43, FastEthernet0/0.6
```

9.16 IPv6 Redistribution

- Redistribute between OSPFv3 and RIPng on R5.
- Ensure each protocol prefers to reach native prefixes using internal paths.
- R6 should be the preferred exit point for each IPv6 routing domain.

Configuration

```
R5:
ipv6 router ospf 1
 redistribute rip RIPNG metric 8
 redistribute connected metric 8
!
ipv6 router rip RIPNG
 redistribute ospf 1 include-connected metric 8

R6:
ipv6 router ospf 1
 distance ospf external 121
```

Verification

Note

When redistributing between two IPv6 IGPs, remember that the command **`redistribute <protocol>`** will not redistribute the locally connected interfaces advertised into <protocol>. For example, on R6, if you configure **`redistribute rip RIPNG`**, only the routes learned by R6 via RIPng will be redistributed (not the local Loopback100, which is advertised into RIPng).

In order to account for directly connected subnets, use the separate command **`redistribute connected`** or use the **`include-connected`** keyword when performing redistributing.

Since OSPFv3 has a better administrative distance than RIPng, adjust the OSPFv3 external distance on R6 to make sure that external OSPFv3 prefixes are not preferred over internal RIPng prefixes.

```
Rack1R6#show ipv6 route rip
IPv6 Routing Table - 22 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   2001:1:0:1234::/64 [120/2]
     via FE80::20D:65FF:FE84:6560, FastEthernet0/0.146
     via FE80::20F:24FF:FE11:4961, FastEthernet0/0.146
R   FC00:1::/61 [120/2]
     via FE80::20D:65FF:FE84:6560, FastEthernet0/0.146
R   FC00:1:0:8::8/128 [120/10]
     via FE80::20D:65FF:FE84:6560, FastEthernet0/0.146
     via FE80::20F:24FF:FE11:4961, FastEthernet0/0.146
R   FC00:1:0:58::/64 [120/10]
     via FE80::20D:65FF:FE84:6560, FastEthernet0/0.146
     via FE80::20F:24FF:FE11:4961, FastEthernet0/0.146
R   FC00:1:0:85::/64 [120/10]
     via FE80::20D:65FF:FE84:6560, FastEthernet0/0.146
     via FE80::20F:24FF:FE11:4961, FastEthernet0/0.146
R   FC00:1:0:88::88/128 [120/10]
     via FE80::20D:65FF:FE84:6560, FastEthernet0/0.146
     via FE80::20F:24FF:FE11:4961, FastEthernet0/0.146
```

```
Rack1R2#show ipv6 route ospf
IPv6 Routing Table - 12 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
O   2001:1:0:1234::3/128 [110/128]
     via FE80::5, Serial0/0
O   2001:1:0:1234::5/128 [110/64]
     via FE80::5, Serial0/0
OI  FC00:1::/56 [110/65]
     via FE80::5, Serial0/0
OE2  FC00:1:0:5::/64 [110/8]
     via FE80::5, Serial0/0
OI  FC00:1:0:37::/64 [110/129]
     via FE80::5, Serial0/0
OI  FC00:1:0:37::1/128 [110/128]
     via FE80::5, Serial0/0
OI  FC00:1:0:37::7/128 [110/129]
     via FE80::5, Serial0/0
O   FC00:1:0:67::/64 [110/130]
     via FE80::5, Serial0/0
```

9.17 IPv6 Filtering

- Configure R3's Frame-Relay interface so that only FTP and HTTP traffic from the users on VLAN67 is permitted to flow access the interface.
- Allow DNS queries and responses from the same subnet, and ensure IPv6 routing is not affected.

Configuration

```
R3:
ipv6 access-list FILTER_OUT
 permit tcp fc00:1:0:67::/64 any eq 80
 permit tcp fc00:1:0:67::/64 any range 20 21
 permit udp fc00:1:0:67::/64 any eq 43
!
ipv6 access-list FILTER_IN
 permit tcp any eq 80 fc00:1:0:67::/64
 permit tcp any range 20 21 fc00:1:0:67::/64
 permit udp any eq 43 fc00:1:0:67::/64
 permit 89 any any
!
interface Serial 1/0
 ipv6 traffic-filter FILTER_OUT out
 ipv6 traffic-filter FILTER_IN in
```

Verification

Note

Cisco IOS supports only extended IPv6 access-lists. The syntax remains the same, except the source/destination specifications. Instead of using subnets and wildcard masks, you specify IPv6 prefixes. Aside from that, the logic of the access-lists remains the same. The access-lists are applied as traffic-filters to the interfaces.

Note that IPv6 access-lists do not have a keyword to permit OSPFv3 explicitly, so you need to use its protocol number, which is 89.

Generate some traffic to test the access-lists from SW1.

```
Rack1SW1#ping 2001:1:0:1234::3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1:0:1234::3, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms

Rack1SW1#ping 2001:1:0:1234::5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1:0:1234::5, timeout is 2
seconds:
AAAAA
Success rate is 0 percent (0/5)
```

Check that OSPFv3 adjacencies are OK on R3:.

```
Rack1R3#show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
150.1.7.7         1   FULL/  -            -        2033            OSPFv3_VL0
150.1.5.5         1   FULL/  -        00:01:48    6               Serial1/0
150.1.7.7         1   FULL/  -        00:00:31    1021            FastEthernet0/0
```

Verify that HTTP filtering is working:

```
Rack1SW1#telnet 2001:1:0:1234::5 80
Trying 2001:1:0:1234::5, 80 ...
% Destination unreachable; gateway or host down

Rack1SW1#telnet 2001:1:0:1234::5 80 /source-interface vlan 67
Trying 2001:1:0:1234::5, 80 ... Open
```

9.18 IPv6 NAT-PT

- Configure R6 so that SW1 can access the IPv4 address 150.X.4.4 as 2000::9601:404.
- SW1 should source IPv6 packets off its VLAN67 interface and R6 should translate it to the IPv4 address 155.X.146.7.

Configuration

```
R6:
interface FastEthernet 0/0.67
 ipv6 nat
!
interface FastEthernet 0/0.146
 ipv6 nat
!
ipv6 nat v6v4 source fc00:1:0:67::7 155.1.146.7
ipv6 nat v4v6 source 150.1.4.4 2000::9601:0404
!
ipv6 nat prefix 2000::/96
!
ipv6 route rip RIPNG
 redistribute connected
```

Verification

Note

IPv6 NAT-PT allows IPv6 and IPv4 domains to communicate by rewriting packet headers and replacing addresses. In contrast to IPv4 NAT, both the source and destination addresses of every packet must be rewritten. Specifically, IPv6 NAT-PT works as follows.

A block of IPv6 addresses is selected to represent the IPv4 address space. This block usually has a /96 prefix length, to cover all 2^32 IPv4 addresses, but otherwise it could be arbitrary. If needed, selected IPv4 addresses are mapped to respective IPv6 addresses, using static translation rules. When an IPv6 host needs to communicate with IPv4 host, it sends an IPv6 packets to the address within the /96 block. The translating router intercepts the packet going to the selected /96 prefix, and replaces the IPv6 header with an IPv4 header, rewriting both the source and the destination IPv6 addresses with IPv4 addresses. When a response IPv4 packet returns, the router performs the reverse translation.

In order to configure IPv6 NAT-PT, you have to specify the following:

1) Rules to translate IPv4 source addresses to IPv6 addresses
2) Rules to translate IPv6 source addresses to IPv4 addresses
3) The /96 prefix to map the IPv4 address space into

To verify this configuration, enable IPv6 NAT-PT debugging on R6 and send packets from SW1 to R4.

```
Rack1R6#debug ipv6 nat detailed
IPv6 NAT-PT detailed debugging is on

Rack1SW1#ping ipv6 2000::9601:0404

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2000::9601:404, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/4/8 ms
Rack1SW1#

Rack1R6#
IPv6 NAT: ipv6nat_find_entry_v4tov6:
         ref_count = 1,
                              usecount = 0, flags = 64, rt_flags = 0,
                              more_flags = 0

IPv6 NAT: icmp src (FC00:1:0:67::7) -> (155.1.146.7), dst
(2000::9601:404) -> (150.1.4.4)
IPv6 NAT: ipv6nat_find_entry_v4tov6:
         ref_count = 1,
                              usecount = 0, flags = 64, rt_flags = 0,
                              more_flags = 0

IPv6 NAT: icmp src (150.1.4.4) -> (2000::9601:404), dst (155.1.146.7) -
> (FC00:1:0:67::7)
IPv6 NAT: ipv6nat_find_entry_v4tov6:
         ref_count = 1,
                              usecount = 0, flags = 64, rt_flags = 0,
                              more_flags = 0
```

9.19 IPv6 MP-BGP

- Configure R1 and R5 in BGP autonomous systems 100 and 500 respectively, and peer them over the Frame-Relay cloud.
- Create new loopback interfaces on R1 with the IPv6 addresses 2003:X:0:1::1/64 and 2003:X:0:11::11/61, and advertise them into BGP.
- Ensure R5 sees only an optimal summary route encompassing both IPv6 prefixes.

Configuration

```
R1:
interface Loopback 101
 ipv6 address 2003:1:0:1::1/64
!
interface Loopback 102
 ipv6 address 2003:1:0:11::11/61
!
router bgp 100
 address-family ipv6 unicast
  neighbor 2001:1:0:1234::5 remote-as 500
  neighbor 2001:1:0:1234::5 activate
  network 2003:1:0:1::/64
  network 2003:1:0:10::/61
  aggregate-address 2003:1::/59 summary-only

R5:
router bgp 500
 address-family ipv6 unicast
  neighbor 2001:1:0:1234::1 remote-as 100
  neighbor 2001:1:0:1234::1 activate
```

Verification

Note

Multi-protocol extensions for BGP adds supports for the IPv6 address family, which essentially means that IPv6 prefixes and attributes can be exchanged over a normal TCP based BGP peering. IPv6 peers are configured under the **`address-family ipv6 unicast`** configuration mode inside of BGP, and then must be "activated", or enabled. Technically this the same process as is used in IPv4 peerings, but the normal global BGP process automatically refers to the **`address-family ipv4 unicast`**, and the command **`bgp default ipv4-unicast`** is enabled by default, which means that all unicast IPv4 peers are automatically "activated".

All IPv6 related options are configured under the address-family, such as the network statement, or applying attributes onto neighbors like route-maps, prefix-lists, filter-lists, etc. Keep in mind that all other BGP operation stays the same, such as advertisement rules, route reflection, confederation, bestpath selection, etc. The only thing that is mainly changing is the format of the prefix and next-hop advertised inside of the BGP update message.

Note that for EBGP peerings the next-hop values recurse to link-local addresses in the routing table, so when peering over multipoint NBMA, link-local IPv6 addresses must be mapped.

Prefix summarization uses the same command as in IPv4 and the same prefix length calculations. In our case, take the mismatched parts of the prefixes (remember, 11 is in hex, thus it is 17 in decimal).

0000 0000 000**0 0001**
0000 0000 000**1 0**

(note that the second prefix length is 61, not 64). To build the summary, shift prefix length left by 5 bits. The resulting prefix is:

2003:1::/59

```
Rack1R1#show bgp ipv6 unicast summary
BGP router identifier 150.1.1.1, local AS number 100
BGP table version is 6, main routing table version 6
3 network entries using 447 bytes of memory
3 path entries using 228 bytes of memory
3/2 BGP path/bestpath attribute entries using 372 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1047 total bytes of memory
BGP activity 3/0 prefixes, 3/0 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
2001:1:0:1234::5
                4   500       4       5        6    0    0 00:00:03
0

Rack1R1#show bgp ipv6 unicast
BGP table version is 6, local router ID is 150.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2003:1::/59      ::                                 32768 i
s> 2003:1:0:1::/64  ::                       0         32768 i
s> 2003:1:0:10::/61 ::                       0         32768 i

Rack1R5#show bgp ipv6 unicast
BGP table version is 8, local router ID is 150.1.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2003:1::/59      2001:1:0:1234::1
                                             0             0 100 i
```

9.20 IPv6 Tunneling

- Create new Loopback interfaces on R2, R5, and R6 with the IP addresses 2001:X:0:Y::Y/64 where Y is the router number and X is your rack number in decimal notation (e.g. 10 for rack 10, 11 for rack 11 etc.)
- Create tunnels between R5 & R6 and R2 & R6. Use the IPv6 addresses 2001:X:0:56::Y/64 for the tunnel between R5 and R6 and the IPv6 address 2001:X:0:26::Y/64 for the tunnel between R2 and R6.
- The tunnel connecting R2 and R6 should use encapsulation suitable for transporting different Layer 3 protocols.
- The tunnel connecting R5 and R6 should use the encapsulated specifically designed for transporting IPv6 over IPv4.
- Use static routing to obtain connectivity.

Configuration

```
R2:
ipv6 unicast-routing
!
interface Loopback100
 ipv6 address 2001:1:0:2::2/64
!
interface tunnel 26
 tunnel source Loopback0
 tunnel destination 150.1.6.6
 ipv6 address 2001:1:0:26::2/64
!
ipv6 route ::/0 Tunnel 26

R5:
ipv6 unicast-routing
!
interface Loopback100
 ipv6 address 2001:1:0:5::5/64
!
interface tunnel 56
 tunnel source Loopback0
 tunnel destination 150.1.6.6
 ipv6 address 2001:1:0:56::5/64
 tunnel mode ipv6ip
!
ipv6 route ::/0 Tunnel 56
```

```
R6:
ipv6 unicast-routing
!
interface Loopback100
 ipv6 address 2001:1:0:6::6/64
!
interface tunnel 26
 tunnel source Loopback0
 tunnel destination 150.1.2.2
 ipv6 address 2001:1:0:26::6/64
!
interface tunnel 56
 tunnel source Loopback0
 tunnel destination 150.1.5.5
 ipv6 address 2001:1:0:56::6/64
 tunnel mode ipv6ip
!
ipv6 route 2001:1:0:5::/64 Tunnel 56
ipv6 route 2001:1:0:2::/64 Tunnel 26
```

Verification

Note

IPv6 can be transported across IPv4 clouds using various tunneling techniques. The most common are static point-to-point tunnels, using either GRE (generic routing encapsulation), or IPv6 in IPv4 encapsulation (a special protocol type to carry only IPv6 packets). GRE is designed to carry a multi-protocol payload, and thus has a slightly larger over than IPv6 in IPv4 encapsulation.

```
Rack1R6#show interfaces tunnel 26
Tunnel26 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 150.1.6.6 (Loopback0), destination 150.1.2.2
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
<snip>
```

```
Rack1R6#show interfaces tunnel 56
Tunnel56 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 150.1.6.6 (Loopback0), destination 150.1.5.5
  Tunnel protocol/transport IPv6/IP
  Tunnel TTL 255
<snip>
```

```
Rack1R2#ping 2001:1:0:5::5 source loopback 100

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1:0:5::5, timeout is 2 seconds:
Packet sent with a source address of 2001:1:0:2::2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 156/167/205
ms
```

☠ Pitfall

IPv6IP tunnels use IP protocol number 41 for transport. This protocol number does not have a keyword shortcut in extended IP access-lists in IOS. Therefore to permit or deny an IPv6IP tunnel, the syntax **`access-list 100 [permit|deny] 41 any any`** is required.

9.21 Automatic 6to4 Tunneling

- Using the IPv4 Loopback0 interfaces create automatic 6to4 tunnels connecting R3, R4, and R5.
- Create additional Loopback interfaces on every router with the subnet number zero and the prefix length of 64 under the respective 6to4 /48 prefix.
- Use static routing to obtain connectivity between the newly allocated subnets.

Configuration

```
R3:
interface Tunnel 345
 tunnel source Loopback0
 tunnel mode ipv6ip 6to4
 ipv6 address 2002:9601:303::3/64
!
ipv6 route 2002::/16 Tunnel 345
!
interface Loopback200
 ipv6 address 2002:9601:303:1::3/64

R4:
interface Tunnel 345
 tunnel source Loopback0
 tunnel mode ipv6ip 6to4
 ipv6 address 2002:9601:404::5/64
!
ipv6 route 2002::/16 Tunnel 345
!
interface Loopback200
 ipv6 address 2002:9601:404:1::4/64

R5:
interface Tunnel 345
 tunnel source Loopback0
 tunnel mode ipv6ip 6to4
 ipv6 address 2002:9601:505::5/64
!
ipv6 route 2002::/16 Tunnel 345
!
interface Loopback200
 ipv6 address 2002:9601:505:1::5/64
```

Verification

Note

Automatic 6to4 tunnels are multipoint by their design. The idea is to allow automatic routing across an IPv4 cloud based on part of the destination IPv6 address. Specifically, the format of the 6to4 IPv6 address is as following:

2002 (16 bits):IPv4 address (32 bits):Subnet ID(16 bits):Interface ID (64 bits)

When a packet is routed across the 6to4 tunnel, the router extracts the IPv4 address embedded in the IPv6 address and uses it to build the IPv4 destination address of the tunnel header. The receiving router strips the header, extracts the IPv6 packet, and routes it based on the IPv6 routing table.

As you can see, 6to4 subnets have some addressing restriction. First, you need to use the 16-bit prefix 2002, as it is the common reservation for all 6to4 deployments. Second, you need to select a public IPv4 address to create the /48 prefix. It is common to pick any interface of the border router and then allocate the /64 subnets to other devices on the network, as long as the address is publicly routable.

6to4 tunnels were intended as a transition mechanism for hosts that don’t have native IPv6 connectivity, allowing them to reach other nodes that have full connectivity to the IPv6 Internet. Due to the multipoint nature of the 6to4 tunnels, only static routing is possible with this technology. It is common to simply route the whole 2002::/16 prefix to the 6to4 tunnel.

In our case, the use of Loopback0 subnets results in the following IPv6 6to4 prefixes:

R3: 150.1.3.3 = 2002:9601:303::/48
R4: 150.1.4.4 = 2002:9601:404::/48
R5: 150.1.5.5 = 2002:9601:505::/48

“9601” in hex corresponds to “150.1” in decimal, and on R3 “303” corresponds to “3.3”.

```
Rack1R5#show interfaces tunnel 345
Tunnel345 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 150.1.5.5 (Loopback0), destination UNKNOWN
  Tunnel protocol/transport IPv6 6to4

  Fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input never, output 00:00:13, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     11 packets output, 1176 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out

Rack1R5#ping ipv6 2002:9601:404:1::4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2002:9601:404:1::4, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/45/52 ms

Rack1R5#ping ipv6 2002:9601:303:1::3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2002:9601:303:1::3, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/70/72 ms
```

9.22 ISATAP Tunneling

- Remove the 6to4 tunnels connecting R3, R4, and R5.
- Using the Loopback0 IPv4 addresses connect R1, SW1, and SW2 via ISATAP tunnels.
- Use the /64 IPv6 prefix 2001:1:0:345::/64 to allocated IPv6 addresses to the tunnel endpoints.
- Create additional IPv6 Loopback interfaces on all three routers with the IPv6 addresses 2001:1:0:Y::Y/64, and use static routing to obtain full connectivity.

Configuration

```
R3:
interface Tunnel345
 ipv6 address 2001:1:0:345::/64 eui-64
 tunnel source Loopback0
 tunnel mode ipv6ip isatap
!
interface Loopback100
 ipv6 address 2001:1:0:3::3/64
!
ipv6 route 2001:1:0:4::/64 2001:1:0:345:0:5efe:9601:404
ipv6 route 2001:1:0:5::/64 2001:1:0:345:0:5efe:9601:505

R4:
interface Tunnel345
 ipv6 address 2001:1:0:345::/64 eui-64
 tunnel source Loopback0
 tunnel mode ipv6ip isatap
!
interface Loopback100
 ipv6 address 2001:1:0:4::4/64
!
ipv6 route 2001:1:0:3::/64 2001:1:0:345:0:5efe:9601:303
ipv6 route 2001:1:0:5::/64 2001:1:0:345:0:5efe:9601:505

R5:
interface Tunnel345
 ipv6 address 2001:1:0:345::/64 eui-64
 tunnel source Loopback0
 tunnel mode ipv6ip isatap
!
interface Loopback100
 ipv6 address 2001:1:0:5::5/64
!
ipv6 route 2001:1:0:4::/64 2001:1:0:345:0:5efe:9601:404
ipv6 route 2001:1:0:3::/64 2001:1:0:345:0:5efe:9601:303
```

Verification

> ✎ **Note**
>
> ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) is a technology that contrasts 6to4 automatic tunnels. While 6to4 automatically generate a /48 prefix out of an IPv4 address, ISATAP uses the IPv4 domain as a “multi-access” media with IPv4 addresses being the rough equivalent of Ethernet MAC addresses.
>
> Specifically, ISATAP constructs the interface identifier (last 64 bits) of the IPv6 address based on the IPv4 address of a host, and using the EUI-64 rules. Thus, given that you already have a /64 IPv6 prefix, you can allocate IPv6 addresses to tunnel endpoints performing simple manipulations over the IPv4 endpoint addresses. Specifically, the interface ID is constructed as follows:
>
> EUI-64 = 0000 (16 bits) + 5EFE (16 bits) + IPv4 Address (32 bits).
>
> Therefore, if you select the prefix 2001:1:0:345::/64, then R3 will have the IP address:
>
> 2001:1:0:345:0:5efe:9601:0303/64
>
> Where 9601 is for 150.1 and 0303 is for last two octets of the IP address. When configuring an IOS router for ISATAP, simply use the **`eui-64`** keyword to automatically generate the interface ID.
>
> Unlike the 6to4 tunnels, ISATAP tunnels cannot automatically extract the destination. Therefore, you must use static routes that point to the exact IPv6 endpoint on the other end of the tunnel.

```
Rack1R5#show interfaces tunnel 345
Tunnel345 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 150.1.5.5 (Loopback0), destination UNKNOWN
  Tunnel protocol/transport IPv6 ISATAP

  <snip>
```

```
Rack1R5#show ipv6 interface tunnel 345
Tunnel345 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::5EFE:9601:505
  Global unicast address(es):
    2001:1:0:345:0:5EFE:9601:505, subnet is 2001:1:0:345::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF01:505
  MTU is 1480 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is not supported
  ND reachable time is 30000 milliseconds
  Hosts use stateless autoconfig for addresses.

Rack1R5#ping 2001:1:0:345:0:5efe:9601:404

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1:0:345:0:5EFE:9601:404, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/44/45 ms

Rack1R5#ping 2001:1:0:345:0:5efe:9601:303

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1:0:345:0:5EFE:9601:303, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/69/73 ms

Rack1R5#ping 2001:1:0:4::4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1:0:4::4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/44/45 ms

Rack1R5#ping 2001:1:0:3::3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1:0:3::3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/69/72 ms
```