Cisco live!

# What You Make Possible

BUILT FOR
THE HUMAN
NETWORK
CISCO

# IPv6 for R&S CCIE Candidates

Johnny Bass
CCIE #6458

BUILT FOR THE HUMAN NETWORK

# About the Presenter

Johnny Bass

- Networking industry since the late 1980s
- CCIE R&S #6458
- CCSI 97168
- Cisco 360 R&S Master Instructor
- Course director for several programs, including Cisco 360 for Route Switch, for Global Knowledge

# Why Are We Here?

- Show of hands, how many of you are currently supporting IPv6?

# CCIE R&S V4.0 IPv6 Section

| **3.00** | **Implement IPv6** |
|---|---|
| 3.10 | Implement IP version 6 (IPv6) addressing and different addressing types |
| 3.20 | Implement IPv6 neighbor discovery |
| 3.30 | Implement basic IPv6 functionality protocols |
| 3.40 | Implement tunneling techniques |
| 3.50 | Implement OSPF version 3 (OSPFv3) |
| 3.60 | Implement EIGRP version 6 (EIGRPv6) |
| 3.70 | Implement filtering and route redistribution |

# Agenda

- **IP version 6 (IPv6) addressing and different addressing types**
- IPv6 neighbor discovery
- Basic IPv6 functionality protocols
- Tunneling techniques
- IPv6 Unicast routing
- Filtering and route redistribution
- IPv6 Multicast
- IPv6 and 3560 switches
- IPv6 NAT protocol translation
- Troubleshooting IPv6
- CCIE R&S
- Q&A

# IPv6 Addressing and Different Addressing Types

- Globally significant addressing
- Link local
- Site local
- Unique local
- Multicast
- Anycast
- Address assignments
  - Static
  - DHCPv6
  - Stateless auto configure
  - IPv4 Compatible

# IPv6 Addressing and Different Addressing Types

| IPv6 Prefix | Allocation |
|---|---|
| 0000::/8 | Reserved by IETF |
| 0100::/8 | Reserved by IETF |
| 0200::/7 | Reserved by IETF |
| 0400::/6 | Reserved by IETF |
| 0800::/5 | Reserved by IETF |
| 1000::/4 | Reserved by IETF |
| 2000::/3 | Global Unicast |
| 4000::/3 | Reserved by IETF |
| 6000::/3 | Reserved by IETF |
| 8000::/3 | Reserved by IETF |
| A000::/3 | Reserved by IETF |
| C000::/3 | Reserved by IETF |
| E000::/4 | Reserved by IETF |
| F000::/5 | Reserved by IETF |
| F800::/6 | Reserved by IETF |
| FC00::/7 | Unique Local Unicast |
| FE00::/9 | Reserved by IETF |
| FE80::/10 | Link Local Unicast |
| FEC0::/10 | Reserved by IETF |
| FF00::/8 | Multicast |

# IPv6 Addressing and Different Addressing Types - Globally Significant Addressing

| | |
|---|---|
| 2001:4200::/23 | AfriNIC |
| 2C00:0000::/12 | AfriNIC |
| 2001:0200::/23 | APNIC |
| 2001:0C00::/23 | APNIC |
| 2001:0E00::/23 | APNIC |
| 2001:4400::/23 | APNIC |
| 2001:8000::/19 | APNIC |
| 2001:A000::/20 | APNIC |
| 2001:B000::/20 | APNIC |
| 2400:0000::/12 | APNIC |
| 2001:0400::/23 | ARIN |
| 2001:1800::/23 | ARIN |
| 2001:4800::/23 | ARIN |
| 2600:0000::/12 | ARIN |
| 2610:0000::/23 | ARIN |
| 2620:0000::/23 | ARIN |
| 2001:1200::/23 | LACNIC |
| 2800:0000::/12 | LACNIC |

| | |
|---|---|
| 2001:0600::/23 | RIPE NCC |
| 2001:0800::/23 | RIPE NCC |
| 2001:0A00::/23 | RIPE NCC |
| 2001:1400::/23 | RIPE NCC |
| 2001:1600::/23 | RIPE NCC |
| 2001:1A00::/23 | RIPE NCC |
| 2001:1C00::/22 | RIPE NCC |
| 2001:2000::/20 | RIPE NCC |
| 2001:3000::/21 | RIPE NCC |
| 2001:3800::/22 | RIPE NCC |
| 2001:4000::/23 | RIPE NCC |
| 2001:4600::/23 | RIPE NCC |
| 2001:4A00::/23 | RIPE NCC |
| 2001:4C00::/23 | RIPE NCC |
| 2001:5000::/20 | RIPE NCC |
| 2003:0000::/18 | RIPE NCC |
| 2A00:0000::/12 | RIPE NCC |

| | |
|---|---|
| 2002:0000::/16 | 6to4 |
| 2001:0000::/23 | IANA |
| 2001:3C00::/22 | IANA |
| 2D00:0000::/8 | IANA |
| 2E00:0000::/7 | IANA |
| 3000:0000::/4 | IANA |

# IPv6 Addressing and Different Addressing Types - Globally Significant Addressing

| Prefix | Subnet ID | Interface/Node ID |
|---|---|---|
| 32 to 56 bits | 32 to 8 bits | 64 bits |

- The interface ID can be either EUI-64, use Privacy Extensions (RFC 3041) or locally configured
  - EUI-64 takes the MAC address and modifies it to fit the 64 bit field

00-DD-011C-DB-80

02- DD-01FF-FE1C-DB-80

2005:0:DEAD:BEEF:2DD:1FF:FE1C:DB80

# IPv6 Addressing and Different Addressing Types - Link Local

| 10 bits | 38 bits | 16 bits | 64 bits |
|---|---|---|---|
| 1111 1110 10 | 0000….0000 | Sub-net ID | Interface ID |

- Link local address (FE80::/10) are assigned to any and all interfaces configured for IPv6, regardless if the interface has a routable address.
- Link local address can be used to speak between devices within the same layer 2 domain
- Link local address are not routable
- They can be used to limit the scope of delivery of packets that not to be routed (such as IGPs)

# IPv6 Addressing and Different Addressing Types - Link Local

| 10 bits | 38 bits | 16 bits | 64 bits |
|---|---|---|---|
| 1111 1110 10 | 0000….0000 | Sub-net ID | Interface ID |

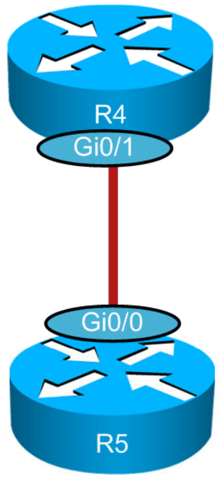
- Link local address are used to source ICMPv6 packets for neighbor discovery and stateless address autoconfiguration
- The interface ID can be either EUI-64, use Privacy Extensions or locally configured

FE80::2DD:1FF:FE1C:DB80

# IPv6 Addressing and Different Addressing Types - Link Local

```
interface GigabitEthernet0/1
 ipv6 enable
```

```
interface GigabitEthernet0/1
 ipv6 address FE80::5 link-local
```



# IPv6 Addressing and Different Addressing Types - Site Local

- IPv6 site-local addresses are similar to IPv4 private addresses. The scope of site local is within an organization.
- Not to be advertised on the Internet
- FEC0::/10

| 10 bits | 38 bits | 16 bits | 64 bits |
|---|---|---|---|
| 1111 1110 11 | 0000….0000 | Sub-net ID | Interface ID |

- Deprecated!

# IPv6 Addressing and Different Addressing Types - Unique Local

- Replaces Site Local
- RFC 4193
- FC00::/7 is to be used within an enterprise
  - FC00::/8 planned to be globally managed
  - FD00::/8 assigned locally by the network administrator
- Not to be advertised on the Internet

| 7 bits | | 40 bits | 16 bits | 64 bits |
|---|---|---|---|---|
| 1111 110 | X | Global ID | Sub-net ID | Interface ID |

# IPv6 Addressing and Different Addressing Types - Multicast

- Multicast - FF00::/8

| 8 bits | 4 bits | 4 bits | 112 bits |
|---|---|---|---|
| 11111111 | Flag | Scope | Group ID |

- Flag bits:

  Only the least significant bit is currently used: 0 is a permanent or well known multicast, 1 is not permanent or transient multicast

- Scope: 1 – Node Local (only useful for multicast to loopback interfaces)
  2 – Link Local
  5 – Site Local
  8 – Organization Local
  E (14) – Global

# IPv6 Addressing and Different Addressing Types - Anycast

- An anycast address is an address that is assigned to more than one interface (typically belonging to different nodes), with the property that a packet sent to an anycast address is routed to the "nearest" interface having that address, according to the routing protocols' measure of distance.
- Subnet anycast address is the prefix with all zeros in the node portion

# IPv6 Addressing and Different Addressing Types - Address Assignments

- Address assignments
  - Static

    ipv6 address 2005:0:DEAD:BEEF::1/64
  - DHCPv6

    ipv6 address dhcp
  - Stateless auto configure

    Ipv6 address autoconfig

# IPv6 Addressing and Different Addressing Types - Address Assignments

- DHCPv6

| | |
|---|---|
| RFC3315 | Dynamic Host Configuration Protocol for IPv6 (DHCPv6) |
| RFC3319 | Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers |
| RFC3633 | IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) |
| RFC3646 | DNS Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6) |
| RFC3898 | NIS Server Option, NIS Domain Option/NIS+ Server Option, NIS+ Domain Option |
| RFC4075 | Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6 |
| RFC4242 | Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6) |
| RFC4649 | DHCPv6 Relay Agent Remote-ID Option |

# IPv6 Addressing and Different Addressing Types - Address Assignments – DHCPv6

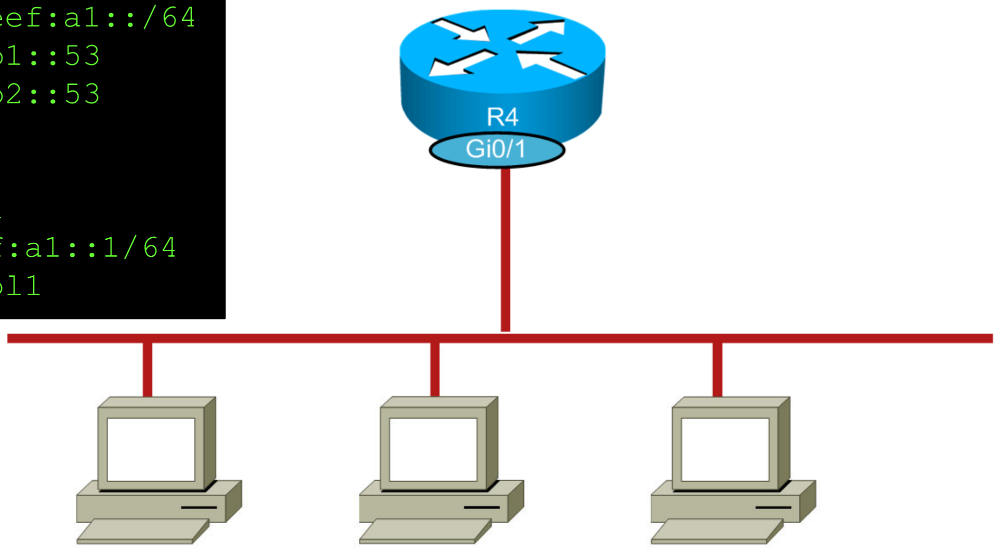
| Message Types | DHCPv4 | DHCPv6 |
|---|---|---|
| Layer 2 MAC address | FF-FF-FF-FF-FF-FF | 33-33-00-00-01-02 |
| Layer 3 address | 255.255.255.255 Broadcast | All on-link DHCPV6-relay agents or servers FF02::1:2 |
| Discovery packet type | DHCP Discovery | Solicit message |
| DHCP server response | DHCP Offer | Advertise message |
| Client to server response to offer | DHCP Request | Request message |
| DHCP server ack to client response | DHCP ACK | Reply message |
| DHCP router relay support | Configured on router to forward DHCP packets | Relay agents use ff05::1:3 All DHCPv6 servers |

# IPv6 Addressing and Different Addressing Types - DHCPv6 Server

- A router can act as a DHCP server
- Operation is similar to IPv4 DHCP
  - Client address are assigned
  - Servers keep a binding table
  - Binding table can be uploaded to another server
- Configuration options
  - DHCP pool name
  - Prefix information
  - Addresses for specific clients
  - DNS servers and domain name

# IPv6 Addressing and Different Addressing Types - DHCPv6 Server

```
ipv6 dhcp pool MyPool1
 address prefix 2005:dead:beef:a1::/64
 dns-server 2005:dead:beef:b1::53
 dns-server 2005:dead:beef:b2::53
 domain-name MyDomain.org
!
interface GigabitEthernet0/1
 ipv6 address 2005:dead:beef:a1::1/64
 ipv6 dhcp server pool MyPool1
```

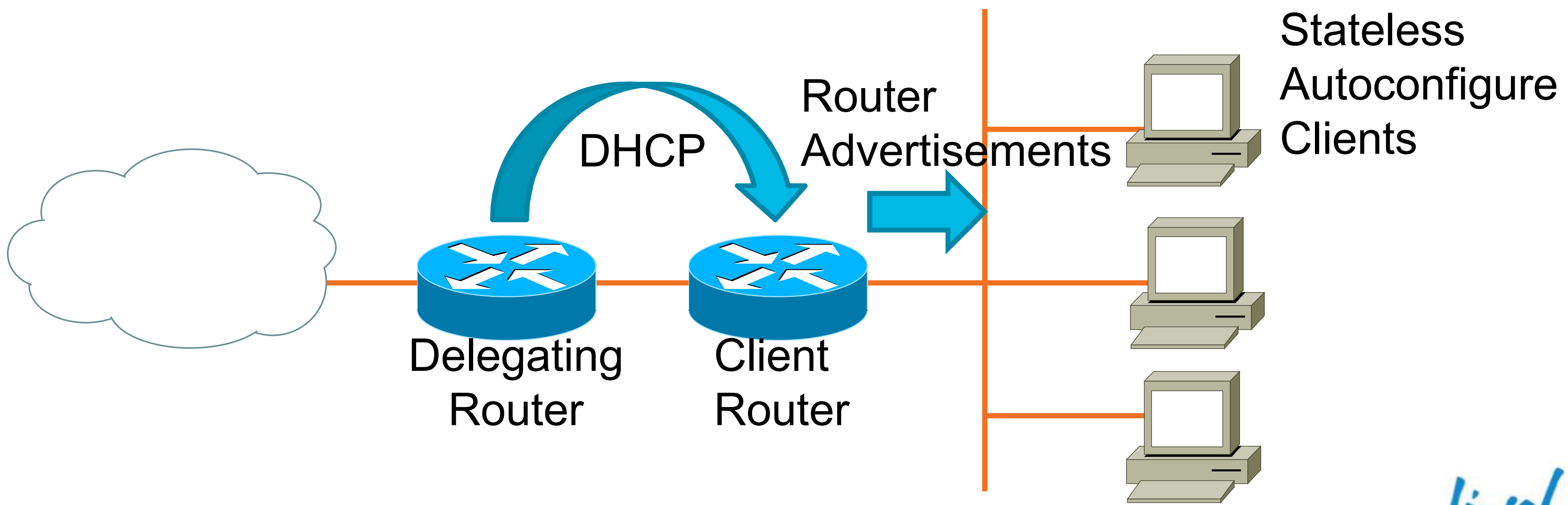


# IPv6 Addressing and Different Addressing Types - DHCPv6 Server

- To store the binding table on another server:
  - `Router(config)#ipv6 dhcp database` *`URL`*
- DHCPv6 client can obtain addressing from a server either through a rapid two-message exchange (solicit, reply) or through a normal four-message exchange (solicit, advertise, request, reply). Default, the four-message exchange is used. When the rapid-commit option is enabled by both client and server, the two-message exchange is used.
  - `Router(config-if)# ipv6 dhcp server MyPool1 rapid-commit`

# IPv6 Addressing and Different Addressing Types - DHCPv6 Delegation

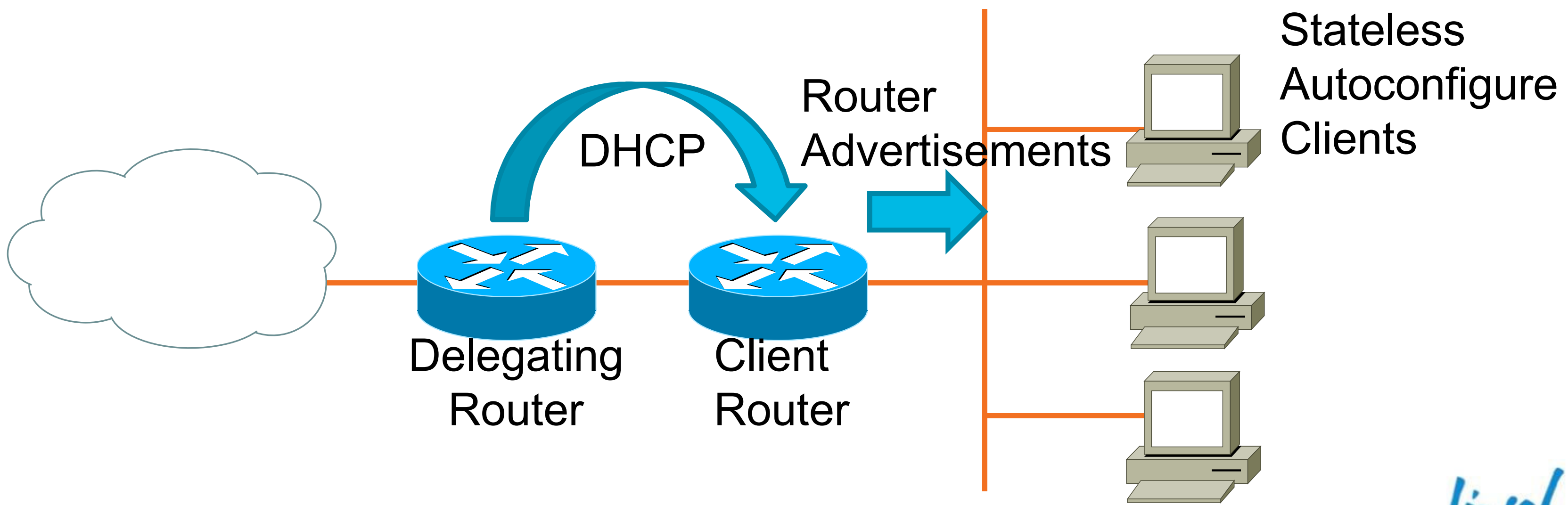
- You can delegate a block of addresses from one server to another to provision
- One device acts as a client to the other, then acts as a server to it's clients



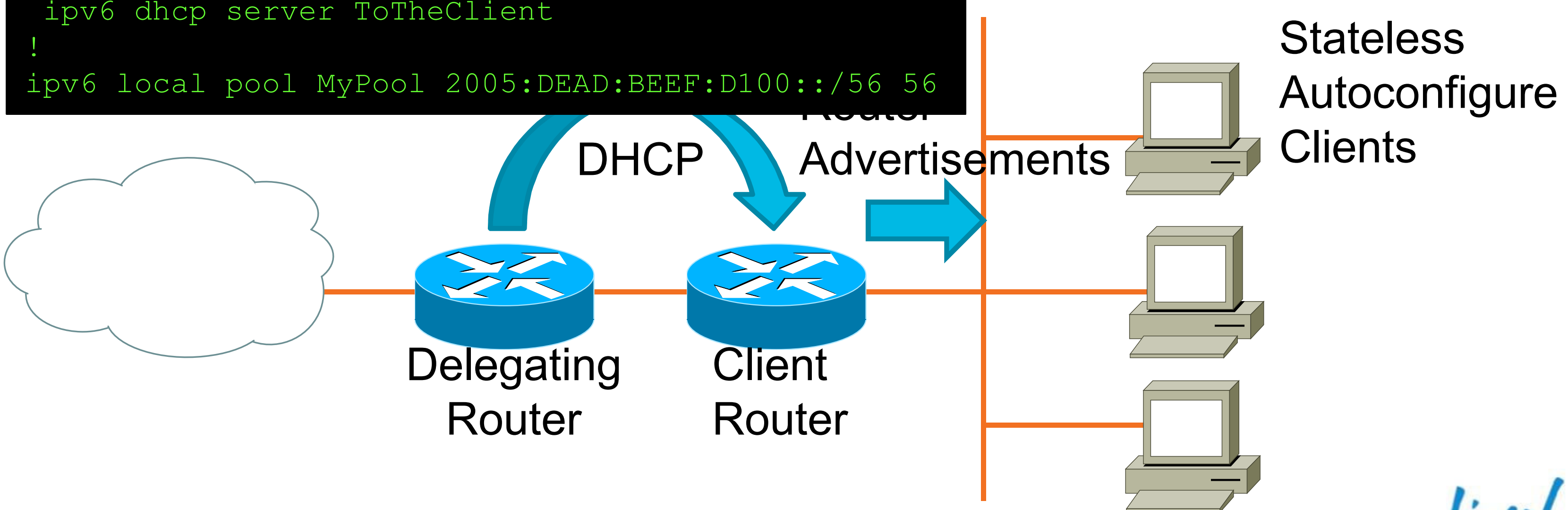
# IPv6 Addressing and Different Addressing Types - DHCPv6 Delegation

- Configure the delegating router
  - Configure a pool of prefixes
  - Enable DHCPv6 on the Client router facing interface



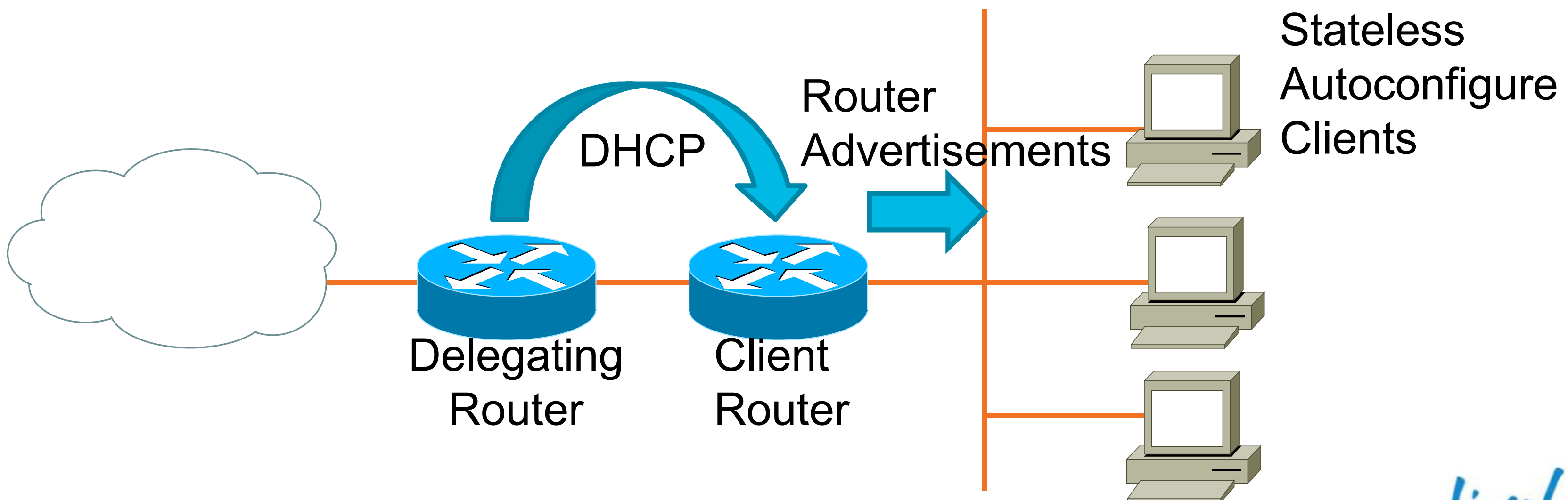
# IPv6 Addressing and Different Addressing Types - DHCPv6 Delegation

```
ipv6 dhcp pool ToTheClient
 prefix-delegation pool MyPool
!
interface GigabitEthernet0/1
 ipv6 address 2005:DEAD:BEEF:C100::1/64
 ipv6 dhcp server ToTheClient
!
ipv6 local pool MyPool 2005:DEAD:BEEF:D100::/56 56
```



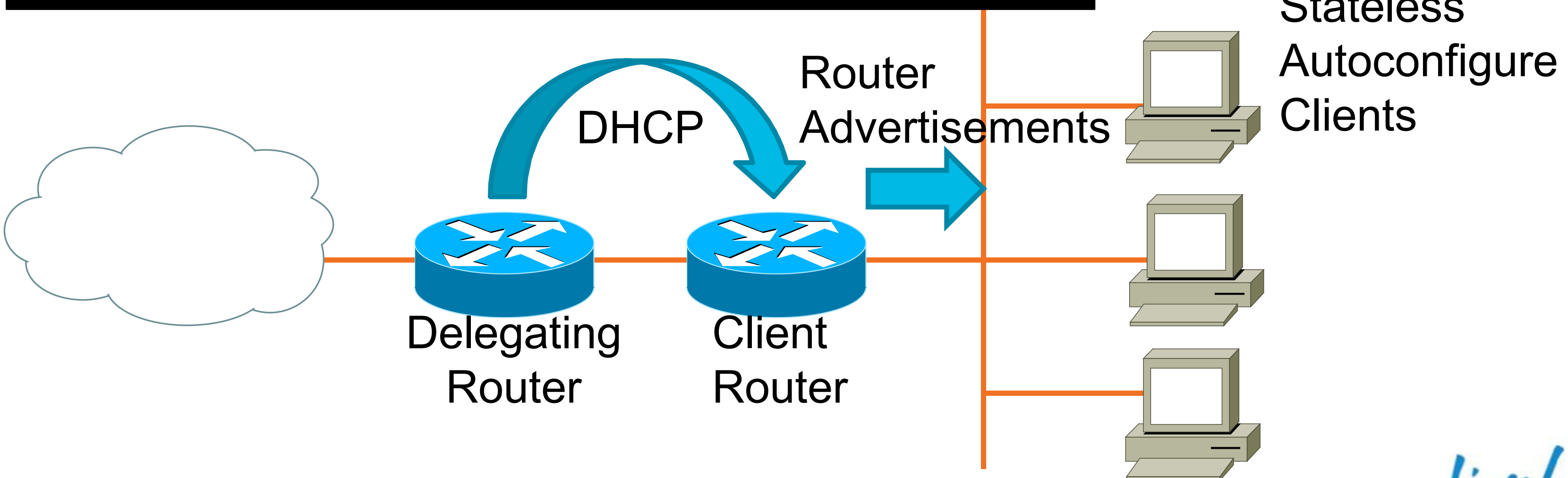
# IPv6 Addressing and Different Addressing Types - DHCPv6 Delegation

- Configure the Client router
  - Interface facing Delegating router is set as DHCP client
  - Host facing interface is rending IPv6 Router Advertisement messages



# IPv6 Addressing and Different Addressing Types - DHCPv6 Delegation

```
interface GigabitEthernet0/0
 ipv6 address 2005:DEAD:BEEF:C100::2/64
 ipv6 dhcp client pd ToTheHost
!
interface GigabitEthernet0/1
 ipv6 address ToTheHost ::1:0:0:0:1/64
```



# IPv6 Addressing and Different Addressing Types - Address Assignments – Autoconfigure

- Stateless Address Autoconfigure (SLAAC)
  - Uses ICMPv6 router discovery
  - Router reports back with prefix(s) for the link and default router

    Options:

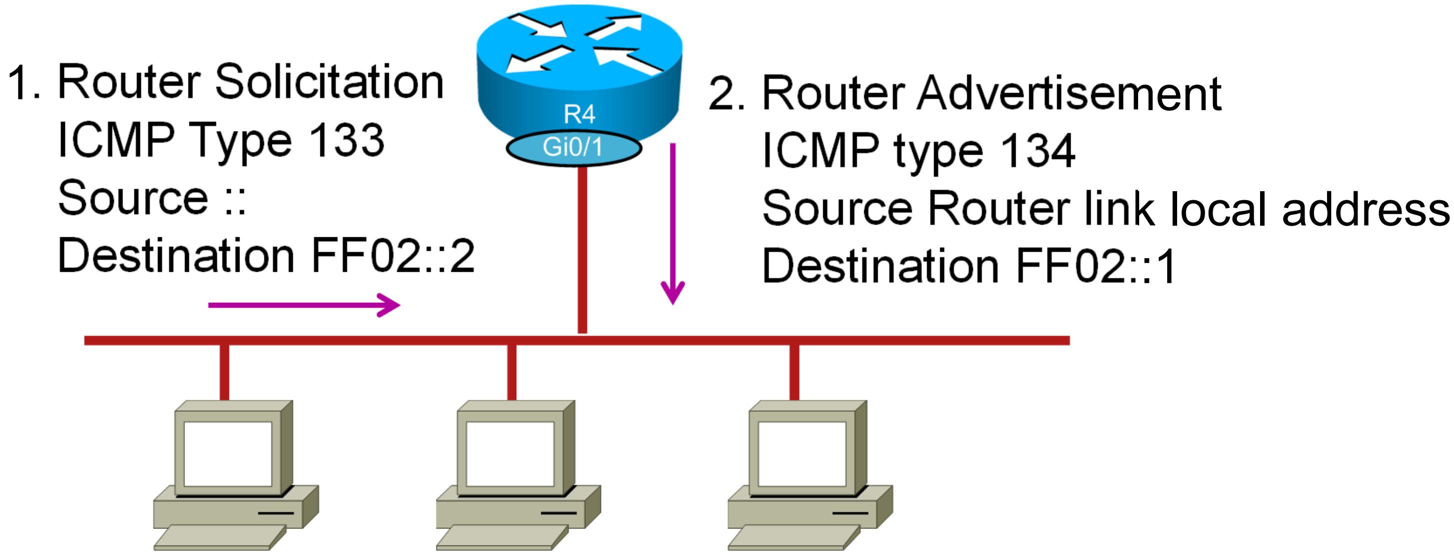
      Lifetime of advertisement

    MTU

    Prefix Length

    Router Priority
  - Host completes IPv6 address with either a preconfigured host portion, use Privacy Extensions or with the IEEE EUI-64 process

# IPv6 Addressing and Different Addressing Types - Address Assignments – Autoconfigure




- Router solicitations are sent by nodes to request router advertisements so to configure their interface

# IPv6 Addressing and Different Addressing Types – IPv4 Compatible

- IPv4-compatible addresses are derived from IPv4 addresses
- This provides a method for connecting IPv6 hosts or sites over the existing IPv4 infrastructure
- IPv6 traffic, when used with IPv4-compatible addresses, does not require the addition of IPv6 routers…Its traffic is encapsulated with an IPv4 header.
- IPv4-compatable Tunnel!
  – We'll get back to this
- ::192.168.14.1

# IPv6 Addressing and Different Addressing Types – IPv4 Verses IPv6

- IPv4
  - 127.0.0.1 (loopback)
  - 0.0.0.0 (undefined)
  - 169.254.0.0/16 (link local)
  - 0.0.0.0/0 (default)
  - 224.0.0.0/8 (link local multicast)
  - 224.0.0.1 (all local nodes)
  - 224.0.0.2 (all local routers)
  - 224.0.0.5 & 6 (OSPFv2)
  - 224.0.0.9 (RIPv2)
  - 224.0.0.10 (EIGRP)
- IPv6
  - ::1 (loopback)
  - :: (undefined)
  - FE80::/10 (link local)
  - ::/0 (default)
  - FF02::/16 (link local multicast)
  - FF02::1 (all local nodes)
  - FF02::2 (all local routers)
  - FF02::5 & 6 (OSPFv3)
  - FF02::9 (RIPng)
  - FF02::A (EIGRPv6)

# Agenda

- IP version 6 (IPv6) addressing and different addressing types
- **IPv6 neighbor discovery**
- Basic IPv6 functionality protocols
- Tunneling techniques
- IPv6 Unicast routing
- Filtering and route redistribution
- IPv6 Multicast
- IPv6 and 3560 switches
- IPv6 NAT protocol translation
- Troubleshooting IPv6
- CCIE R&S
- Q&A

# IPv6 Neighbor Discovery

- How do we find out the layer 2 address when we have an IPv4 address?
  – ARP
- How do we find out the layer 2 address when we have an IPv6 address?
  – Neighbor discovery
- Neighbor discovery handles the following:
  – Duplicate address detection (DAD)
  – Layer 2 addressing for neighbors
  – Finding neighbors on a link
- Neighbor discovery uses ICMPv6 messages, sent to multicast addresses

# IPv6 Neighbor Discovery

- ICMPv6 messages used for Neighbor discovery are:
  - Type 133: Router Solicitation
  - Type 134: Router Advertisement
  - Type 135: Neighbor Solicitation
  - Type 136: Neighbor Advertisement
  - Type 137: Redirect Message

# IPv6 Neighbor Discovery

- IPv6 Address we are looking for:

| Prefix | Interface ID | Lower 24 bits |
|---|---|---|

- Solicited-Node Multicast Address:

| FF02 | 0 | 0001 | FF | Lower 24 bits |
|---|---|---|---|---|

- Solicited-node address is a link local multicast that has the target node’s least significant 24 bits within it

# IPv6 Neighbor Discovery

- IPv6 Address we are looking for:

| | | |
|---|---|---|
| 2005:DEAD:BEEF:00A1: | 02DD:01FF:FE | 00:0DB8 |

- Solicited-Node Multicast Address:

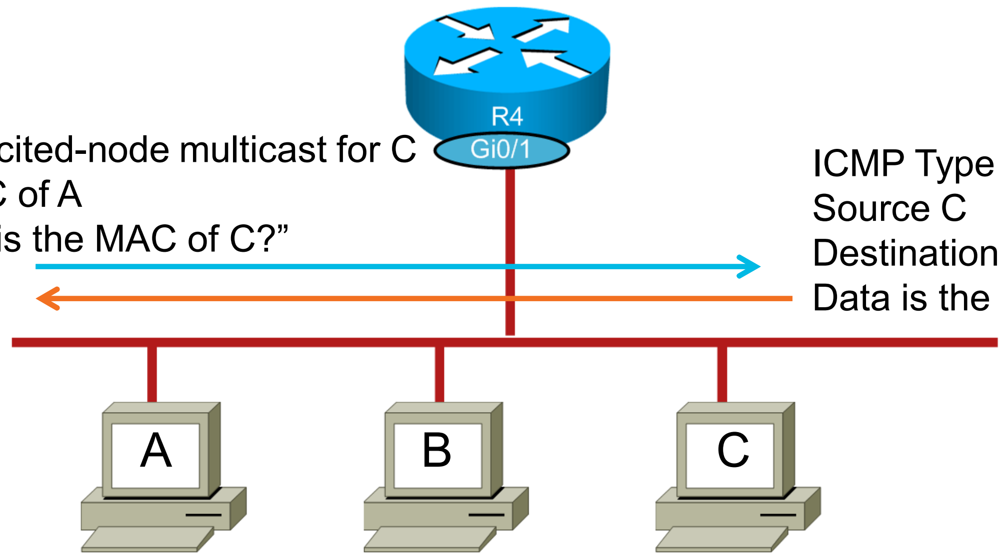
| | | | | |
|---|---|---|---|---|
| FF02 | 0 | 0001 | FF | 00 0DB8 |

- The conversion of layer 3 multicast to layer 2 is to take the last 32 bits of the IPv6 address and prepend 33 33

| | | | | | |
|---|---|---|---|---|---|
| 33 | 33 | FF | 00 | 0D | B8 |

# IPv6 Neighbor Discovery

ICMP Type 135
Source A
Destination Solicited-node multicast for C
Data is the MAC of A
Query is “What is the MAC of C?”




ICMP Type 136
Source C
Destination A
Data is the MAC of C

# IPv6 Neighbor Discovery

- Neighbor Advertisement message:
  – R flag indicates sender is a router
  – S flag is the solicited flag, this is a response to a neighbor solicitation
  – O flag is the override flag and indicates this advertisement should override existing neighbor cache

# Agenda

- IP version 6 (IPv6) addressing and different addressing types
- IPv6 neighbor discovery
- **Basic IPv6 functionality protocols**
- Tunneling techniques
- IPv6 Unicast routing
- Filtering and route redistribution
- IPv6 Multicast
- IPv6 and 3560 switches
- IPv6 NAT protocol translation
- Troubleshooting IPv6
- CCIE R&S
- Q&A

# Basic IPv6 Functionality Protocols

- DHCP
- FHRP
  - HSRP
  - GLBP
- MTU Path Discovery
- SSH
- Telnet
- Ping
- Traceroute
- DNS

# ICMPv6

- ICMPv6 Error messages:
  - Type 1: Destination Unreachable
  - Type 2: Packet too Big
  - Type 3: Time Exceeded
    - Code 0: Hop Limit Exceeded
    - Code 1: Fragment Reassembly Time Exceeded
  - Type 4: Parameter Problem

# First Hop Redundancy Protocols

- First Hop Redundancy Protocols are used to provide gateway redundancy
- IPv4 has: HSRP (v1 and v2), GLBP, VRRP
- IPv6 has: HSRP and GLBP
  - HSRP for IPv6 must run version 2
- HSRP has one active forwarding device per group
- GLBP has up to four active forwarding devices per group

# FHRP – HSRP

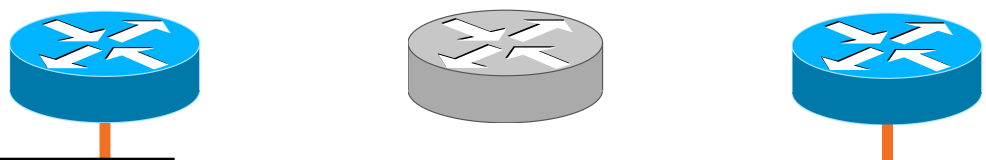
- HSRP version 6 for IPv6
  - This version uses link-local addresses
  - Multicast router announcement messages are transmitted to hosts in the subnet with a virtual link-local address for the default router
- Virtual MAC is derived from the HSRP group
- Virtual link-local address is derived from the virtual HSRP MAC address
- Periodic router announcement messages are sent with the HSRP virtual link-local address of the default router
- HSRP virtual MAC address range is:
  - 0005.73a0.0000 to 0005.73a0.0fff

# FHRP – HSRP

- Priority is a mechanism to control the election of active router within the HSRP group
  - Default is 100; higher is better
  - If priority is equal, then first router up or one with the highest IPv6 address wins
- Preemption can be used to guarantee the active router for a group
- The standby router monitors the status of the group
- HSRP can track objects to control priority (and therefore the active router)
  - Interfaces
  - Routes
  - IP SLA objects

# FHRP – HSRP

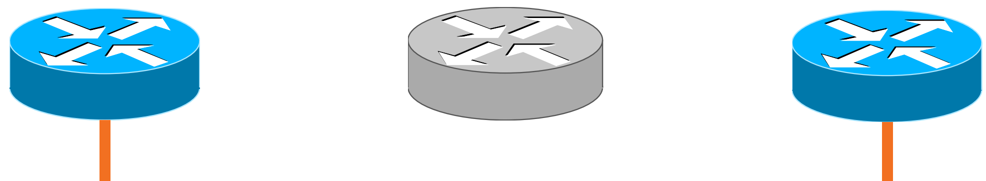
- HSRP group 66 is configured with the following parameters:
  - Automatic virtual IPv6 address
  - Priority
  - Preemption
  - Version



```
interface GigabitEthernet0/0
 ipv6 address 2005:DEAD:BEEF:C100::1/64
 standby version 2
 standby 66 ipv6 autoconfig
 standby 66 priority 110
 standby 66 preempt
```

```
interface GigabitEthernet0/0
 ipv6 address 2005:DEAD:BEEF:C100::2/64
 standby version 2
 standby 66 ipv6 autoconfig
 standby 66 priority 105
 standby 66 preempt
```

# FHRP – HSRP



```
R1# show standby
 GigabitEthernet0/0 - Group 66 (version 2)
   State is active
      2 state changes, last state change 2d23h
   Virtual IP address is FE80::5:73FF:FFA0:42
   Active virtual MAC address is 0005.73a0.0042
     Local virtual MAC address is 0005.73a0.0042 (v2 IPv6 default)
   Hello time 3 sec, hold time 10 sec
     Next hello sent in 1.543 secs
   Preemption enabled
   Active router is local
   Standby router is FE80::200:CFF:FEF6:52CD, priority 105 (expires in 10.464 sec)
   Priority 110 (configured 110)
   Group name is "hsrp-Gi0/1-66" (default)
```

# FHRP – GLBP

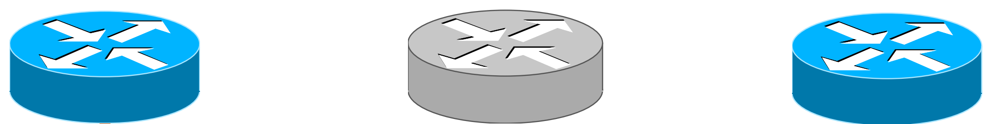
- Gateway Load Balancing Protocol allows for multiple active forwards per group
- You can have up to four active routers within a group (active virtual forwarders (AVF))
- One active virtual gateway (AVG) controls the group
- All routers share a virtual IPv6 address
- Each active forwarder has a unique virtual MAC address
- Same as IPv4

# FHRP – GLBP

- GLBP gateway priority
  - Controls the election of AVG and standby virtual gateway (SVG)
  - Preemption needs to be enable to control which router becomes AVG
    - Preemption is disabled by default
- GLBP gateway weight
  - Weight is used to elect AVF
  - Weight is set with an acceptable range
  - If weight drops below the range, the AVF stops forwarding packets
    - Think tacking
  - Preemption is enable by default

# FHRP – GLBP

- Basic GLBP group 66 is configured with the following parameters:
  - Automatic virtual IPv6 address
  - Priority and Preemption for AVG

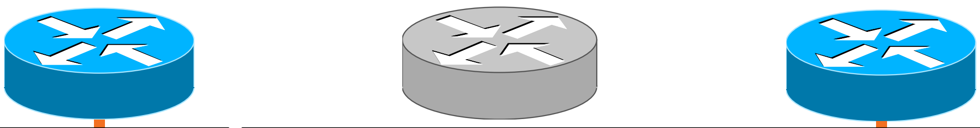


```
interface GigabitEthernet0/0
 ipv6 address 2005:DEAD:BEEF:C100::1/64
 glbp 66 ipv6 autoconfig
 glbp 66 priority 110
 glbp 66 preempt
 glbp 66 weighting 110 lower 95 upper 110
 glbp 66 weighting track 2 decrement 15
!
track 2 interface serial 0/0/0 line-protocol
```

```
interface GigabitEthernet0/0
 ipv6 address 2005:DEAD:BEEF:C100::2/64
 glbp 66 ipv6 autoconfig
 glbp 66 priority 105
 glbp 66 preempt
```

# FHRP – GLBP

- A bit more complex GLBP

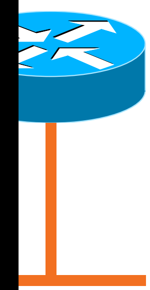


```
interface GigabitEthernet0/0
 ipv6 address 2005:DEAD:BEEF:C100::1/64
 glbp 66 ipv6 autoconfig
 glbp 66 priority 110
 glbp 66 preempt
 glbp 66 weighting 110 lower 95 upper 110
 glbp 66 weighting track 2 decrement 15
!
track 2 interface serial 0/0/0 line-protocol
```

```
interface GigabitEthernet0/0
 ipv6 address 2005:DEAD:BEEF:C100::2/64
 glbp 66 ipv6 autoconfig
 glbp 66 priority 105
 glbp 66 preempt
 glbp 66 weighting 110 lower 95 upper 110
 glbp 66 weighting track 2 decrement 15
!
track 2 interface serial 0/0/0 line-protocol
```

# FH

```
R1# show glbp
GigabitEthernet0/0 - Group 66
  State is Active
    2 state changes, last state change 23:50:33
  Virtual IP address is FE80::7:B4FF:FE04:200 (auto-configured)
  Hello time 5 sec, hold time 18 sec
    Next hello sent in 4.300 secs
  Redirect time 600 sec, forwarder time-out 7200 sec
  Preemption enabled, min delay 60 sec
  Active is local
  Standby is FE80::200:CFF:FEF6:52CD
  Priority 110 (configured)
  Weighting 115 (configured 110), thresholds: lower 95, upper 110
    Track object 2 state Up
  Load balancing: host-dependent
  There are 2 forwardera (1 active)
  Forwarder 1
    State is Active
      1 state change, last state change 23:50:15
    MAC address is 0007.b400.4201 (default)
    Owner ID is 0000.0cf6.52cd
    Redirection enabled
    Preemption enabled, min delay 60 sec
    Active is local, weighting 110
... omitted ...
```



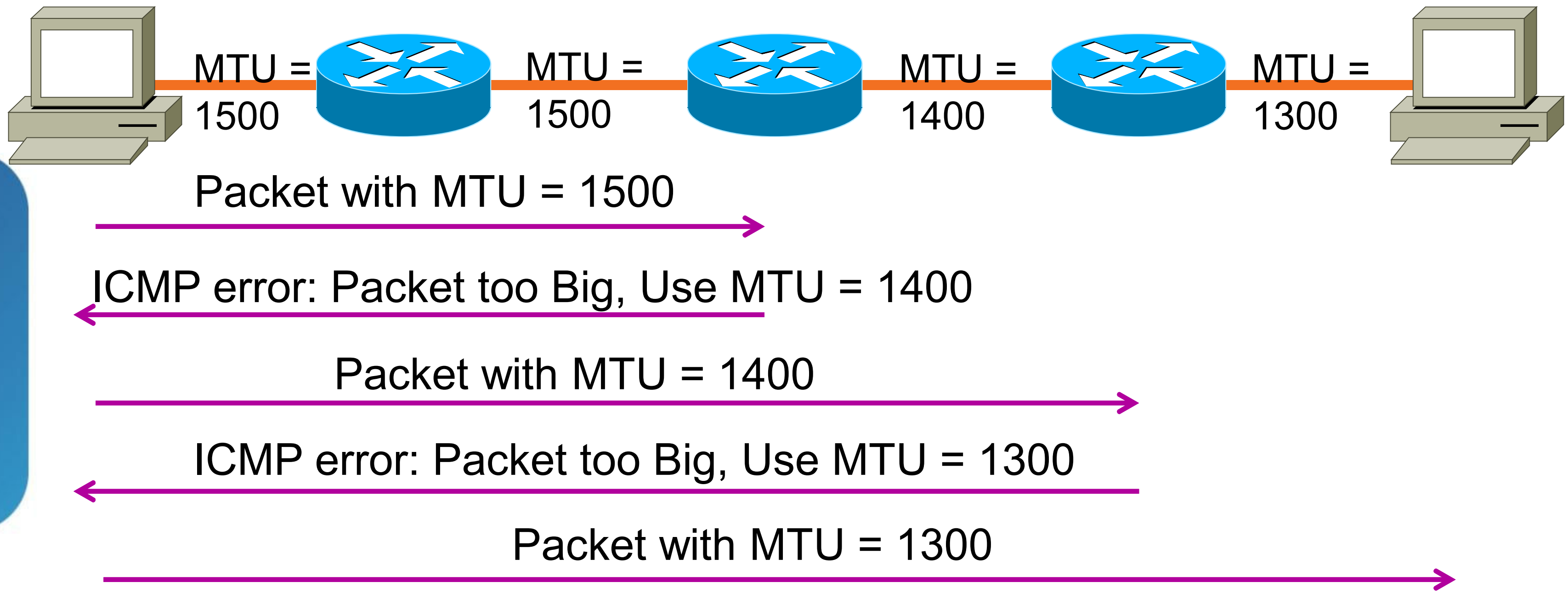
# Basic IPv6 Functionality Protocols

- DHCP
- FHRP
  - HSRP
  - GLBP
- MTU Path Discovery
- SSH
- Telnet
- Ping
- Traceroute
- DNS

# Path MTU Discovery

- Fragmentation is not done by routers (unless they originate the packet), it's done by end devices
- Minimum assumed MTU for IPv6 is 1280 bytes (IPv4 is 68, 576 minimum that can be processed at host)
- Routers send Packet too big message to the sender if the packet is too large
  – ICMPv6 Type 2

# Path MTU Discovery



# Basic IPv6 Functionality Protocols

- DHCP
- FHRP
  - HSRP
  - GLBP
- MTU Path Discovery
- SSH
- Telnet
- Ping
- Traceroute
- DNS

# Basic IPv6 Functionality Protocols

- SSH
  - ssh –l cisco:2005:dead:beef:c100::1
- Telnet
  - telnet 2005:dead:beef:c100::1
- Ping
  - Ping 2005:dead:beef:c100::1
- Traceroute
  - Traceroute 2005:dead:beef:c100::1

ipv6 host Router1 2005:dead:beef:c100::1

# Basic IPv6 Functionality Protocols

- DHCP
- FHRP
  - HSRP
  - GLBP
- MTU Path Discovery
- SSH
- Telnet
- Ping
- Traceroute
- DNS

# Domain Name System

- DNS had to be modified to support the larger address space
  - AAAA for 128 bit address
  - PTR records are for reverse lookups, which is also for IPv4, but uses a new nibble format
  - Current recommendation is to not to automatically generate the PTR database as done with IPv4 to save on memory

# Basic IPv6 functionality protocols

- DHCP
- FHRP
  - HSRP
  - GLBP
- MTU Path Discovery
- SSH
- Telnet
- Ping
- Traceroute
- DNS

# Agenda

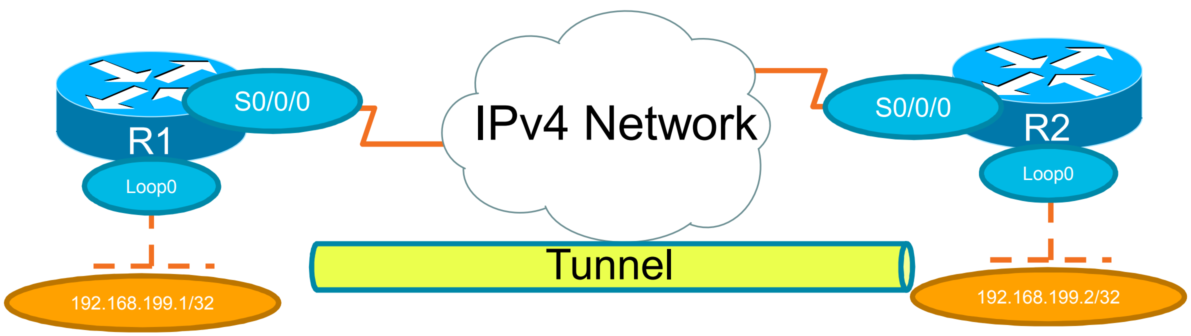
- IP version 6 (IPv6) addressing and different addressing types
- IPv6 neighbor discovery
- Basic IPv6 functionality protocols
- **Tunneling techniques**
- IPv6 Unicast routing
- Filtering and route redistribution
- IPv6 Multicast
- IPv6 and 3560 switches
- IPv6 NAT protocol translation
- Troubleshooting IPv6
- CCIE R&S
- Q&A

# Tunneling Techniques

- IPv4 Compatible tunnel
- 6to4 tunnel
- ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) tunnel
- Teredo
- IPv6 Rapid Deployment (6RD)

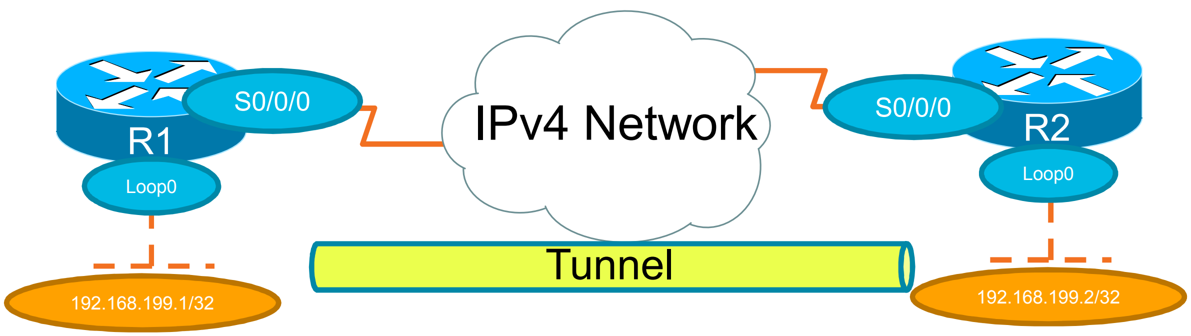
# Tunneling Techniques – IPv4 Compatible Tunnel

- To connecting IPv6 hosts or sites over the existing IPv4 infrastructure
- The addressing for tunnel destination is derived from the lower order 32 bits of the IPv4-compatable address
  – Assuming IPv4 reachability



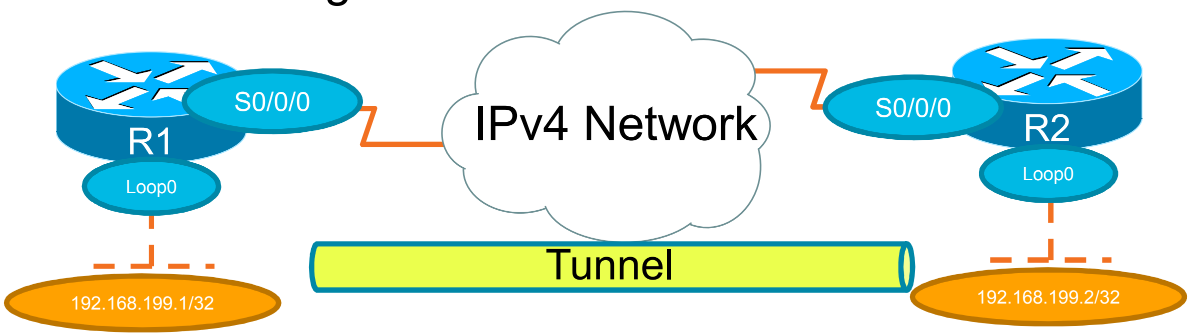
# Tunneling Techniques – IPv4 Compatible Tunnel

```
interface loopback 0
 ip address 192.168.199.1 255.255.255.255
!
interface tunnel 0
 tunnel source loopback 0
 tunnel mode ipv6ip auto-tunnel
!
ipv6 route ::/0 ::192.168.199.2
```



# Tunneling Techniques – 6to4 Tunnel

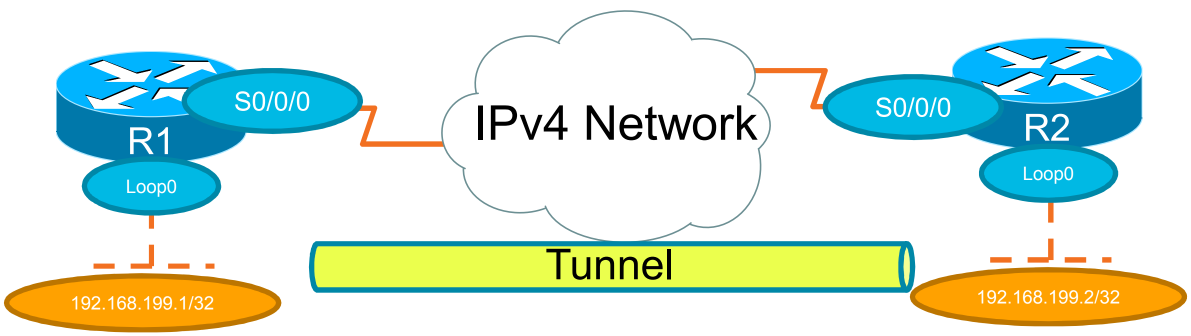
- 6to4 tunnels embeds the IPv4 address as part of the IPv6 address for the tunnel interface
- 6to4 tunnel address must start with 2002 for it's prefix
  - The next 32 bits of the IPv6 address is the IPv4 address in hex
- 6to4 tunnels are considered point to multipoint dynamic tunnels
- There is no defined destination within the tunnel, it's defined with neighbor statements or through static routes.



# Tunneling Techniques – 6to4 Tunnel

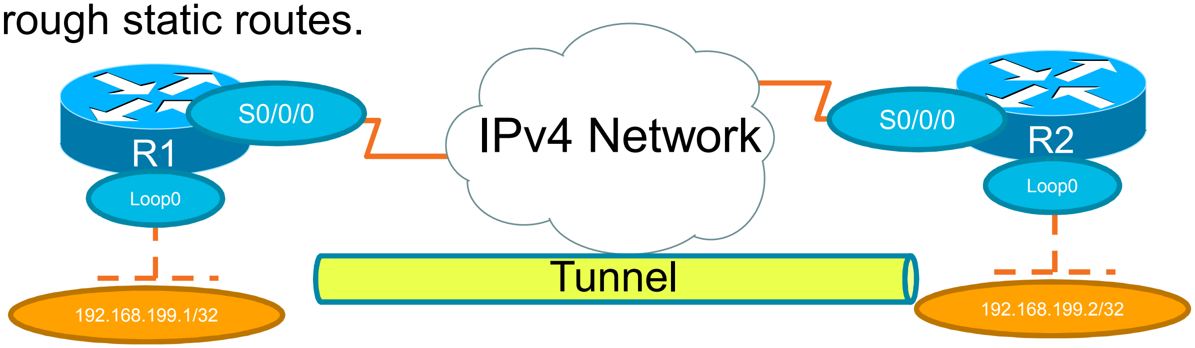
```
interface loopback 0
 ip address 192.168.199.1 255.255.255.255
!
interface tunnel 0
 ipv6 address 2002:C0A8:C701::/48
 tunnel source loopback 0
 tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 tunnel 0
ipv6 route ::/0 2002:C0A8:C702::
```

192 = 0xC0
168 = 0xA8
199 = 0xC7
1 = 0x01



# Tunneling Techniques – ISATAP Tunnel

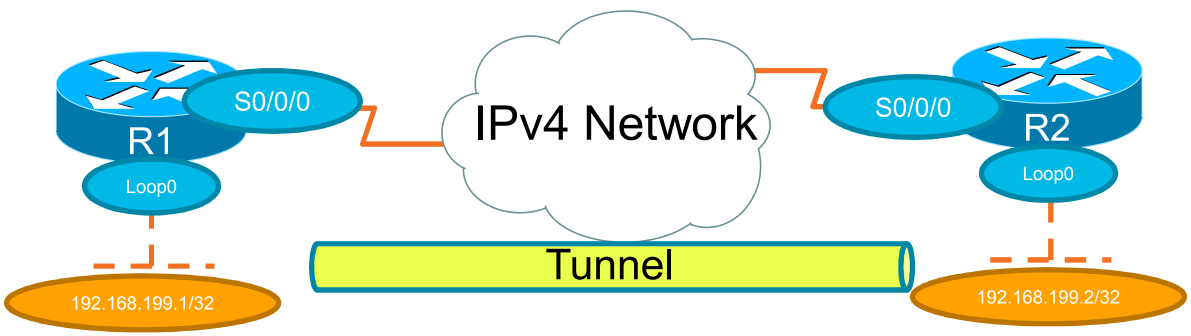
- Like 6to4 tunnels, ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) embeds the IPv4 address as part of the IPv6 address for the tunnel interface
- The prefix can be any assigned /64 prefix
- The last 64 bits are made up of 0000:5EFE and then the IPv4 address in hex
  - If you use EUI-64 with your IPv6 address, the router will configure your last 64 bits for you
- ISATAP tunnels are considered point to multipoint dynamic tunnels
- There is no defined destination within the tunnel, it's defined with neighbor statements or through static routes.



# Tunneling Techniques – ISATAP Tunnel

```
interface loopback 0
 ip address 192.168.199.1 255.255.255.255
!
interface tunnel 0
 ipv6 address 2005:DEAD:BEEF:0:5EFE:C0A8:C701/64
 ipv6 address FE80::5EFE:C0A8:C701 link-local
 tunnel source loopback 0
 tunnel mode ipv6ip ISATAP
!
ipv6 route ::/0 2005:DEAD:BEEF:0:5EFE:C0A8:C702
```

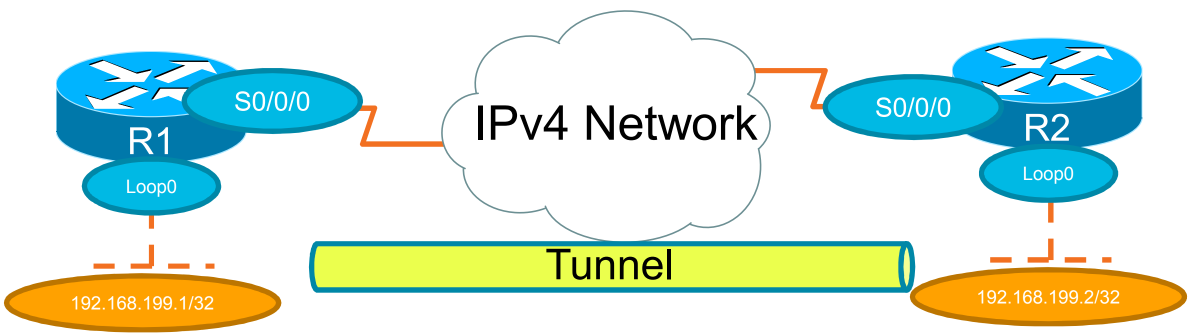
192 = 0xC0
168 = 0xA8
199 = 0xC7
1 = 0x01



# Tunneling Techniques – ISATAP Tunnel

```
interface loopback 0
 ip address 192.168.199.1 255.255.255.255
!
interface tunnel 0
 ipv6 address 2005:DEAD:BEEF::/64 EUI-64
 tunnel source loopback 0
 tunnel mode ipv6ip ISATAP
!
ipv6 route ::/0 2005:DEAD:BEEF:0:5EFE:C0A8:C702
```

| | |
|---|---|
| 192 | = 0xC0 |
| 168 | = 0xA8 |
| 199 | = 0xC7 |
| 1 | = 0x01 |



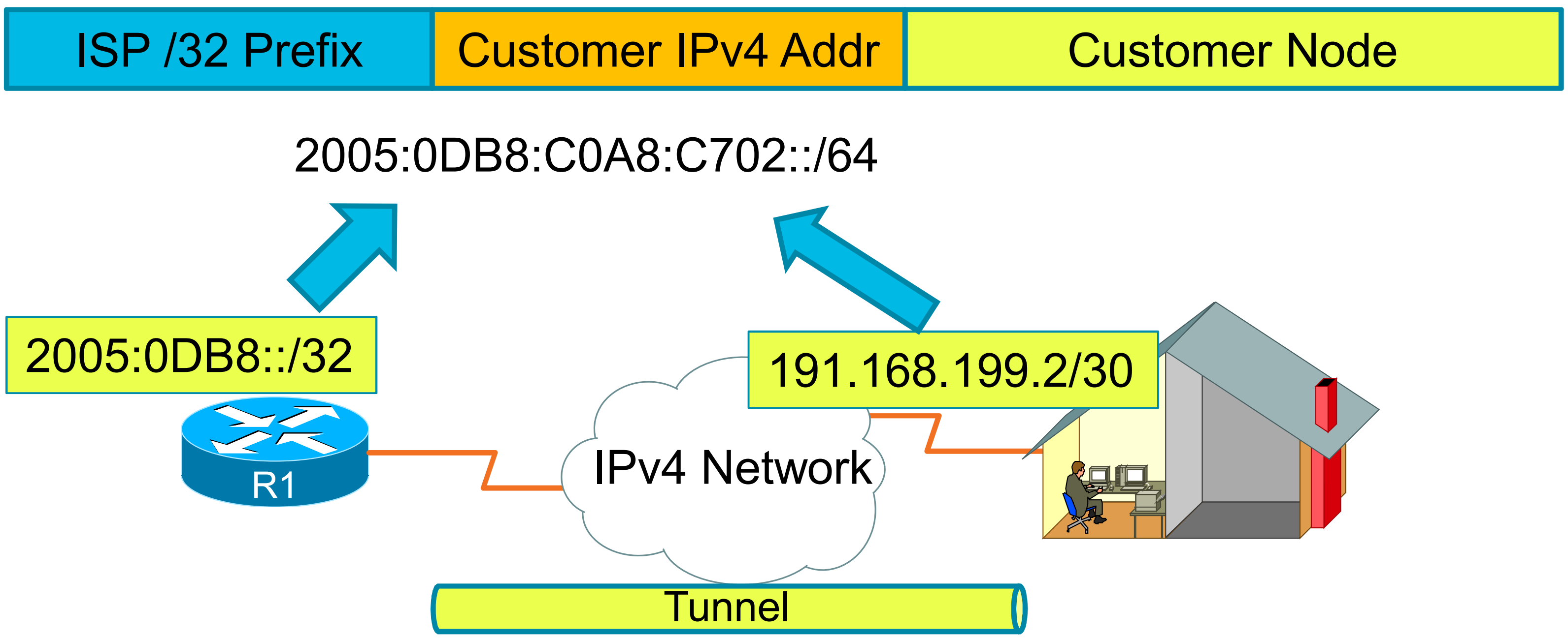
# Tunneling Techniques – Teredo

- **Teredo** is a transition technology that gives IPv6 connectivity across a IPv4
- Teredo operates using a platform independent tunneling protocol encapsulating IPv6 packets within IPv4 User Datagram Protocol (UDP) packets.
- These packets can be routed through an IPv4 network and through NAT devices.

# Tunneling Techniques – 6RD

- IPv6 Rapid Deployment (6RD)
  - Designed to speed up the process to extend IPv6 across the first and last mile
  - Modified 6to4 tunnel
  - Have to have the core setup to support IPv6
  - Intended for ISP to support SOHO customers
  - ISP must have /32 address space to feed from
  - Client has to be configured with the 6RD endpoint (IPv4 address)
  - Client then does a modified SLAAC
    - Gets the /32 from the ISP
    - Next 32 bits is the client's IPv4 address in hex
    - Fills in the last 64 bit normally

# Tunneling Techniques – 6RD



# Agenda

- IP version 6 (IPv6) addressing and different addressing types
- IPv6 neighbor discovery
- Basic IPv6 functionality protocols
- Tunneling techniques
- **IPv6 Unicast routing**
- Filtering and route redistribution
- IPv6 Multicast
- IPv6 and 3560 switches
- IPv6 NAT protocol translation
- Troubleshooting IPv6
- CCIE R&S
- Q&A

# IPv6 Unicast Routing

- Static
- RIPng
- OSPF version 3 (OSPFv3)
- EIGRP version 6 (EIGRPv6)
- BGP
- PBR

What about IS-IS? Yes, but it's not on the blueprint, therefore it does not exist!

# IPv6 Unicast Routing – Static

- Directly attached Static Routes

```
ipv6 route 2005:DEAD:BEEF:A100::/64 GigabitEthernet0/1
```

- Recursive Static Routes

```
ipv6 route 2005:DEAD:BEEF:A100::/64 2005:DEAD:BEEF:C100::1
```

- Fully Specific Static Routes

```
ipv6 route 2005:DEAD:BEEF:A100::/64 GigabitEthernet0/1 2005:DEAD:BEEF:C100::1
```

- Floating Static Routes

```
ipv6 route 2005:DEAD:BEEF:A100::/64 2005:DEAD:BEEF:C100::1 130
```

# IPv6 Unicast Routing – RIPng

- RIPng (next generation) for IPv6
  - Detailed in RFC 2080
  - Based on RIPv2, but not interchangeable
  - 15 hop limit
  - Split horizon
  - Same timers, minus the hold down (180 sec flush rather than 240 sec)
  - Runs UDP port 521 on top of IPv6
  - Multicast updates on FF02::9
  - No network command, interface level
  - Named, can have up to four instances per router
  - UP to 64 ECMP (default is 4)

# IPv6 Unicast Routing – RIPng

- RIPv6 has to be enabled per interface

```
R1(config-if)# ipv6 rip name enable
```

- The name has local significance only. It is a process identifier
- Global options for the RIP process like redistribution, timers, etc. can be configured under the RIP routing process

```
R1(config)# ipv6 router rip name
R1(config-rtr)#
```

- Default is generated at the interface

```
R1(config-if)# ipv6 rip name default-information originate
```

# IPv6 Unicast Routing – RIPng

```
ipv6 unicast-routing
!
interface Loopback0
 ipv6 address 2005:DEAD:BEEF:2::2/64
 ipv6 rip MyRip enable
!
interface GigabitEthernet0/1
 ipv6 address 2005:DEAD:BEEF:C001::1/64
 ipv6 rip MyRip enable
!
ipv6 router rip MyRip
```

# IPv6 Unicast Routing – RIPng

- What if they ask you to change the destination address of UDP port number for RIPng

```
ipv6 router rip MyRip
 port 3521 multicast-group ff02::521
```

# IPv6 Unicast Routing – OSPFv3

- OSPF for IPv6
- Based on OSPFv2, with enhancements
  - Multi-area with Area 0 as backbone
  - Stub and NSSA
  - Summarization on the area border routers
  - Basic packet types (Hello, DBD, LSR, LSU, LSA)
  - Mechanisms for neighbor discovery and adjacency formation
  - Interface types (P2P, P2MP, Broadcast, NBMA, Virtual links)
  - LSA flooding and aging
  - Nearly identical LSA types
  - Router ID is still 32 bit number, written in four octets

# IPv6 Unicast Routing – OSPFv3

- Distributes IPv6 prefixes using Type Length Value (TLV) fields
- Runs directly over IPv6
- Ships-in-the-night with OSPFv2
- Adjacencies and next-hop use link-local addresses
  – except for virtual links
- Enabled at the interface, no network (and no wildcard masks)
- LSA has flooding scope
  – Link-local
  – Area
  – Autonomous system
- Uses link-local multicast
  – FF02::5 – All OSPF Routers
  – FF02::6 – All OSPF designated routers

# IPv6 Unicast Routing – OSPFv3

- Two LSA’s have been renamed:
  – Type 3 LSA is now Interarea Prefix LSA
  – Type 4 LSA is now Interarea Router LSA
- Two new LSAs have been added
  – Type 8 LSA is now Link LSA
  – Type 9 LSA is now Intra-Area Prefix LSA

# IPv6 Unicast Routing – OSPFv3

- OSPFv3 has to be enabled per interface

```
R1(config-if)# ipv6 ospf <process-id> area <area>
```

- Global options for the OSPFv3 process can be configured under the OSPFv3 routing process

```
R1(config)# ipv6 router ospf <process-id>
R1(config-rtr)# router-id ?
  A.B.C.D  OSPF router-id in IP address format
```

- When no IPv4 addresses are available you must configure a router-id manually

# Cisco IOS OSPFv3 Specific Attributes

- Configuring area range

```
R1(config-router)#[no] area <area ID> range <prefix>/<prefix length>
```

- Auto summary at the ABR

```
R1(config-router)#[no] area {ipv6-prefix/prefix length} [cost cost]
```

- Configuring summary addresses for external routes

```
R1(config-router)#[no] summary-address <prefix>/<prefix length>
```

- Static neighbor configuration

```
R1(config-if)#[no] ipv6 ospf neighbor ipv6-link-local-address [priority]
[poll-interval seconds] [cost cost] [database-filter all out]
```

# IPv6 Unicast Routing – OSPFv3 Authentication

- Authentication and encryption are used to secure routing updates and prevent attacks
- OSPFv3 uses IPv6's built in security
  - IPsec AH for authentication
  - IPsec ESP for encryption of payload
- Security policy definition on the router is required
  - Key
  - Security parameter index (SPI) value

# IPv6 Unicast Routing – OSPFv3 Authentication

- Configuring authentication between routers

```
R1(config-if)#ipv6 ospf authentication ipsec spi spi md5 key
```

- Authentication at the area level

```
R1(config-router)#area area-id authentication ipsec spi spi md5 key
```

- To configure encryption on an interface between routers

```
R1(config-if)#ipv6 ospf encryption {ipsec spi spi esp encryptin-
algorithm [[key-encryption-type] key] authentication-algorithm
[[key-encryption-type] key]}
```

- Encryption at the area level

```
R1(config-router)#area area-id encryption {ipsec spi spi esp
encryptin-algorithm [[key-encryption-type] key] authentication-
algorithm [[key-encryption-type] key]}
```

# IPv6 Unicast Routing – OSPFv3 Authentication

```
ipv6 unicast-routing
!
interface Loopback0
 ipv6 address 2005:DEAD:BEEF:2::2/64
 ipv6 ospf 1 area 0
!
interface GigabitEthernet0/1
 ipv6 address 2005:DEAD:BEEF:C001::1/64
 ipv6 ospf 1 area 0
!
ipv6 router ospf 1
 router-id 1.1.1.1
```

# IPv6 Unicast Routing – EIGRPv6

- Advanced distance vector protocol
- EIGRP is an integrated routing protocol
  - IPv4
  - IPv6
  - IPX
  - Appletalk
- Supports IPv6 as a separate routing context
- Dynamic or static neighbor discovery
- Built in reliability
- Incremental updates
- Protocol-dependent modules

# IPv6 Unicast Routing – EIGRPv6

- Defusing Update Algorithm (DUAL)
- Three sets of protocol specific tables:
  - Neighbor
  - Topology
  - Routing
- Composite metric:
  - Bandwidth + delay (by default)
    - Slowest link in the path is used (BW = 2.56Tb/slowest link BW)
    - Cumulative delay (delay = 256 * all interface delays in path in 10s microseconds)
- No network commands, interface level configuration

# IPv6 Unicast Routing – EIGRPv6

- Create the EIGRPv6 routing process

```
R1(config)# ipv6 router eigrp as-number
```

- You have to enable the process

```
R1(config-rtr)# no shutdown
```

- To advertise a default route

```
R1(config-rtr)# default-information originate [route-map route-map]
```

- To change the maximum paths for the routing table

```
R1(config-rtr)# maximum-paths number
```

# IPv6 Unicast Routing – EIGRPv6

- Enable EIGRPv6 at the interface

```
R1(config-if)# ipv6 eigrp as-number
```

- To configure a summary

```
R1(config-if)# ipv6 summary-address eigrp as-number prefix/mask [AD]
```

- To disable split horizon

```
R1(config-if)# no ipv6 split-horizon eigrp as-number
```

- To change the bandwidth percentage for EIGRP

```
R1(config-rif)# ipv6 bandwidth-percent eigrp as-number percent
```

# IPv6 Unicast Routing – EIGRPv6

- Multi-Protocol BGP for IPv6 (et al)
- Defined in RFC 2545
- Uses new Address Family in BGP
  - NEXT_HOP and NLRI are expressed as IPv6 addresses and prefix.
  - Address Family Information (AFI) = 2 (IPv6)
- Underlying Protocol can be either IPv6 or IPv4
  - Uses TCP Port 179
- Attributes and Route Selection are similar to BGP for IPv4

# IPv6 Unicast Routing – BGP

- Configure peerings using IPv6 (or IPv4)

```
R1(config)# router bgp 1
R1(config-router)# neighbor 2005:DEAD:BEEF:C001::2 remote-as 2
```

- And activate them for IPv6

```
R1(config-router)# address-family ipv6
R1(config-router-af)# 2005:DEAD:BEEF:C001::2 activate
```

- Then configure the prefixes you want to advertise under the IPv6 address-family

```
R1(config-router)# address-family ipv6
R1(config-router-af)# network 2005:DEAD:BEEF:A100::/64
```

# IPv6 Unicast Routing – BGP

```
router bgp 1
 no synchronization
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 neighbor remote-as 1
 neighbor 2005:DEAD:BEEF:2::2 update-source Loopback0
 no neighbor 2005:DEAD:BEEF:2::2 activate
 neighbor 2005:DEAD:BEEF:C001::2 remote-as 2
 no neighbor 2005:DEAD:BEEF:C001::2 activate
 no auto-summary
 !
 address-family ipv6
 neighbor 2005:DEAD:BEEF:2::2 activate
 neighbor 2005:DEAD:BEEF:C001::2 activate
 network 2005:DEAD:BEEF:C001::/64
 exit-address-family
```

# IPv6 Unicast Routing – BGP

```
interface FastEthernet0/0
 ip address 180.40.7.66 255.255.255.224
!
interface Tunnel0
 no ip address
 no ip redirects
 ipv6 address 2002:B428:742::/64
 tunnel source FastEthernet0/0
 tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 Tunnel0

router bgp 100
 neighbor 2002:b428:701:: remote-as 200
 neighbor 2002:b428:701 :: ebgp-multihop 2
```

# IPv6 Unicast Routing – Policy Based Routing

- PBRv6 has the same basic functionality as PBR for IPv4.
- Create a route map
- Match conditions
  - Access-list
  - Prefix list
  - Ingress interface
  - Size
- Set conditions
  - Next hop address
  - Egress interface
  - IPv6 precedence

# IPv6 Unicast Routing – Policy Based Routing

```
interface GigabitEthernet1/1
ipv6 policy route-map V6PBR
!
route-map V6PBR permit 10
match ipv6 address ThroughR1
set ipv6 next-hop FE80::20F:90FF:FEFB:12A0
!
route-map V6PBR permit 20
match ipv6 address ThroughR3
set ipv6 next-hop FE80::21C:F6FF:FE85:260
!
route-map V6PBR permit 30
set ipv6 next-hop FE80::21C:F6FF:FE85:260 FE80::20F:90FF:FEFB:12A0
!
ipv6 access-list ThroughR1
permit ipv6 2001:213:112:31::/64 any
!
ipv6 access-list ThroughR3
permit ipv6 2001:212:104:11::/64 any
```

# Agenda

- IP version 6 (IPv6) addressing and different addressing types
- IPv6 neighbor discovery
- Basic IPv6 functionality protocols
- Tunneling techniques
- IPv6 Unicast routing
- **Filtering and route redistribution**
- IPv6 Multicast
- IPv6 and 3560 switches
- IPv6 NAT protocol translation
- Troubleshooting IPv6
- CCIE R&S
- Q&A

# Filtering and Route Redistribution

- IPv6 access lists
- IPv6 prefix lists
- Route maps
- Redistribution issues

# IPv6 Access Lists

- All IPv6 ACLs are:
  - Named
  - Extended
- Very similar to IPv4 named ACL
- Implicit deny all at the end of the ACL is not really a deny all
  - ICMPv6 for neighbor discovery is still enabled, even if not explicitly stated
- No wildcard masks
- IPv6 access lists are sequenced:
  - Individual entries can be added and removed
  - ACLs cannot be resequenced

# IPv6 Access Lists

- IPv6 ACLs can also filter on some of the additional header fields or some of the extended headers:
  - DSCP value
  - Flow label value
  - Fragmentation header (if it is present)
  - Routing header (if it is present and type)
  - Mobility header (if it is present and type)
  - Destination option header (if it is present and type)
  - Authentication header (if it is present)

# IPv6 Access Lists

- To get into the access list configuration
  - **ipv6access-list** *access-list-name*
- Then the match conditions:
  - **permit** *protocol* {*source-ipv6-prefixlprefix-length* | **any** | **host** *source-ipv6-address* | **auth**} [*operator* [*port-number*] ] {*destination-ipv6-prefixlprefix-length* | **any** | **host** *destination-ipv6-address* | **auth**} [*operator* [*port-number*] ] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**hbh**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**reflect** *name* [**timeout** *value*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]

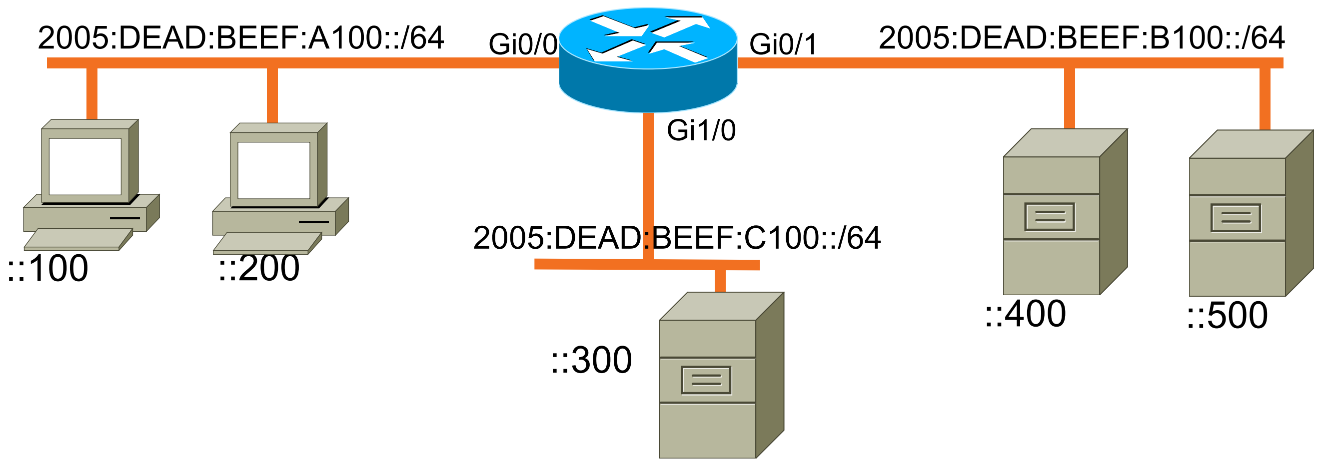
# IPv6 Access Lists

- For ICMP
  - **permit icmp** {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address* | **auth**} [*operator* [*port-number*] ] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address* | **auth**} [*operator* [*port-number*] ] [*icmp-type* [*icmp-code*] | *icmp-message*] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**hbh**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
- TCP
  - **permit tcp** {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address* | **auth**} [*operator* [*port-number*] ] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address* | **auth**} [*operator* [*port-number*] ] [**ack**] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**established**] [**fin**] [**flow-label** *value*] [**fragments**] [**hbh**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**neq** {*port* | *protocol*}] [**psh**] [**range** {*port* | *protocol*}] [**reflect** *name* [**timeout** *value*]] [**routing**] [**routing-type** *routing-number*] [**rst**] [**sequence** *value*] [**syn**] [**time-range** *name*] [**urg**]

# IPv6 Access Lists

- And UDP
  - **permit udp** {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address* | **auth**} [*operator* [*port-number*] ] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address* | **auth**} [*operator* [*port-number*] ] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**hbh**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**neq** {*port* | *protocol*}] [**range** {*port* | *protocol*}] [**reflect** *name* [**timeout** *value*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]

# IPv6 Access Lists

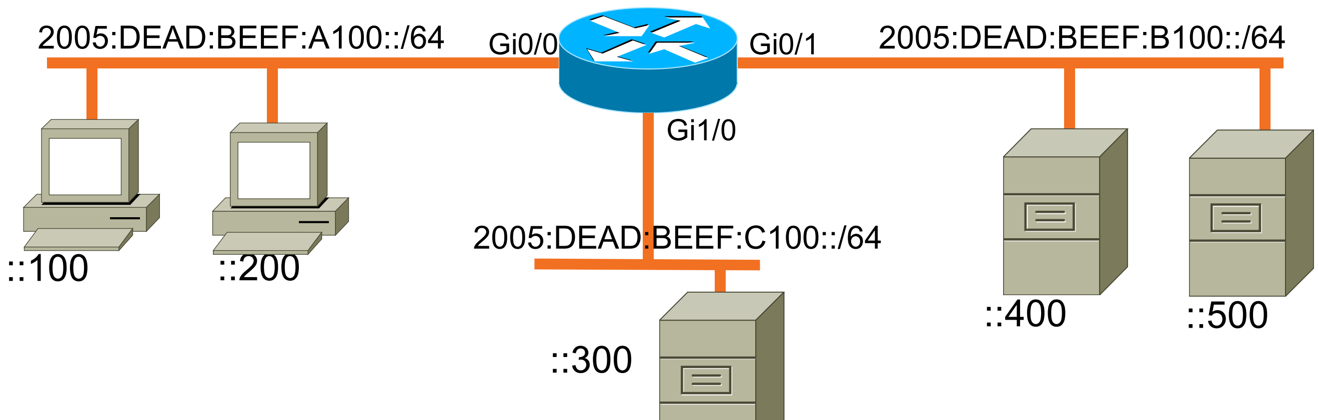



```
Ipv6 access-list MyWebList
 permit tcp 2005:dead:beef:a100::/64 host 2005:dead:beef:b100::400 eq 80
!
Interface GiabitEthernet0/0
 ipv6 traffic-filter MyWebList
```

# IPv6 Access Lists

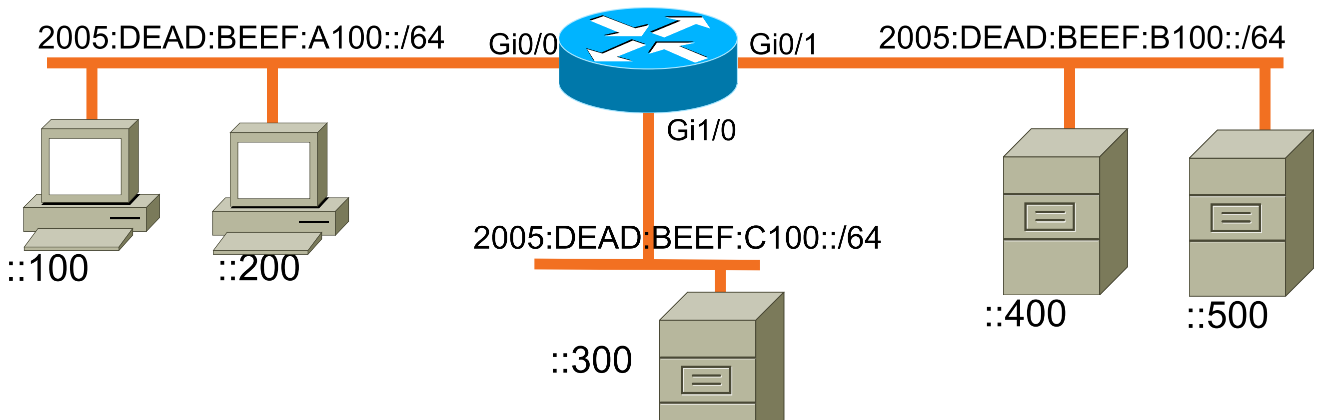
- Reflexive and Time based ACLs are the same as IPv4
- Reflexive ACLs provide access by, based on session initiation
  - Router tracks outbound traffic and automatically creates temporary reverse path rule
- Time based ACLs permit or deny traffic that are linked to a configured time range

# IPv6 Access Lists




```
Ipv6 access-list MyReflectOut
 permit tcp 2005:dead:beef:a100::/64 any eq 80 reflect ref-web
 permit udp 2005:dead:beef:a100::/64 any reflect ref-udp
!
Ipv6 access-list MyReflectIn
 evaluate ref-web
 evaluate ref-udp
!
Interface GiabitEthernet0/1
 ipv6 traffic-filter MyReflectOut out
 ipv6 traffic-filter MyReflectIn in
```

# IPv6 Access Lists




```
Ipv6 access-list MyTimeOut
 permit tcp 2005:dead:beef:a100::/64 any eq 80 time-range Lunch
 deny tcp 2005:dead:beef:a100::/64 any eq 80
 permit ipv6 any any
!
Interface GiabitEthernet0/1
 ipv6 traffic-filter MyTimeOut out
```

# IPv6 Prefix Lists

- IPv6 prefix list operate the same way as prefix list for IPv4 do
- Used for route filtering
- Distribute list for IPv6 can only call prefix list, no ACLs!

- **ipv6 prefix-list** *list-name* [**seq** *seq-number*] {**deny** *ipv6-prefix*/*prefix-length* | **permit** *ipv6-prefix*/*prefix-length* | **description** *text*} [**ge** *ge-value*] [**le** *le-value*]

# Configuring Prefix-Lists

- What will be matched by:
  A. ipv6 prefix-list A permit ::/0 ge 128
  B. ipv6 prefix-list B permit FEC0::/10 ge 11
  C. ipv6 prefix-list C permit ::/0 le 128
  D. ipv6 prefix-list D permit ::/0
  E. ipv6 prefix-list E permit xxxx:xxxx:xxxx:xxxx::/64

# Configuring Prefix-Lists

- What will be matched by:
  A. ipv6 prefix-list A permit ::/0 ge 128
  B. ipv6 prefix-list B permit FEC0::/10 ge 11
  C. ipv6 prefix-list C permit ::/0 le 128
  D. ipv6 prefix-list D permit ::/0
  E. ipv6 prefix-list E permit xxxx:xxxx:xxxx:xxxx::/64

**A. All host routes**
**B. Any site local address space**
**C. All routes**
**D. Just the default route**
**E. A specific prefix with a length of 64 bits**

# Route Maps

- Route maps can be your friend
  - They can be used for Policy based routing
  - Manipulation of attributes
  - Filtering

# Route-maps

- To act as a filter:
  - Match a deny statement

    Route-map MyMap deny 10

     match ipv6 prefix-list MyList

  - Match a permit statement with null

    Route-map MyMap permit 10

     match ipv6 prefix-list MyList

     set interface null0

  - No Match statement*

    Route-map MyMap permit 10

     match ipv6 prefix-list MyList

    Ipv6 prefix-list MyList permit 2005:dead:beef::/48 ge 64 le 64

* Does not work as a filter with PBR

# Redistribution

- Used when you have more than one protocol or static route or connected routes that you would like to inject into a protocol
- Remember, except for BGP, there are no network commands
- Redistribution will not include directly connected interfaces by default
- When redistributing into OSPFv3, metric type can be set
  – Type 2 is default

# Redistribution

- Redistribute connected
  - Only locally connected routes
  - Route maps can be used to control and manipulate attributes

```
R1(config-rtr)# redistribute connected [route-map route-map]
```

- When redistributing static, you can use a route-map to control and some protocols will let you set a tag without the route-map

```
R1(config-rtr)# redistribute static [route-map route-map] [tag tag]
```

- With RIP and EIGRP, you can redistribute with or without the connected interfaces

```
R1(config-rtr)# redistribute rip [include-connected][route-map route-map]
R1(config-rtr)# redistribute eigrp as-number [include-connected][route-map
route-map]
```

# Redistribution

- To redistribute OSPFv3 routes, you can match on the route type
  - Interface, external, nssa-external
  - Type 1 or type 2
- Route maps again can be used for greater control

```
R1(config-rtr)# redistribute ospf process-id [match {internal | external
[1|2] | nssa-external [1|2]}] [route-map route-map]
```

# Agenda

- IP version 6 (IPv6) addressing and different addressing types
- IPv6 neighbor discovery
- Basic IPv6 functionality protocols
- Tunneling techniques
- IPv6 Unicast routing
- Filtering and route redistribution
- **IPv6 Multicast**
- IPv6 and 3560 switches
- IPv6 NAT protocol translation
- Troubleshooting IPv6
- CCIE R&S
- Q&A

# IPv6 Multicast

- Multicast - FF00::/8

| 8 bits | 4 bits | 4 bits | 112 bits |
|---|---|---|---|
| 11111111 | Flag | Scope | Group ID |

Flag bits:

Only the least significant bit is currently used for multicast traffic: 0 is a permanent or well known multicast, 1 is not permanent or transient multicast

Scope:

1 – Node Local

2 – Link Local

5 – Site Local

8 – Organization Local

E (14) – Global

# IPv6 Multicast

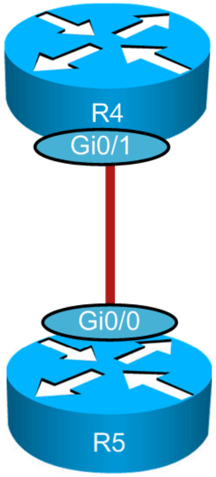
| Service | IPv4 | IPv6 |
|---|---|---|
| Address Range | 32 bit, Class D, 28 bits for groups | 128 bit, 112 bits for groups |
| Routing | Protocol independent multicast (PIM) and MBGP | PIM and MBGP |
| Forwarding | PIM: Dense mode (DM), Sparse mode (SM), bidirectional and source specific multicast (SSM) | PIM: SM, SSM, and bidirectional |
| Group Management | IGMPv1, v2, v3 | MLDv1, v2 |
| Domain control | Boundary border | Scope identifier |
| Interdomain | MSDP with RP in independent PIM domains | Single RP within globally shared domains |

# IPv6 Multicast

- Rendezvous Points
  - Static RP for PIM-SM and Bidir-PIM, no redundancy yet
  - BSR for PIM-SM and Bidir-PIM, with RP redundancy
  - Embedded-RP for PIM-SM only, no redundancy yet

# IPv6 Multicast – Static RP

```
ipv6 multicast-routing
!
interface loopback 0
 no ip address
 ipv6 address 2005:DEAD:BEEF:FFFF::40/128
!
ipv6 pim rp-address 2005:DEAD:BEEF:FFFF::40
```




```
ipv6 multicast-routing
!
ipv6 pim rp-address 2005:DEAD:BEEF:FFFF::40
```

# IPv6 Multicast – Embedded RP

- Embeds the RP address into a multicast group address
- Redefines what was an 8 bit reserved field into a 4 bit reserved and 4 bit R field:
  - – R field allows provision of 16 RPs on embedded address
  - – 32 bit group ID field provides for 232 multicast groups per RP

| 8 | 4 | 4 | 4 | 4 | 8 | 64 | 32 |
|---|---|---|---|---|---|---|---|
| FF | Flags | Scope | Rsvd | RPadr | Plen | Network-Prefix | Group-ID |

  - – Flags – 0RPT (0111 for Embedded RP Address)

Plen = Prefix Length

# IPv6 Multicast – Embedded RP

- Address example:
  - 16 RP addresses per network prefix
  - 232 multicast groups per RP

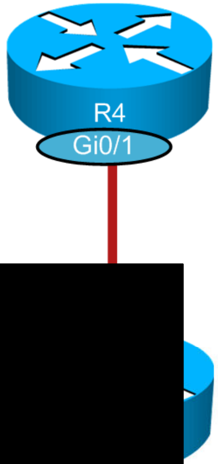
| 8 | Flags | Scope | Rsvd | Rpadr | Plen | Network-Prefix | Group-ID |
|---|---|---|---|---|---|---|---|
| FF | 7 | E | 0 | 1 | 64 | 2005:DEAD:BEEF:00C1 | 1001:4321 |

Embedded multicast: FF7E:0164:2005:DEAD:BEEF:00C1:1001:4321

RP Address:  2005:DEAD:BEEF:00C1::1

# IPv6 Multicast – Embedded RP

- RP to be used as an embedded RP needs to be configured with address/group range
- All other non-RP routers require no special configuration




```
ipv6 multicast-routing
!
ipv6 pim rp-address 2005:DEAD:BEEF:00C1::1 ERP
!
ipv6 access-list ERP
 permit ipv6 any FF7E:0164:2005:DEAD:BEEF:00C1:1001::/96
```

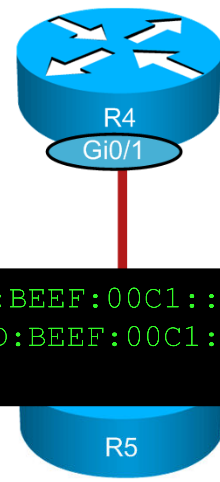
# IPv6 Multicast – Boot Strap Router

- Boot Strap Router (BSR) allows for the dynamic distribution of the RP address to other routers.
- BSR allows for redundant RPs

**ipv6 pim** [**vrf** *vrf-name*] **bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** *seconds*] [**scope** *scope-value*] [**bidir**]

**ipv6 pim** [**vrf** *vrf-name*] **bsr candidate bsr***ipv6-address* [*hash-mask-length*] [**priority** *priority-value*]

# IPv6 Multicast – Boot Strap Router




```
R4(config)# ipv6 pim bsr candidate rp 2005:DEAD:BEEF:00C1::1 priority 0
R4(config)# ipv6 pim bsr candidate bsr 2005:DEAD:BEEF:00C1::1 124 priority 10
```

# IPv6 Multicast – Multicast Listener Discovery

- MLD uses ICMPv6 messages to perform the same functions as IGMP for IPv4 multicast
  - Type 130: Multicast Listener Query
  - Type 131: MLDv1 Multicast Listener Remote
  - Type 132: MLDv1 Multicast Listener Done
  - Type 143: MLDv2 Multicast Listener Report
- MLD version 1 is similar to IGMPv2
- MLD version 2 is similar to IGMPv3

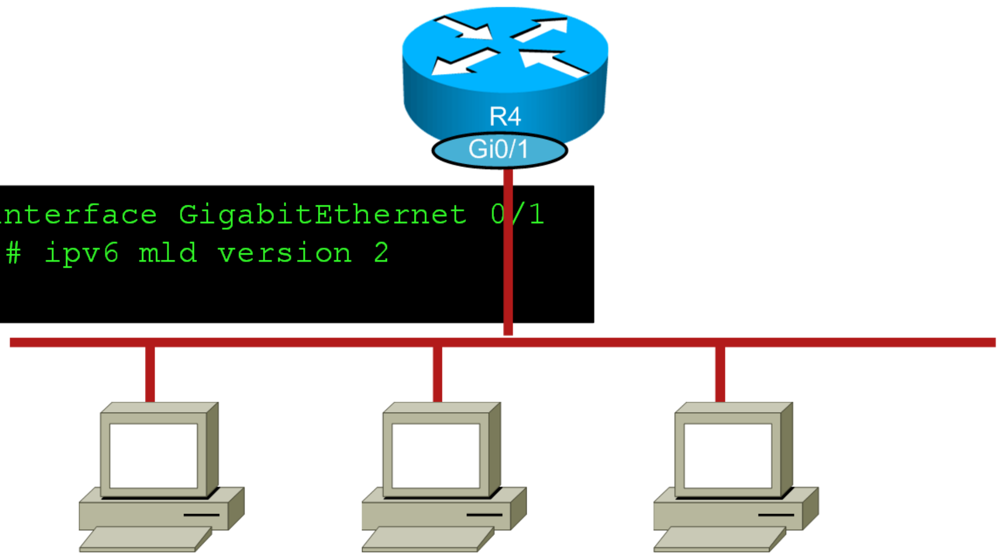
# IPv6 Multicast – MLDv1

| Type | Code | Checksum | |
|---|---|---|---|
| Max Response Delay | | Reserved | |
| Multicast Address | | | |

# IPv6 Multicast – MLDv2

- Support for source filtering
- Interoperable with MLDv1
- Message types:
  - General query
  - Multicast address-specific query
  - Multicast address and source-specific query
  - Current state report
  - State change report

# IPv6 Multicast – MLDv2




```
R4(config)# interface GigabitEthernet 0/1
R4(config-if)# ipv6 mld version 2
```

# Agenda

- IP version 6 (IPv6) addressing and different addressing types
- IPv6 neighbor discovery
- Basic IPv6 functionality protocols
- Tunneling techniques
- IPv6 Unicast routing
- Filtering and route redistribution
- IPv6 Multicast
- **IPv6 and 3560 switches**
- IPv6 NAT protocol translation
- Troubleshooting IPv6
- CCIE R&S
- Q&A

# IPv6 and 3560 switches

- Default is that IPv6 is NOT supported
- Have to change the SDM Template
  - To configure the ASIC to allocate resources for IPv6
  - IPv4-and-IPv6 Default

    or
  - IPv4-and-IPv6 Routing if PBRv6 is needed
  - Reload of the switch is needed

# Agenda

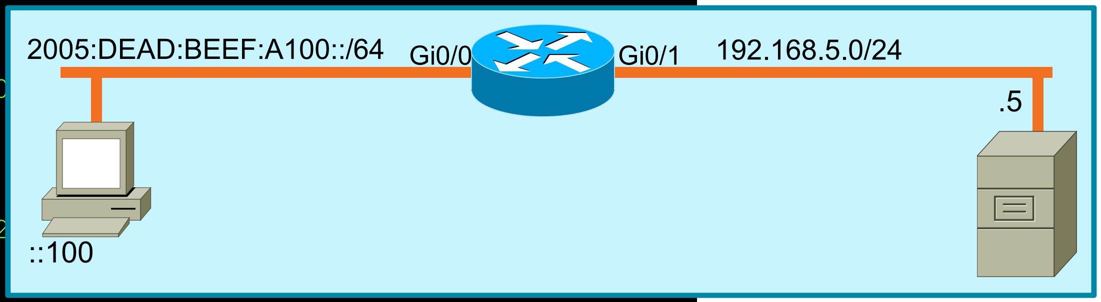
- IP version 6 (IPv6) addressing and different addressing types
- IPv6 neighbor discovery
- Basic IPv6 functionality protocols
- Tunneling techniques
- IPv6 Unicast routing
- Filtering and route redistribution
- IPv6 Multicast
- IPv6 and 3560 switches
- **IPv6 NAT protocol translation**
- Troubleshooting IPv6
- CCIE R&S
- Q&A

# IPv6 NAT Protocol Translation

- Network Address Translation--Protocol Translation (NAT-PT) is an IPv6 to IPv4 translation mechanism, as defined in RFC 2765 and RFC 2766, allowing IPv6-only devices to communicate with IPv4-only devices and vice versa.
- You somehow have to inject a IPv4 route into the IPv4 domain
- You somehow have to inject a IPv6 route into the IPv6 domain
- You have to set aside a /96 prefix for the IPv4 to IPv6 translation
- You identify the interfaces involved
- You set the translation rule for IPv4 to IPv6
- You set the translation rule for IPv6 to IPv4

# IPv6 NAT Protocol Translation

```
interface Loopback1
description To inject into IPv4 domain
ip address 192.168.55.1 255.255.255.0
!
interface GigabitEthernet0/0
 ipv6 address 2005:DEAD:BEEF:A100
 ipv6 nat
!
interface GigabitEthernet0/1
 ip address 192.168.5.1 255.255.2
 ipv6 nat
!
ipv6 router ospf 1
distribute-list prefix-list NATPT out connected
redistribute connected
!
ipv6 nat v4v6 source 192.168.5.5 2006:5:5::5
ipv6 nat v6v4 source list NATPT pool MyPool
ipv6 nat v6v4 pool MyPool 192.168.55.2 192.168.55.100 prefix-length 24
ipv6 nat prefix 2005:DEAD:BEEF:5::/96
!
ipv6 prefix-list NATPT seq 5 permit 2005:DEAD:BEEF:5::/96
!
ipv6 access-list NATPT
 permit ipv6 any any
```



# Agenda

- IP version 6 (IPv6) addressing and different addressing types
- IPv6 neighbor discovery
- Basic IPv6 functionality protocols
- Tunneling techniques
- IPv6 Unicast routing
- Filtering and route redistribution
- IPv6 Multicast
- IPv6 and 3560 switches
- IPv6 NAT protocol translation
- **Troubleshooting IPv6**
- CCIE R&S
- Q&A

# Troubleshooting IPv6

- Show ipv6 cef
- Show ipv6 interface [brief]
- Show ipv6 protocols
- Show ipv6 neighbors

# Troubleshooting RIPng

- Show ipv6 rip
- Show ipv6 rip database
- Show ipv6 route rip
- Debug ipv6 rip
- Clear ipv6 rip database

# Troubleshooting – OSPFv3 Show Commands

- Show ipv6 ospf [*process-id*] [*area-id*] interface [*int*]
- Show ipv6 ospf [*process-id*] [*area-id*]
- Show ipv6 ospf [*process-id*] [*area-id*] neighbor
- Show ipv6 ospf [*process-id*] [*area-id*] database
- Clear ipv6 ospf [*process-id*] {process | force-spf | redistribution | counters [neighbor [neighbor-interface]]}

# Troubleshooting – OSPFv3 Debug Commands

- debug ipv6 ospf adj
- debug ipv6 ospf hello
- debug ipv6 ospf spf
- debug ipv6 ospf flooding
- debug ipv6 ospf events
- debug ipv6 ospf lsa-generation
- debug ipv6 ospf database-timer
- debug ipv6 ospf packets
- debug ipv6 ospf retransmission
- debug ipv6 ospf tree

# Troubleshooting – EIGRPv3

- Show ipv6 eigrp topology
- Show ipv6 eigrp neighbors
- Show ipv6 eigrp interface
- Show ipv6 route eigrp
- Debug ipv6 eigrp
- Debug eigrp packet

# Troubleshooting – BGP

- Show bgp ipv6 unicast [*prefix/length*]
- Show bgp ipv6 unicast summary
- Show bgp ipv6 unicast neighbors [*address*]
- Debug bgp ipv6 unicast

# Troubleshooting – Multicast

- Show ipv6 mroute
- Show ipv6 mld interface
- Show ipv6 mld groups
- Show ipv6 pim interface
- Show ipv6 pim neighbors
- Debug ipv6 mfib
- Debug ipv6 mld
- Debug ipv6 pim

# Agenda

- IP version 6 (IPv6) addressing and different addressing types
- IPv6 neighbor discovery
- Basic IPv6 functionality protocols
- Tunneling techniques
- IPv6 Unicast routing
- Filtering and route redistribution
- IPv6 Multicast
- IPv6 and 3560 switches
- IPv6 NAT protocol translation
- Troubleshooting IPv6
- **CCIE R&S**
- Q&A

# CCIE R&S

- The CCIE Route Switch exam has now had IPv6 as a component since 2005 (the written has had it since 2007)
- IPv6 is becoming more of a core competency within the configuration and troubleshooting sections
- Point values can vary from exam to exam
- Points are likely to increase for this topic, while other topics reduce or are eliminated

# Agenda

- IP version 6 (IPv6) addressing and different addressing types
- IPv6 neighbor discovery
- Basic IPv6 functionality protocols
- Tunneling techniques
- IPv6 Unicast routing
- Filtering and route redistribution
- IPv6 Multicast
- IPv6 and 3560 switches
- IPv6 NAT protocol translation
- Troubleshooting IPv6
- CCIE R&S
- **Q&A**

# Q&A

# Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Receive 20 Passport points for each session evaluation you complete.
- Complete your session evaluation online now (open a browser through our wireless network to access our portal) or visit one of the Internet stations throughout the Convention Center.

Don't forget to activate your Cisco Live Virtual account for access to all session material, communities, and on-demand and live activities throughout the year. Activate your account at the Cisco booth in the World of Solutions or visit www.ciscolive.com

# Final Thoughts

- Get hands-on experience with the Walk-in Labs located in World of Solutions, booth 1042
- Come see demos of many key solutions and products in the main Cisco booth 2924
- Visit www.ciscoLive365.com after the event for updated PDFs, on-demand session videos, networking, and more!
- Follow Cisco Live! using social media:
  - Facebook: https://www.facebook.com/ciscoliveus
  - Twitter: https://twitter.com/#!/CiscoLive
  - LinkedIn Group: http://linkd.in/CiscoLI

BUILT FOR
THE HUMAN
NETWORK

CISCO