

IPv6 Intro Part 1: Overview and Addressing Basics



Cisco | Networking Academy®
Mind Wide Open™



Objectives

Part 1

- Describe IPv4 issues and workarounds.
- Describe IPv6 features and benefits.
- Describe the IPv6 header structure.
- Describe the basics of IPv6 addressing.

Part 2

- Describe IPv6 address types and special addresses.
- Describe IPv6 Unicast addresses and assignment methods.
- Explain the stateless autoconfiguration process.
- Describe IPv6 Multicast addresses and their use.
- Describe IPv6 subnetting and aggregation
- Configure and verify IPv6 addressing on networking devices.

Part 3

- Overview of static routing

Part 4

- Transitioning IPv4 to IPv6

PART 1:

IPv4 Issues and IPv6 Benefits





The Motivation for Moving to IPv6

- The ability to scale networks for future demands requires a large supply of IP addresses and improved mobility.
 - IPv6 combines expanded addressing with a more efficient header.
 - IPv6 satisfies the complex requirements of hierarchical addressing.



The Internet Is Growing ...

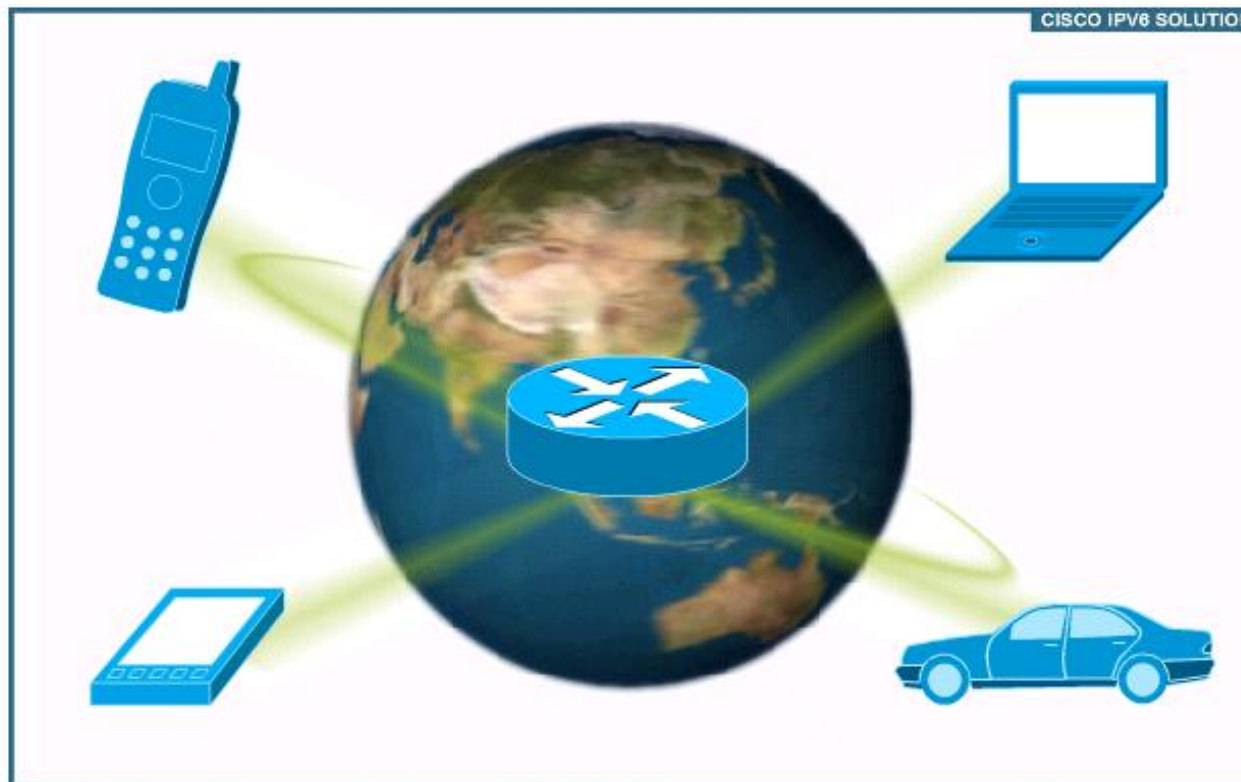
- In 2009, only 21% of the world population was connected.
- The Internet is growing rapidly. As of 2009, Africa, with a population of nearly one billion, had a penetration rate of only 5.3 percent. The adoption rate worldwide will continue to increase as **underdeveloped** countries get connected.





Explosion of New IP-Enabled Devices

More and more IP-enabled devices are connecting to the Internet. Devices include cell phones, consumer products (blue ray players, TVs), surveillance and transportation communication systems.





IPv4 Address Depletion

The green 45.x.x.x/8 address block was previously owned by Interop, the sponsors of the major annual international networking conference. They recently returned the bulk of these 16+ million address to the pool of available addresses. These are now available for assignment but won't last for long!

IPv4 address space as of October 20, 2010



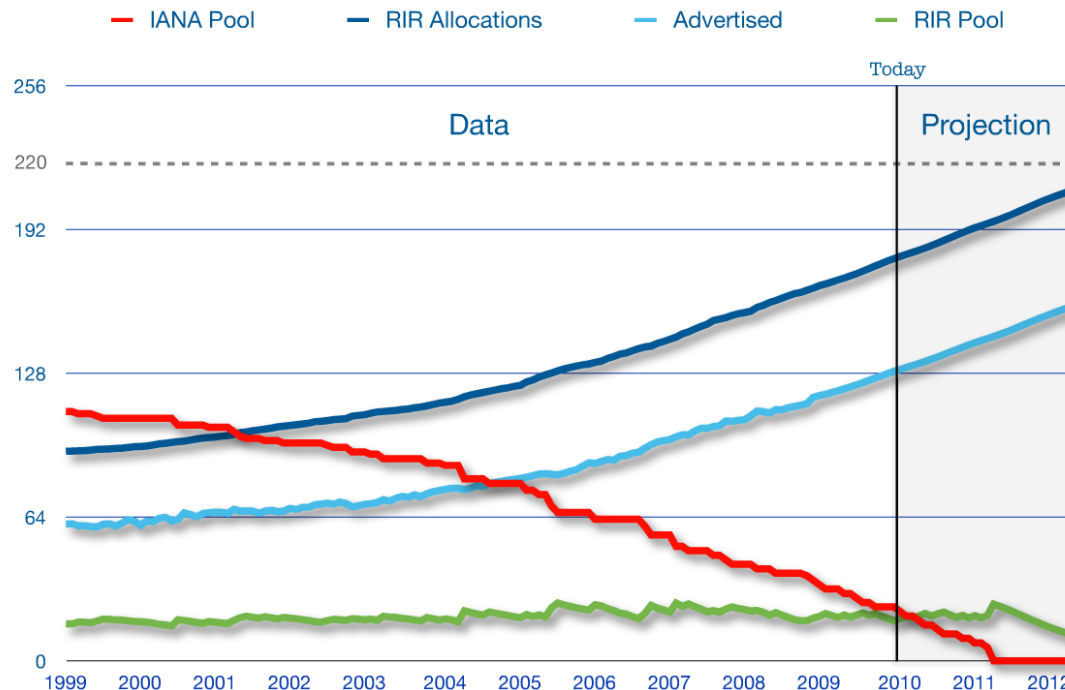
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255



IPv4 Address Depletion

- NAT, VLSM and CIDR were developed as workarounds and have helped to extend the life of IPv4.
- In October 2010, less than 5% of the public IPv4 addresses remained unallocated.

IPv4 address pool projections





Other IPv4 Issues

- The Internet routing tables continue to grow which means Internet core routers require more processing power, memory, and overhead
- Lack of true end-to-end model due to NAT



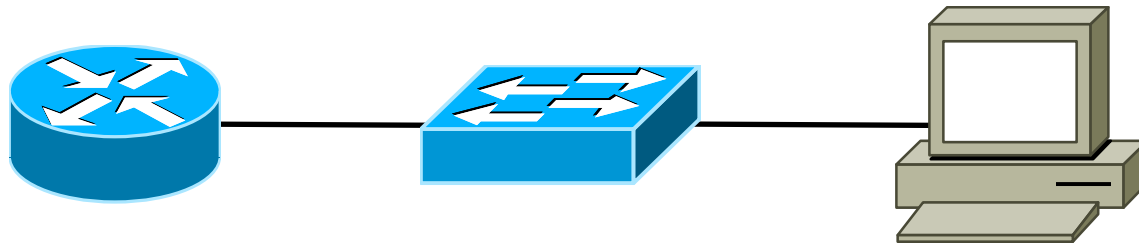
What Happened to IPv5?

- The **Internet Stream Protocol (ST)** was developed in the 1970s to experiment with voice, video and distributed simulation.
- Newer ST2 packets used IP version number 5 in the header.
- Although not officially known as IPv5, ST2 is considered to be the closest thing.
- The next Internet protocol became IPv6.



Features and Benefits of IPv6

- Larger address space - *with a 128-bit There are enough IPv6 addresses to allocate more than the entire IPv4 Internet address space to everyone on the planet*
- Elimination of public-to-private NAT
- Elimination of broadcast addresses
- Simplified header for improved router efficiency
- Support for mobility and security
- Many devices and applications already support IPv6





Features and Benefits of IPv6 - Continued

- Prefix renumbering simplified
- Multiple addresses per interface
- Address autoconfiguration
 - No requirement for DHCP - although stateless autoconfiguration can satisfy most IPv6 addressing needs, DHCPv6 is available and can still be used to assign addresses statefully if the network administrator desires more control over addressing
- Link-local and globally routable addresses
 - IPv6 link-local addresses are used as the next hop when IGPs are exchanging routing updates and can be automatically configured
- Multiple-level hierarchy by design
 - More efficient route aggregation
- Transition mechanisms from IPV4 to IPV6

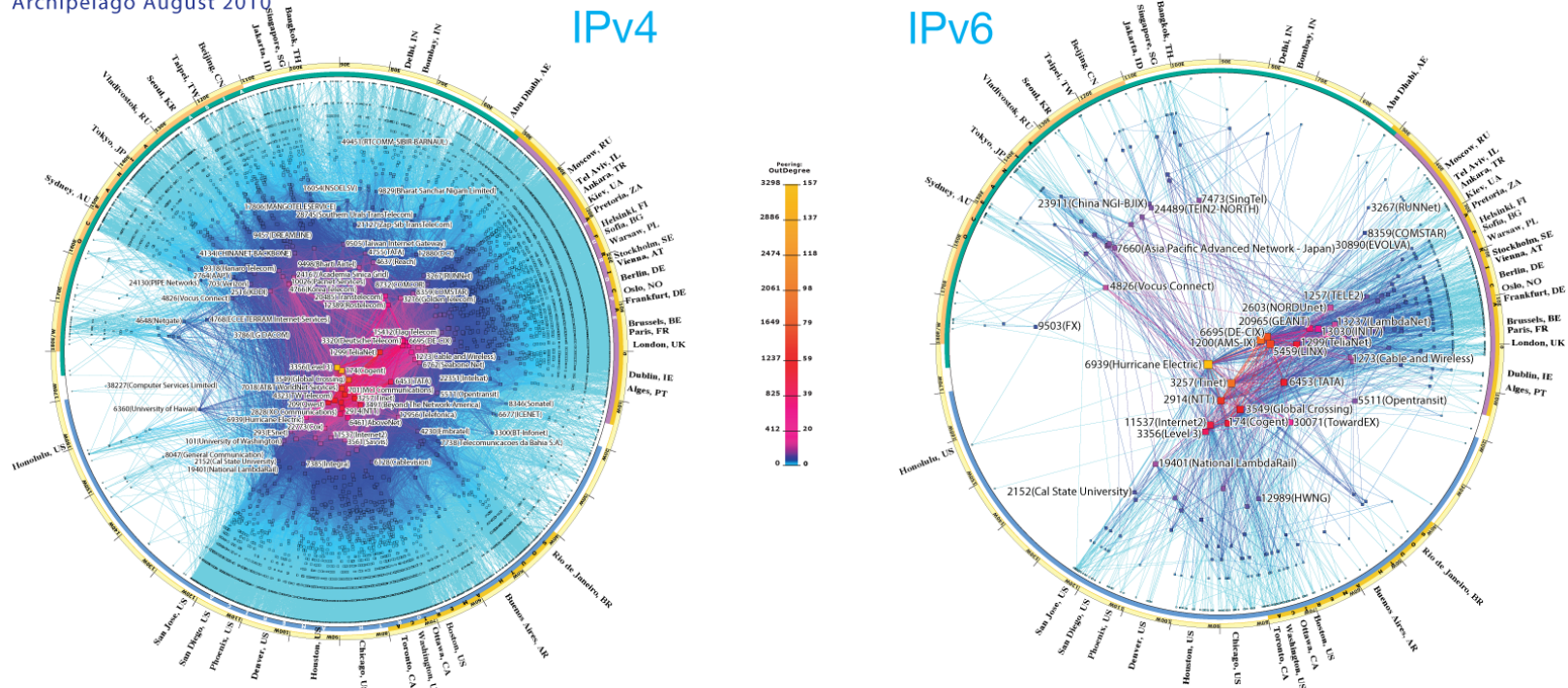


Who is Using IPv6?

- Governments, Corporations, Universities
- Internet Service Providers
- Google, Facebook

CAIDA's IPv4 & IPv6 AS Core AS-level INTERNET GRAPH

Archipelago August 2010



copyright © 2010 UC Regents. all rights reserved.



IP Address Space Allocated to ARIN

IPv6 address Example

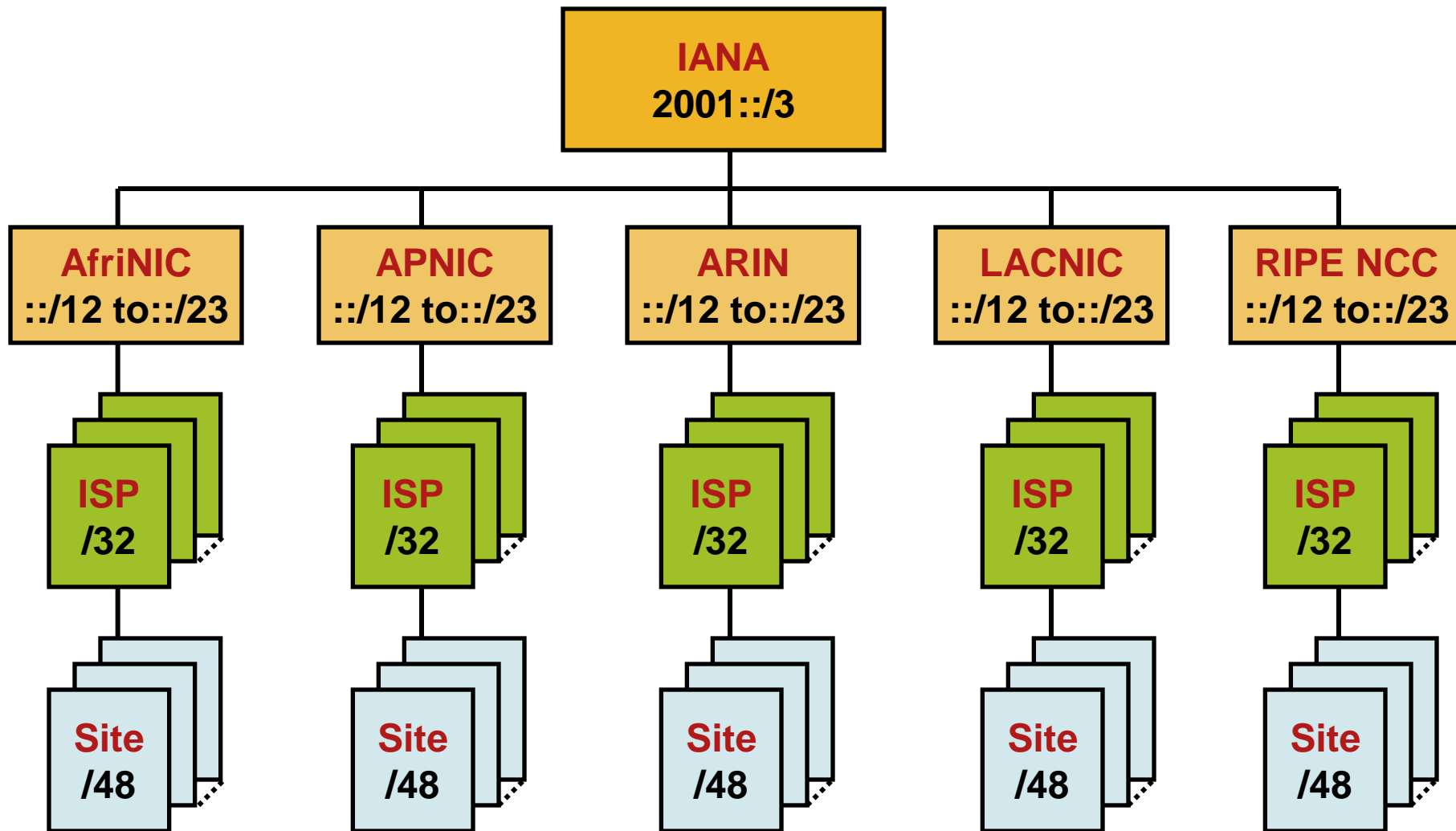
Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7	Field 8
0010 0000 0000 0001	0011 0100 0101 0110	0000 0000 0000 0000	0000 0000 0000 0000	0111 1000 1001 1010	1011 1100 1101 1110	0000 0000 0000 0000	0000 0000 0000 0000
2 0 0 1	: 3 4 5 6	: 0 0 0 0	: 0 0 0 0	: 7 8 9 A	: B C D E	: 0 0 0 0	: 0 0 F 0

IPv6 Allocation Blocks

- 2001:0400::/23
 - 2001:1800::/23
 - 2001:4800::/23
 - 2600:0000::/12
 - 2610:0000::/23
 - 2620:0000::/23
- Internet Assigned Numbers Authority (IANA) coordinates the global IPv4, IPv6 and autonomous system (AS) number space, and allocates these to Regional Internet Registries (RIRs).



IPv6 Prefix Allocation Hierarchy and Policy Example



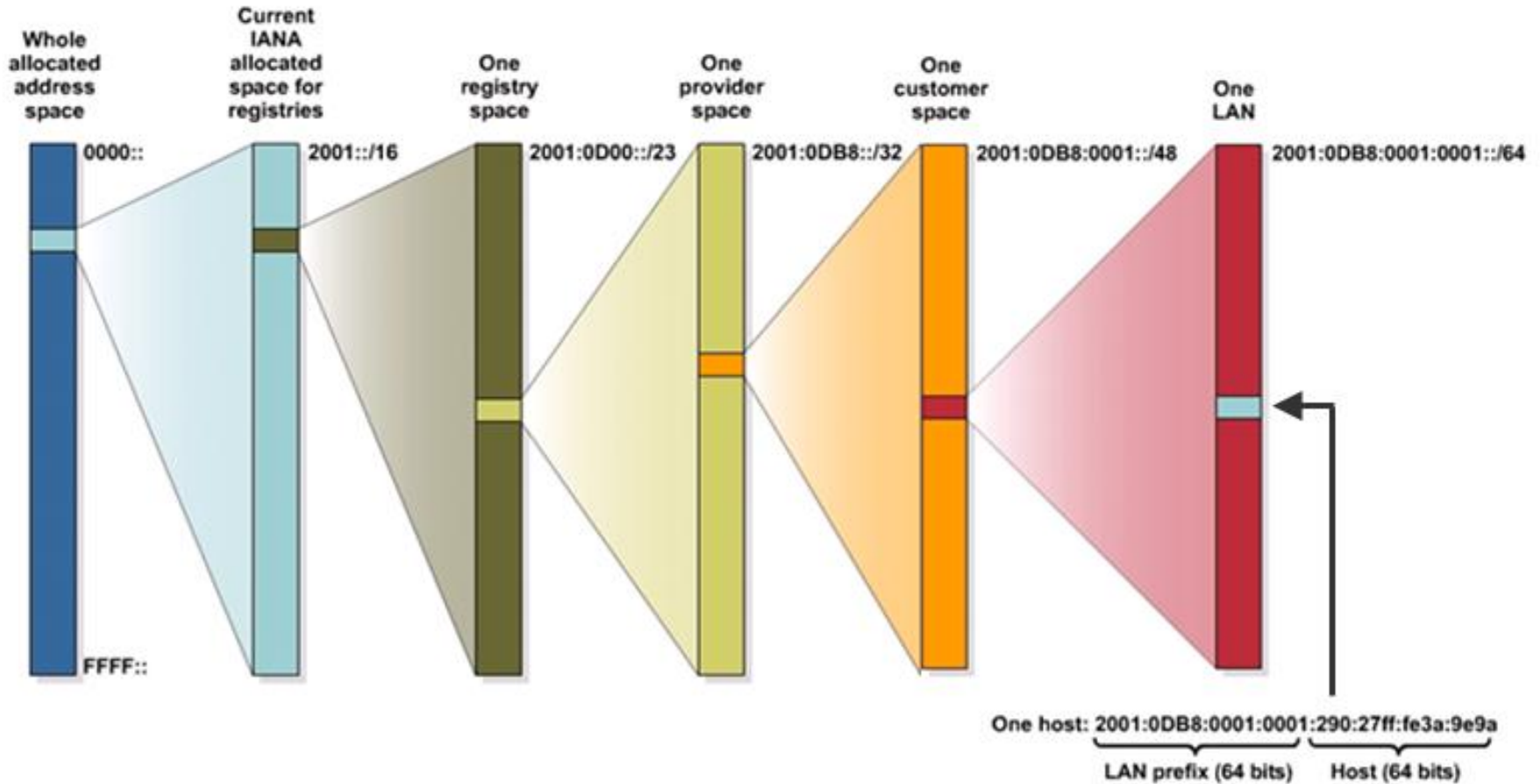


IPv6 Prefix Allocation Hierarchy and Policy Example

- IANA allocates from the total $2001::/3$ global IPv6 unicast address space to regional registries such as APNIC and RIPE
- Each regional registry gets a $/23$ prefix from IANA (although a larger address space can be requested)
- The registry typically allocates a $/32$ prefix to an IPv6 ISP
- Then the ISP allocates a $/48$ prefix to each customer (or potentially $/64$ for a local network or $/128$ for a single host)



IPv6 Address Allocation Process





Is IPv4 Obsolete?

- IPv4 is in no danger of disappearing overnight.
- It will coexist with IPv6 and then gradually be replaced.
- IPv6 provides several transition options including:
 - Dual stack
 - Tunneling mechanisms
 - NAT-PT (Deprecated)

NOTE: Although NAT-PT (defined RFC 2766) is listed as a transition option here for completeness, it is important to note that, due to numerous issues, it has been obsoleted by RFC 4966 and deprecated to historic status.

IPv6 Header Structure



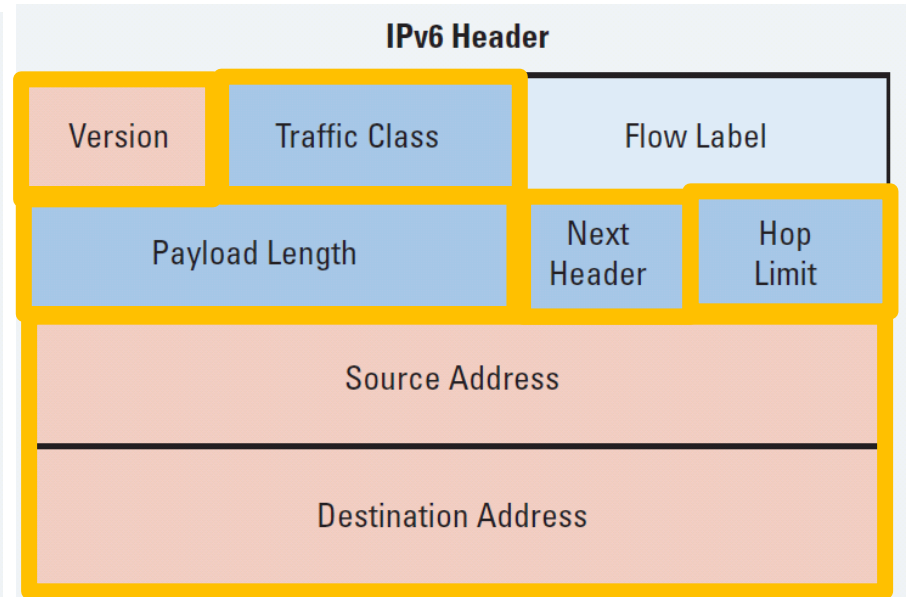
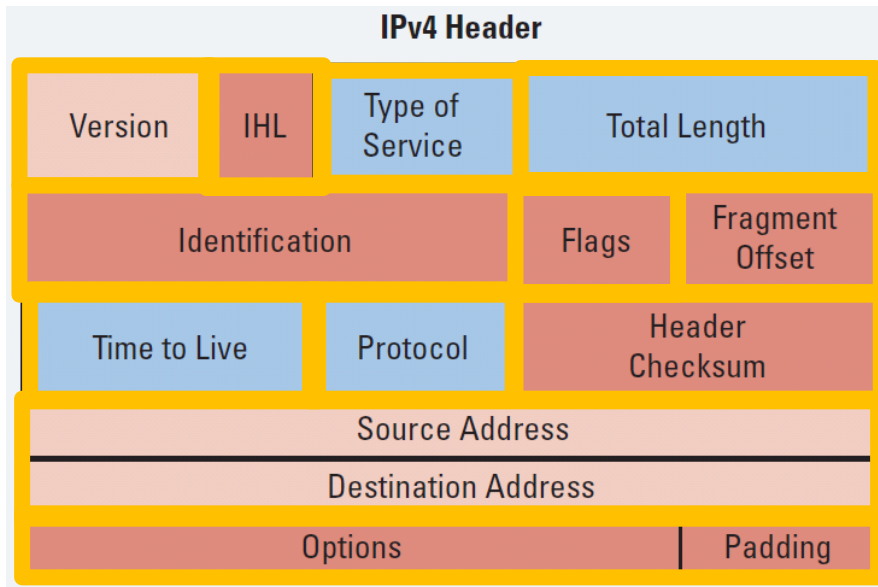


IPv6 Header Improvements

- Improved routing efficiency
- No requirement for processing checksums
- Simpler and more efficient extension header mechanisms
- Flow labels for per-flow processing



IPv4 Header vs. IPv6 Header



- Field's name kept from IPv4 to IPv6
- Fields not kept in IPv6

- Name and position changed in IPv6
- New field in IPv6



IPv6 Header

- **Version**—A 4-bit field, the same as in IPv4. For IPv6, this field contains the number 6;
- **Traffic class**—An 8-bit field similar to the type of service (ToS) field in IPv4. This field tags the packet with a traffic class that it uses in differentiated services (DiffServ) quality of service (QoS). These functionalities are the same for IPv6 and IPv4.
- **Flow label**—This 20-bit field is new in IPv6. It can be used by the source of the packet to tag the packet as being part of a specific flow, allowing multilayer switches and routers to handle traffic on a per-flow basis rather than per-packet, for faster packet-switching performance. This field can also be used to provide QoS.
- **Payload length**—This 16-bit field is similar to the IPv4 total length field.
- **Next header**—The value of this 8-bit field determines the type of information that follows the basic IPv6 header. It can be a transport-layer packet, such as Transmission Control Protocol (TCP) or User Datagram Protocol (UDP), or it can be an extension header. The next header field is similar to the protocol field of IPv4.
- **Hop limit**—This 8-bit field specifies the maximum number of hops that an IP packet can traverse. Similar to the time to live (TTL) field in IPv4, each router decreases this field by one.
- **Source address**—This field has 16 octets or 128 bits. It identifies the source of the packet.
- **Destination address**—This field has 16 octets or 128 bits. It identifies the destination of the packet.
- **Extension headers**—The extension headers, if any, and the data portion of the packet follow the other eight fields. The number of extension headers is not fixed, so the total length of the extension header chain is variable.



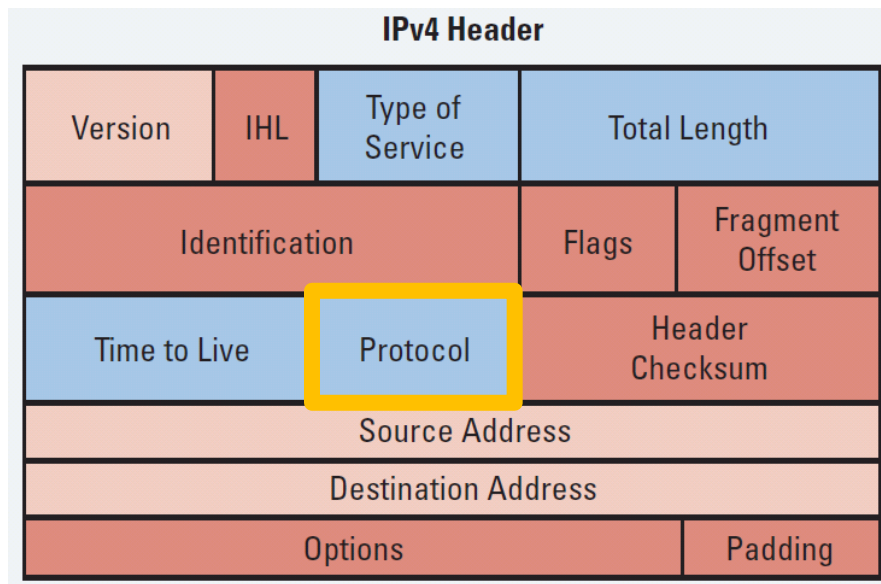
IPv6 Header

- Notice that the IPv6 header does not have a header checksum field. Because link-layer technologies perform checksum and error control and are considered relatively reliable, an IP header checksum is considered to be redundant. Without the IP header checksum, upper-layer checksums, such as within UDP, are mandatory with IPv6.

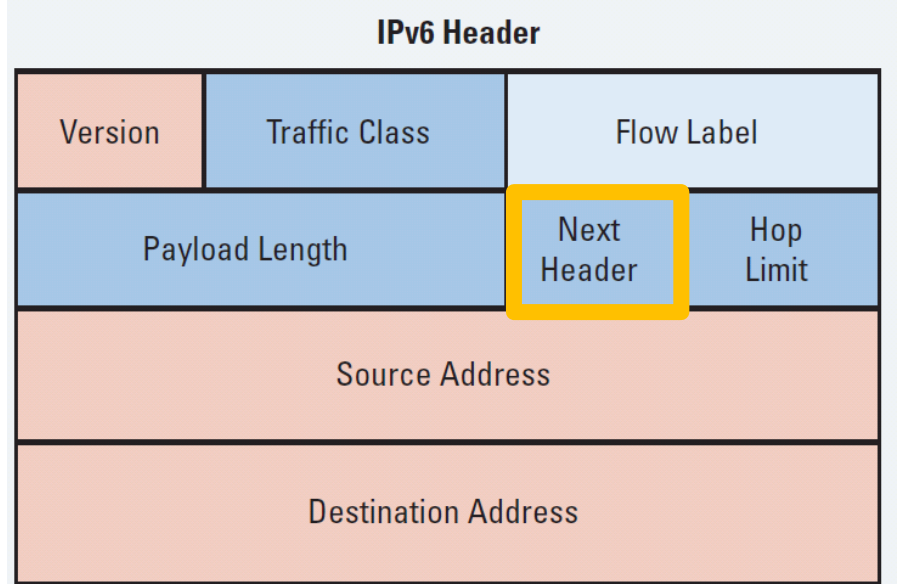


Protocol and Next Header Fields

- In IPv4 the Protocol field is used to identify the next level protocol (e.g., TCP, UDP, ICMP, ...)
- In IPv6, this field is called the "Next Header" field and serves the same purpose



- Field's name kept from IPv4 to IPv6
- Fields not kept in IPv6

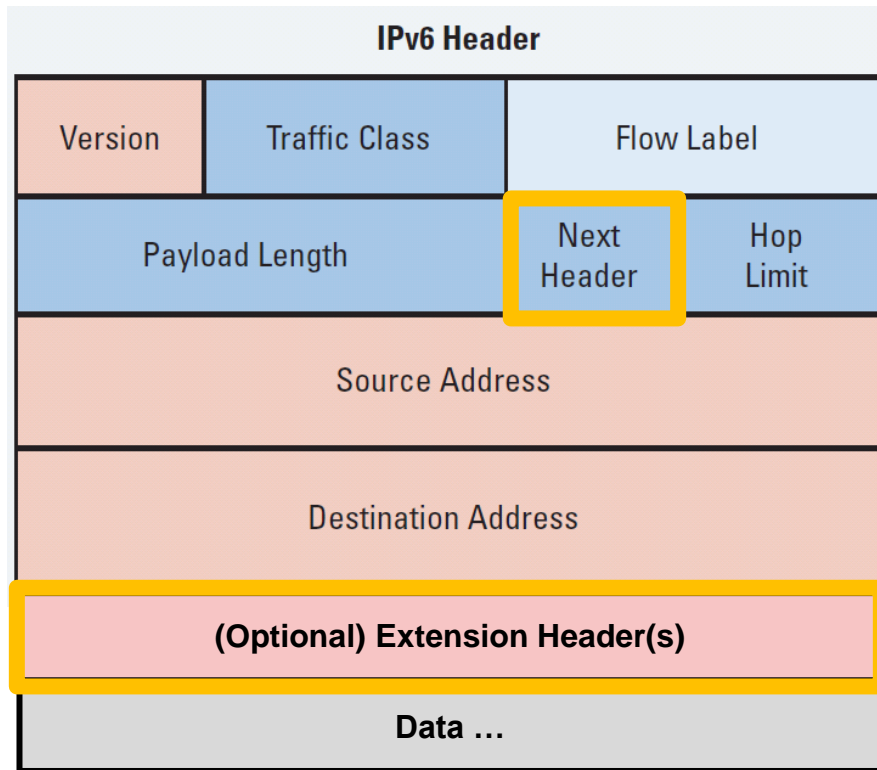


- Name and position changed in IPv6
- New field in IPv6



Extension Headers

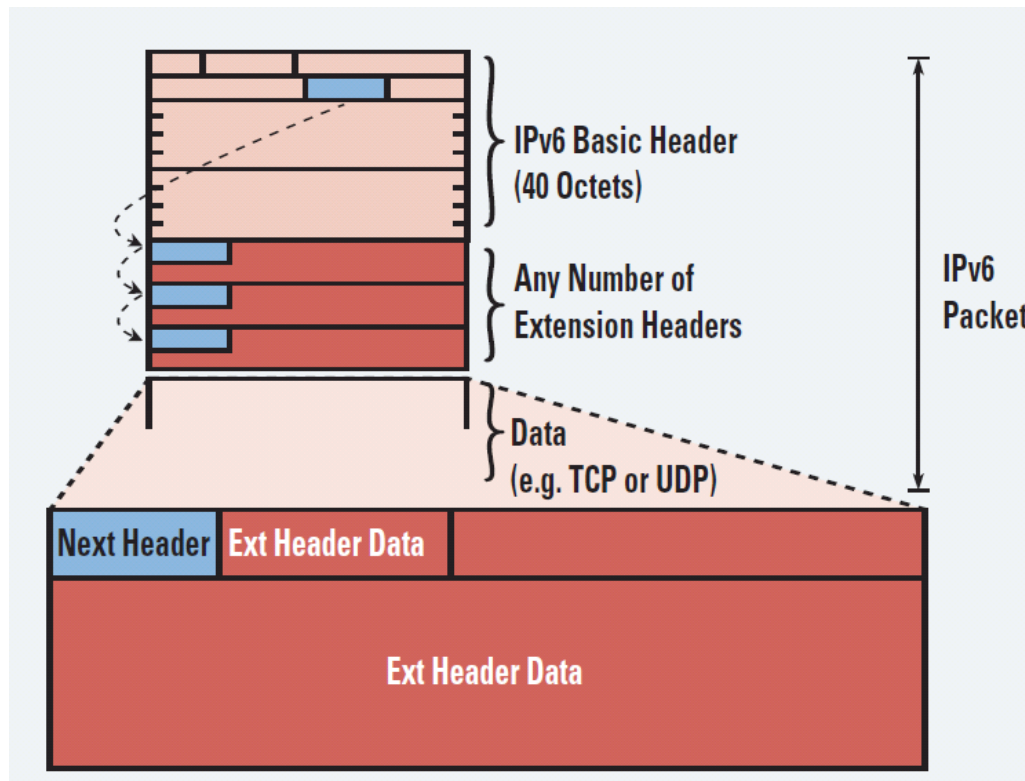
- The Next Header field identifies what follows the Destination Address field:





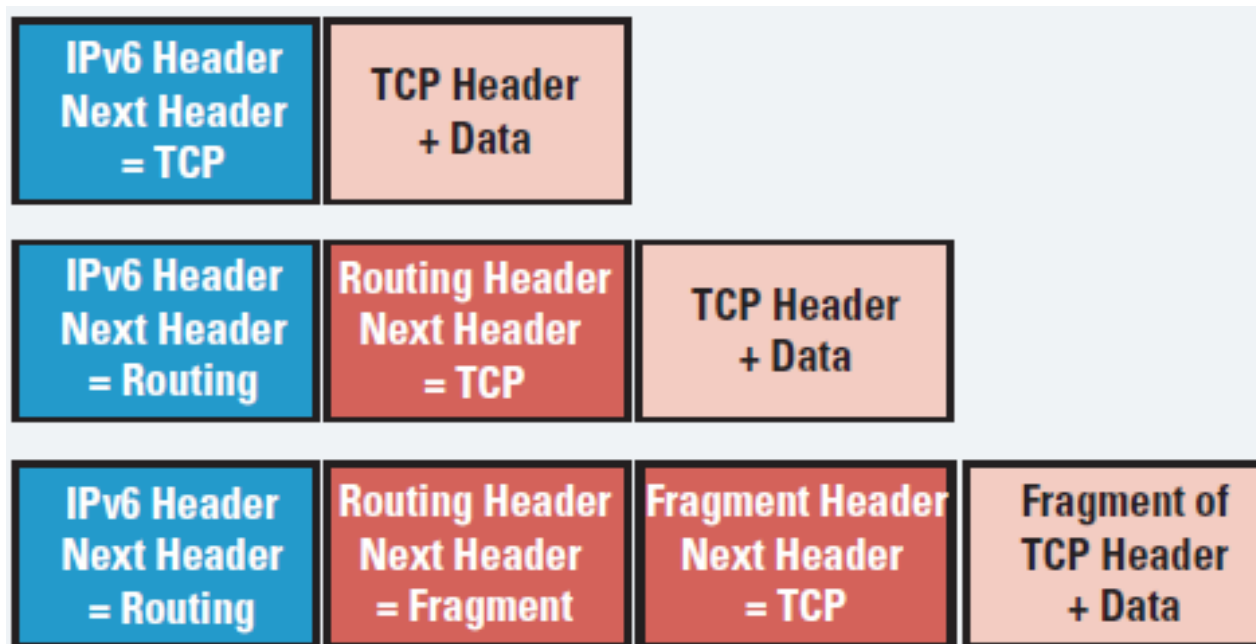
Extension Headers

- The destination node examines the first extension header (if any).





Extension Header Options





Extension Header Chain Order

Process Order	Extension Header	Next-header value (protocol #)
1	Hop-by-hop options header	0
2	Destination options header	60
3	Routing header	43
4	Fragment header	44
5	Authentication header (AH) and ESP header	ESP = 50 AH = 51
6	Upper-layer header: TCP UDP	TCP = 6 UDP = 17

IPv6 Addressing Overview

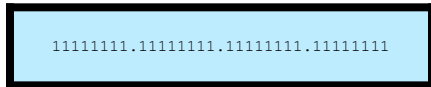




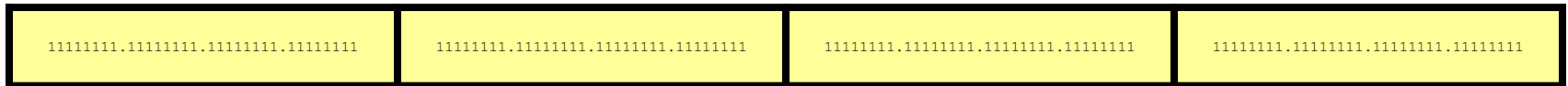
IPv6 Addressing Overview

- IPv6 increases the number of address bits by a factor of 4, from 32 to 128, providing a very large number of addressable nodes.

IPv4 = 32 bits



IPv6 = 128 bits





IPv6 Address Specifics

- The 128-bit IPv6 address is written using 32 hexadecimal numbers.
- The format is **x:x:x:x:x:x:x:x**, where **x** is a 16-bit hexadecimal field, therefore each **x** represents four hexadecimal digits.
- Example address:
 - **2035:0001:2BC5:0000 : 0000:087C:0000:000A**



Abbreviating IPv6 Addresses

- Leading 0s within each set of four hexadecimal digits can be omitted.
 - **09c0** = **9c0**
 - **0000** = **0**
- A pair of colons (“**::**”) can be used, *once* within an address, to represent any number (“a bunch”) of successive zeros.



IPv6 Address Abbreviation Example

2031:0000:130F:0000:0000:09C0:876A:130B

2031:0:130F:0:0:9C0:876A:130B

2031:0:130F:0:0:9C0:876A:130B

2031:0:130F:::9C0:876A:130B



More IPv6 Address Abbreviation Examples

FF01:0000:0000:0000:0000:0000:0000:1

= FF01:0:0:0:0:0:0:1

= FF01::1

E3D7:0000:0000:0000:51F4:00C8:C0A8:6420

= E3D7::51F4:C8:C0A8:6420

3FFE:0501:0008:0000:0260:97FF:FE40:EFAB

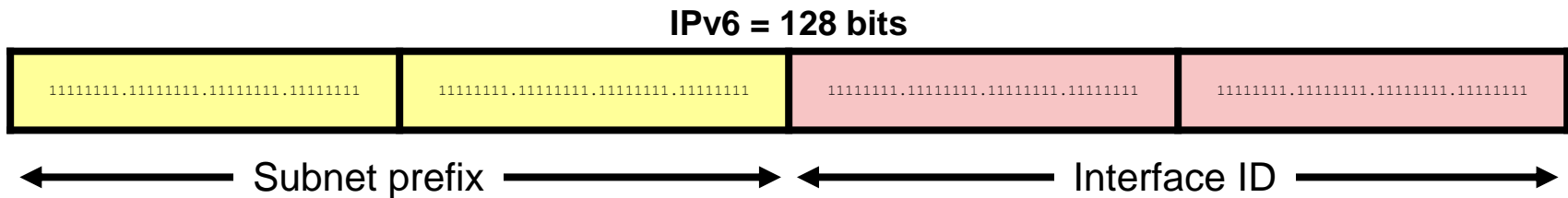
= 3FFE:501:8:0:260:97FF:FE40:EFAB

= 3FFE:501:8::260:97FF:FE40:EFAB



IPv6 Address Components

- An IPv6 address consists of two parts:
 - A *subnet prefix*
 - An *interface ID*





Subnet Prefix

- IPv6 uses CIDR notation (*/prefix*) to denote the number of bits that represent the subnet.

Example:

FC00:0:0:1::1234/64

is really

FC00:0000:0000:0001:0000:0000:0000:1234/64

- The first 64-bits (**FC00:0000:0000:0001**) forms the address prefix.
- The last 64-bits (**0000:0000:0000:1234**) forms the Interface ID.



Subnet Prefix

- The prefix length is almost always /64.
 - However, IPv6 rules allow for either shorter or longer prefixes
- Deploying a /64 IPv6 prefix on a device recommended.
 - Allows Stateless Address Auto Configuration (SLAAC)

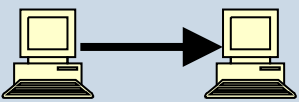
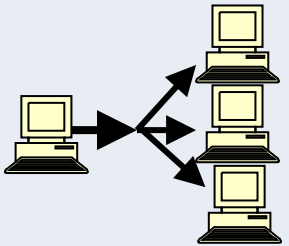
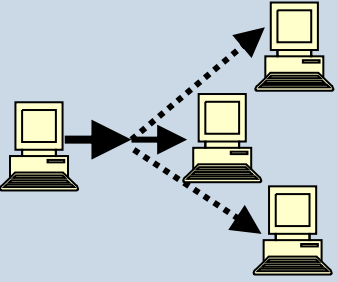


Interface Identifiers

- IPv6 addresses on a link must be unique.
- Using the link prefix length, IPv6 hosts can automatically create a unique IPv6 address.
- The following Layer 2 protocols can dynamically create the IPv6 address interface ID:
 - Ethernet
 - PPP
 - HDLC
 - NBMA, Frame Relay
- Note that the Cisco IOS does not support autoconfiguration of IPv6 addresses with all Layer 2 protocols, only the more common ones



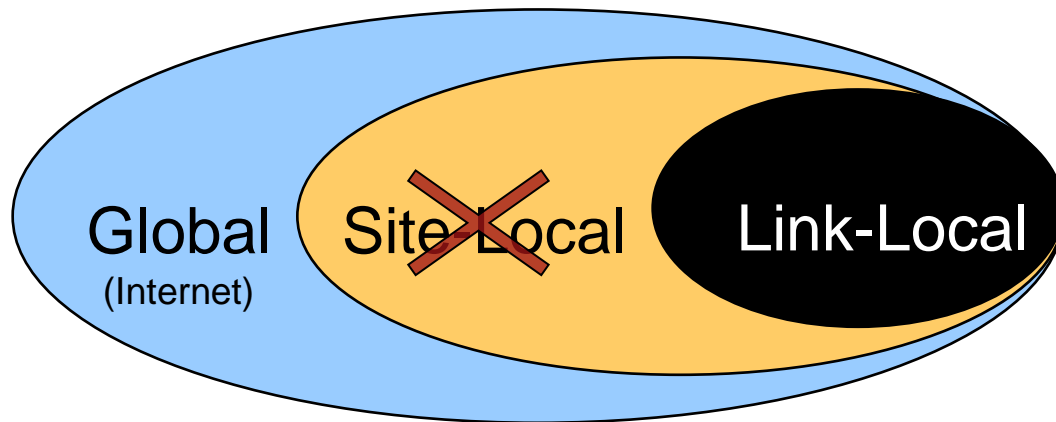
IPv6 Address Types

Address Type	Description	Topology
<p style="text-align: center;">Unicast</p>	<p>“One to One”</p> <ul style="list-style-type: none"> • An address destined for a single interface. • A packet sent to a unicast address is delivered to the interface identified by that address. 	
<p style="text-align: center;">Multicast</p>	<p>“One to Many”</p> <ul style="list-style-type: none"> • An address for a set of interfaces (typically belonging to different nodes). • A packet sent to a multicast address will be delivered to all interfaces identified by that address. 	
<p style="text-align: center;">Anycast</p>	<p>“One to Nearest” (Allocated from Unicast)</p> <ul style="list-style-type: none"> • An address for a set of interfaces. • In most cases these interfaces belong to different nodes. • created “automatically” when a single unicast address is assigned to more than one interface. • A packet sent to an anycast address is delivered to the closest interface as determined by the IGP. 	



IPv6 Unicast Address Scopes

- Address types have well-defined destination scopes:
 - Link-local address
 - Site-local address (replaced by Unique-local addresses)
 - Global unicast address

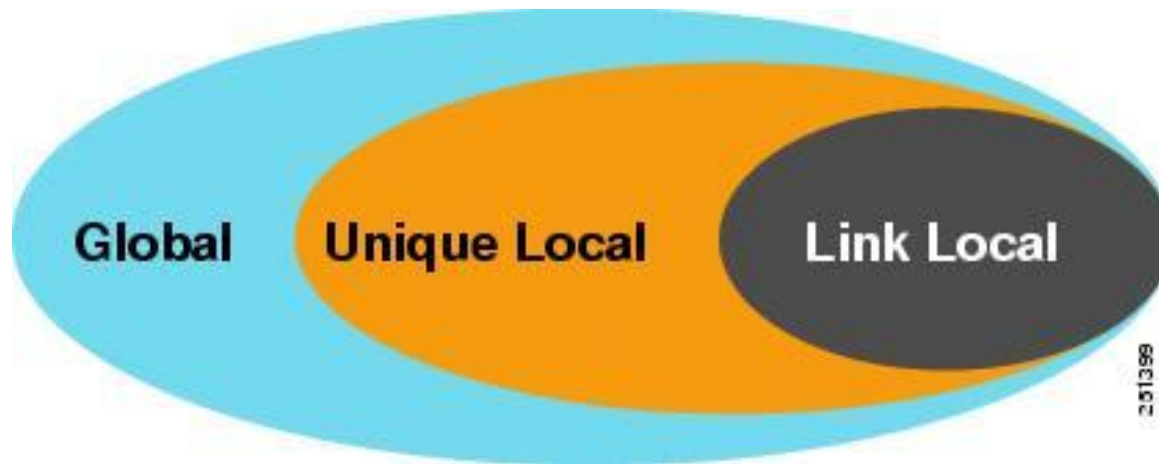


- **Note:** Site-Local Address are deprecated in RFC 3879.



IPv6 Unicast Address Scopes

- Link-local addresses—only on single link, not routed
 - **FE80:: prefix**
- Unique-local addresses—routed within private network
 - **FC00:: prefix**
- Global unicast addresses—globally routable
 - **2001:: prefix most common**





Site-Local Addresses - Deprecated

- Site-local addresses allowed devices in the same organization, or site, to exchange data.
 - Site-local addresses start with the prefix **FEC0::/10**.
- They are analogous to IPv4's private address classes.
 - However, using them would also mean that NAT would be required and addresses would again not be end-to-end.
- Site-local addresses are **no longer supported** (deprecated by RFC 3879).



Multiple IPv6 Addresses per Interface

- An interface can have multiple global IPv6 addresses.
- Typically, an interface is assigned a link-local and one (or more) global IPv6 address.
- For example, an Ethernet interface can have:
 - **Link-local address**
(FE80::21B:D5FF:FE5B:A408)
 - **Global unicast address**
(2001:8:85A3:4289:21B:D5FF:FE5B:A408)
- The Link-local address is used for local device communication.
- The Global address is used to provide Internet reachability.

PART 2:

IPv6 Intro Part 2: Address Types and Application



Cisco | Networking Academy®
Mind Wide Open™

IPv6 Address Types





IPv6 Address Space Overview

Prefix Hex Value	Use
0000 to 00FF	<ul style="list-style-type: none"> • Unspecified • Loopback • IPv4-compatible
0100 to 01FF	Unassigned (0.38 % of IPv6 space)
0200 to 03FF	NSAP Network Service AP)
0400 to 1FFF	Unassigned (~11% of IPv6 space)
2000 to 3FFF	Aggregatable global unicast (12.5%)
4000 to FE7F (Huge)	Unassigned (~75% of IPv6 space)
FE80 to FEBF	Link-local
FC00 to FCFF	Unique-local
FF00 to FFFF	Multicast

Note: IPv6 Internet uses 2001::/3 which is < 2% of IPv6 address space

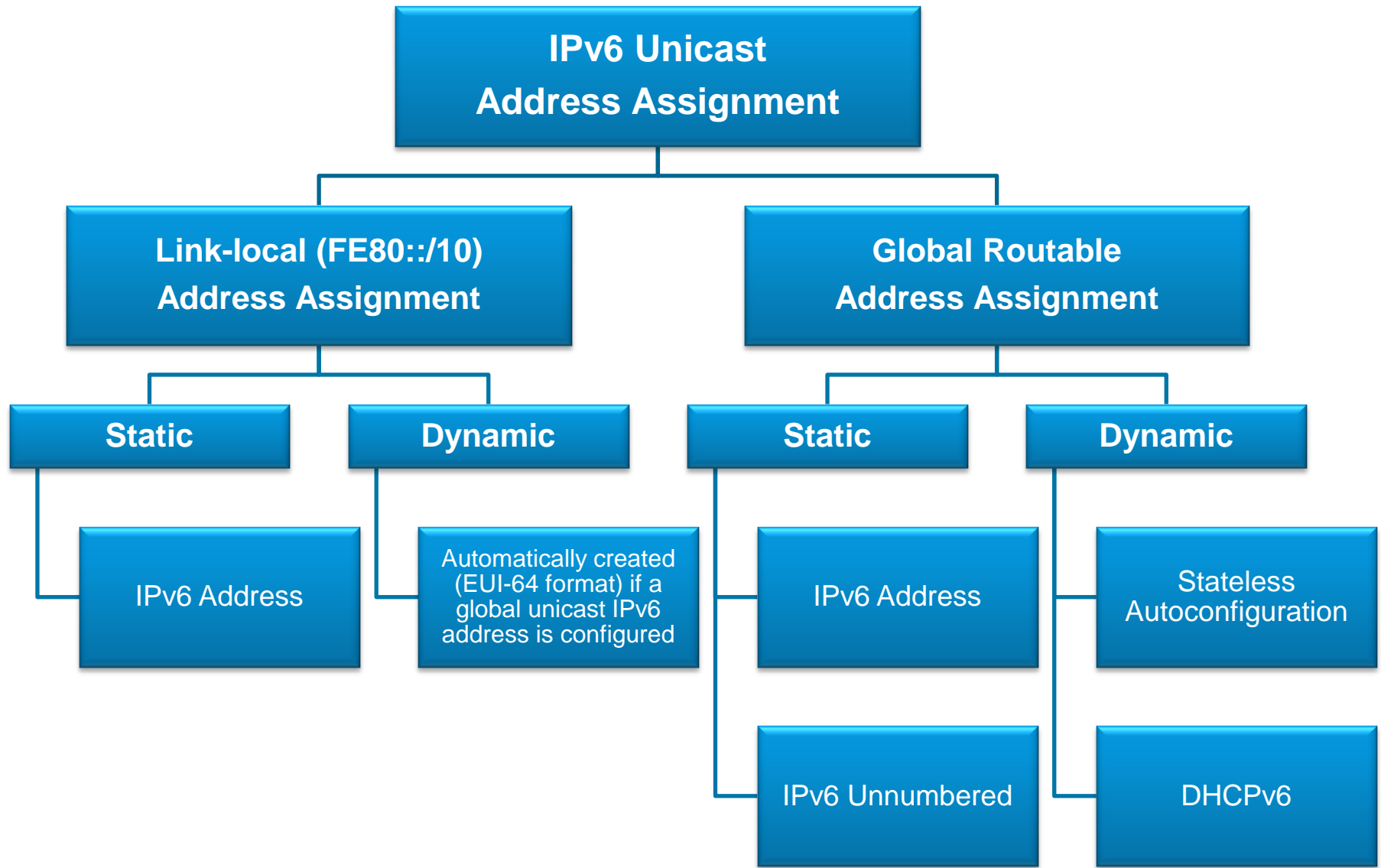


Special IPv6 Addresses

IPv6 Address	Description
::/0	<ul style="list-style-type: none"> • All networks and used when specifying a default static route. • It is equivalent to the IPv4 quad-zero (0.0.0.0)
::/128	<ul style="list-style-type: none"> • Unspecified address and is initially assigned to a host when it first resolves its local link address
::1/128	<ul style="list-style-type: none"> • Loopback address of local host • Equivalent to 127.0.0.1 in IPv4
FE80::/10	<ul style="list-style-type: none"> • Link-local unicast address • Similar to the Windows autoconfiguration IP address of 169.254.x.x
FF00::/8	<ul style="list-style-type: none"> • Multicast addresses
All other addresses	<ul style="list-style-type: none"> • Global unicast address



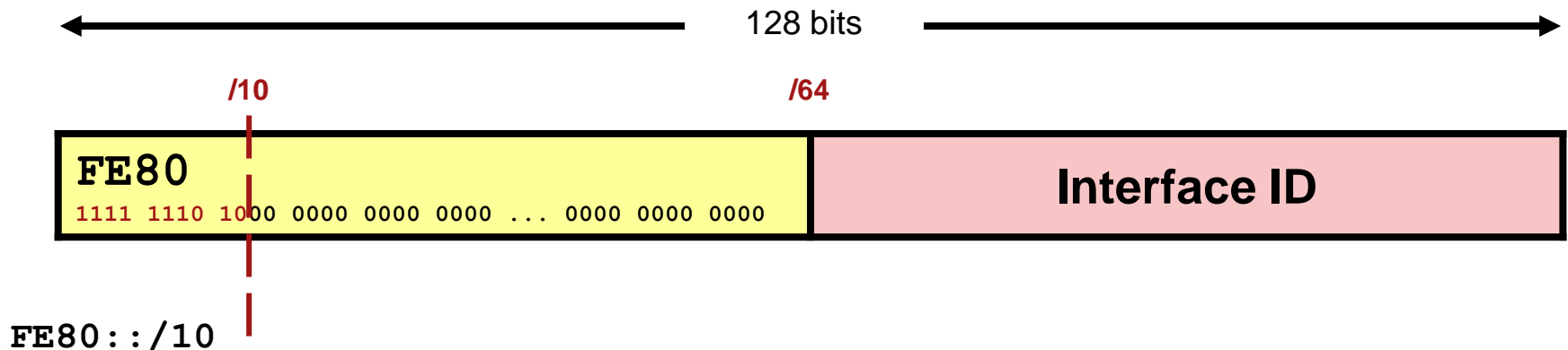
IPv6 Unicast Addresses





IPv6 Link-Local Unicast Address

- Link-local addresses play a crucial role in the operation of IPv6.
- They are dynamically created using a link-local prefix of **FE80::/10** and a 64-bit interface identifier.





IPv6 Link-Local Unicast Address

- When pinging another device using Cisco IOS and a link-local address, the outgoing interface must be specified.

```
R2# ping FE80::202:16FF:FEEB:3D01
```

```
Output Interface: serial0/0/0
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to FE80::202:16FF:FEEB:3D01, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 31/34/47 ms
```



IPv6 Link-Local Unicast Address Example

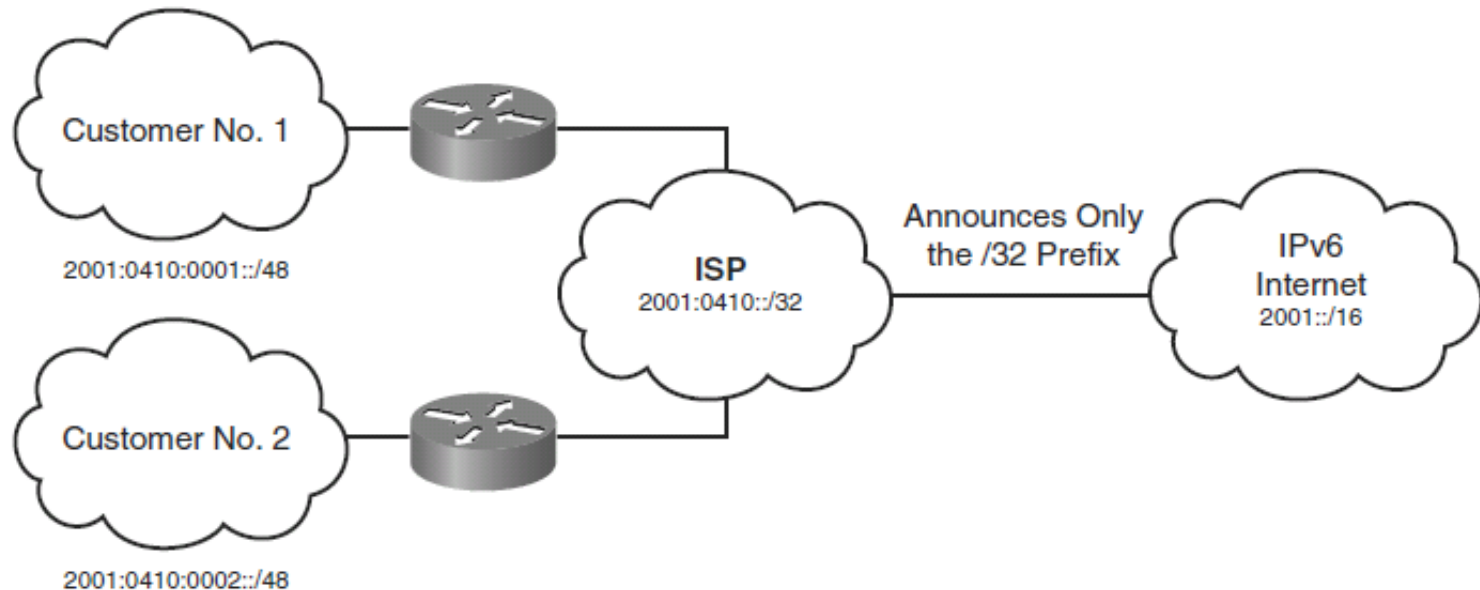
```

R1# show ipv6 interface loopback 100
Loopback100 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::222:55FF:FE18:7DE8
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:8:85A3:4290:222:55FF:FE18:7DE8, subnet is 2001:8:85A3:4290::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF18:7DE8
  MTU is 1514 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is not supported
  ND reachable time is 30000 milliseconds (using 31238)
  Hosts use stateless autoconfig for addresses.
R1#
  
```



IPv6 Global Unicast Address

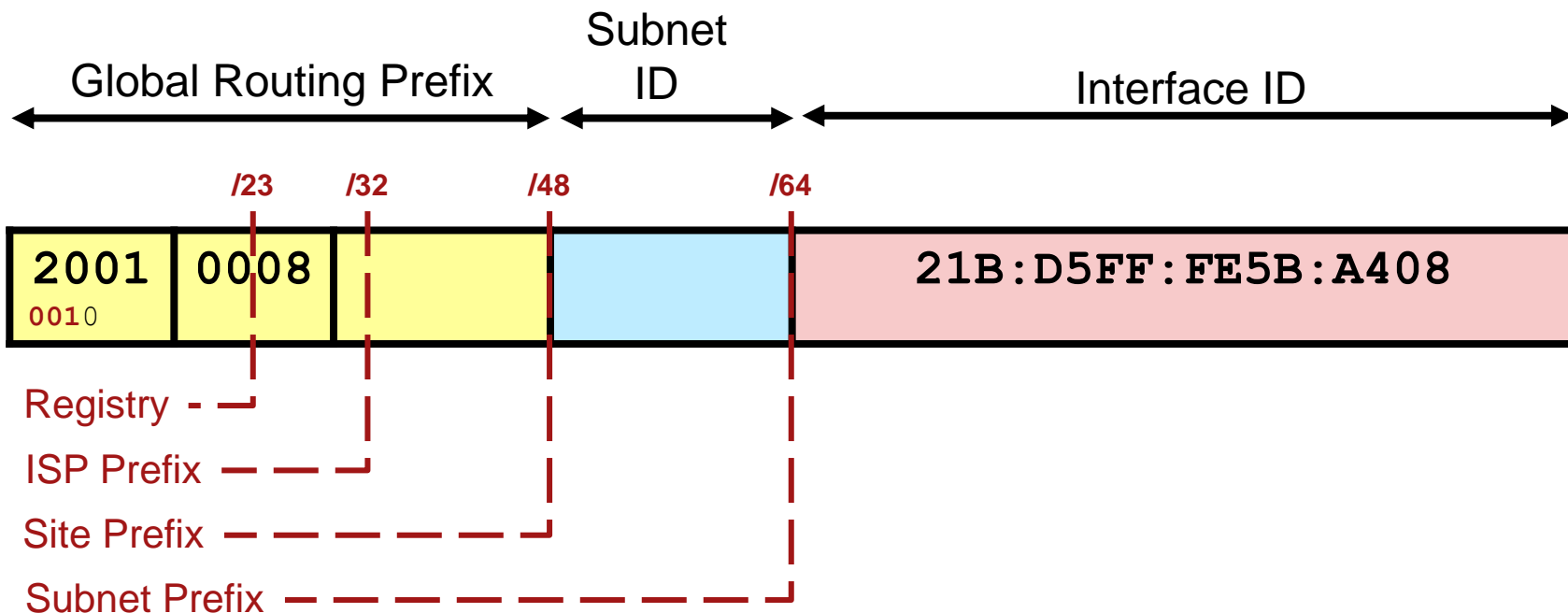
- A global unicast address is an IPv6 address from the global public unicast prefix (2001::/16).
- These addresses are routable on the global IPv6 Internet.
- Global unicast addresses are aggregated upward through organizations and eventually to the ISPs.





IPv6 Global Unicast Address

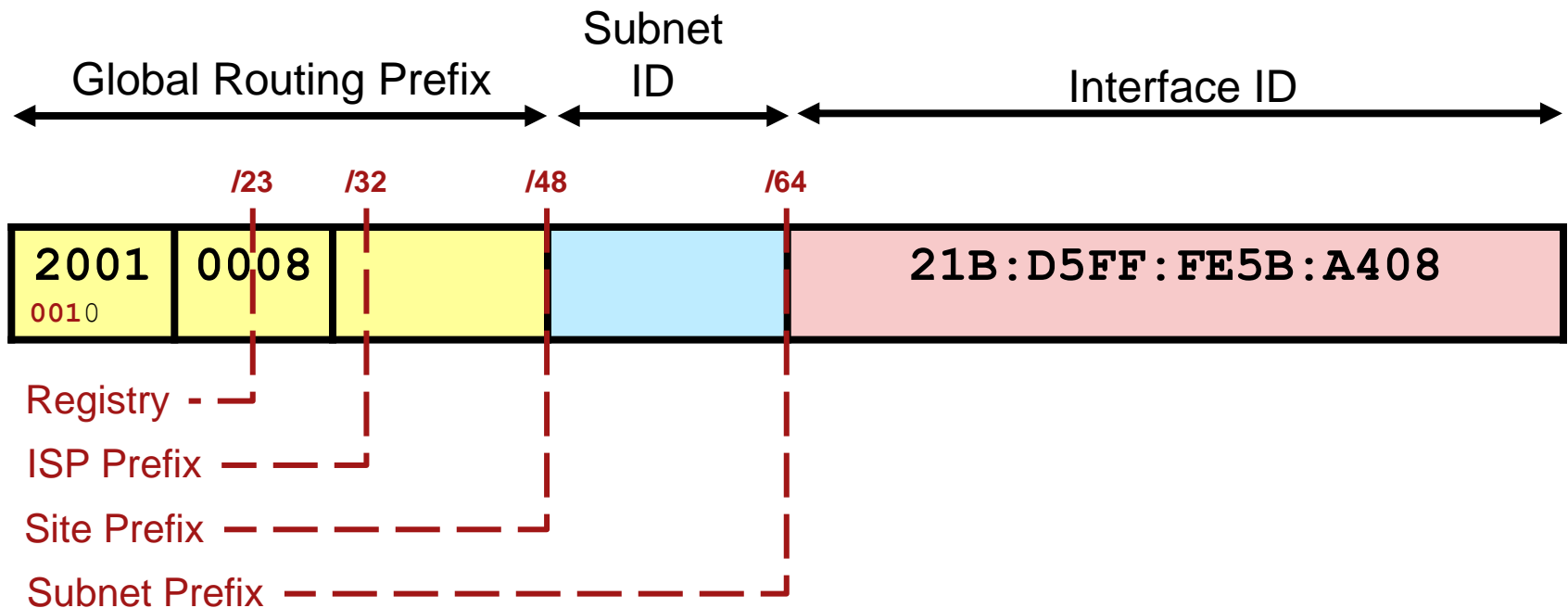
- The global unicast address consists of:
 - **A 48-bit global routing prefix** - assigned by an ISP and is derived from a /32 ISP prefix
 - **A 16-bit subnet ID** - may be use internally by an organization to subdivide its networks
 - **A 64-bit interface ID**





IPv6 Global Unicast Address

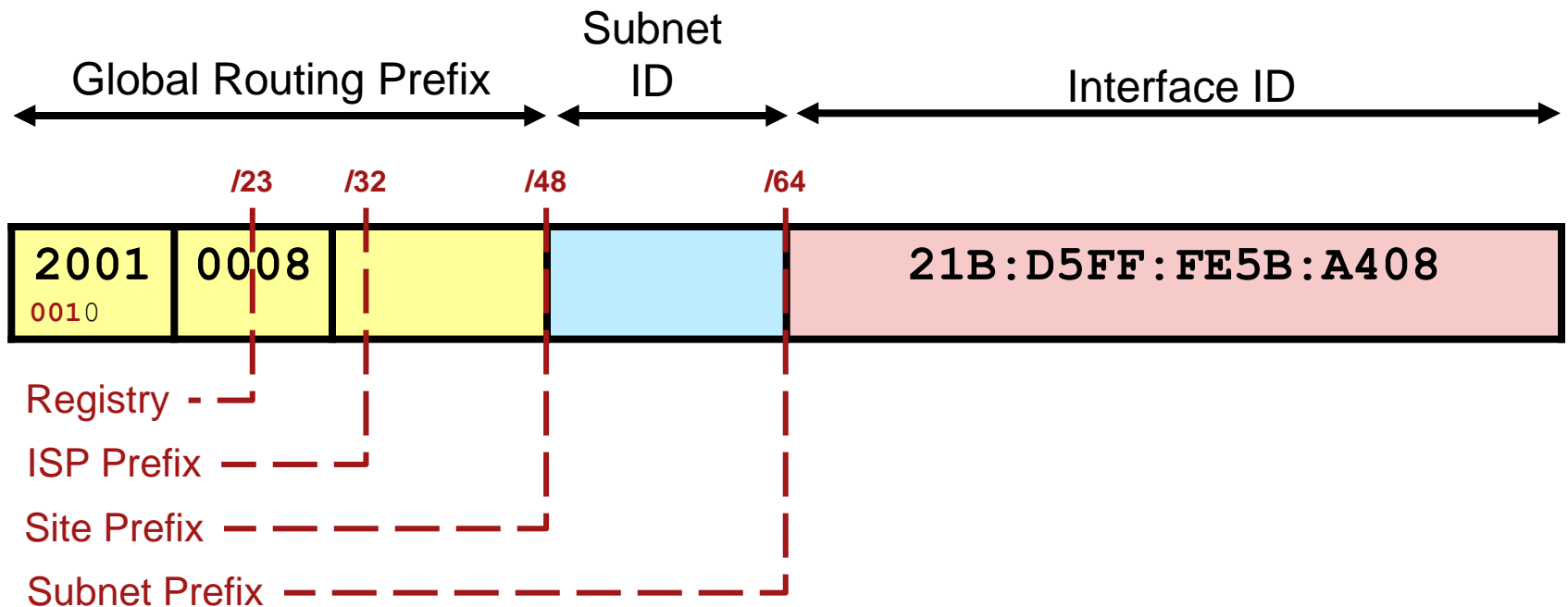
- The current IANA global routing prefix uses the range that starts with binary 0010 (2000::/3).





IPv6 Global Unicast Address

- The subnet ID can be used by an organization to create their own local addressing hierarchy.





IPv6 Global Unicast Address Example

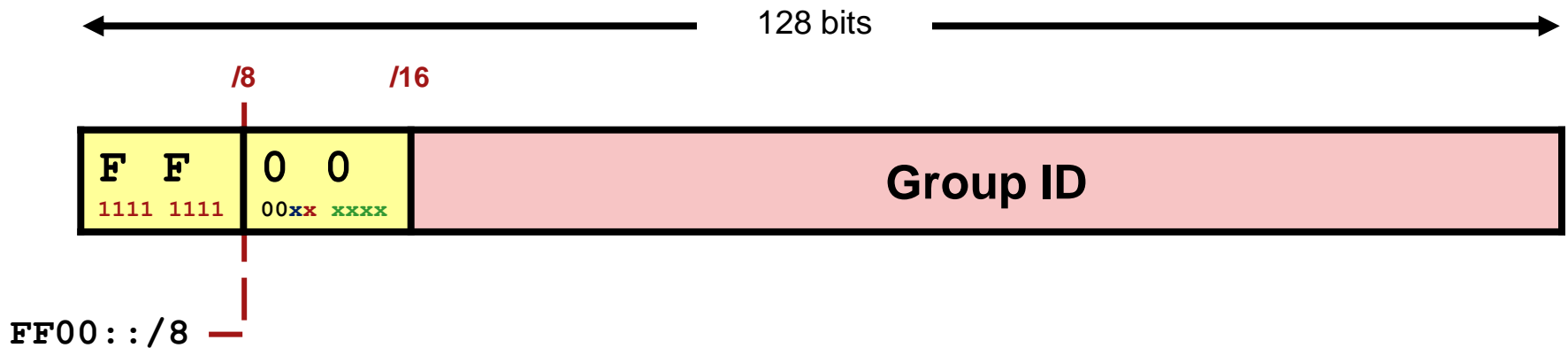
```

R1# show ipv6 interface loopback 100
Loopback100 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::222:55FF:FE18:7DE8
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:8:85A3:4290:222:55FF:FE18:7DE8, subnet is 2001:8:85A3:4290::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF18:7DE8
  MTU is 1514 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is not supported
  ND reachable time is 30000 milliseconds (using 31238)
  Hosts use stateless autoconfig for addresses.
R1#
  
```




IPv6 Multicast Addresses

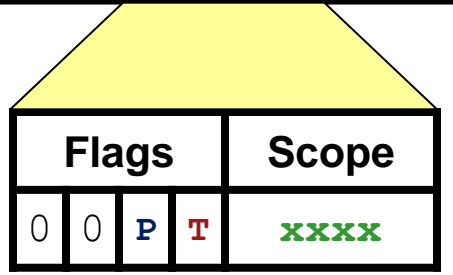
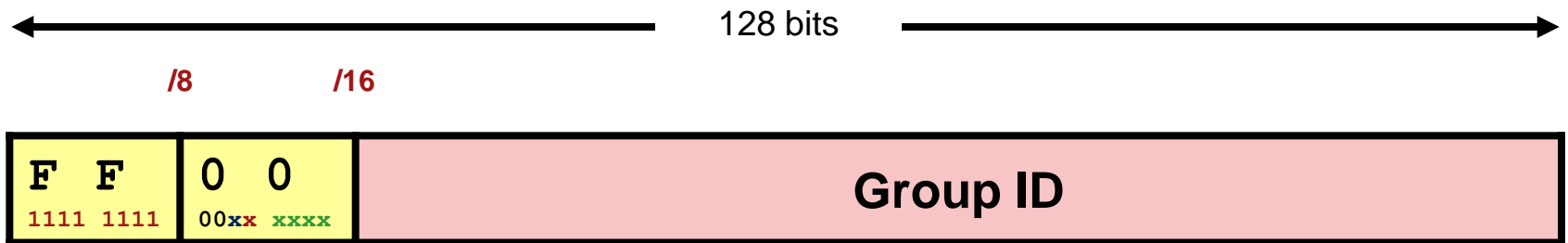
- Multicasting is at the core of many IPv6 functions and is a replacement for the broadcast address.
- They are defined by the prefix **FF00::/8**.
- An interface may belong to any number of multicast groups.





IPv6 Multicast Address

- The second octet of the address contains the prefix and transient (lifetime) flags, and the scope of the multicast address.



← 8 bits →

Flags:

- **P** = Prefix for unicast-based assignments
- **T** = **0** if permanent, **1** if temporary

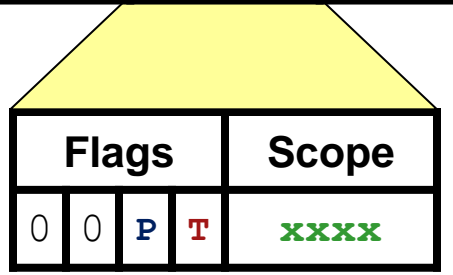
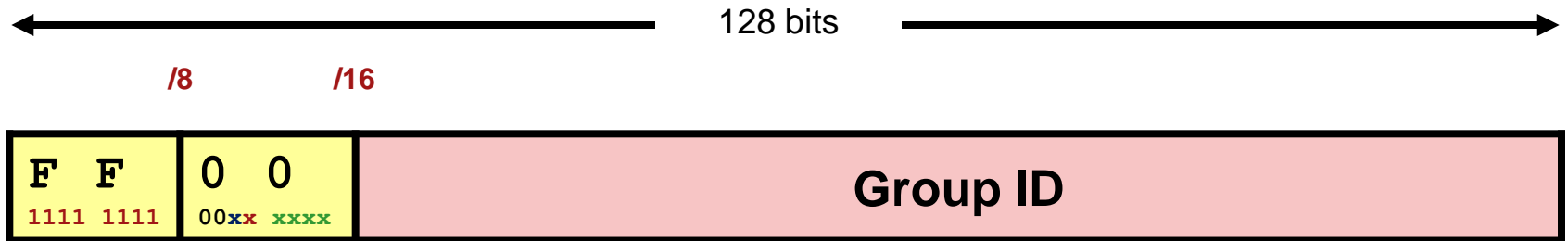
Scope:

- **1** (0001) = Node
- **2** (0010) = Link
- **5** (0101) = Site
- **8** (1000) = Organization
- **E** (1110) = Global



IPv6 Multicast Address

- The multicast addresses **FF00::** to **FF0F::** are permanent and reserved.



← 8 bits →

Flags:

- P** = Prefix for unicast-based assignments
- T** = **0** if permanent, **1** if temporary

Scope:

- 1** (0001) = Node
- 2** (0010) = Link
- 5** (0101) = Site
- 8** (1000) = Organization
- E** (1110) = Global



Reserved IPv6 Multicast Addresses

Reserved Multicast Address	Description
FF02::1	<ul style="list-style-type: none"> All nodes on a link (link-local scope).
FF02::2	<ul style="list-style-type: none"> All routers on a link.
FF02::9	<ul style="list-style-type: none"> All routing information protocol (RIP) routers on a link.
FF02::1:FFxx:xxxx	<ul style="list-style-type: none"> All solicited-node multicast addresses used for host autoconfiguration and neighbor discovery (similar to ARP in IPv4). The xx:xxxx is the far right 24 bits of the corresponding unicast or anycast address of the node.
FF05::101	<ul style="list-style-type: none"> All Network Time Protocol (NTP) servers.



IPv6 Multicast Address Example

```

R1# show ipv6 interface loopback 100
Loopback100 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::222:55FF:FE18:7DE8
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:8:85A3:4290:222:55FF:FE18:7DE8, subnet is 2001:8:85A3:4290::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF18:7DE8
  MTU is 1514 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is not supported
  ND reachable time is 30000 milliseconds (using 31238)
  Hosts use stateless autoconfig for addresses.
R1#

```



Solicited-Node Multicast Addresses

- The solicited-node multicast address (FF02::1:FF) is used for:
 - Neighbor discovery (ND) process
 - Stateless address autoconfiguration
- The Neighbor discovery (ND) process is used to:
 - Determine the local-link address of the neighbor
 - Determine the routers on the link and default route
 - Keep track of neighbor reachability
 - Send network information from routers to hosts



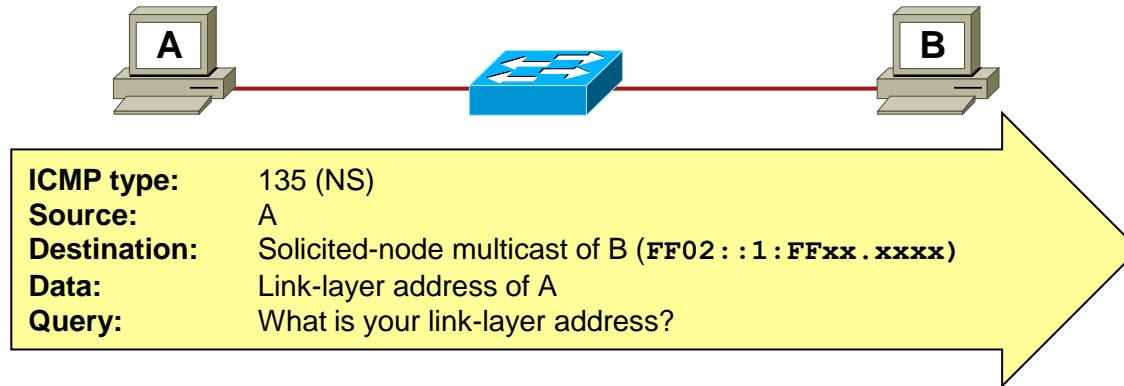
Neighbor Discovery ICMPv6 Packet Types

- Neighbor Discovery uses four ICMPv6 packet types

ICMPv6 Message	Type	Description
Neighbor Solicitation (NS)	135	<ul style="list-style-type: none"> Sent by a host to determine the link-layer address of a neighbor. Used to verify that a neighbor is still reachable. An NS is also used for Duplicate Address Detection (DAD).
Neighbor Advertisement (NA)	136	<ul style="list-style-type: none"> A response to a NS message. A node may also send unsolicited NA to announce a link-layer address change.
Router Advertisement (RA)	134	<ul style="list-style-type: none"> RAs contain prefixes that are used for on-link determination or address configuration, a suggested hop limit value and MTU value. RAs are sent either periodically, or in response to a RS message.
Router Solicitation (RS)	133	<ul style="list-style-type: none"> When a host is booting it sends out an RS requesting routers to immediately generate an RA rather than wait for their next scheduled time.



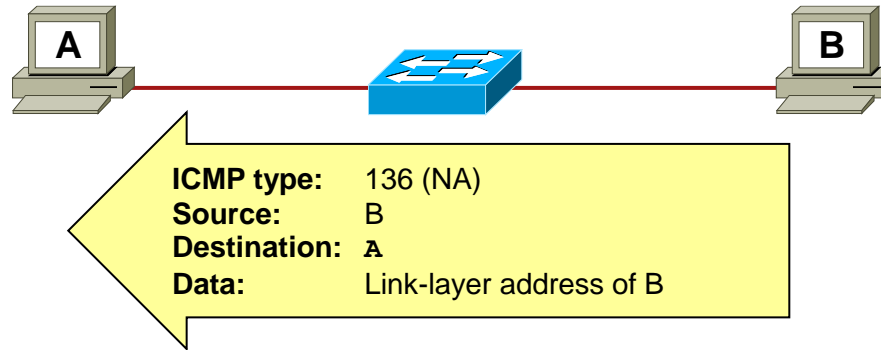
Neighbor Solicitation Example



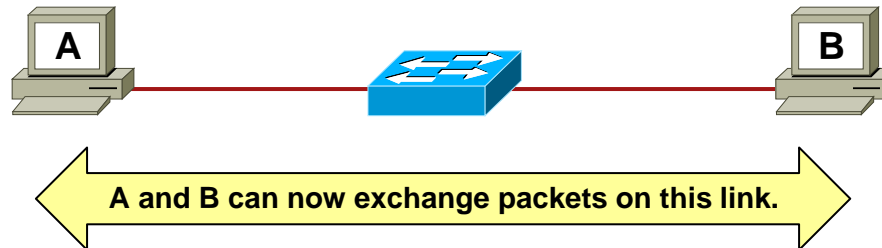
- ICMPv6 Neighbor Solicitation (NS) is similar to IPv4 ARP.
- For Host A to send a packet to Host B it needs the MAC address of Host B.



Neighbor Advertisement Example



- Each destination node that receives the NS responds with an ICMPv6 message type 136, NA, including Host B.





Stateless Address Autoconfiguration (SLAAC)

- Every IPv6 system is able to build its own unicast global address.
 - Enables new devices to easily connect to the Internet.
 - No configuration or DHCP server is required.
- **IPv6 Router** - sends network info on local link.
 - IPv6 prefix
 - Default IPv6 route
- **IPv6 Hosts** - listen on local link and configure themselves.
 - IP Address (EUI-64 format)
 - Default route



Stateless Address Autoconfiguration

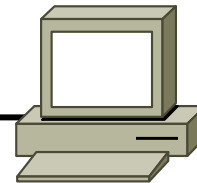
An IPv6 address must be configured on the router gateway interface.

IPv6 Router



Local Link

IPv6 Host



MAC Address
00:14:BF:7A:3C:E5

RA

Router sends network info
(IPv6 Prefix and Default IPv6 Route)

Autoconfiguration Address
(IPv6 Prefix + Link-Layer EUI Address)



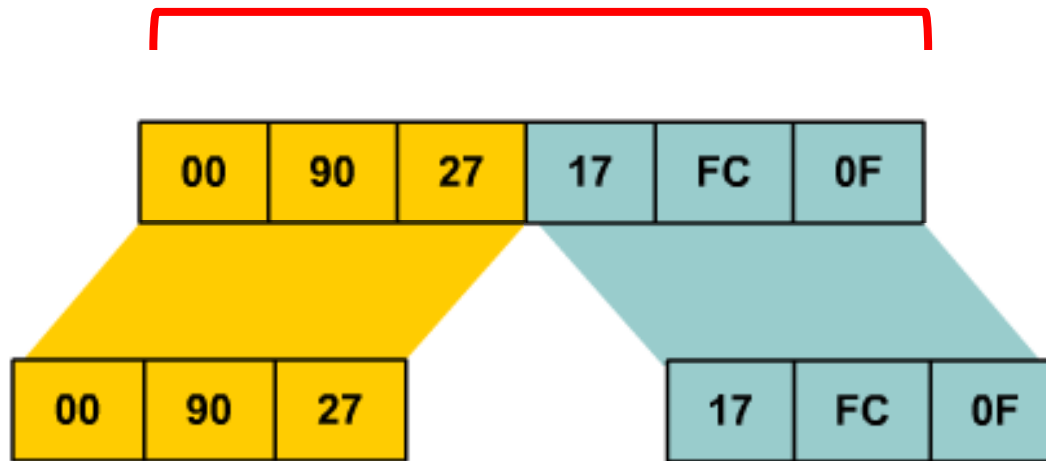
Ethernet EUI-64 IPv6 Addresses

- The first 64 bits are the network portion of the address and are statically specified or learned via SLAAC.
- The interface ID (second 64-bits) is the host portion of the address and is automatically generated by the router or host device.
- The interface ID on an Ethernet link is based on the 48-bit MAC address of the interface with an additional 16-bit **0xFFFE** inserted in the middle of the MAC address.



EUI-64 IPv6 Interface Identifier

48-bit MAC Address



64-bit IPv6 EUI-64 Interface ID



Stateless Autoconfiguration Process



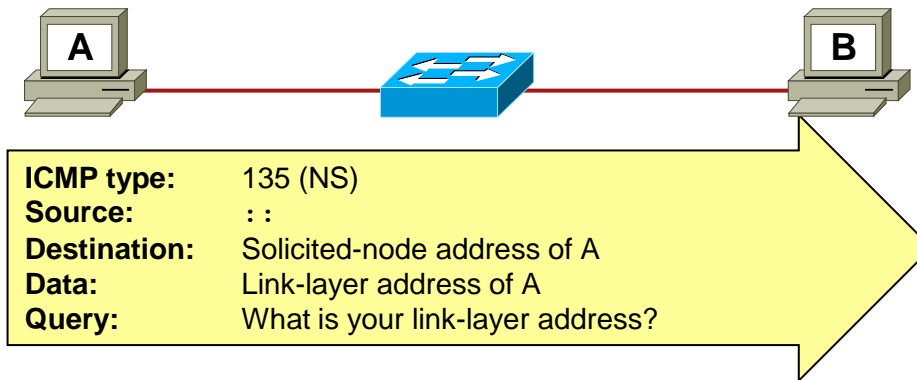
ICMP type: 133 (RS)
Source: ::
Destination: All routers multicast address (FF02::2)
Query: Please send RA



ICMP type: 134 (RA)
Source: R1 link-local address
Destination: All nodes multicast address (FF02::1)
Data: Options, prefixes, lifetime, ...



Stateless Autoconfiguration Process



- Host A creates an IPv6 address using the RA supplied by the router.
- Host A verifies that it's new IPv6 address is unique using DAD process.

IPv6 Subnetting and Aggregation





IPv6 Subnetting Overview

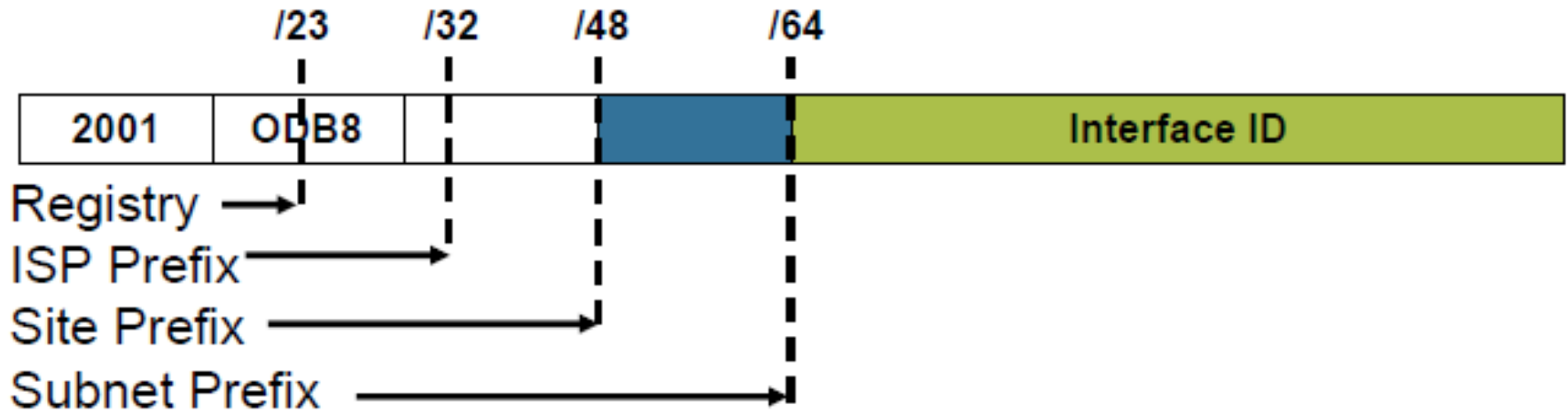
- Not the same as IPv4
- IPv6 does **NOT use subnet masks**
- CIDR notation is used
 - IPv6 address is in Hex
 - Network mask is in decimal
- Number of subnet bits set to 1 define network prefix
- All other bits are for nodes
- There are no reserved addresses (network or broadcast)

2001:25:12:AB12:3456:DFB5:712:45FF/64



Prefix Length, Allocation of Bits

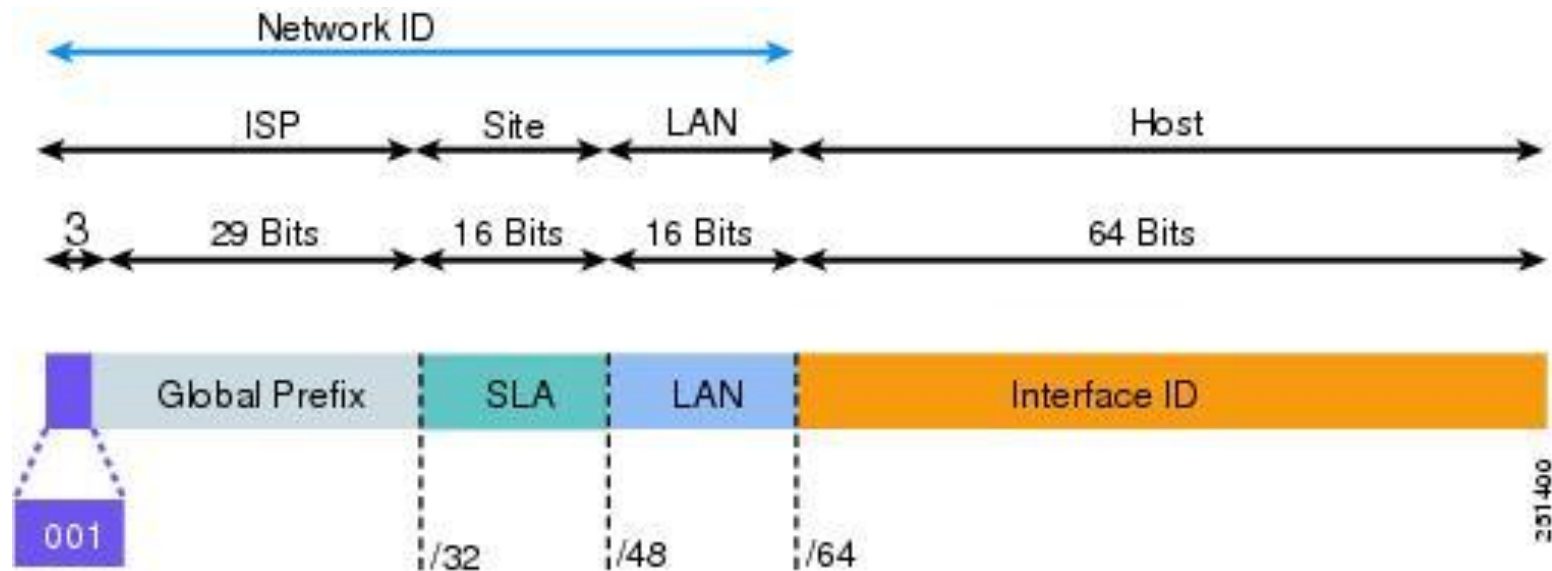
- Example: 2001:DB8:0:2F00:2AA:FF:FE28:9C5A/64
- Prefix length (total number of network bits) is 64
- 16 subnet bits allow 65,535 LANs
- Usually 64 bits are used for hosts in IPv6





IPv6 Subnetting with Global Unicast Addresses

- The global routing prefix is assigned to a service provider by IANA (/32).
- The site level aggregator (SLA) is assigned by the ISP (/48).
- The LAN ID represents individual subnets within the customer site and is administered by the customer (/64).

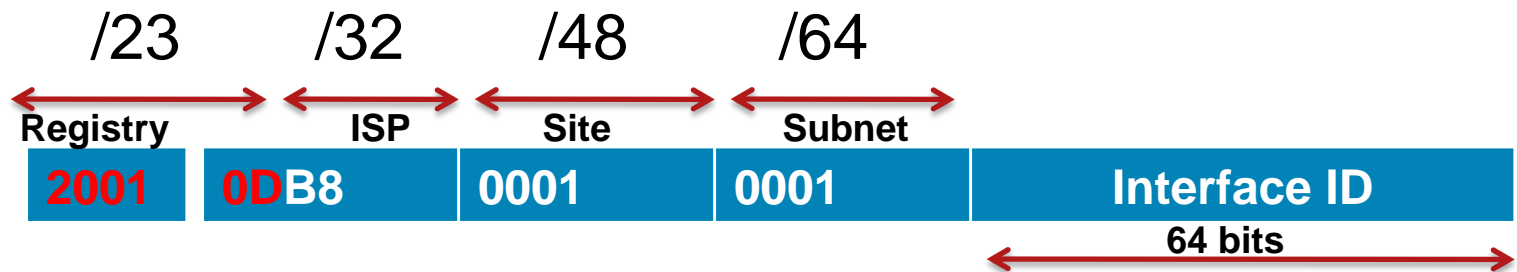




IPv6 Subnetting with Global Unicast Addresses

■ Default Subnets

- /23 Registry
- /32 ISP Prefix
- /48 Site Prefix
 - Bits 49 to 64 are for subnets
 - $2^{16} = 65,535$ subnets available
- /64 Default Subnet prefix
 - Bits 65 to 128 for Hosts
 - Host bits are either statically assigned, EUI-64, DHCP or random number generated.





IPv6 Subnetting with Global Unicast Addresses

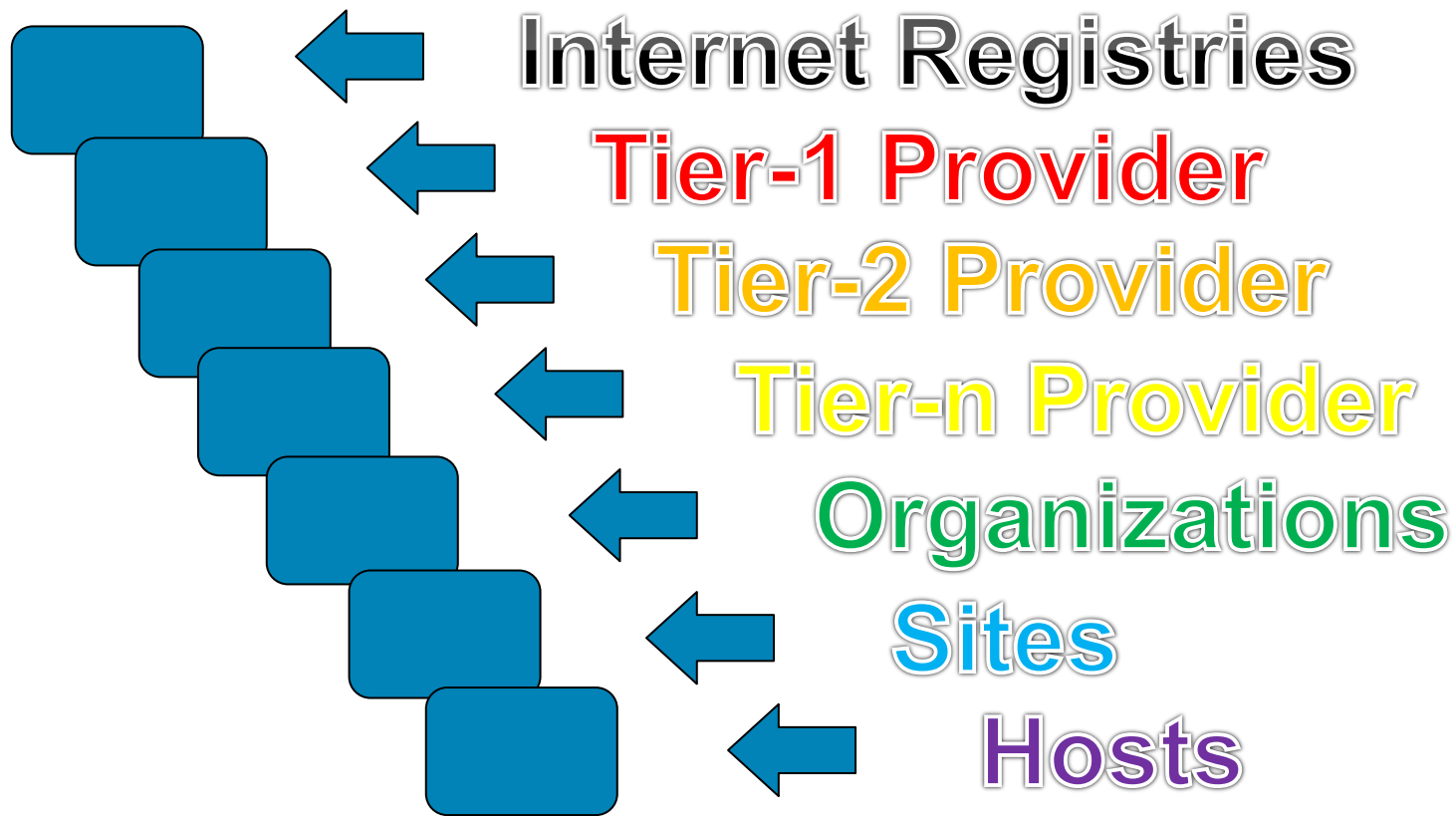
IPv6 prefix	# of Subnets	# of Hosts
2001:520:1:1::3FFF /128	1	1
2001:520:0:1:: /64 (default prefix for subnet)	1	2 ⁶⁴
2001:520:0:: /48 (default Site prefix)	2 ¹⁶	2 ⁶⁴ per subnet

- 61 Global Network bits and 64 Host bits
- No more “bit borrowing” as with IPv4
- 2⁶⁴ hosts possible in a single broadcast domain
- Autoconfiguration will take care of most of them
- VLANs become the method of isolation



IPv6 Address Hierarchy

- Large address space
- Allows for multiple levels





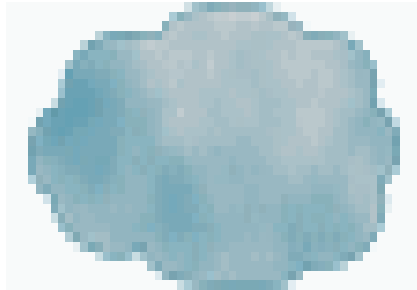
IPv6 Address Aggregation

- Large prefix assigned to an organization
 - Can handle even the largest networks
- ISPs summarize routes
 - All customer prefixes into one prefix
 - Make it available to the Internet
- Aggregation provides:
 - Efficient routing
 - Scalable routing
 - Fewer routes in global IPV6 routing table



Aggregation Example

Global Routing Table
2001:051A::/35



ISP A
AS 60000
2001:051A::/35

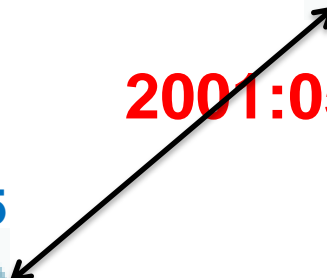


Routing Table
2001:051A:A1::/48 AS 60000
2001:051A:A2::/48 AS 60000

Customer A1



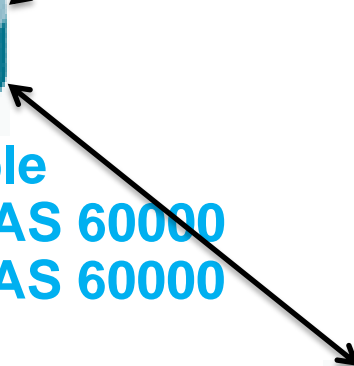
2001:051A:A1::/48



Customer A2



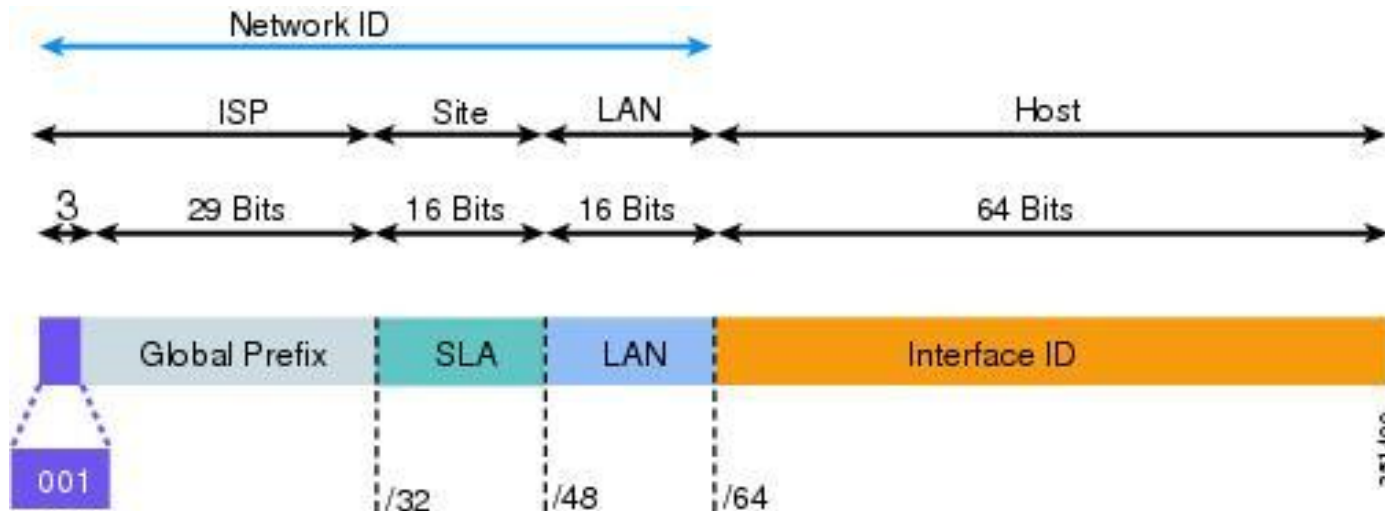
2001:051A:A2::/48





Subnetting Global Addresses

- **2001:05a8::0/32**
- Subnets the ISP can create:
 - $2^{16} = 65,536$
 - 2001:05a8:**0001**::0 – 2001:05a8:**ffff**::0/48





Subnetting Global Addresses

- Customer now has 16 network bits with which to create 2^{16} networks (or 2^8 if ISP used /56)
 - 2001:05a8:0001:0001::0 – 2001:05a8:0001:ffff::0/64
 - 2001:05a8:94ad:0001::0 – 2001:05a8:94ad:ffff::0/64
 - 2001:05a8:b002:0001::0 – 2001:05a8:b002:ffff::0/64
 - 2001:05a8:ffff:0001::0 – 2001:05a8:ffff:ffff::0/64
- For a smaller organization, the ISP might assign a /56 prefix instead of a /64, which would still allow the customer to create 256 subnets ($2^8 = 256$).



Subnetting Global Addresses

- Apply 1 of these /64 prefixes to a router interface
 - 2001:05a8:0001:00a1::0 /64
- Hosts per subnet (a /64 is a single host)
 - 2001:05a8:0001:00a1::1 –
2001:05a8:01a1:00a1:FFFF:FFFF:FFFF:FFFF/64

Implementing IPv6





Verifying IPv6

Command	Description
<pre>show ipv6 interface [brief] [type number] [prefix]</pre>	<p>Displays the status of interfaces configured for IPv6.</p> <ul style="list-style-type: none"> • The brief keyword displays a brief summary. • The prefix keyword displays the IPv6 neighbor discovery prefixes that are configured on a specified interface.
<pre>show ipv6 routers [interface-type interface-number] [conflicts]</pre>	<p>Displays IPv6 router advertisement information received from on-link routers (those locally reachable on the link).</p> <ul style="list-style-type: none"> • The conflicts keyword displays information about routers advertising parameters that differ from the advertisement parameters configured for the specified interface on which the advertisements are received.
<pre>show ipv6 neighbors [interface-type interface-number ipv6- address ipv6-hostname statistics]</pre>	<p>Displays IPv6 neighbor discovery cache information for the specified neighbors.</p> <ul style="list-style-type: none"> • The optional statistics parameter displays neighbor discovery cache statistics.



Troubleshooting IPv6

Command	Description
<pre>debug ipv6 nd</pre>	<p>Displays messages associated with ICMPv6 neighbor discovery.</p> <ul style="list-style-type: none"> • ICMPv6 neighbor discovery is the IPv6 replacement for the IPv4 ARP.
<pre>debug ipv6 packet [access-list access- list-name] [detail]</pre>	<p>Displays information associated with IPv6 packet processing.</p> <ul style="list-style-type: none"> • When an IPv6 access list is specified, only packets permitted by the ACL are displayed. • The <code>detail</code> keyword displays more information.



Enable IPv6 Routing

- Enable the forwarding of IPv6 unicast datagrams.

```
Router(config) #
```

```
ipv6 unicast-routing
```

- This command is required before configuring any form of IPv6 routing (static or dynamic).
- Also required to support autoconfiguration of end devices.
- The **no ipv6 unicast-routing** command disables IPv6 routing capabilities of the router.



Enable CEF for IPv6

- Enable Cisco Express Forwarding (CEF) for IPv6 (CEFv6).

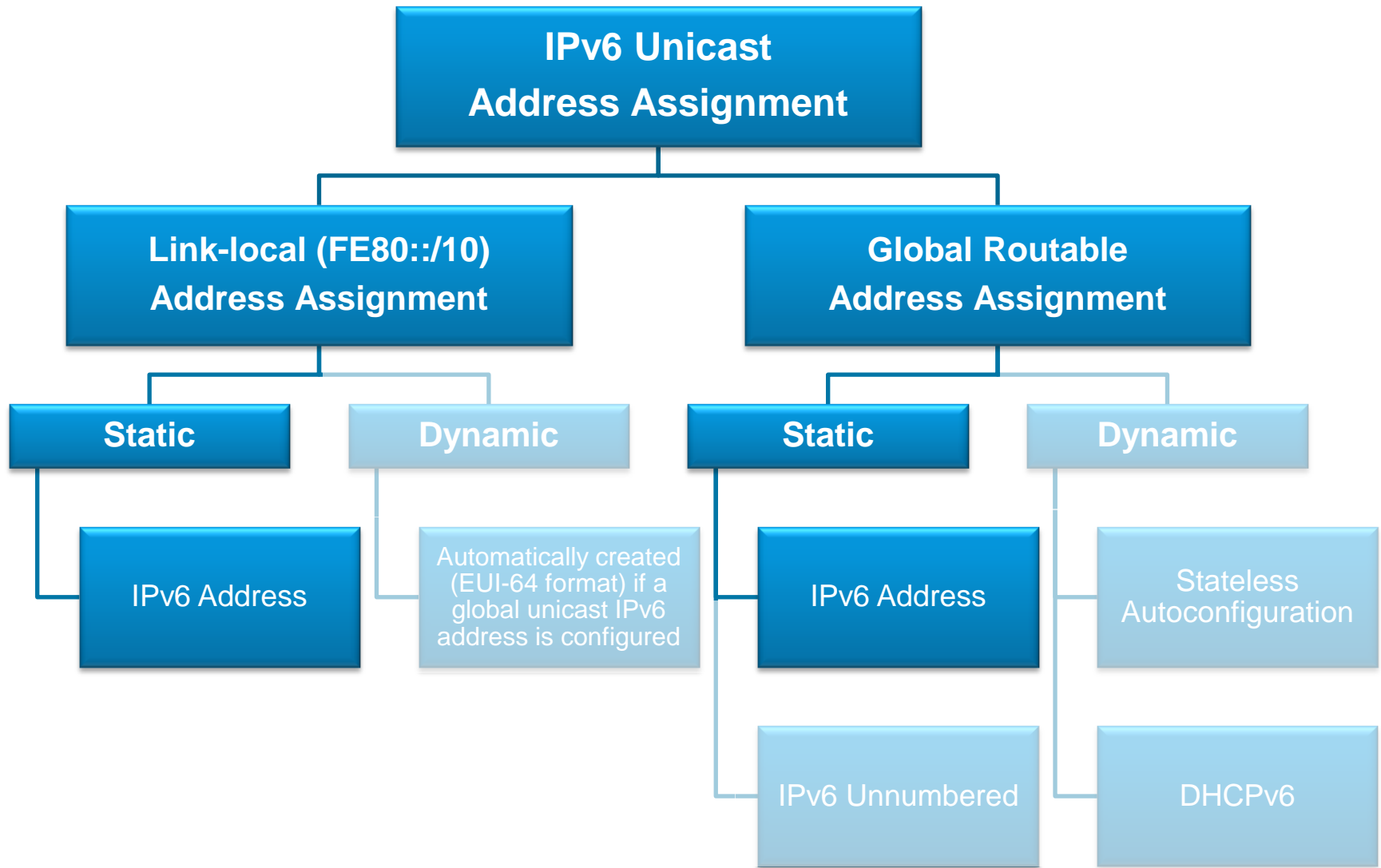
```
Router(config) #
```

```
ipv6 cef
```

- An optional command, CEFv6 is an advanced Layer 3 IP switching technology for the forwarding of IPv6 packets.
 - It is required for some other IPv6 features to operate.
 - When enabled, network entries in the IPv6 routing table are reflected in the Forwarding Information Bases (FIBs).
 - The IPv6 adjacency tables maintain Layer 2 next-hop addresses for all entries that are in each FIB.



Configuring IPv6 Unicast Addresses





Enable IPv6 on an Interface

- Configure an IPv6 address and prefix.

```
Router(config-if) #
```

```
ipv6 address address/prefix-length [link-local | eui-64]
```

- Command is used to statically configure an IPv6 address and prefix on an interface.



Assigning a Link-Local Address



```
R1(config)# interface fa0/0
R1(config-if)# ipv6 address FE80::1 ?
link-local use link-local address
R1(config-if)# ipv6 address FE80::1 link-local
R1(config-if)# end
R1#
```

- Link-local addresses are created:
 - Automatically using EUI-64
 - Manually by specifying an interface ID (as in this example)
- The prefix mask is not required because they are not routed.



Assigning a Static Link-Local Address



```

R1# show ipv6 interface fa0/0
FastEthernet0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::1 [TEN]
  No global unicast address is configured
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
  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
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
R1(config-if)#
  
```



Assigning a Static Global Unicast Address

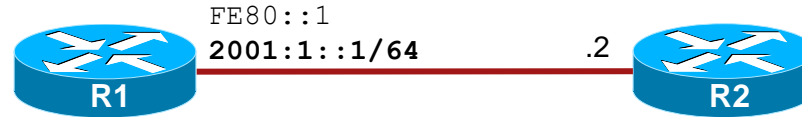


```
R1(config)# ipv6 unicast-routing
R1(config)# interface fa0/0
R1(config-if)# ipv6 address 2001:1::1/64
R1(config-if)#
```

- Global Unicast IPv6 addresses are assigned by omitting the `link-local` parameter.



Assigning a Static Global Unicast Address



```

R1# show ipv6 interface fa0/0
FastEthernet0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::1 [TEN]
Global unicast address(es):
  2001:1::1, subnet is 2001:1::/64 [TEN]
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF00:1
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
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
R1#

```



Configuring an EUI-64 IPv6 Global Address

```

R1# config t
R1(config)# int fa0/1
R1(config-if)# ipv6 add 2001::/64 eui-64
R1(config-if)# do show ipv6 interface fa0/1

FastEthernet0/1 is administratively down, line protocol is down
IPv6 is enabled, link-local address is FE80::211:92FF:FE54:E2A1 [TEN]
Global unicast address(es):
2001::211:92FF:FE54:E2A1, subnet is 2001::/64 [EUI/TEN]
Joined group address(es):
FF02::1
FF02::2
FF02::1:FF54:E2A1
MTU is 1500 bytes

<output omitted>

```



EUI-64 IPv6 Global Address – Example 2

```

R1(config)# interface loopback 100
R1(config-if)# ipv6 address 2001:8:85a3:4289::/64 eui-64

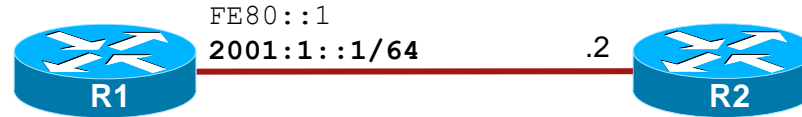
<output omitted>

R1# show ipv6 interface loopback 100
Loopback100 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::21B:D5FF:FE5B:A408
Global unicast address(es):
  2001:8:85A3:4289:21B:D5FF:FE5B:A408, subnet is 2001:8:85A3:4289::/64 [EUI]
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF5B:A408
MTU is 1514 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.

```




Assigning Multiple IPv6 Addresses



```

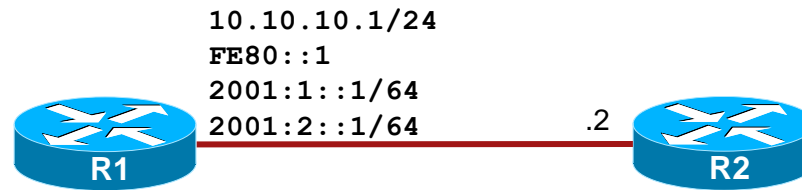
R1(config)# interface fa0/0
R1(config-if)# ip address 10.20.20.1 255.255.255.0
R1(config-if)# ip address 10.10.10.1 255.255.255.0
R1(config-if)# ipv6 address 2001:1::1/64
R1(config-if)# ipv6 address 2002:1::1/64
R1(config-if)# end
R1#

```

- Interfaces can have multiple IPv6 addresses assigned to them. These addresses can be used simultaneously.
- What would happen if we configured 2 different IPv4 addresses and 2 different IPv6 addresses on the same interface?



Assigning Multiple IPv6 Addresses



```

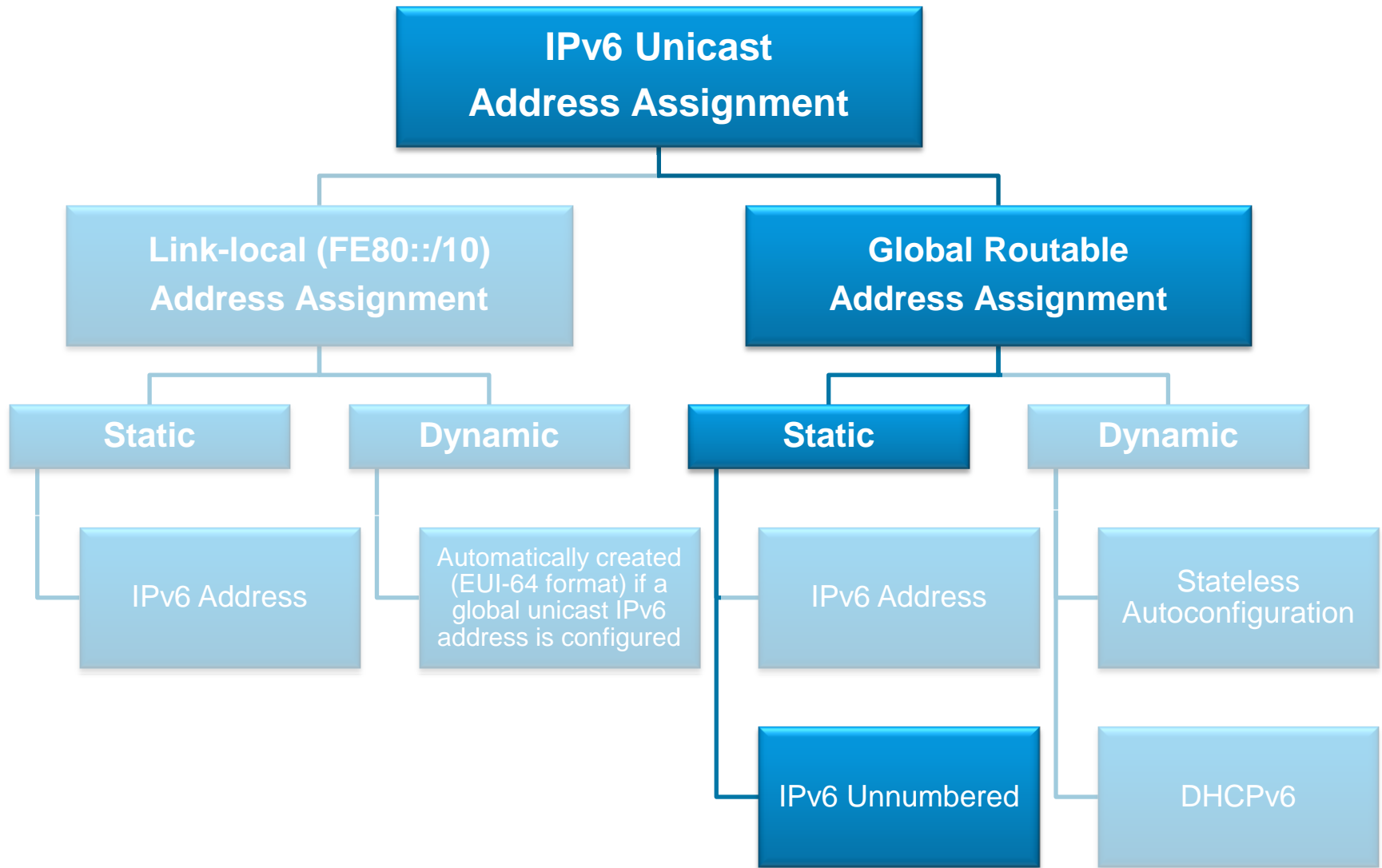
R1# show run interface fa0/0
Building configuration...
Current configuration : 162 bytes
!
interface FastEthernet0/0
ip address 10.10.10.1 255.255.255.0
duplex auto
speed auto
ipv6 address 2001:1::1/64
ipv6 address 2002:1::1/64
ipv6 address FE80::1 link-local
end
R1#

```

- The second IPv4 entry replaced the first entry.
- However, both IPv6 addresses have been assigned to the Fa0/0 interface.



Configuring IPv6 Unnumbered Addresses





Enable IP Unnumbered

- Enable IPv6 on an interface without an explicit IPv6 address.

```
Router(config-if) #
```

```
ipv6 unnumbered interface-type interface-number
```



Assigning IPv6 Unnumbered Interfaces



```

R1(config)# interface loopback 10
R1(config-if)# ipv6 address 2001:1::10/64
R1(config-if)# exit
R1(config)#
R1(config)# interface s0/0/0
R1(config-if)# ipv6 unnumbered loopback 10
R1(config-if)# no shut
R1(config-if)#
  
```



Assigning IPv6 Unnumbered Interfaces

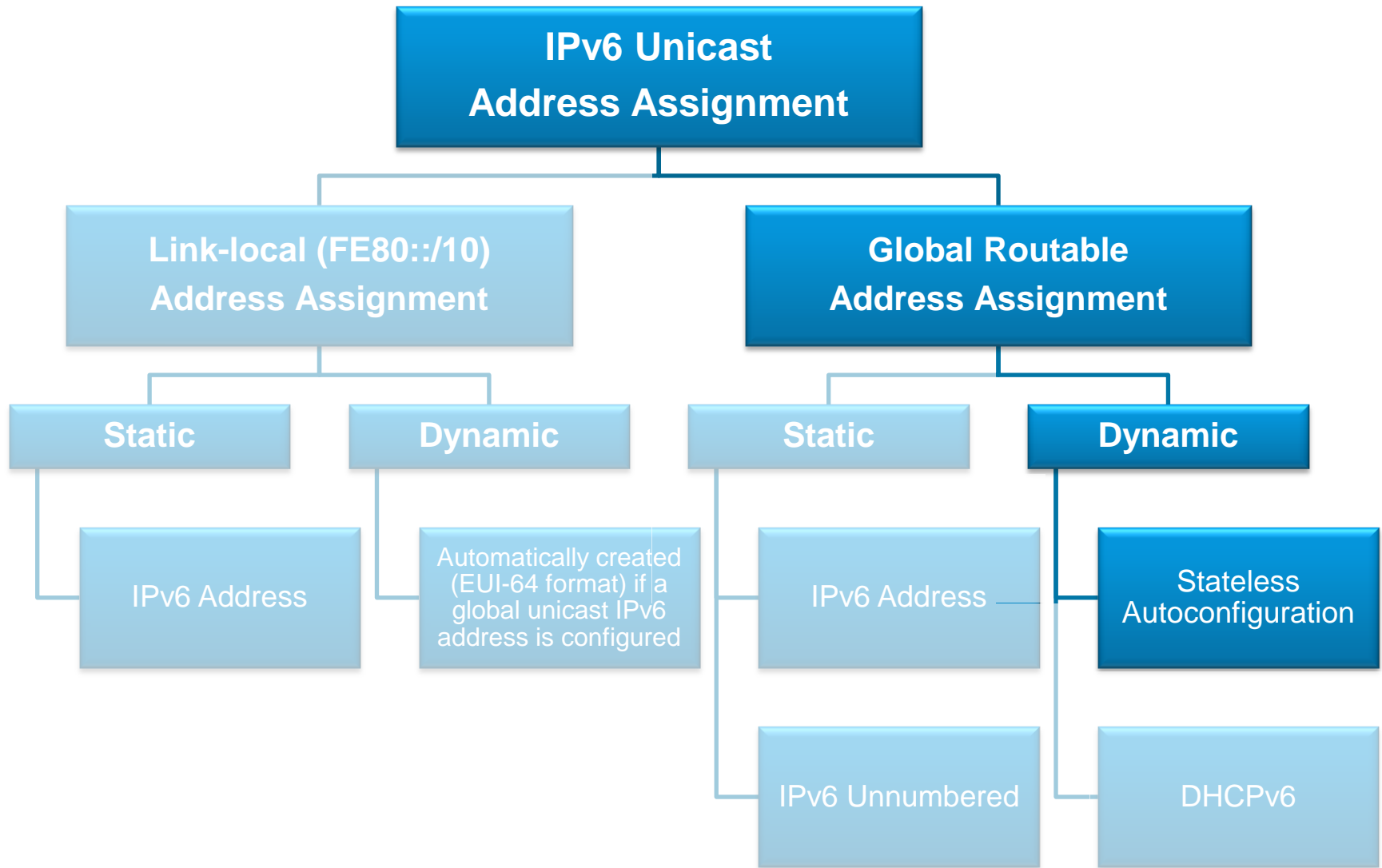


```

R1# show ipv6 interface s0/0/0
Serial0/0/0 is up, line protocol is up
 IPv6 is enabled, link-local address is FE80::222:55FF:FE18:7DE8
No Virtual link-local address(es):
Interface is unnumbered. Using address of Loopback10
No global unicast address is configured
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF18:7DE8
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ICMP unreachable are sent
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds (using 16238)
Hosts use stateless autoconfig for addresses.
R1#
  
```



Configuring IPv6 Unicast Addresses





Enable Autoconfiguration of a Router Interface.

- Enable the automatic configuration of an IPv6 address.

```
Router(config-if) #
```

```
ipv6 address autoconfig [default]
```

- Enables stateless autoconfiguration which:
 - Automatically configures an IPv6 address using the interface.
 - Enables the IPv6 processing on the interface.
- Addresses are configured depending on the prefixes received in RA messages from other routers.



Alter the Neighbor Detection Timeframe

- Alter the neighbor detection parameter.

```
Router(config-if) #
```

```
ipv6 nd reachable-time milliseconds
```

- Specifies the number of milliseconds that a remote IPv6 node is considered reachable.
- Enables a router to detect unavailable neighbors more quickly.



Statically Add a Neighbor

- Add a neighbor router to the neighbor discovery cache.

Router(config) #

```
ipv6 neighbor ipv6-address interface-type interface-  
number hardware-address
```



Router EUI-64 Autoconfig Example

Only the network part of the address is supplied in the `ipv6 address` command

```
R1 (config)# ipv6 unicast-routing
```

```
R1 (config)# int fa0/1
```

```
R1 (config-if)# ipv6 addr 2001:db8::/64 eui-64
```

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

```
R1 (config-if)# no shut
```



Router Interface EUI-64 Example

Router's fa0/1 interface generates its link-local address and global unicast address

-

```
Router#sho ipv6 int bri
FastEthernet0/0      [administratively down/down]
FastEthernet0/1      [up/up]
FE80::201:42FF:FE44:3C02
2001:DB8::201:42FF:FE44:3C02
```



Good Practice in IPv6 Addressing

- Hosts should have globally routable addresses created with stateless autoconfiguration
 - Use 2001 prefix
 - Use /64 EUI-64 to create them
- Serial links between routers should not use globally routable addresses
 - Use FC00 (Unique-local) prefix and static addressing
 - Use a prefix length /64
 - However, the prefix length could also be, for example, /112

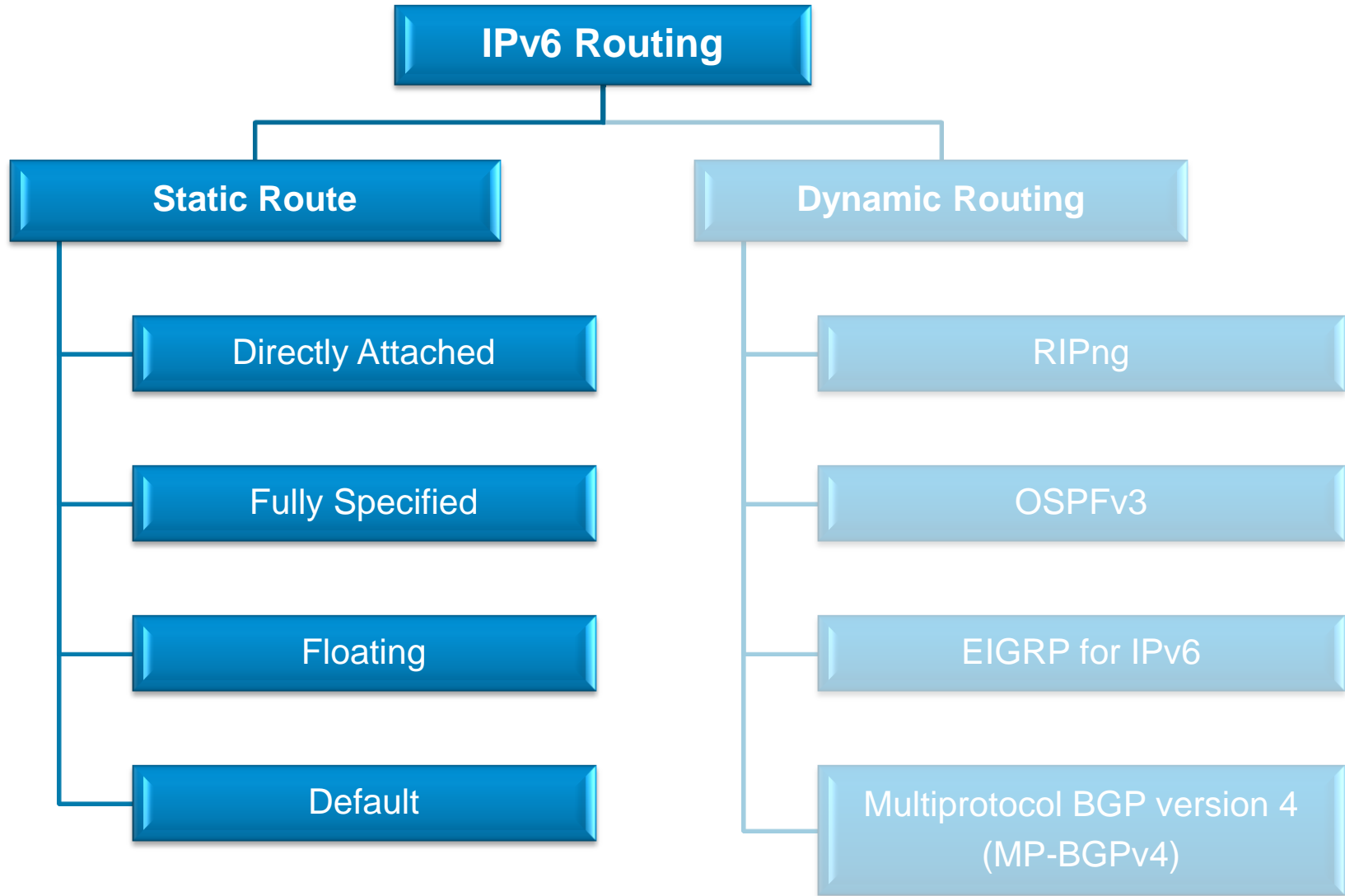
PART 3:

IPv6 Static Routes



Overview

Cisco | Networking Academy®
Mind Wide Open™





IPv6 Static Routes

- Static routes are manually configured and define an explicit path between two networking devices.

- Configuring an IPv6 static route is very similar to IPv4 except that the command is now **ipv6 route**.

- The following must be configured before entering a static IPv6 route:
 - **ipv6 unicast-routing**
 - IPv6 enabled on at least one interface
 - An IPv6 address on that interface.



Complete IPv6 Static Route Syntax

Router(config) #

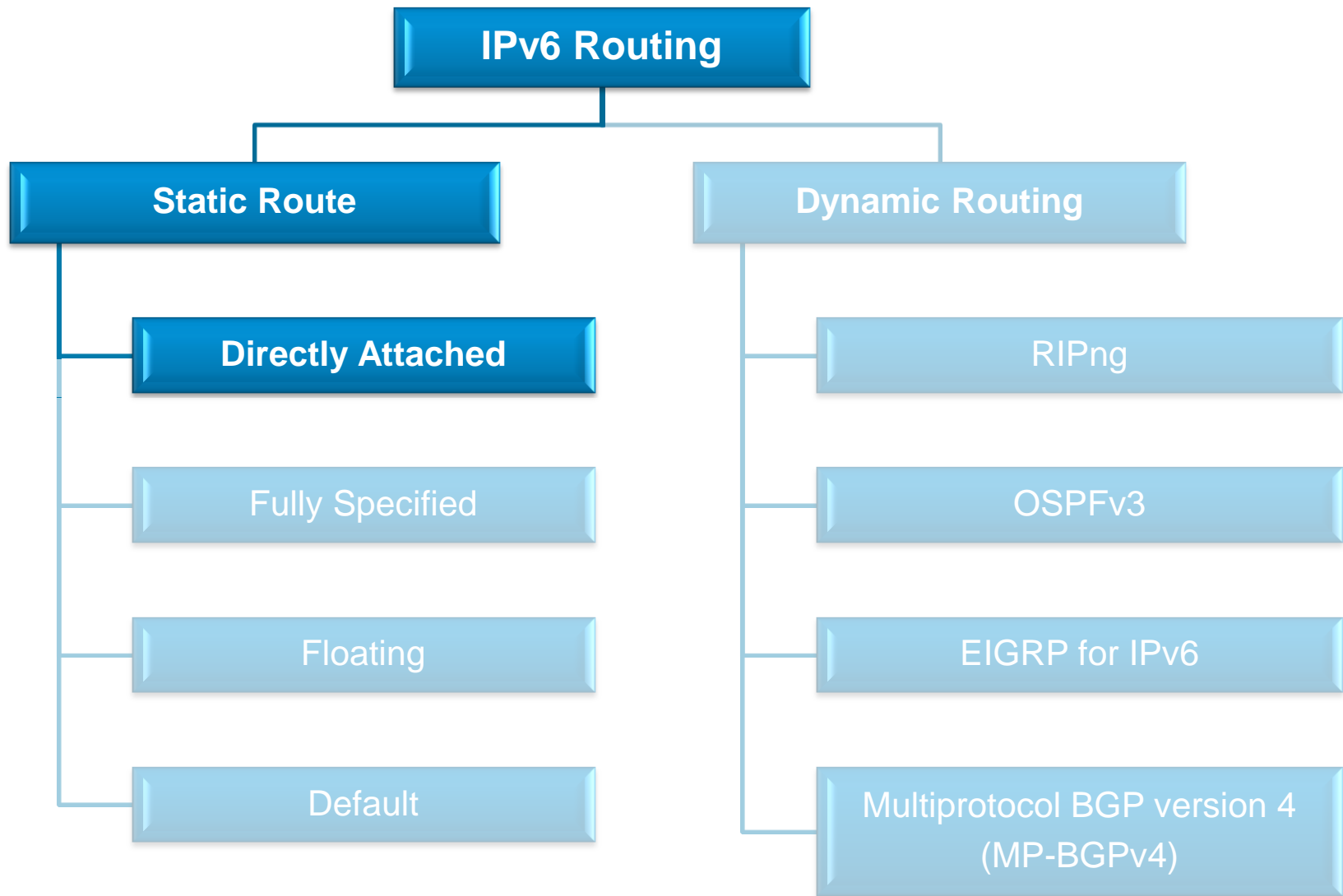
```

ipv6 route                ipv6-prefix/prefix-length
    {ipv6-address | interface-type interface-number [ipv6-address]}

    [administrative-distance]
    
```

- The syntax of the IPv6 command contains more parameters than the IPv4 version.

- The following command parameters are not required to configure directly attached, fully specified, floating and default static routes.
 - Refer to cisco.com for more information on these parameters.





Directly Attached IPv6 Static Route

Router(config)#

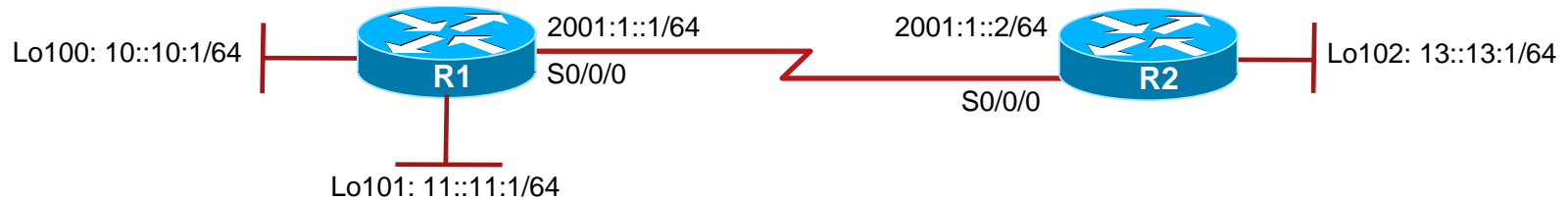
```

ipv6 route ipv6-prefix/prefix-length
  {ipv6-address | interface-type interface-number [ipv6-address]}
  [administrative-distance]
  
```

- A directly attached IPv6 static route is created when specifying only outgoing interface.
- The *ipv6-prefix/prefix-length* parameter identifies the destination IPv6 network and its prefix length.
- The *interface-type interface-number* parameter specifies the interface through which the destination network can be reached.



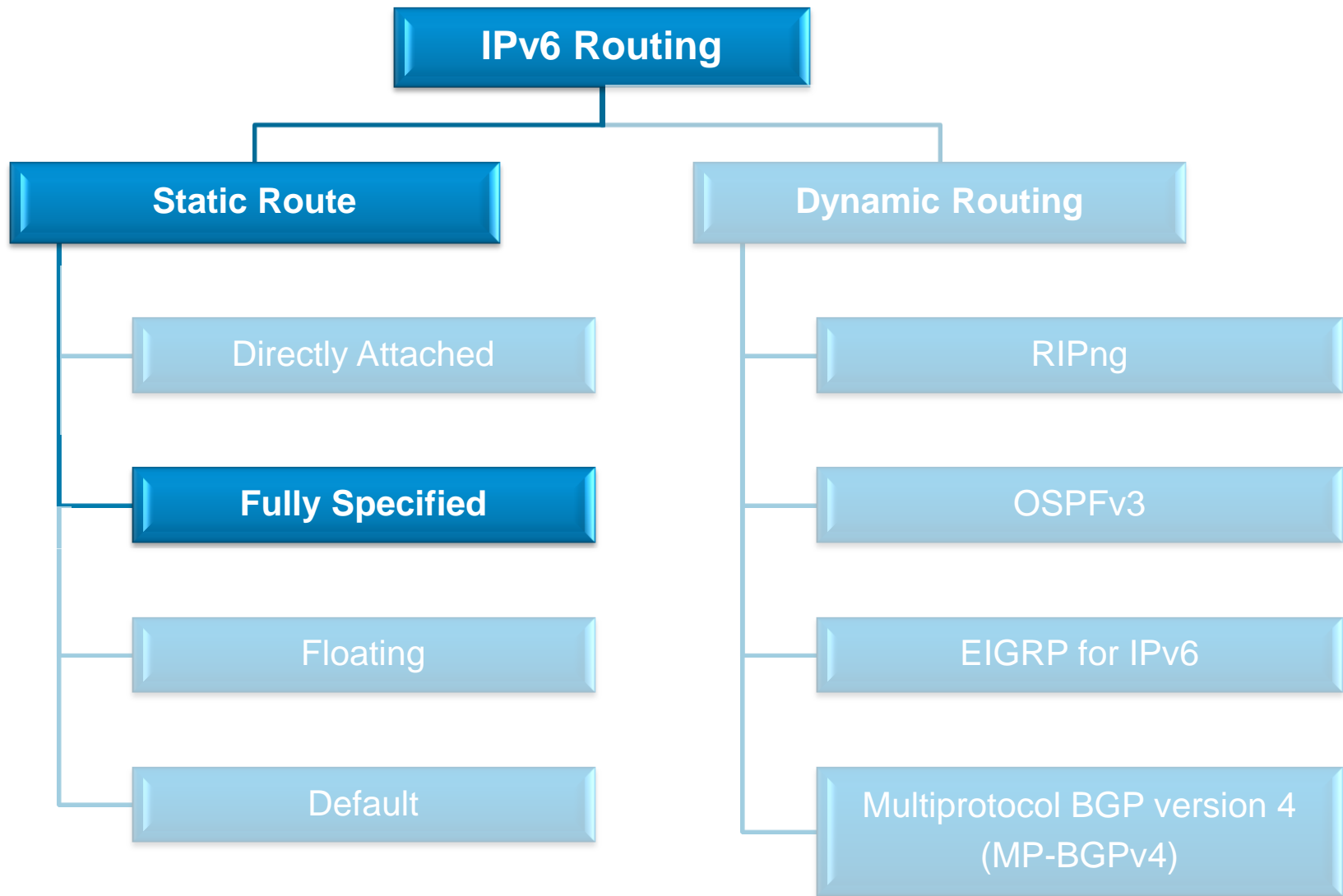
Directly Attached IPv6 Static Route Example



```

R1# config t
R1(config)# ipv6 route 13::/64 s0/0/0
R1(config)# exit
R1# show ipv6 route static
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
S      13::/64 [1/0]
       via ::, Serial0/0/0
R1#
  
```

- A directly attached static route to the 13::13:1/64 network is configured on router R1.





Fully Specified IPv6 Static Route

Router(config) #

```

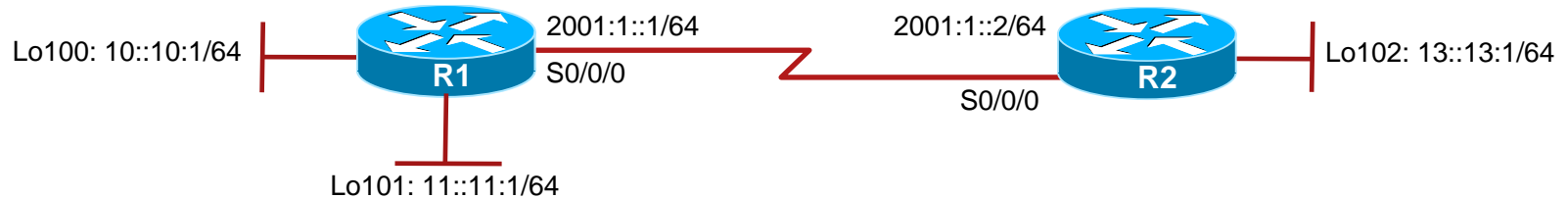
ipv6 route ipv6-prefix/prefix-length
  { ipv6-address | interface-type interface-number [ipv6-address] }
  [administrative-distance]
  
```

- A fully specified static route is created when specifying:
 - The outgoing interface
 - And the next hop IP address.

- This method avoids a recursive lookup.



Fully Specified IPv6 Static Route Example



```

R1# config t
R1(config)# ipv6 route 13::/64 s0/0/0 2001:1::2
R1(config)# exit
R1# show ipv6 route static
IPv6 Routing Table - Default - 8 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   13::/64 [1/0]
    via 2001:1::2, Serial0/0/0
R1#
  
```

- A fully specified static route to the 13::13:1/64 network is configured on router R1.



Note: Recursive IPv6 Static Route

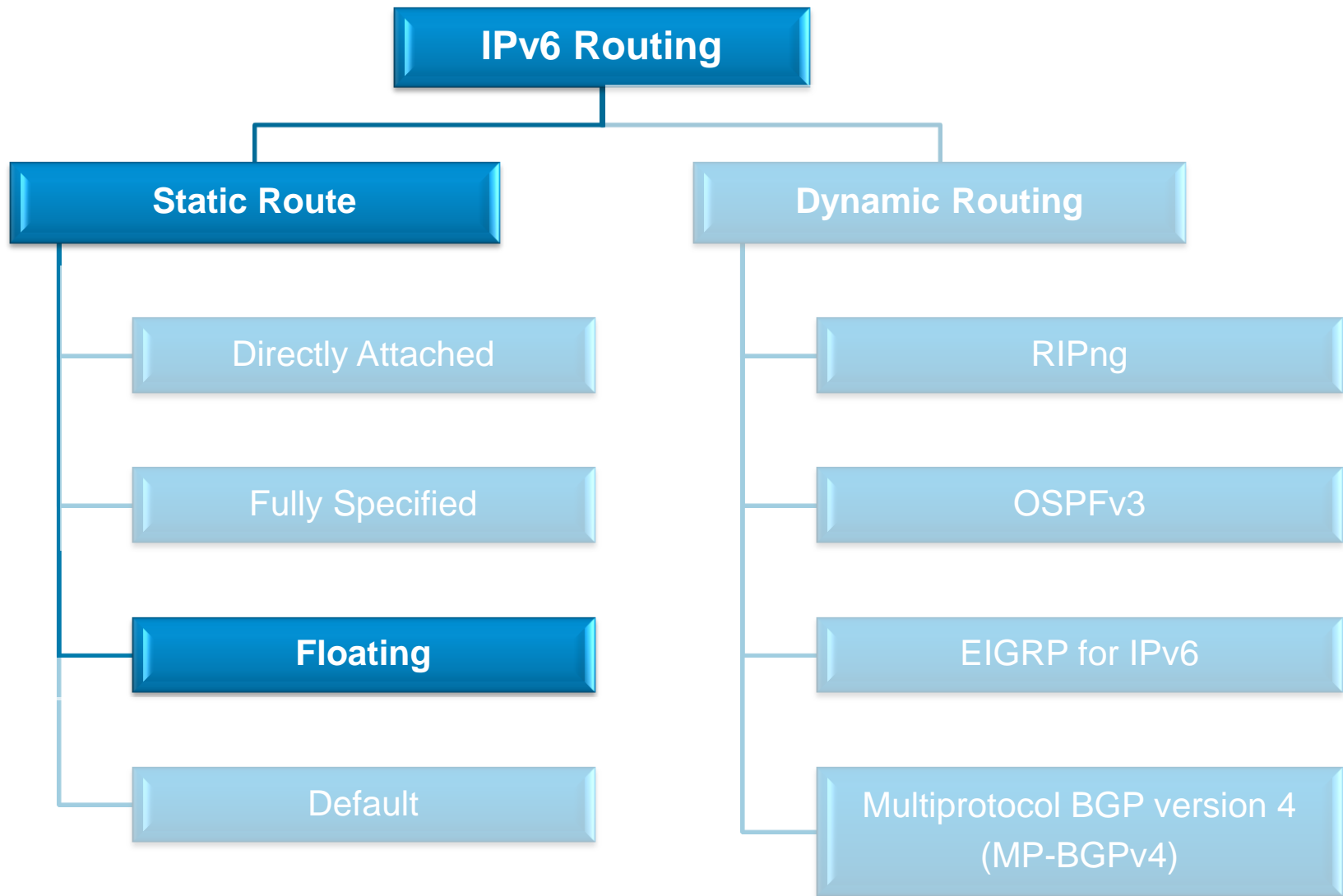
Router(config)#

```

ipv6 route ipv6-prefix/prefix-length
  {ipv6-address | interface-type interface-number [ipv6-address]}
  [administrative-distance]
  
```

- A recursive static route is configured when specifying the next hop IP address of the neighbor.
 - This makes the router perform a second route lookup to resolve the outgoing interface to the specified next hop address.

- Typically, recursive static routes should be avoided.





Floating IPv6 Static Route

Router(config) #

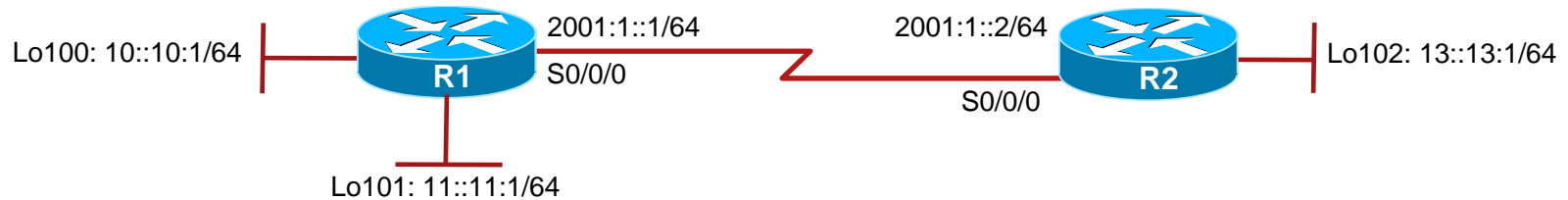
```

ipv6 route ipv6-prefix/prefix-length
  {ipv6-address | interface-type interface-number [ipv6-address]}
  [administrative-distance]
  
```

- A floating static route is usually configured when there are multiple paths to a destination network and a standby backup route is required to support IGP discovered routes.
 - It will only be added to the routing table if the IGP entry is deleted.
- The *administrative-distance* parameter specifies the value of the route, which should be higher than the IGP in the routing table.
 - The default value is 1, which is why static routes have precedence over any other type of route except connected routes.

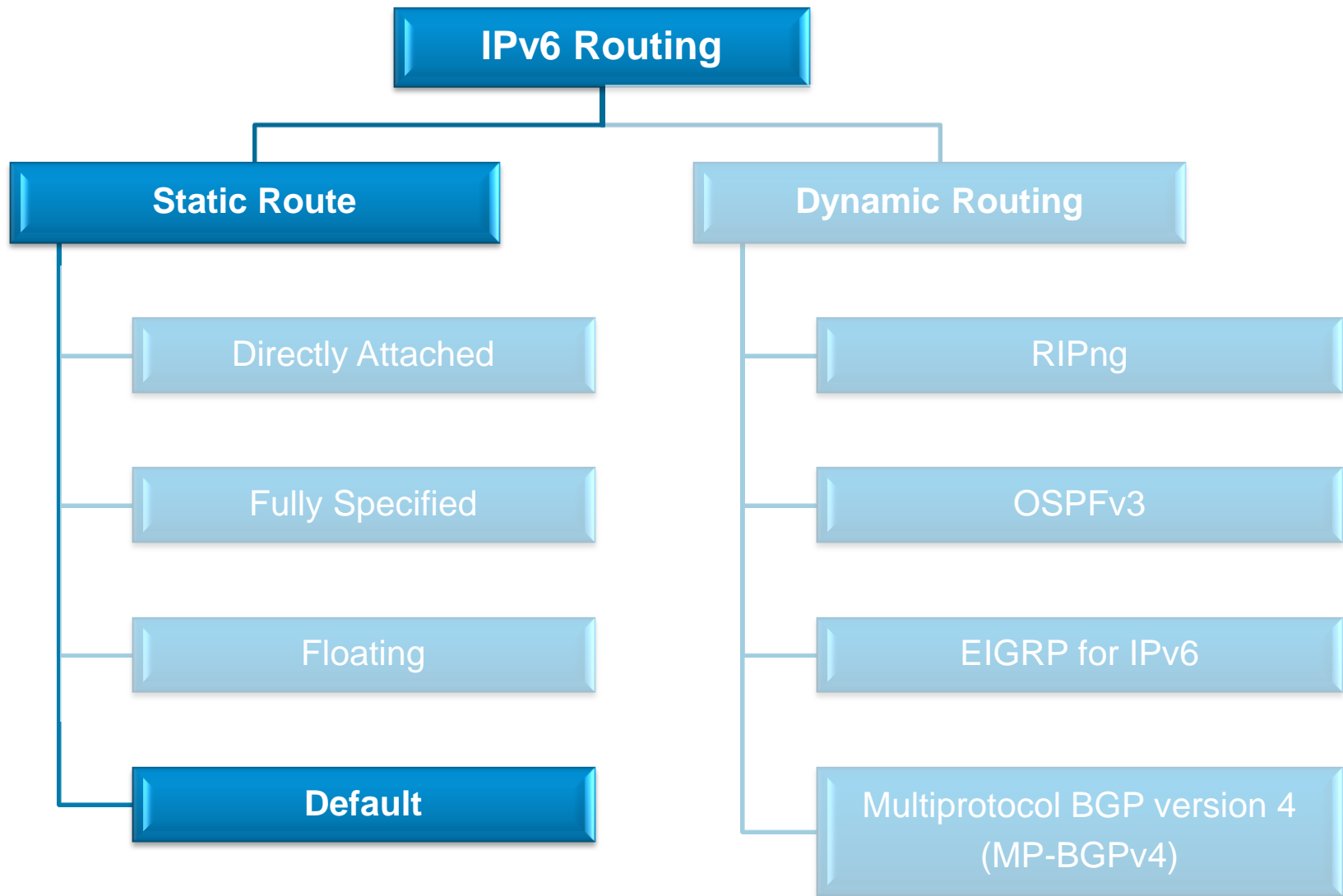


Floating IPv6 Static Route Example



```
R1# config t
R1(config)# ipv6 route 13::/64 130
R1(config)# exit
R1#
```

- For example, R1 is configured with a floating static route specifying an administrative distance of 130 to the R2 LAN.
 - If an IGP already has an entry in the IPv6 routing table to this LAN, then the static route would only appear in the routing table if the IGP entry was removed.





Default IPv6 Static Route

Router(config) #

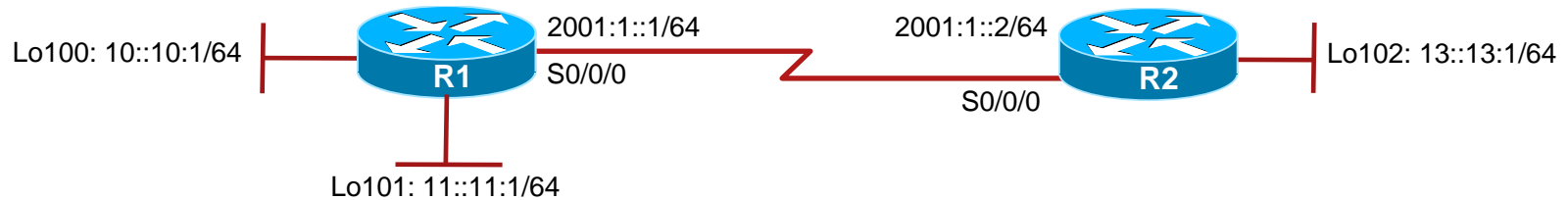
```

ipv6 route ::/0
  {ipv6-address | interface-type interface-number [ipv6-address]}
  [administrative-distance]
  
```

- IPv6 also has a default static route similar to the IPv4 quad zero (0.0.0.0) static default route.
- Instead, the IPv6 command uses the `::/0` notation to specify all networks.



Default IPv6 Static Route Example



```

R2# config t
R2(config)# ipv6 route ::/0 s0/0/0
R2(config)# exit
R2# show ipv6 route static
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
S      ::/0 [1/0]
       via ::, Serial0/0/0
R2#
  
```

- For example, a default static route as specified by the “: : /0” entry is configured on router R2 to reach all other networks connected to R1.



PART 4:

Transitioning IPv4 to IPv6

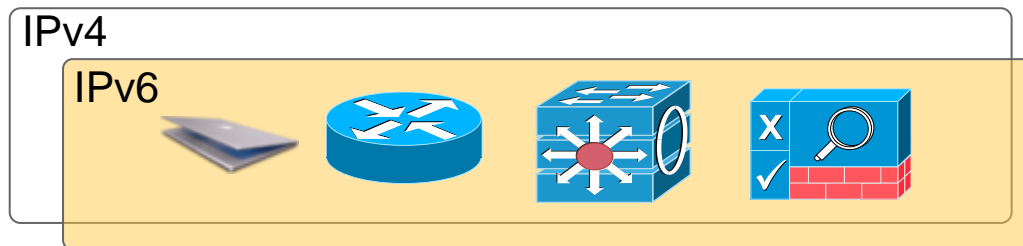


Cisco | Networking Academy®
Mind Wide Open™



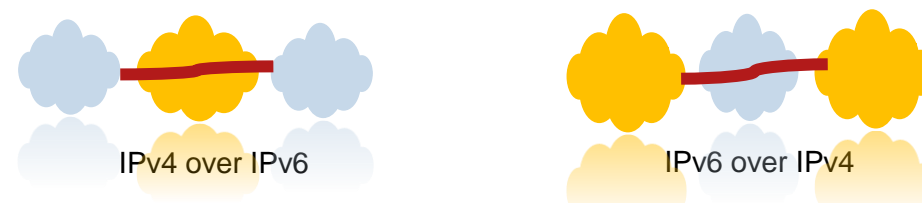
IPv6 Co-existence Solutions

Dual-Stack



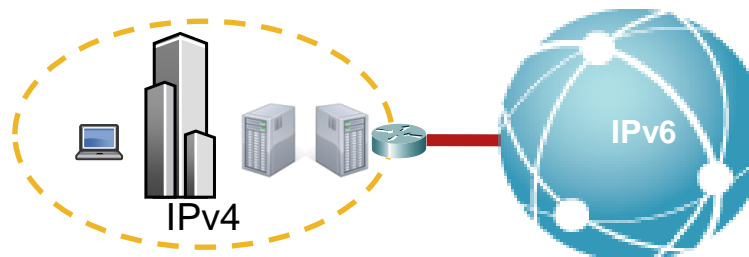
Enterprise Co-existence strategy

Tunneling Services



Connect Islands of IPv6 or IPv4

Translation Services



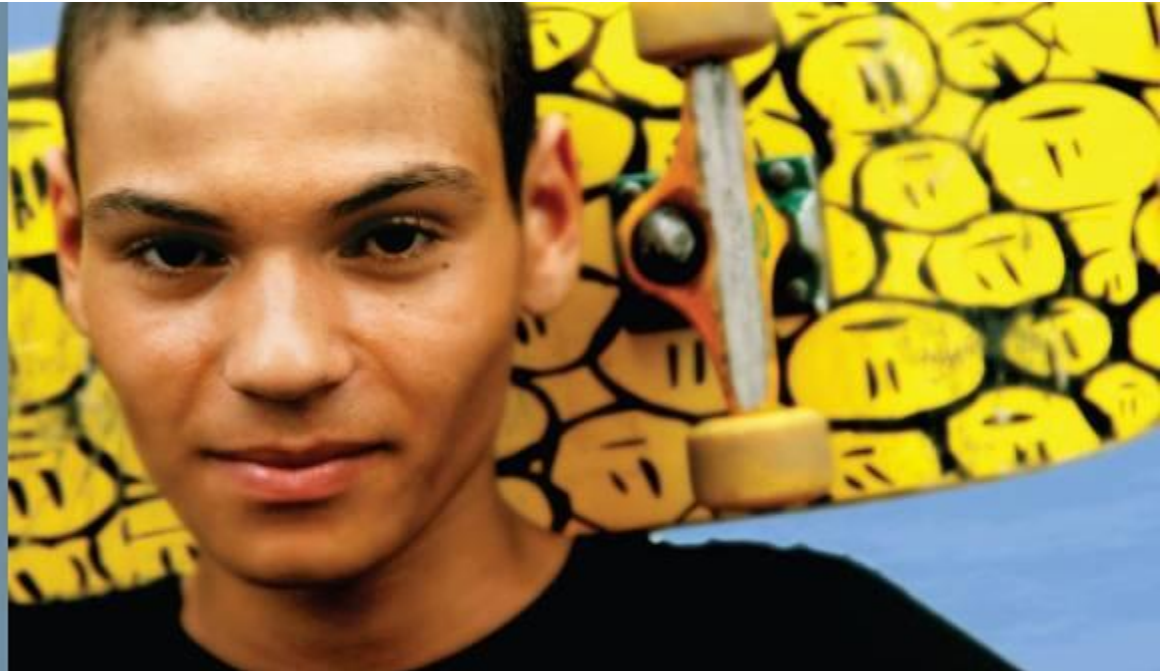
Connect to the IPv6 community



Transitioning IPv4 to IPv6

- IPv4 and IPv6 will coexist for some time
- A wide range of techniques are available for the period of transition between IPv4 and IPv6
- These techniques can be grouped into three categories:
 - Dual-stack techniques
 - Tunneling services
 - Translation services
- This presentation will describe the configuration of transition mechanisms and provide sample topologies and configuration commands
- The techniques covered will include: dual-stack, manual tunnels, 6to4 tunnels, ISATAP tunnels and NAT-PT
- **Note:** NAT-PT has been moved to historical status with RFC 4966.

Dual Stack





Dual-Stack Techniques

- Hosts and network devices run both IPv4 and IPv6 at the same time.
 - This technique is useful as a temporary transition, but it adds overhead and uses many resources.
- Cisco IOS Software is IPv6 ready.
 - As soon as IPv4 and IPv6 configurations are complete, the interface is dual stacked and it forwards both IPv4 and IPv6 traffic.
- Drawback of dual stacking includes:
 - The additional resources required to keep and process dual routing tables, routing protocol topology tables, etc.
 - The higher administrative overhead, troubleshooting, and monitoring, is more complex.



Dual-Stack Example



```
R1(config)# interface fa0/0
R1(config-if)# ip address 10.10.10.1 255.255.255.0
R1(config-if)# ipv6 address 2001:12::1/64
R1(config-if)# ^Z
R1#
```

- The FastEthernet 0/0 interface of R1 is dual stacked.
 - It is configured with an IPv4 and an IPv6 address.
 - Also notice that for each protocol, the addresses on R1 and R2 are on the same network.



Dual-Stack Example



```
R1# show ip interface fa0/0
FastEthernet0/0 is up, line protocol is up
 Internet address is 10.10.10.1/24
 Broadcast address is 255.255.255.255
 Address determined by setup command
 MTU is 1500 bytes
 Helper address is not set
 Directed broadcast forwarding is disabled
 Outgoing access list is not set
 Inbound access list is not set
 Proxy ARP is enabled
 Local Proxy ARP is disabled
 Security level is default
 Split horizon is enabled
 ICMP redirects are always sent
 ICMP unreachable are always present
```

<output omitted>

- The output confirms that the Fa0/0 interface is operational and uses the IPv4 address.



Dual-Stack Example



```

R1# show ipv6 interface fa0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::219:56FF:FE2C:9F60
  Global unicast address(es):
    2001:12::1, subnet is 2001:12::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF2C:9F60
  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

<output omitted>
  
```

- The output confirms that the Fa0/0 interface is operational and also uses the IPv6 address.

Tunneling





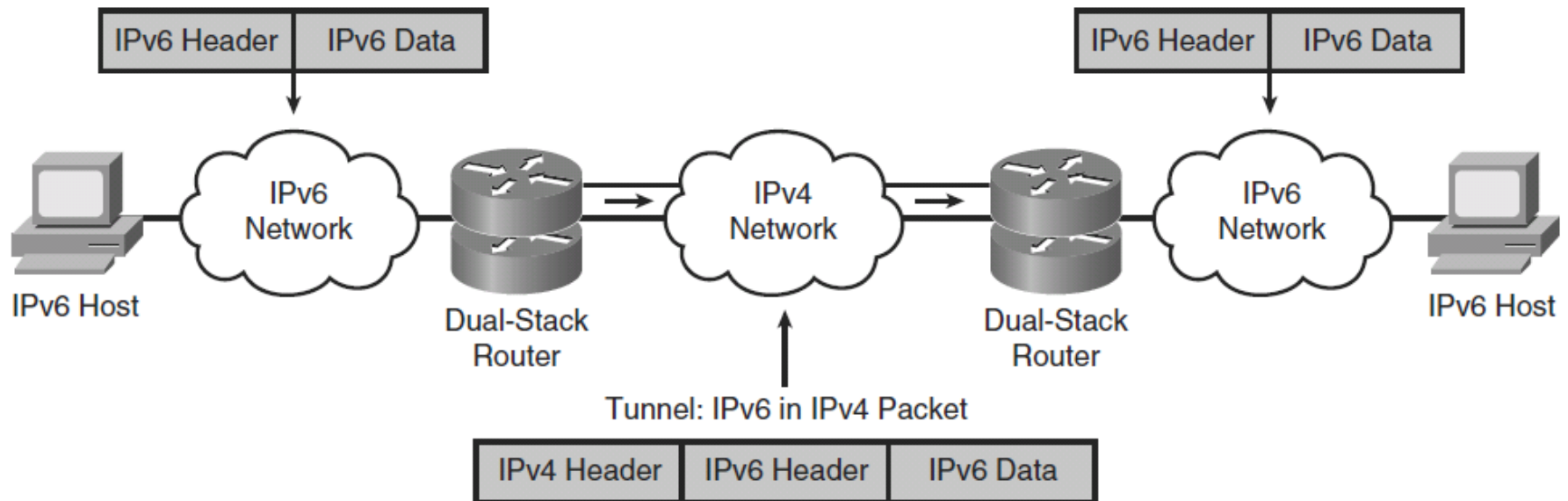
Tunneling Techniques

- Isolated IPv6 networks are connected over an IPv4 infrastructure using tunnels.
- The edge devices are the only ones that need to be dual-stacked.
- Scalability may be an issue if many tunnels need to be created.
 - Tunnels can be either manually or automatically configured, depending on the scale required and administrative overhead tolerated.



Tunneling Techniques

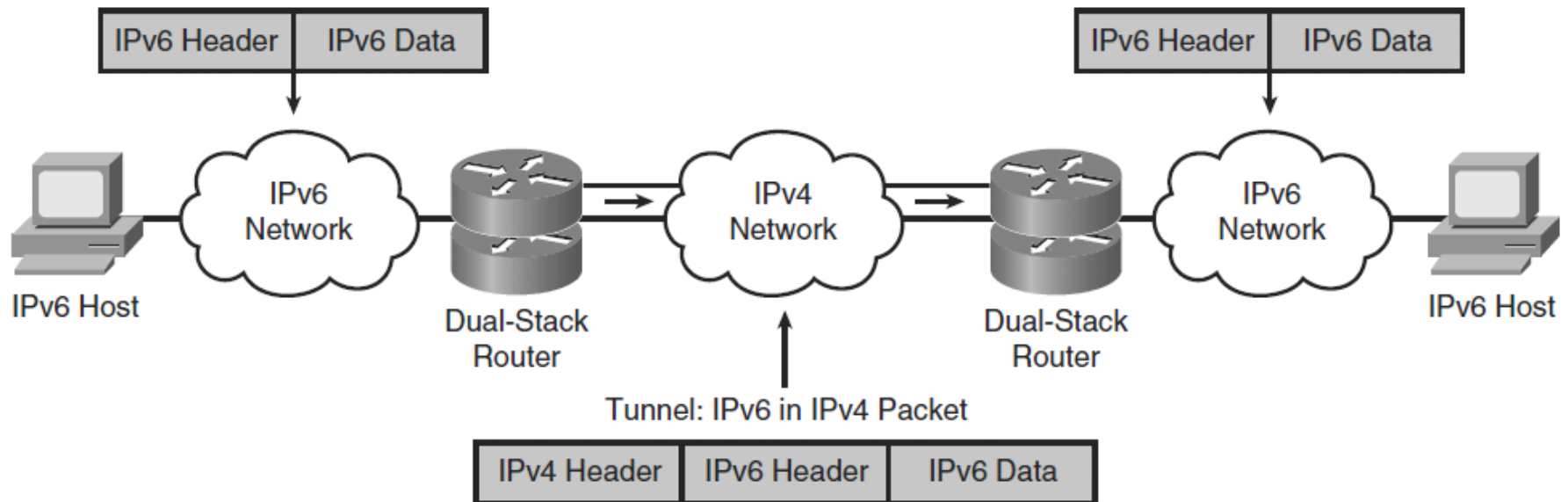
- For IPv6, tunneling is an integration method in which an IPv6 packet is encapsulated within IPv4.
- This enables the connection of IPv6 islands without the need to convert the intermediary network to IPv6.





Tunneling Techniques

- In this example, the tunnel between sites is using:
 - IPv4 as the transport protocol (the protocol over which the tunnel is created).
 - IPv6 is the passenger protocol (the protocol encapsulated in the tunnel and carried through the tunnel).
 - GRE is used to create the tunnel, and is known as the tunneling protocol.





Types of Tunnels

- Tunnels can be created manually using:
 - Manual IPv6 tunnels
 - GRE IPv6 tunnels (not covered in this presentation)
- Tunnels can also be created automatically using:
 - IPv4-Compatible IPv6 Tunnels (now deprecated)
 - 6to4 tunnels
 - ISATAP Tunnels

Manual Tunnels





Manual Tunnel Configuration

- Create a tunnel interface.

Router (config) #

```
interface tunnel number
```

- Creates a tunnel interface which is virtual.
- Once in interface configuration mode, configure the tunnel parameters including:
 - IP address
 - Tunnel source
 - Tunnel destination
 - Tunnel mode (type of tunnel)



Tunnel Configuration Commands

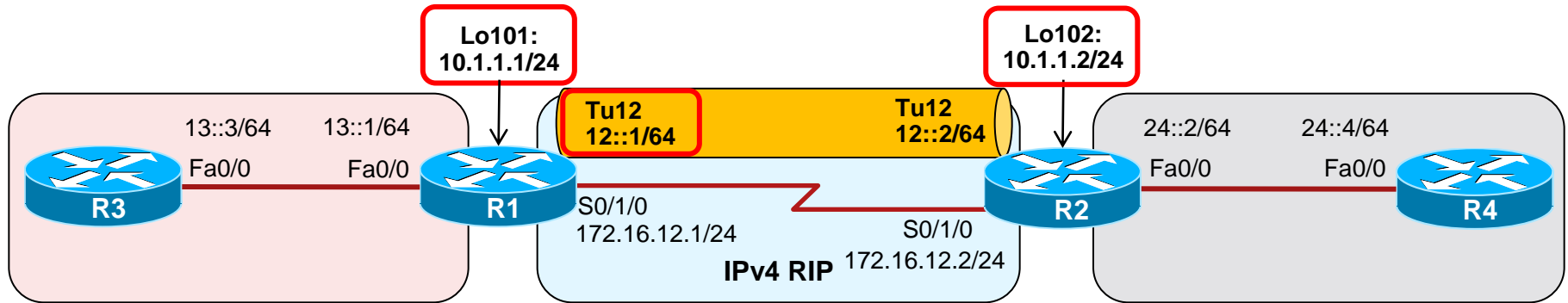
Command	Description
tunnel source <i>interface-type interface-number</i>	An interface configuration command that sets the source address for a tunnel interface as the address of the specified interface
tunnel destination <i>ip-address</i>	An interface configuration command that specifies the destination address for a tunnel interface. In this case the <i>ip-address</i> parameter is an IPv4 address
tunnel mode ipv6ip	An interface configuration command that sets the encapsulation mode for the tunnel interface to use IPv6 as the passenger protocol, and IPv4 as both the encapsulation and transport protocol.



Tunnel Troubleshooting Commands

Command	Description
<code>debug tunnel</code>	EXEC command that enables the display of the tunnel encapsulation and decapsulation process.
<code>debug ip packet detail</code>	EXEC command that enables the display of details about IP packets traversing the router.

Manual IPv6 Tunnel Example

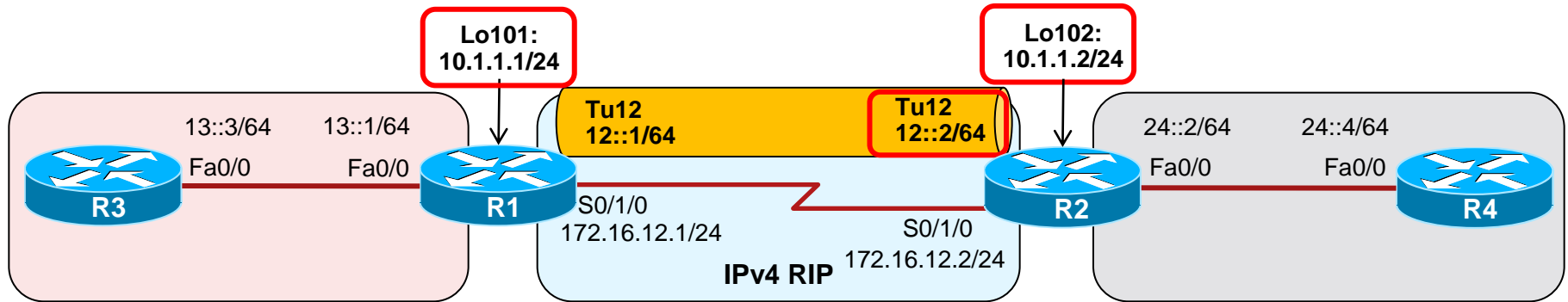


```

R1(config)# interface tunnel 12
R1(config-if)#
*Aug 16 09:34:46.643: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12,
changed state to down
R1(config-if)# no ip address
R1(config-if)# ipv6 address 12::1/64
R1(config-if)# tunnel source loopback 101
R1(config-if)# tunnel destination 10.1.1.2
R1(config-if)#
*Aug 16 09:36:52.051: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12,
changed state to up
R1(config-if)# tunnel mode ipv6ip
R1(config-if)#
  
```

- R1 is configured with the manual tunnel configuration.

Manual IPv6 Tunnel Example



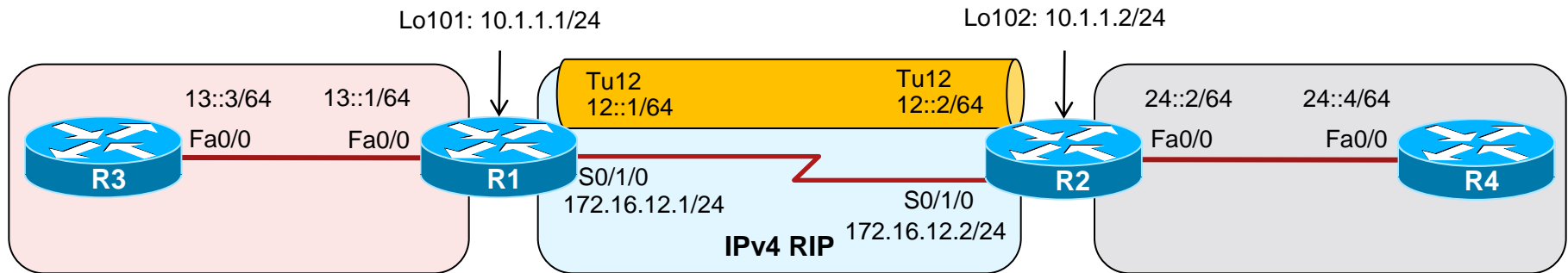
```

R2(config)# interface tunnel 12
R2(config-if)#
*Aug 16 09:38:47.532: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12,
changed state to down
R2(config-if)# no ip address
R2(config-if)# ipv6 address 12::2/64
R2(config-if)# tunnel source loopback 101
R2(config-if)# tunnel destination 10.1.1.1
R2(config-if)#
*Aug 16 09:39:24.056: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12,
changed state to up
R2(config-if)# tunnel mode ipv6ip
R2(config-if)#
  
```

- R2 is configured with the manual tunnel configuration.



Manual IPv6 Tunnel Example



```

R1# show interface tunnel 12
Tunnel12 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit/sec, DLY 500000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.1.1.1 (Loopback101), destination 10.1.1.2
  Tunnel protocol/transport IPv6/IP
  Tunnel TTL 255
  Fast tunneling enabled

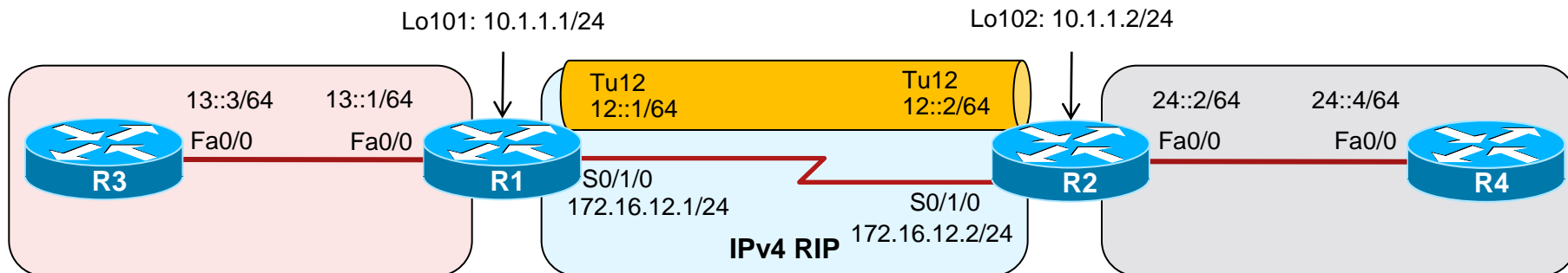
<output omitted>

```

- The tunnel interface is examined.
- Next, RIPng will be configured to cross the tunnel.



Manual IPv6 Tunnel Example



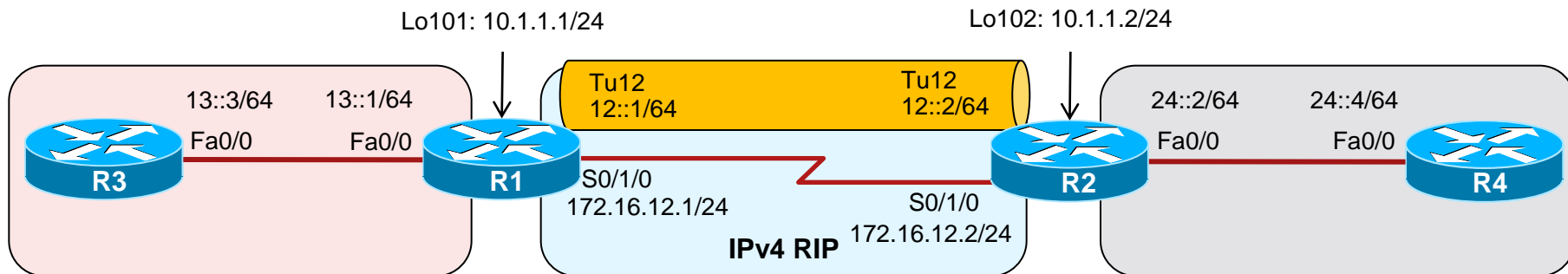
```
R1 (config) # ipv6 unicast-routing
R1 (config) # interface tunnel 12
R1 (config-if) # ipv6 rip RIPv6 enable
R1 (config-if) # interface fa0/0
R1 (config-if) # ipv6 rip RIPv6 enable
R1 (config-if) #
```

```
R2 (config) # ipv6 unicast-routing
R2 (config) # interface tunnel 12
R2 (config-if) # ipv6 rip RIPv6 enable
R2 (config-if) # interface fa0/0
R2 (config-if) # ipv6 rip RIPv6 enable
R2 (config-if) #
```

- RIPng is enabled on the tunnel interfaces and on the FastEthernet interfaces of R1 and R2.



Manual IPv6 Tunnel Example



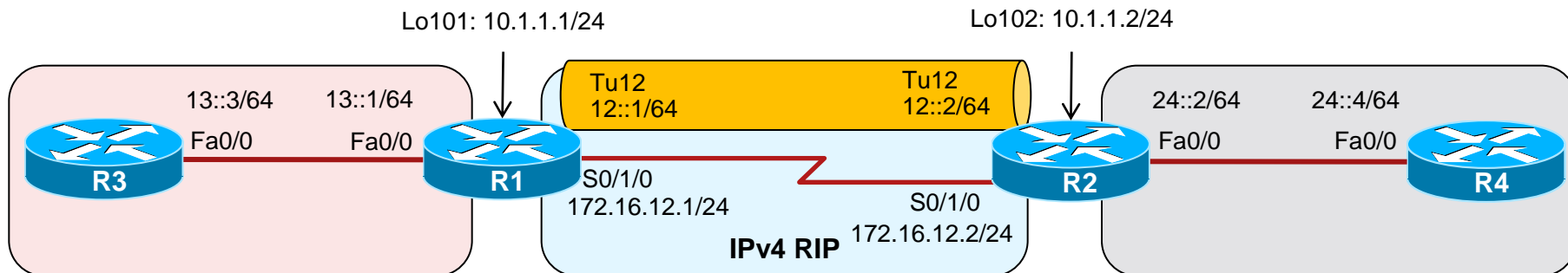
```
R3 (config) # ipv6 unicast-routing
R3 (config) # interface fa0/0
R3 (config-if) # ipv6 rip RIPv6 enable
R3 (config-if) #
```

```
R4 (config) # ipv6 unicast-routing
R4 (config) # interface fa0/0
R4 (config-if) # ipv6 rip RIPv6 enable
R4 (config-if) #
```

- RIPng is enabled on the FastEthernet interfaces of R3 and R4.
- Now end-to-end connectivity should be achieved.



Manual IPv6 Tunnel Example



```
R4# show ipv6 route rip
```

```
<output omitted>
```

```
R 12::/64 [120/2]
   via FE80::2, FastEthernet0/0
R 13::/64 [120/3]
   via FE80::2, FastEthernet0/0
```

```
R4#
```

```
R3# ping 24::4
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 24::4, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/18/20 ms
```

```
R3#
```



Manual IPv6 Tunnel Summary

- Manual tunnels are simple to configure, and are therefore useful for a small number of sites.
- However, for large networks manual tunnels are not scalable, from both a configuration and management perspective.
- The edge routers on which the tunnels terminate need to be dual stacked, and therefore must be capable of running both protocols and have the capacity to do so.

6to4 Tunnels



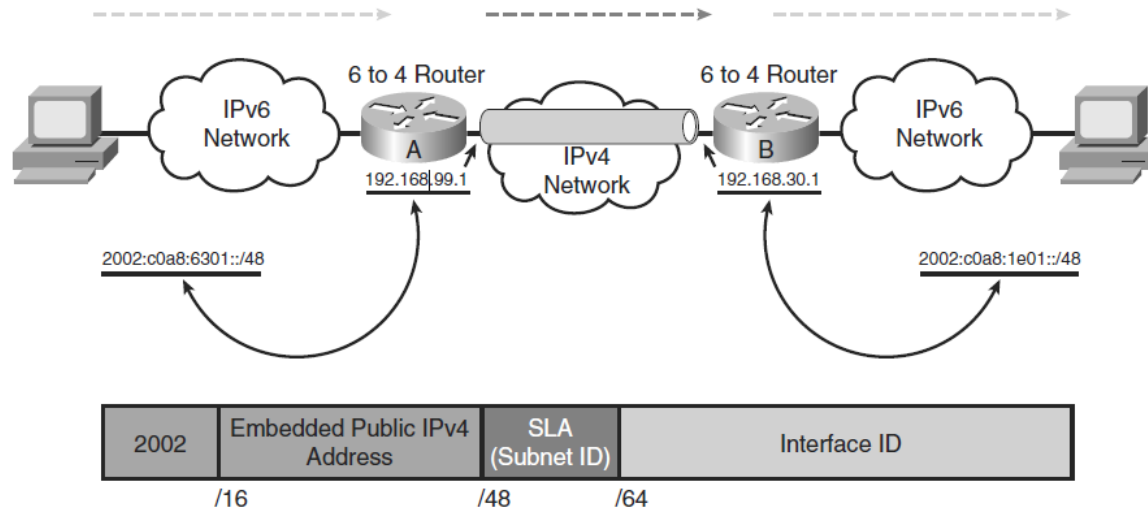


6to4 Tunnels

- 6to4 tunnels, also known as a 6-to-4 tunnel, is an automatic tunneling method.
- 6to4 tunnels are point-to-multipoint, rather than the point-to-point tunnels.
- The 6to4 tunnels are built automatically by the edge routers, based on embedded IPv4 address within the IPv6 addresses of the tunnel interfaces on the edge routers.
- 6to4 tunnels enable the fast deployment of IPv6 in a corporate network without the need for public IPv6 addresses from ISPs or registries.



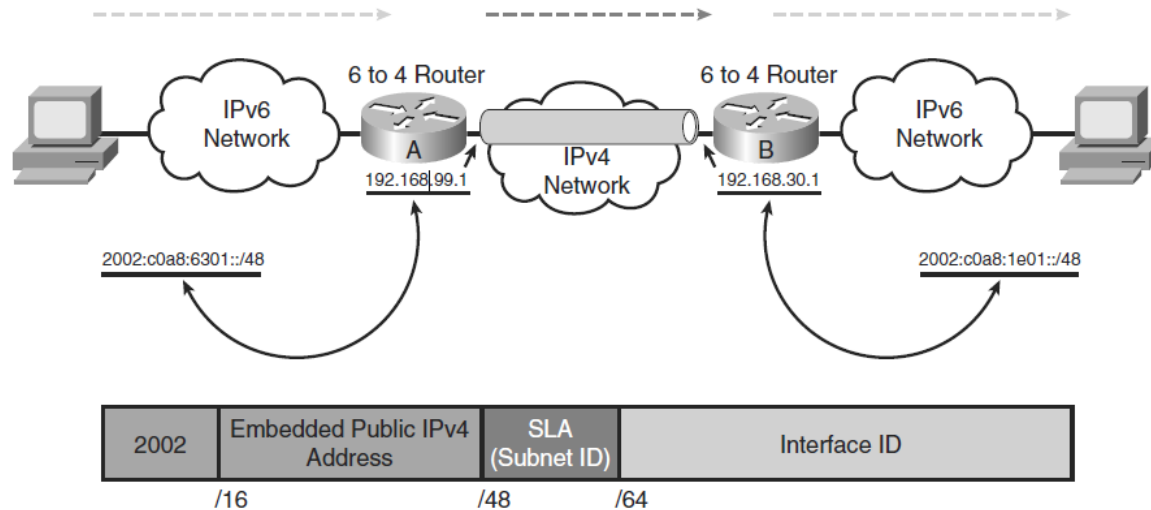
6to4 Tunnel Example



- When Router A receives an IPv6 packet with a destination address in the range of 2002::/16 (the address 2002:c0a8:1e01::/48 in the example), it determines that the packet must traverse the tunnel.
 - The router extracts the IPv4 address embedded in the third to sixth octets, inclusively, in the IPv6 next-hop address.
 - In this example, these octets are c0a8:1e01 which is therefore 192.168.30.1.
- This IPv4 address is the IPv4 address of the 6to4 router at the destination site, Router B.



6to4 Tunnel Example



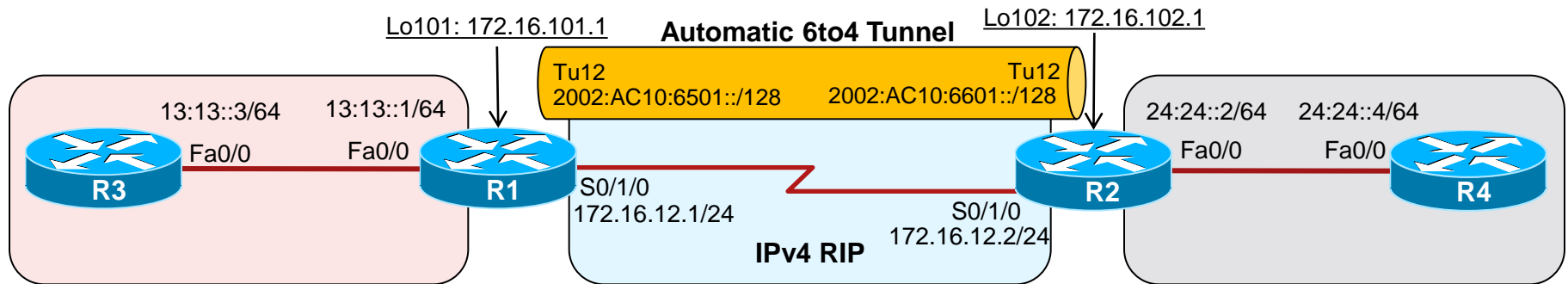
- Router A encapsulates the IPv6 packet in an IPv4 packet with Router B's extracted IPv4 address as the destination address.
 - The packet passes through the IPv4 network.
- Router B, decapsulates the IPv6 packet from the received IPv4 packet and forwards the IPv6 packet to its final destination.



6to4 Limitations

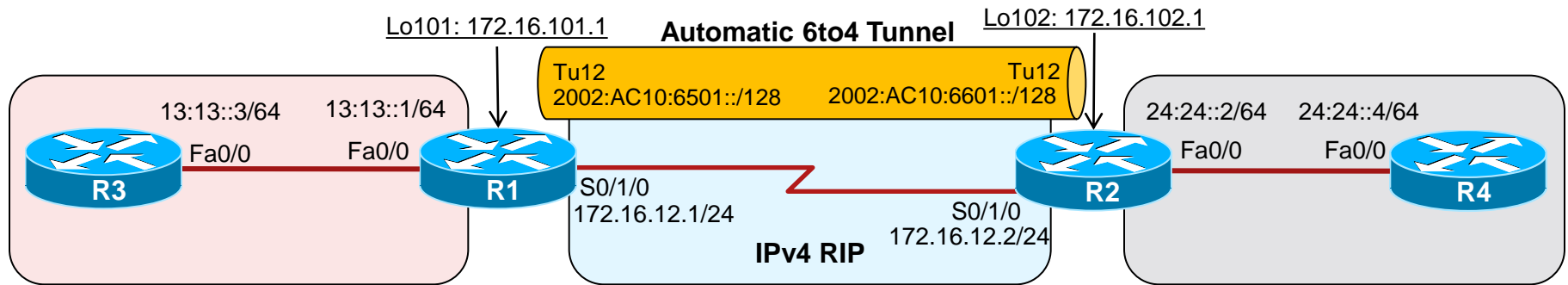
- Only static routes or BGP are supported.
 - This is because the other routing protocols use link-local addresses to form adjacencies and exchange updates and these do not conform to the address requirements for 6to4 tunnels.
 - The example presented here will use static routes.
- NAT cannot be used along the IPv4 path of the tunnel, again because of the 6to4 address requirements.

6to4 Tunnel Example



- In this example, there are two IPv6 networks separated by an IPv4 network.
- The objective of this example is to again provide full connectivity between the IPv6 islands over the IPv4-only infrastructure.
- The first step is to configure routers R1 and R2 so that they can establish the 6to4 tunnel between them.

6to4 Tunnel Example

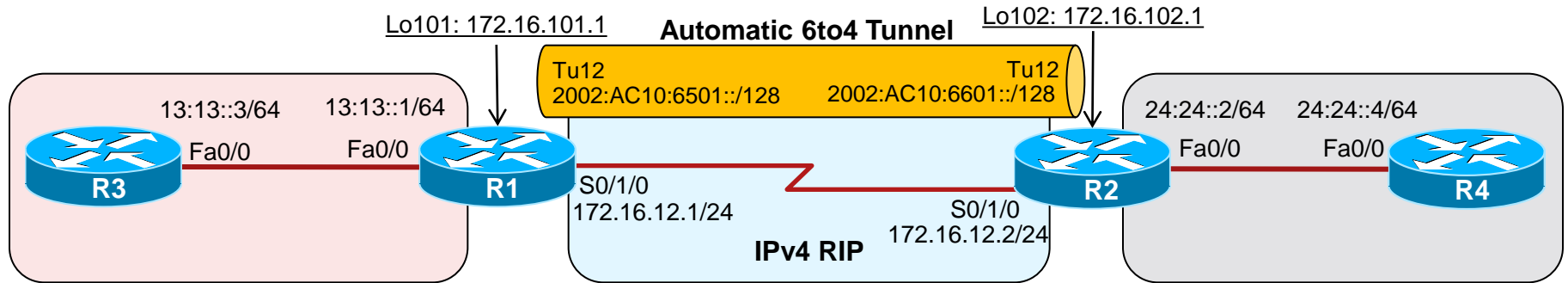


```

R1(config)# interface tunnel 12
R1(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12, changed state to down
R1(config-if)# no ip address
R1(config-if)# ipv6 address 2002:AC10:6501::/128
R1(config-if)# tunnel source loopback 101
R1(config-if)# tunnel mode ipv6ip 6to4
R1(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12, changed state to up
R1(config-if)# exit
  
```

- R1 is configured with the 6to4 tunnel.
 - Notice that the configuration is similar to the manual tunnel configurations except that the tunnel destination is not specified.

6to4 Tunnel Example

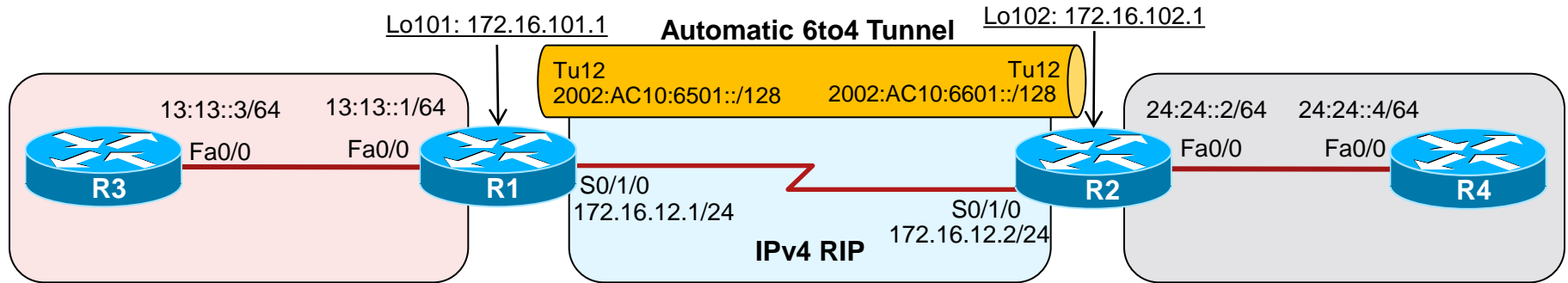


```
R1 (config) # ipv6 route 2002::/16 tunnel 12
R1 (config) # ipv6 route 24::/64 2002:AC10:6601::
R1 (config) #
```

- R1 is configured with static routes.



6to4 Tunnel Example

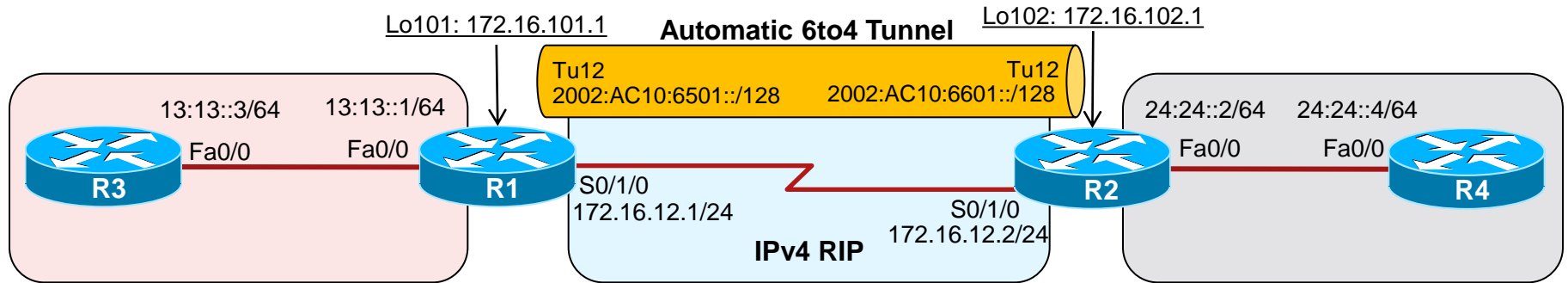


```

R2 (config) # interface tunnel 12
R2 (config-if) #
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12, changed state to down
R2 (config-if) # no ip address
R2 (config-if) # ipv6 address 2002:AC10:6601::/128
R2 (config-if) # tunnel source loopback 102
R2 (config-if) # tunnel mode ipv6ip 6to4
R2 (config-if) #
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12, changed state to up
R2 (config-if) # exit
  
```

- R2 is configured with the 6to4 tunnel.

6to4 Tunnel Example



```
R2 (config) # ipv6 route 2002::/16 tunnel 12
R2 (config) # ipv6 route 13::/64 2002:AC10:6501::
R2 (config) #
```

- R2 is configured with static routes.

ISATAP Tunnels



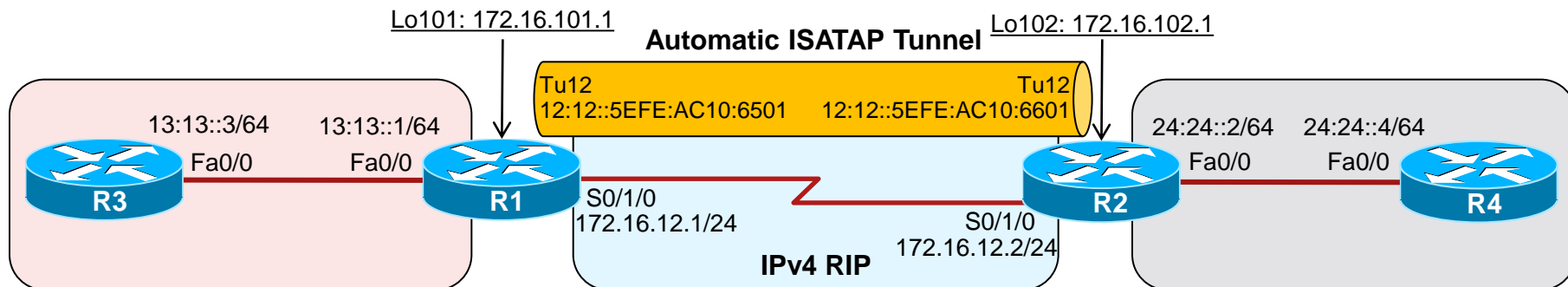


ISATAP Tunnels

- An Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) tunnel is very similar to a 6to4 IPv6 tunnel.
 - It is used to connect IPv6 domains over an IPv4 network.
 - It embeds an IPv4 address within the IPv6 address.
- The goal of ISATAP is to provide connectivity for IPv6 hosts to a centralized IPv6-capable router, over an IPv4-only access network.
- ISATAP was designed to transport IPv6 packets within a site (hence the “intra-site” part of its name).
 - It can still be used between sites, but its purpose is within sites.
- ISATAP tunnels use IPv6 addresses consisting of a 64-bit prefix concatenated to a 64-bit interface ID in EUI-64 format.

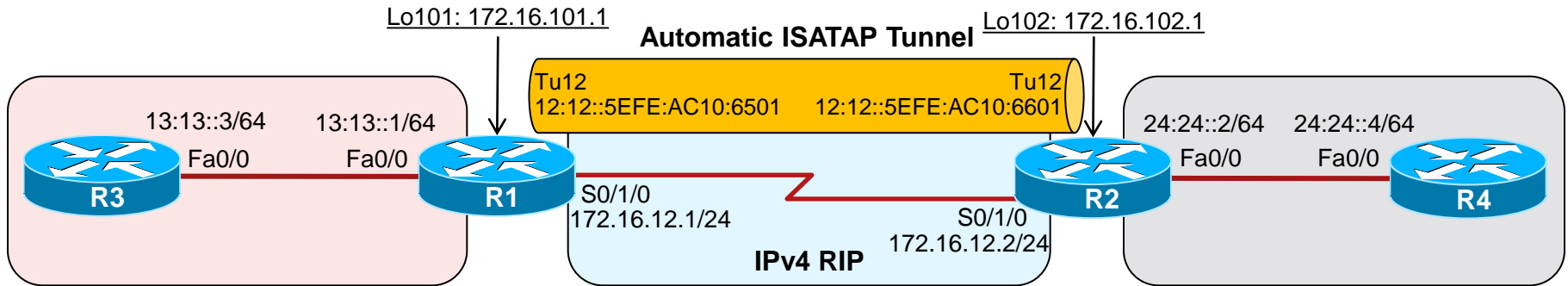


ISATAP Tunnel Example



- In this example, there are two IPv6 networks separated by an IPv4 network.
- The objective of this example is to again provide full connectivity between the IPv6 islands over the IPv4-only infrastructure.
- The first step is to configure routers R1 and R2 so that they can establish the ISATAP tunnel between them.

ISATAP Tunnel Example



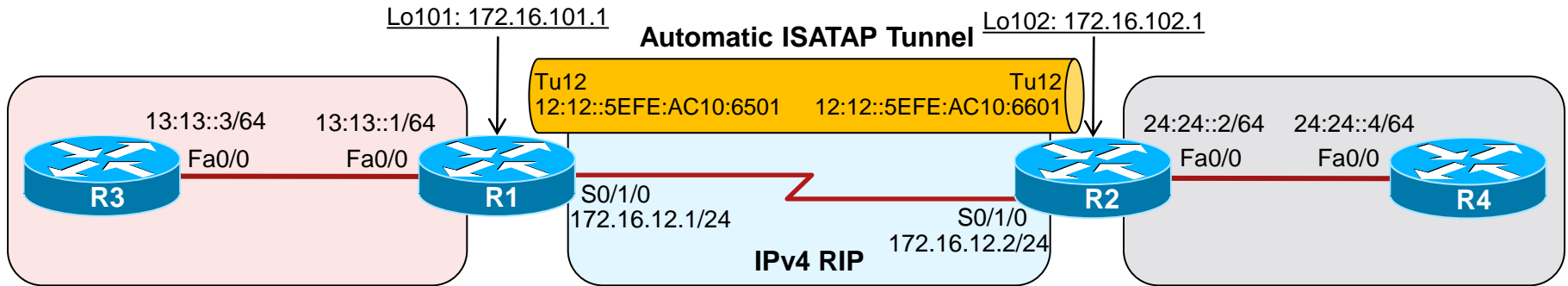
```

R1(config)# interface tunnel 12
R1(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12, changed state to down
R1(config-if)# no ip address
R1(config-if)# ipv6 address 12:12::/64 eui-64
R1(config-if)# tunnel source loopback 101
R1(config-if)# tunnel mode ipv6ip isatap
R1(config-if)# exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel12, changed state to up
R1(config)# ipv6 route 24::/64 tunnel12 FE80::5EFE:AC10:6601
R1(config)#
  
```

- R1 is configured with the ISATAP tunnel and a static route.
 - Notice that the configuration is similar to the manual and GRE tunnel configurations except that the tunnel destination is not specified.



ISATAP Tunnel Example

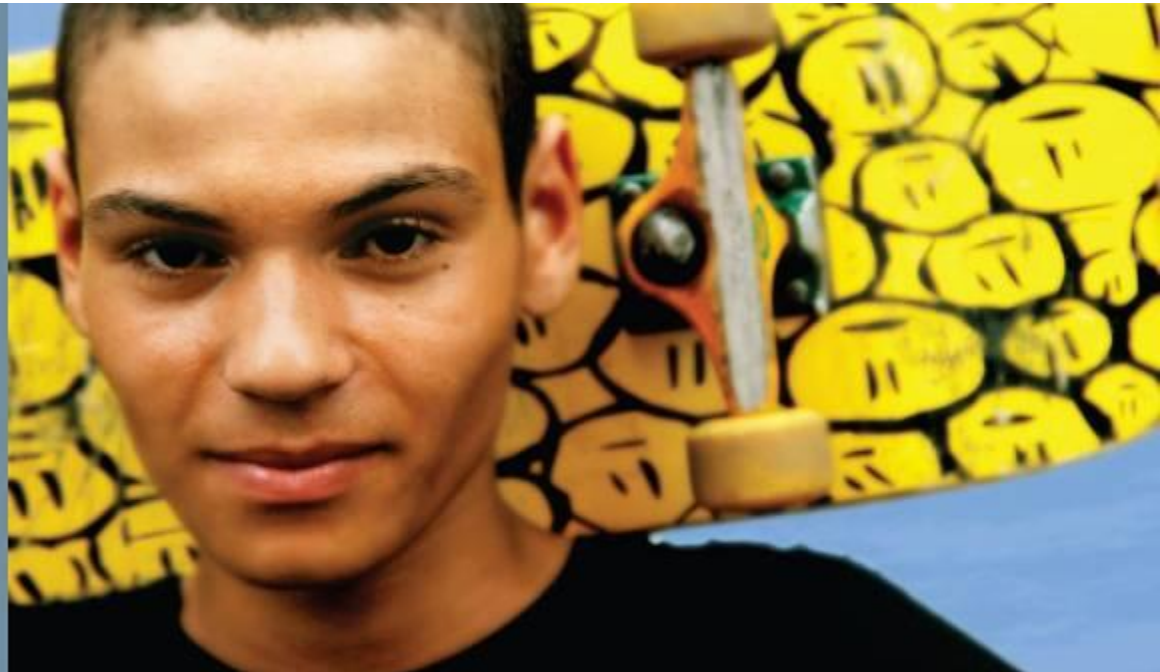


```

R2 (config) # interface tunnel 12
R2 (config-if) #
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnell12, changed state to down
R2 (config-if) # no ip address
R2 (config-if) # ipv6 address 12:12::/64 eui-64
R2 (config-if) # tunnel source loopback 102
R2 (config-if) # tunnel mode ipv6ip isatap
R2 (config-if) # exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnell12, changed state to up
R2 (config) # ipv6 route 13::/64 tunnel12 FE80::5EFE:AC10:6501
R2 (config) #
  
```

- R2 is configured with the ISATAP tunnel and a static route.

Translation Using NAT-PT





NAT-PT

- NAT-PT is a transition technique, but is not a replacement for dual stack or tunneling.
 - It can be used in situations where direct communication between IPv6-only and IPv4-only networks is desired.
 - It would not be appropriate in situations where connectivity between two IPv6 networks is required, because two points of translation would be necessary, which would not be efficient or effective.
- With NAT-PT, all configuration and translation is performed on the NAT-PT router.
 - The other devices in the network are not aware of the existence of the other protocol's network, nor that translations are occurring.
- **Note:** NAT-PT has been moved to historical status with RFC 4966.



Resources

- <http://ipv6.beijing2008.cn/en>
- <http://www.iana.org/numbers/>
- <http://www.cisco.com/go/ipv6>
- <http://www.iana.org/numbers/>
- <http://www.cisco.com/go/ipv6>

- IP address tools (which also support IPv6):
 - IPAT <http://nethead.de/index.php/ipat>
 - ipv6gen <http://techie.devnull.cz/ipv6/ipv6gen/>
 - freeipdb <http://home.globalcrossing.net/~freeipdb/>



Resources - Continued

- Cisco IPv6

<http://www.cisco.com/web/solutions/netsys/ipv6/index.html>

- Cisco IOS IPv6 Configuration Guide

http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/12_4/ipv6_12_4_book.html

- Dual-Stack At-A-Glance

http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6553/at_a_glance_c45-625859.pdf

- Implementing Tunneling for IPv6

<http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-tunnel.html>

- RFC 4966

<http://www.apps.ietf.org/rfc/rfc4966.html>



CISCO