Internet Protocol version 6

ROUTE Module 8

Agenda

- Introduction
- IPv6 Header
- Addressing Schema
- IPv6 Routing protocols
- OSPFv3
- Cooexistence of IPv4 and IPv6

Motivation of IPv6

- *Is it necessary?*
  - IPv4 is 30 years old (first IPv6 RFC was presented 16 years ago)
  - Some features in IPv4 are consider as unused or dangerous (*However same situation is in IPv6 protocol*)
    - Some features in IPv4 are missing (same in IPv6)
    - *30 years of protocol developing should provide enough information how to do it right in IPv6…*
    - *…unfortunately it is not the case of IPv6*
  - IPv6 has bigger addressing space

Objective Design of IPv6

1. **Larger IP address space**
2. **Better end-to-end connectivity**
3. **Ability for autoconfiguring devices**
4. **Simplified header structures**
5. **Better security (IPSEC – ESP, AH)**
6. **Better quality of services**
7. **Better multicast and anycast abilities**
8. **Mobility features**
9. **Ease of administration**
10. **Smooth transition from IPv4**

source: http://ipv6.com/

Situation After Several Years of Development

- ✓ Larger IP address space
- ✓ Better end-to-end connectivity
- ✗ Ability for autoconfiguring devices
- ~ Simplified header structures
- ~ Better security (IPSEC – ESP, AH)
- ~ Better quality of services
- ~ Better multicast and anycast abilities
- ~ Mobility features
- ✗ Ease of administration
- ✗ Smooth transition from IPv4

IPv6 Packet

Simplify IPv6 Header

IPv4 Header

Version	IHL	Type of Service	Total Length
Identification		Flags	Fragment Offset
Time to Live	Protocol	Header Checksum	
Source Address			
Destination Address			
Options			Padding

IPv6 Header

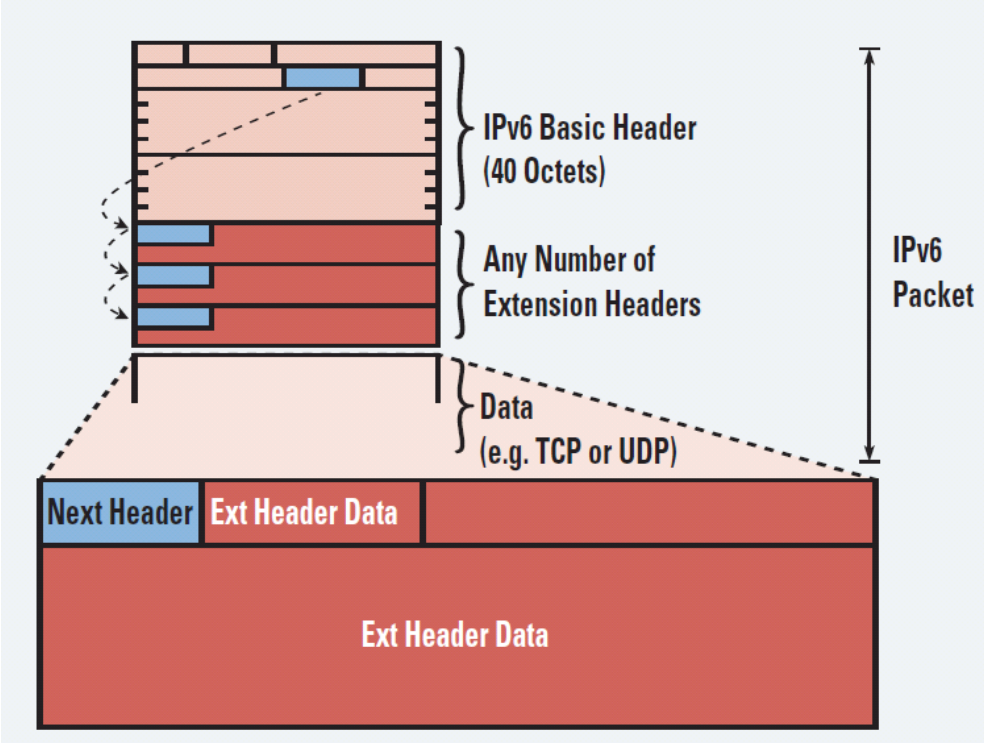
Version	Traffic Class	Flow Label
Payload Length	Next Header	Hop Limit
Source Address		
Destination Address		

Legenda

- 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

- Several IPv4 features have been moved to extension headers
- **Extension headers** offers future extension, however processing chain of extension headers is complicated

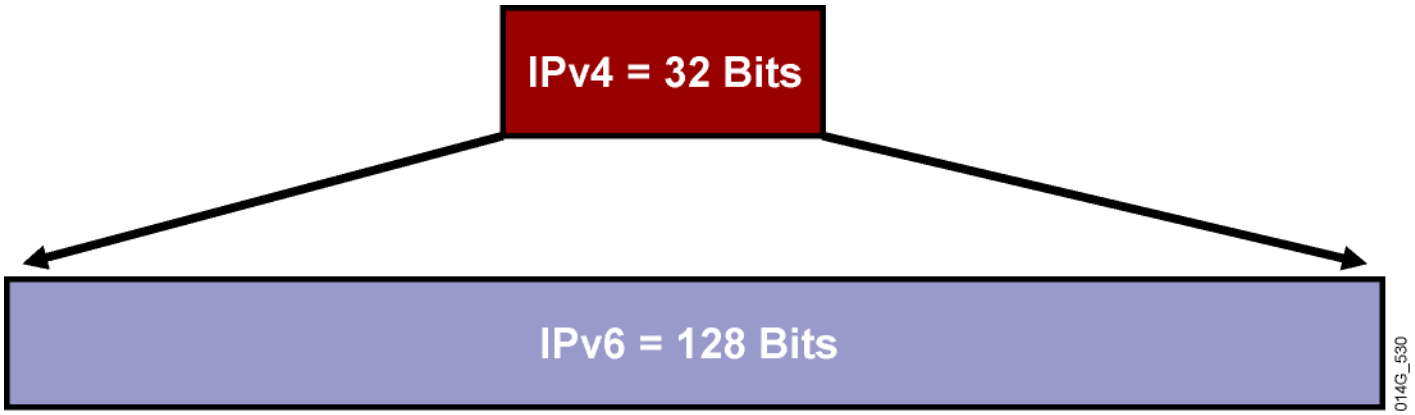


MTU Issues

- Minimum link MTU for IPv6 is 1280B (vs. 68B for IPv4).
  - On links with MTU < 1280, link-specific fragmentation and reassembly must be used
  - IPv6 routers do not implement packet fragmentation
  - `IF` fragmentation is necessary `THEN` end node does it
- Implementations are expected to perform **path MTU discovery (PMTUD)** to send packets bigger than 1280
- Minimal implementation can omit PMTUD as long as all packets sizes are kept ≥ 1280 octets
- A hop-by-hop option supports transmission of **jumbograms** with up to 2^{32} octets of payload

IPv6 Addressing

Larger Address Space



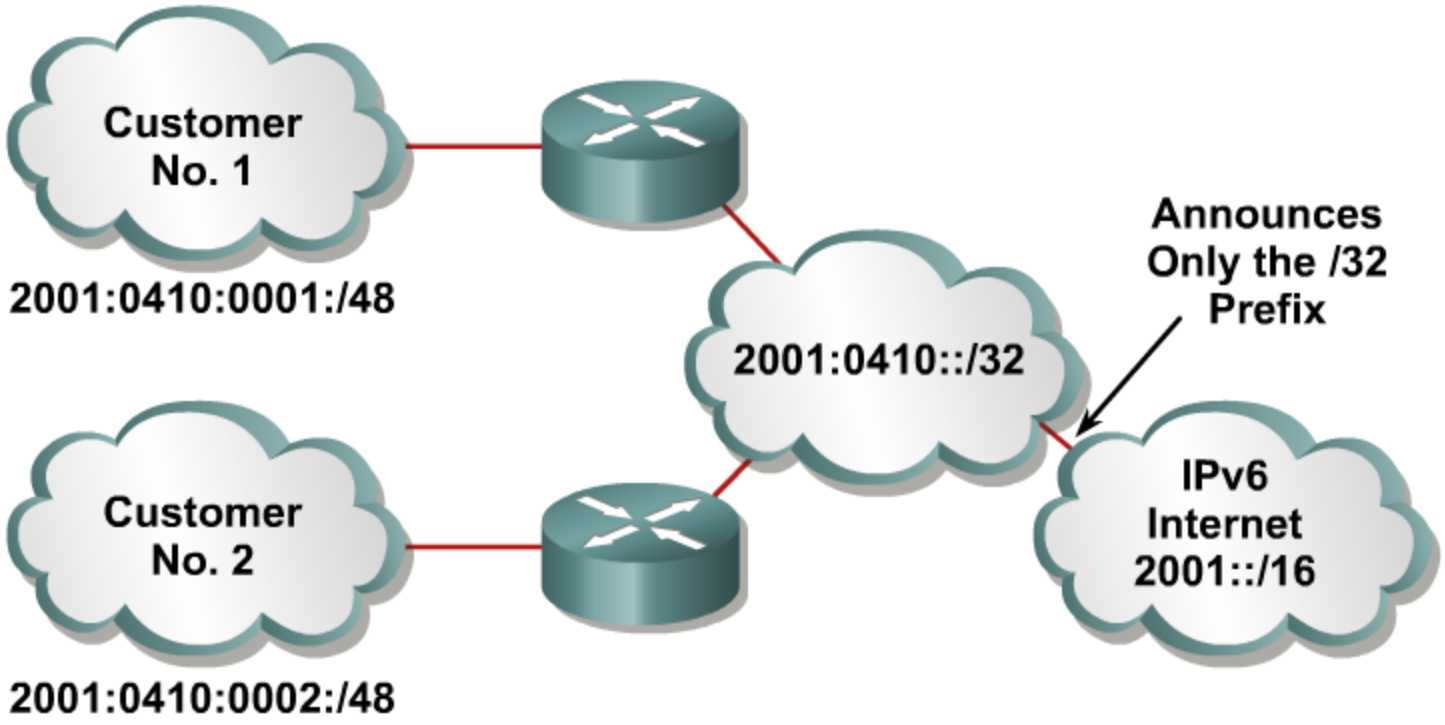
IPv4

- 32 bits or 4 bytes long
  - 4,200,000,000 possible addressable nodes

IPv6

- 128 bits or 16 bytes: four times the bits of IPv4
  - $3.4 * 10^{38}$ possible addressable nodes
  - 340,282,366,920,938,463,374,607,432,768,211,456
  - $5 * 10^{28}$ addresses per person

Address Aggregation



- Aggregation of prefixes announced in the global routing table – similar to classful routing
- Efficient and scalable routing

IPv6 Address Representation

- $x:x:x:x:x:x:x:x$, where x is a 16-bit hexadecimal field
- Leading zeros in a field are optional:
  - 2031:0:130F:0:0:9C0:876A:130B
- Only once per address successive fields of 0 can be represented as ::

Examples:

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

2031:0:130f::9c0:876a:130b

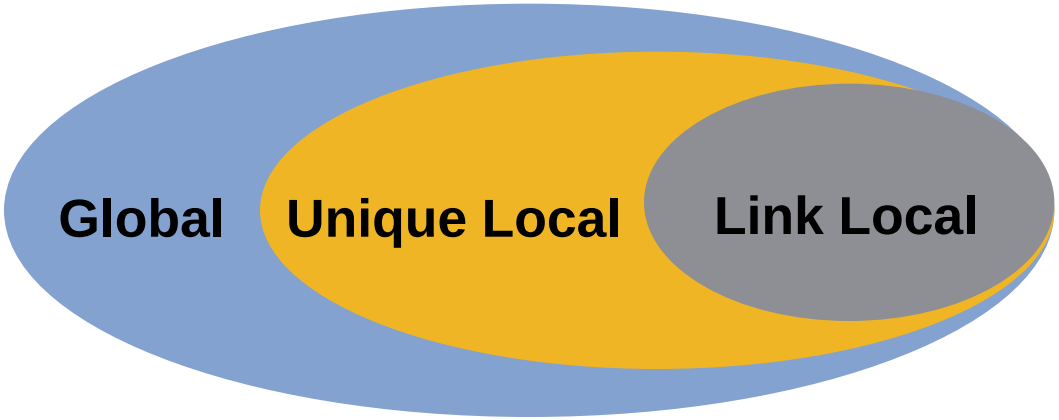
FF01:0:0:0:0:0:0:1 >>> FF01::1

0:0:0:0:0:0:0:1 >>> ::1

0:0:0:0:0:0:0:0 >>> ::

Addressing Model

- Addresses are assigned to interfaces
- Interface has multiple addresses
  - In IPv4, interface has (usually) only 1 address
- Addresses have scope
  - Link Local
  - Global
- Addresses have lifetime
  - Valid and preferred lifetime
  - Used in autoconfiguration



IPv6 Address Types

- **Unicast**
  - Address is for a single interface
  - IPv6 has several types (for example, global and IPv4 mapped)
- **Multicast**
  - One-to-many
  - Enables more efficient use of the network
  - Uses a larger address range
- **Anycast**
  - One-to-nearest (allocated from unicast address space)
  - Multiple devices share the same address
  - All anycast nodes should provide uniform service
  - Source devices send packets to anycast address
  - Routers decide on closest device to reach that destination
  - Suitable for load balancing and content delivery services

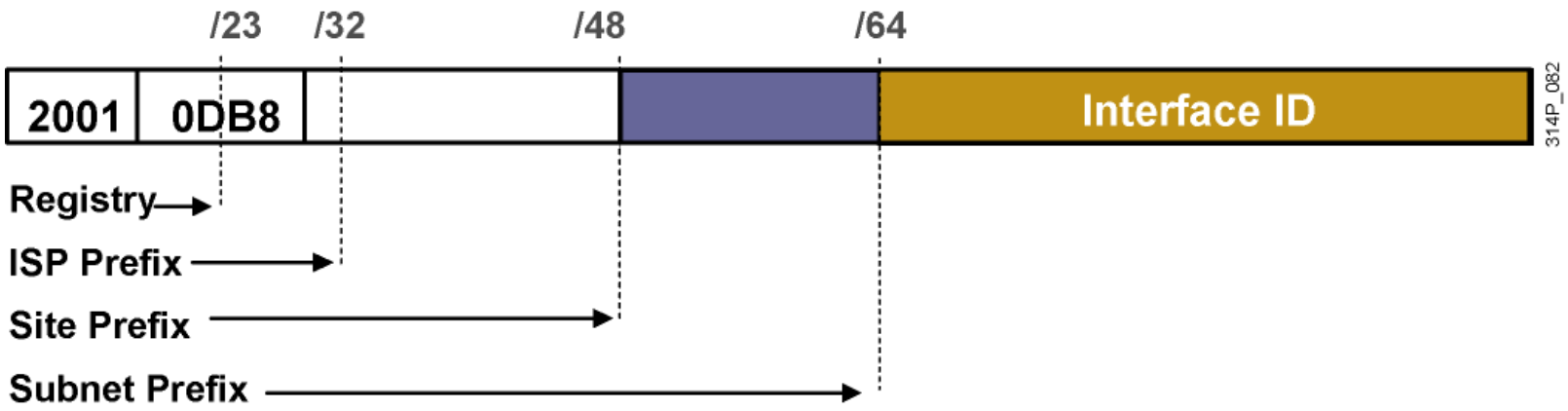
Addressed Overview

- As described in RFC 4291:
  - **::/128** Not specified address
  - **::0/0** Default-gateway
  - **::1/128** Loopback
  - **FF00::/8** Multicast
  - **FE80::/10** Link-Local Unicast
  - **FC00::/7** Unique Local Unicast, RFC 4193
  - **::A.B.C.D** IPv4-compatible addresses (not recommended)
  - **::FFFF:A.B.C.D** IPv4-mapped address
  - All others Global Unicast

Global Unicast and Anycast Addresses ①

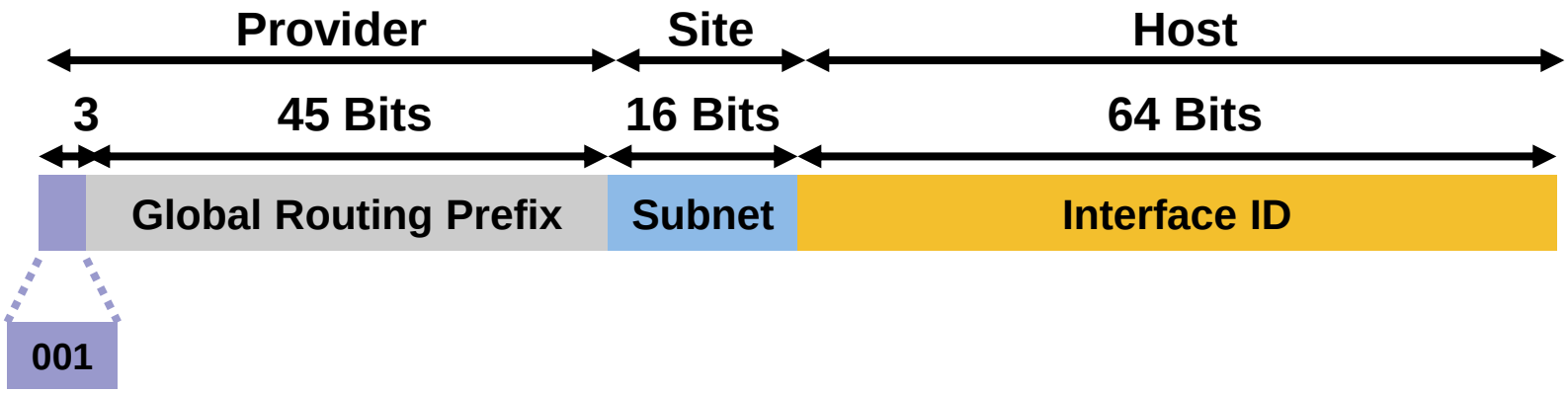
- Global Unicast and Anycast addresses have the same format
  - Network prefix allows reasonable aggregation
- Number of addresses that can be assigned to an interface is not limited
  - Link-local address must be assigned to every IPv6 interface
  - Several unique link or global addresses can be assigned to one interface
  - Anycast address is an address assigned to several interfaces on different devices

Global Unicast and Anycast Addresses ②



- Global Unicast and Anycast address
  - Global Routing Prefix
  - Subnet ID
  - Interface ID
- GRP and SID structure is not fixed
  - Recommendation is to use Interface ID 64bits long
  - Global Unicast address with network prefix /48 is usually assigned
    - Subnet ID has 16 bits

IPv6 Global Unicast Address



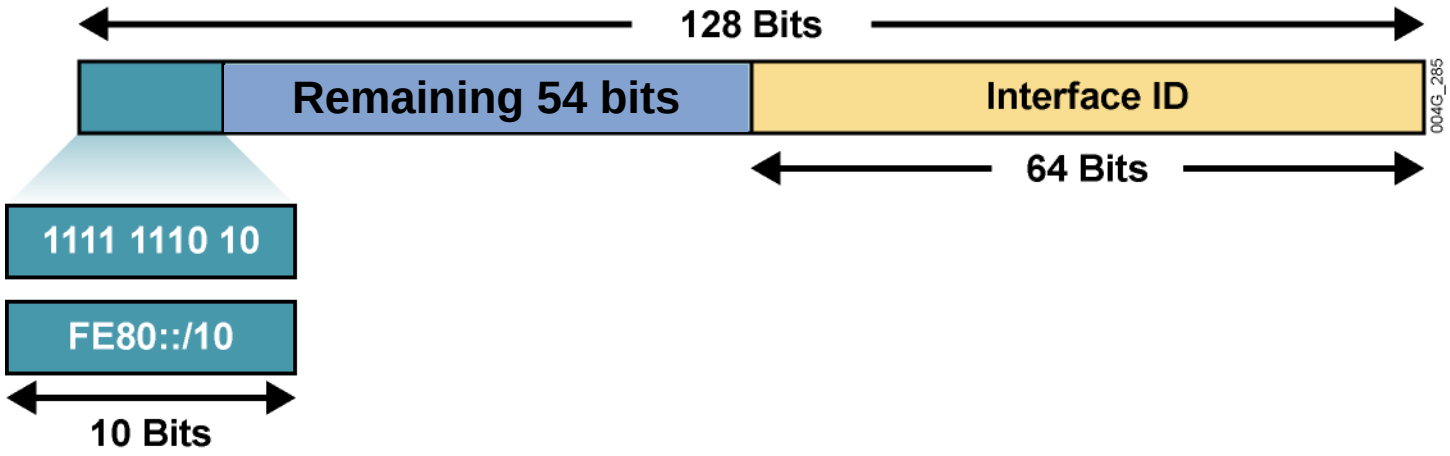
- Structured as a hierarchy to keep the aggregation
- Addresses for generic use of IPv6 – global unique addresses

IPv6 Interface ID

- Cisco uses the **extended universal identifier EUI-64** format to do stateless autoconfiguration (SLAAC)
  - Modified EUI-64 is 64bits long and is used as Interface ID
- Modified EUI-64 expands the 48-bit MAC address to 64 bits by:
  1. Inserting two bytes FF:FE between OUI and S/N
  2. The universal/local bit is inverted

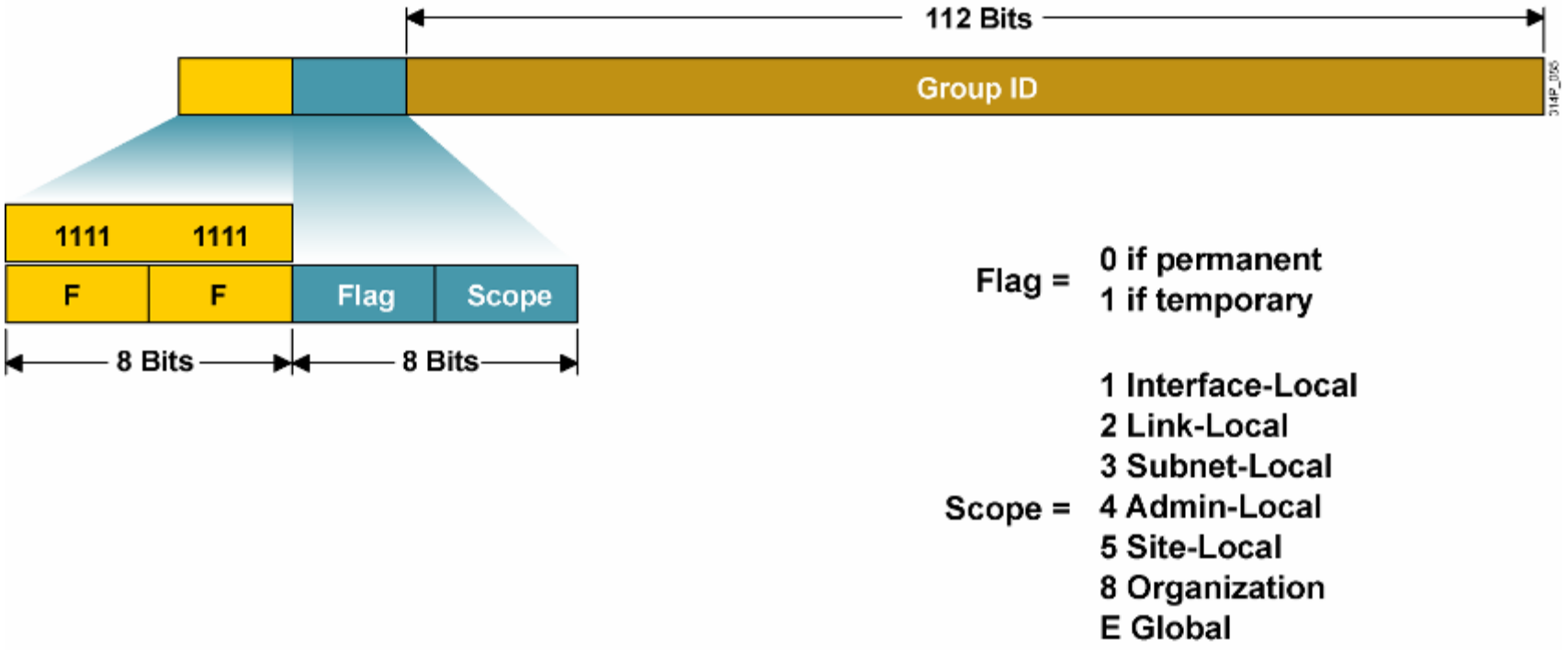
MAC address to EUI-64

Link-Local Address



- **Link-local address** has specific FE80::/10, random 54 bits (usually zero) and Interface ID in EUI-64 format or created by Privacy extensions
- Mandatory address for communication between two IPv6 devices
- Automatically assigned by router as soon as IPv6 is enabled
- Also used for next-hop calculation in routing protocols
- Unique and valid only in one broadcast domain
- Remaining 54 bits could be zero or any manual configured value

IPv6 Multicast Addresses

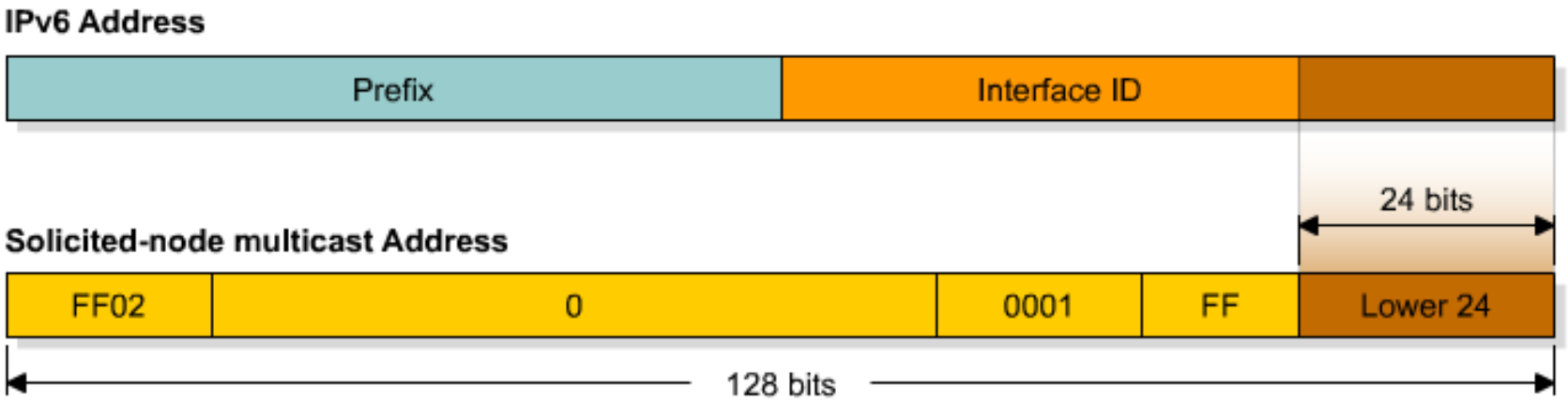


- **Multicast address** is frequently used in IPv6
  - Replaces broadcast
  - Has prefix FF00::/8

Examples of Permanent Multicast Addresses

	Meaning	Scope
FF02::1	All nodes 	Link-local
FF02::2	All routers 	Link-local
FF02::9	All RIP routers 	Link-local
FF02::1:FFXX:XXXX	Solicited-node 	Link-local
FF05::101	All NTP servers 	Site-local

Solicited-Node Multicast Address



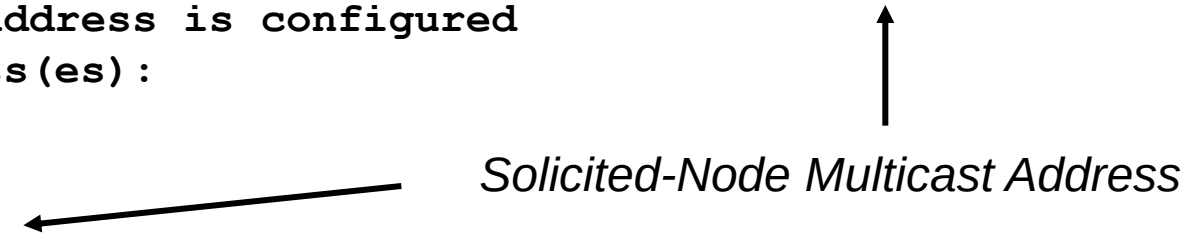
- **Solicited-node multicast address** consists of prefix FF02::1:FF:/104 + lower 24 bits corresponding unicast or anycast address of the node
- Used by ICMPv6
  - ICMPv6 is encapsulated in IPv6 packet, Solicited-Node address is used as destination IPv6 address
- Address with link-local scope

ICMPv6

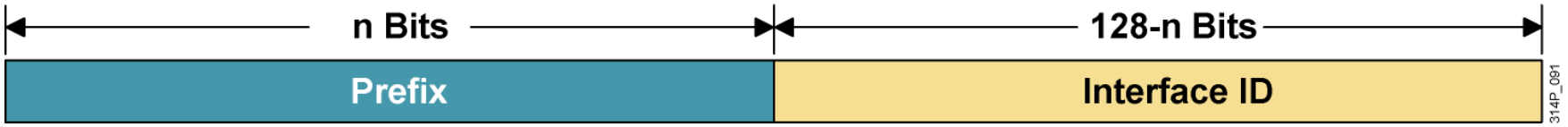
- *ICMPv6 is more important than ICMP for IPv4!!!*
- Supports all messages as ICMP
  - Destination Unreachable, Packet Too Big, Time Exceeded, Parameter Problem, Echo/Echo Reply, Redirect
  - Add new messages
    - Router Solicitation, Advertisement (plug-an-play configuration)
    - Neighbor Solicitation, Advertisement (ARP replacement)
- Added features are used for:
  - Finding routers
  - IPv6 plug-an-play configuration
  - IPv6 address to MAC address translation
  - Duplicate IPv6 address detection

IPv6 Addresses on Interface

```
R1# show ipv6 interface e0
Ethernet0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C203:DFF:FEAC:0
  No global unicast address is configured
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FFAC:0
  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#
```

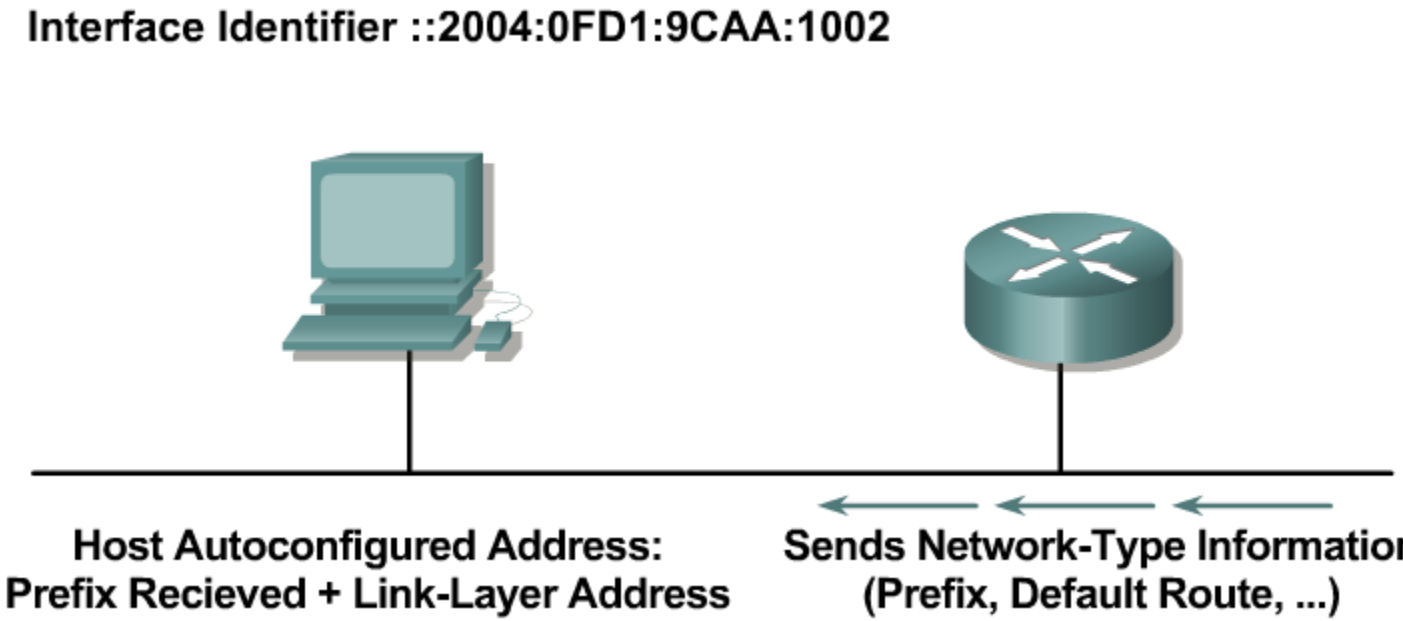


IPv6 Anycast Address



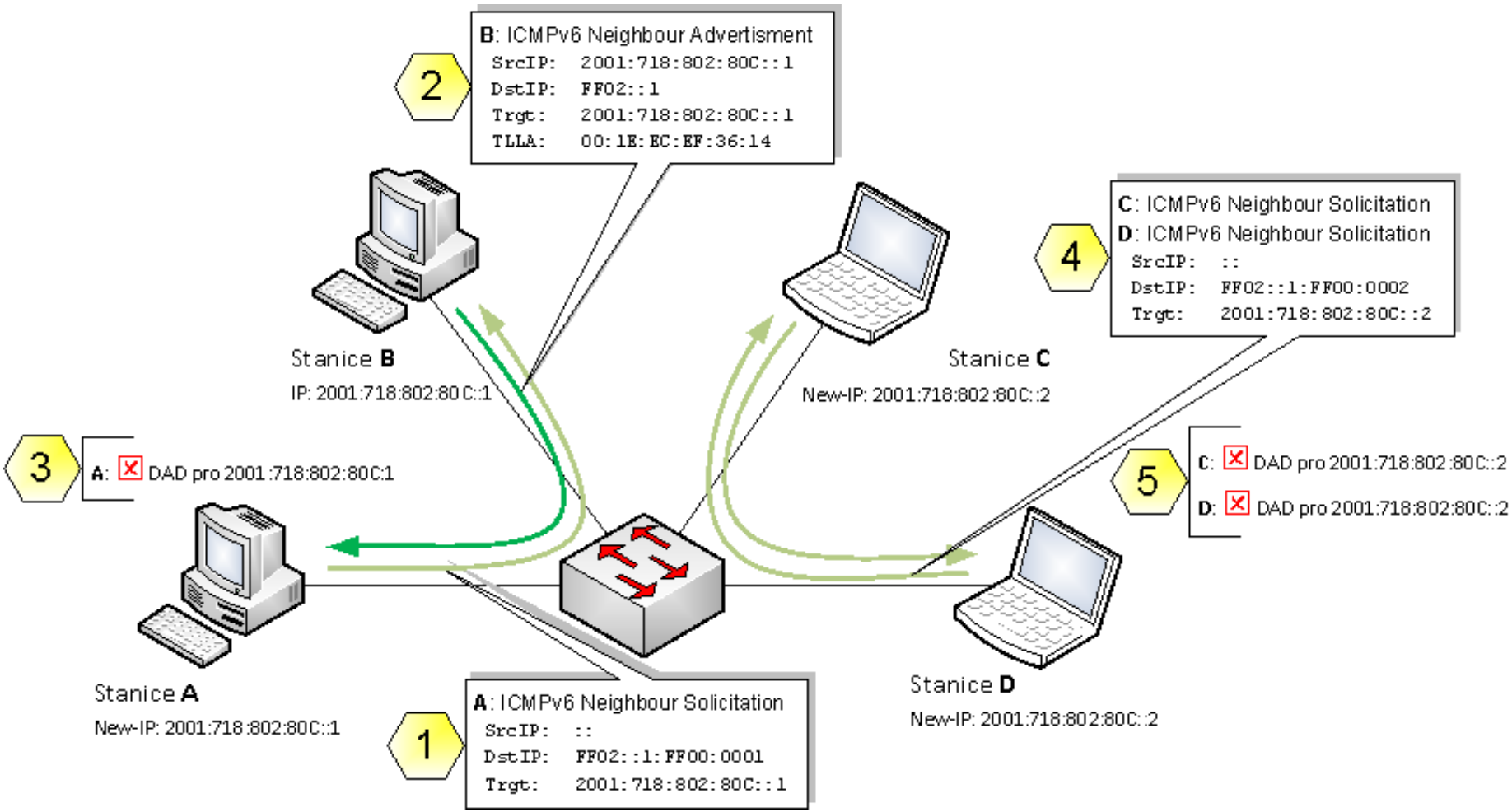
- An IPv6 **anycast address** is a global unicast address that is assigned to more than one interface

Stateless Autoconfiguration ①

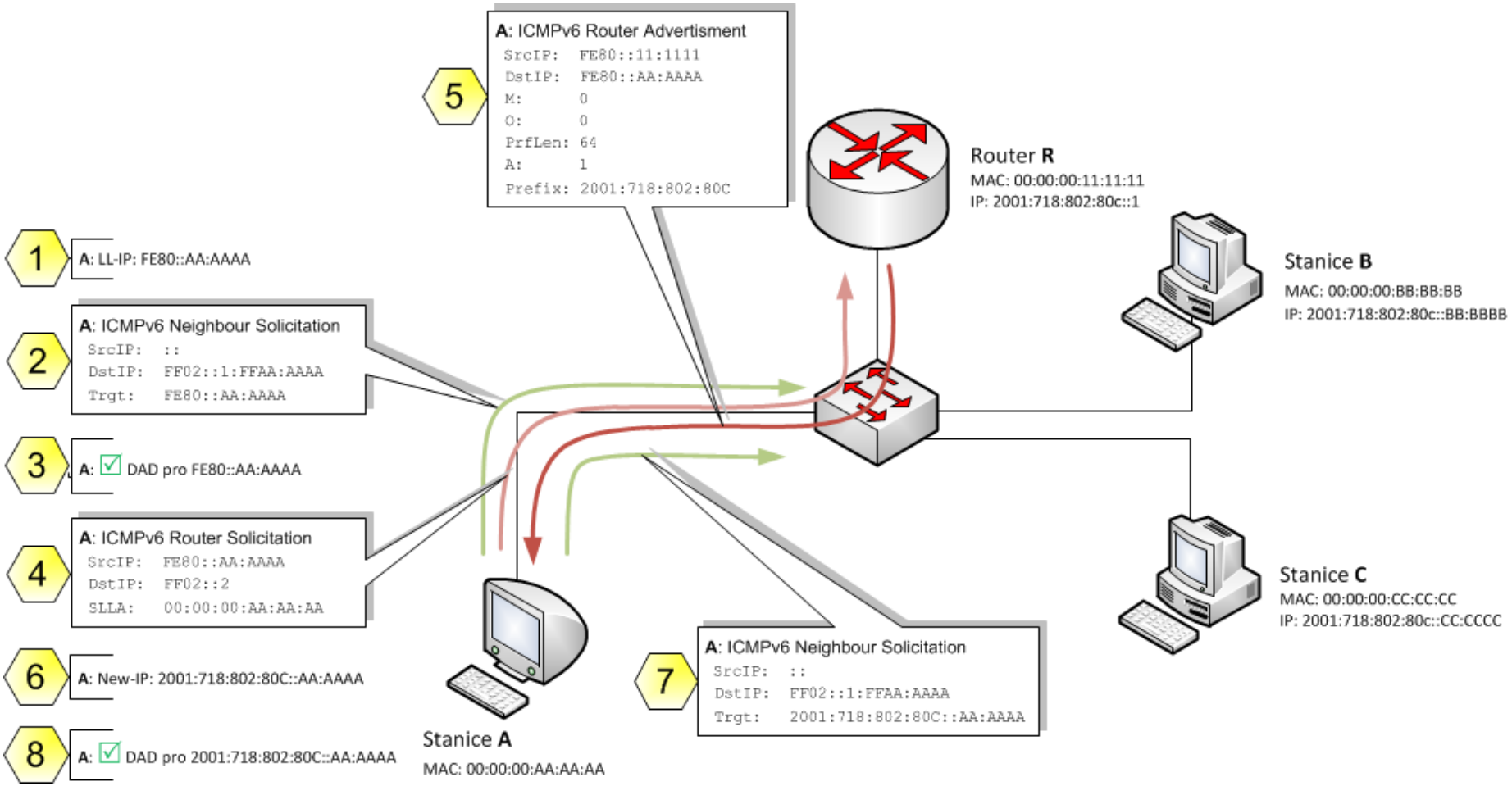


- A router sends network information to all the nodes on the local link (**Router Advertisement messages, RA**)
- A host can autoconfigure itself by appending its IPv6 interface identifier (64-bit format) to the local link prefix (64 bits)
- The result is a full 128-bit address that is usable and guaranteed to be globally unique

Stateless Autoconfiguration ②



Stateless Autoconfiguration ③

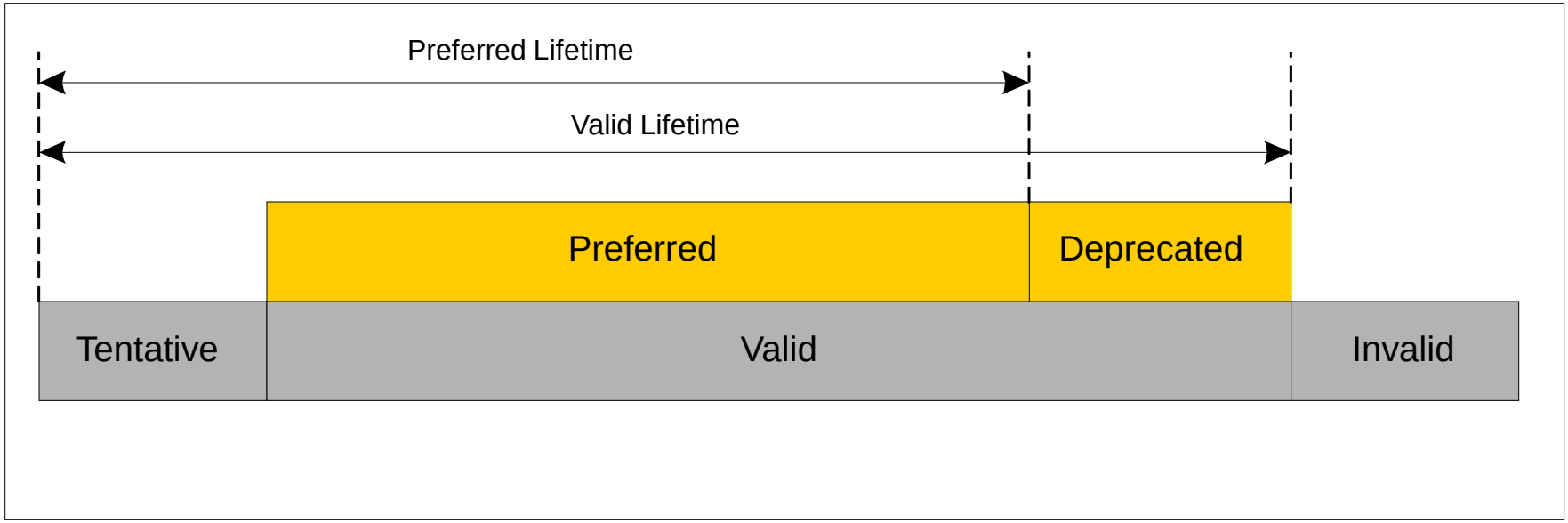


Autoconfigured Addresses States ①

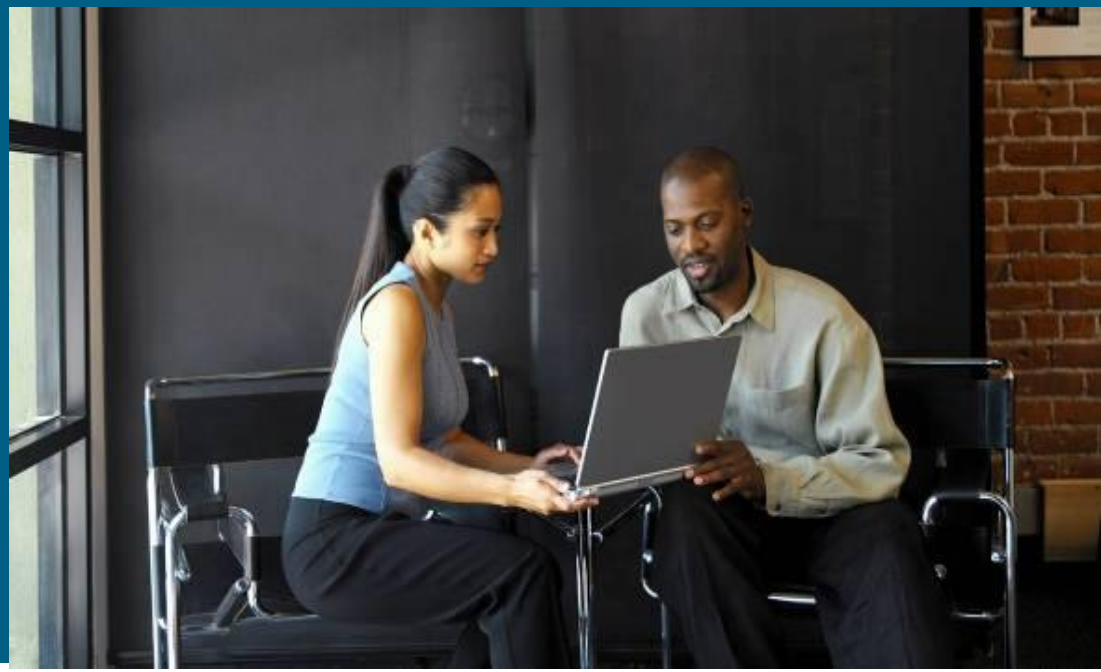
- **Tentative**
  - Duplicate address Detection phase
  - Unicast communication is forbidden
  - Multicast communication – only Neighbor Advertisement
- **Valid**
  - Address is unique and can be used
  - Valid contains another two states: **Preferred** and **Deprecated**
    - Preferred – address is valid
    - Deprecated – address is valid, but can not be used for a new connection, only existing connection can be accepted
- **Invalid**
  - After expiration of Valid Lifetime timer
  - Address can not be used

Autoconfigured Addresses States ②

- Autoconfigured address reset timers if RA message is received
- Autoconfigured addresses are mainly for end stations – not for routers or servers



IPv6 Configuration



Basic Configuration

- IPv6 has to be enabled first

```
Router(config)# ipv6 unicast-routing
Router(config)# ipv6 cef
```

- IPv6 is configured in the same manner as IPv4
  - All commands use **ipv6** instead of **ip**

```
Router(config)# ipv6 route 2000::/3 2001:4118:300:122::1
Router(config)# interface fa0/0
Router(config-if)# ipv6 address 2001:4118:300:123::1/64
```

IPv6 Address Configuration ①

- Static address:

```
Router(config-if)# ipv6 address 2001:4118:300:123::1/64
```

- IPv6 Address using EUI-64:

```
Router(config-if)# ipv6 address 2001:4118:300:123::/64 eui-64
```

- Stateless autoconfiguration (RS/RA):

```
Router(config-if)# ipv6 address autoconfig
```

IPv6 Address Configuration ②

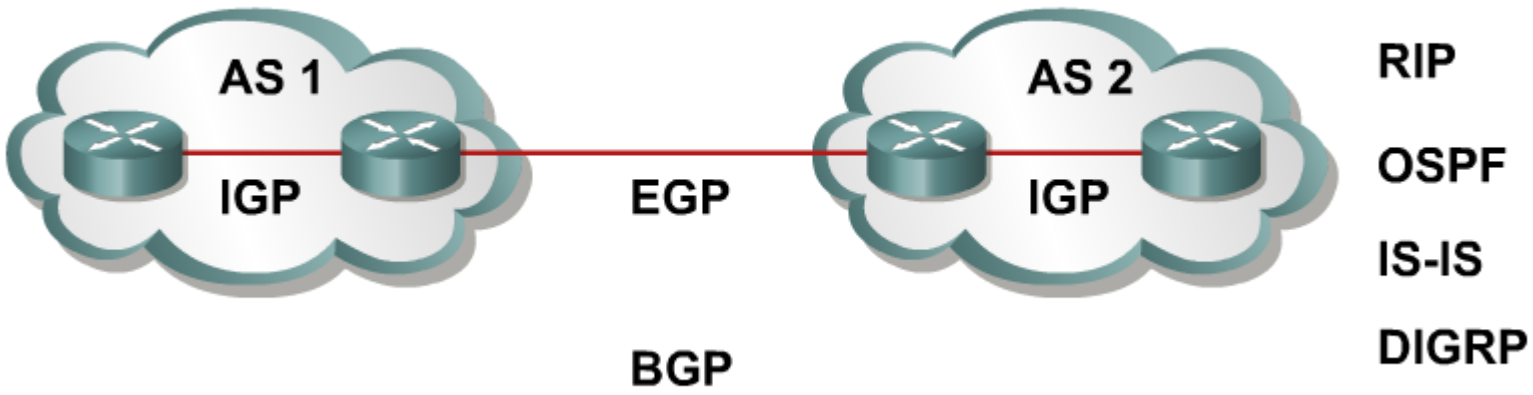
- IPv6 activation on an interface without setting the IPv6 address

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

- The command activate IPv6 support on the interface
  - Interface generates link-local address
  - Can be used e.g. for transit links (similar to IP Unnumbered), because link local addresses are sufficient for routing protocols

IPv6 Routing Protocols

IPv6 Routing Protocols



- IPv6 route types:
  - **Static**
  - **RIPng** (RFC 2080)
  - **OSPFv3** (RFC 2740)
  - **IS-IS** for IPv6
  - **MP-BGP4** (RFC 2545 and RFC 2858)
  - **EIGRP** for IPv6
- All IGP routing protocols are activated on interfaces not with **`network`** command

RIPng

- Same as IPv4
  - Distance-vector, radius of 15 hops, split-horizon and poison reverse
  - Based on RIPv2
- Updated features for IPv6
  - IPv6 prefix, next-hop IPv6 address
  - Uses the multicast group FF02::9, the all-rip-routers multicast group, as the destination address and UDP port 521 for RIP updates
- Does not support **neighbor** command and passive interfaces
- Split-horizon can be disabled only for whole RIPng process

```
Router(config-if)# ipv6 rip NAME enable
```

EIGRP for IPv6

- EIGRP for IPv6 is available since IOS 12.4T
- 4-bytes RouterID is required
  - `IF` RouterID cannot be discovered (no IPv4 address on loopback or interface) `THEN` it has to be specified manually
- Routing protocol is disabled by default, it has to be enabled

```
Router(config)# interface fa0/0
Router(config-if)# ipv6 eigrp 64512
Router(config-if)# exit
Router(config)# ipv6 router eigrp 64512
Router(config-rtr)# router-id 158.193.138.255
Router(config-rtr)# no shutdown
```

OSPF Version 3 (OSPFv3)

- Similar to OSPV for IPv4:
  - Same mechanisms, but a major rewrite of the internals of the protocol
- 4-bytes RouterID is required
  - `IF` RouterID cannot be discovered (no IPv4 address on loopback or interface) `THEN` it has to be specified manually
- Updated features will be described later

```
Router(config)# interface fa0/0
Router(config-if)# ipv6 ospf 1 area 0
Router(config-if)# exit
Router(config)# ipv6 router ospf 1
Router(config-rtr)# router-id 158.193.138.255
```

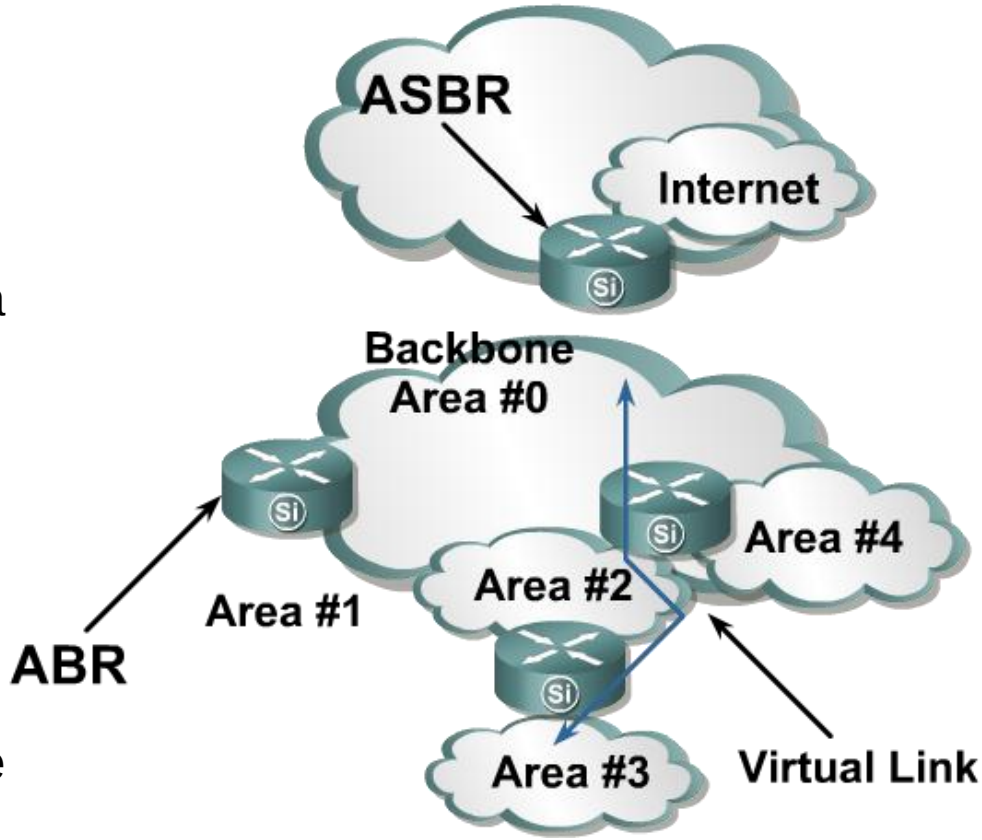
Multiprotocol Border Gateway Protocol (MP-BGP)

- Enables protocols other than IPv4
- New identifier for the address family
- IPv6 specific extensions:
  - NEXT_HOP contains a global IPv6 address and potentially a link-local address (only when there is a link-local reachability with the peer).
  - NEXT_HOP and NLRI (Network Layer Reachability Information) are expressed as IPv6 addresses and prefix in the multiprotocol attritubes

OSPFv3

Hierarchical Structure

- Autonomous system is divided into areas
- Topology of an area is invisible from outside of the area:
  - LSA flooding is bounded by area
  - SPF calculation is performed separately for each area
- Backbones must be contiguous
- All areas must have a connection to the backbone:
  - Otherwise a virtual link must be used to connect to the backbone



Similarities with OSPFv2 ①

- OSPFv3 is OSPF for IPv6 (RFC 2740):
  - Based on OSPFv2, with enhancements
  - Distributes IPv6 prefixes
  - Runs directly over IPv6
- OSPFv3 & v2 can be run concurrently, because each address family has a separate SPF
- OSPFv3 uses the same basic packet types as OSPFv2:
  - Hello
  - Database description blocks (DDB)
  - Link state request (LSR)
  - Link state update (LSU)
  - Link state acknowledgement (LSAck)

Similarities with OSPFv2 ②

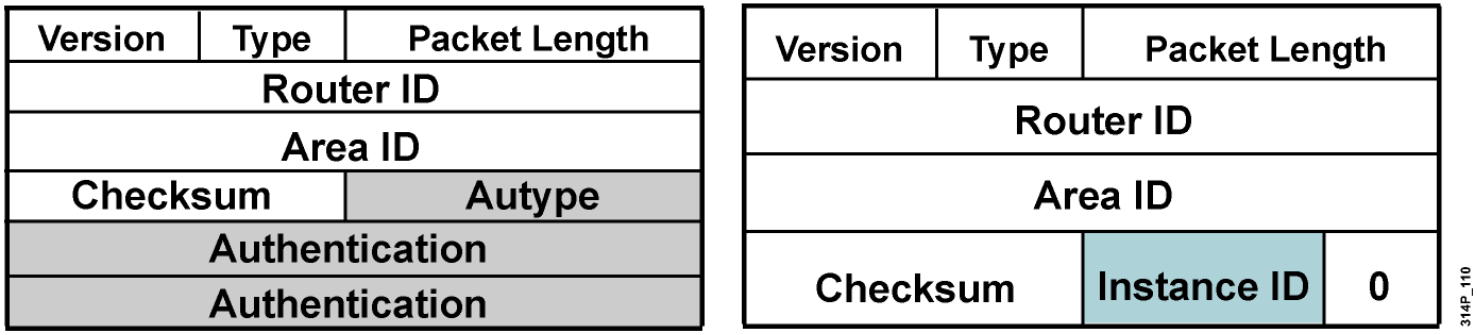
- Neighbor discovery and adjacency formation mechanism is identical
- RFC compliant NBMA and point-to-multipoint topology modes are supported – also supports other modes from Cisco such as point-to-point and broadcast, including the interface
- LSA flooding and aging mechanisms are identical

Differences from OSPFv2 ①

- OSPFv3 has the same five packet types, but some fields have been changed.

Packet Type	Description
1	Hello
2	Database Description
3	Link State Request
4	Link State Update
5	Link State Acknowledgement

- All OSPFv3 packets have a 16-byte header versus the 24-byte header in OSPFv2.



Differences from OSPFv2 ②

- OSPFv3 protocol processing per-link, not per-subnet
- IPv6 connects interfaces to links
- Multiple IPv6 subnets can be assigned to a single link
- Two nodes can talk directly over a single link, even though they do not share a common subnet
- The terms **“network”** and **“subnet”** are being replaced with **“link”**
- An OSPF interface now connects to a link instead of a subnet

Differences from OSPFv2 ③

- Multiple OSPFv3 protocol instances can now run over a single link:
  - This allows for separate autonomous systems, each running OSPF, to use a common link. A single link could belong to multiple areas
  - Instance ID is a new field that is used to have multiple OSPFv3 protocol instances per link
  - In order to have two instances talk to each other, they need to have the same instance ID. By default it is 0, and for any additional instance it is increased

Differences from OSPFv2 ④

- Multicast addresses:
  - FF02::5 – Represents all OSPF routers on the link local scope; equivalent to 224.0.0.5 in OSPFv2
  - FF02::6 – Represents all DR routers on the link local scope; equivalent to 224.0.0.6 in OSPFv2
- Removal of address semantics:
  - IPv6 addresses are no longer present in OSPF packet header (part of payload information)
  - Router LSA (LSA1) and network LSA (LSA2) do not carry IPv6 addresses
  - Router ID, area ID, and link-state ID remains at 32 bits
  - DR and BDR are now identified by their router ID and no longer by their IP address
- Security:
  - OSPFv3 uses IPv6 AH and ESP extension headers instead of variety of mechanisms defined in OSPFv2

LSA Overview

	LSA Function Code	LSA Type
Router-LSA	1	0x2001
Network-LSA	2	0x2002
Inter-Area-Prefix-LSA	3	0x2003
Inter-Area-Router-LSA	4	0x2004
AS-External-LSA	5	0x4005
Group-Membership-LSA	6	0x2006
Type-7-LSA	7	0x2007
Link-LSA	8	0x2008
Intra-Area-Prefix-LSA	9	0x2009

Renamed

New

OSPFv3 Configuration

Configuration

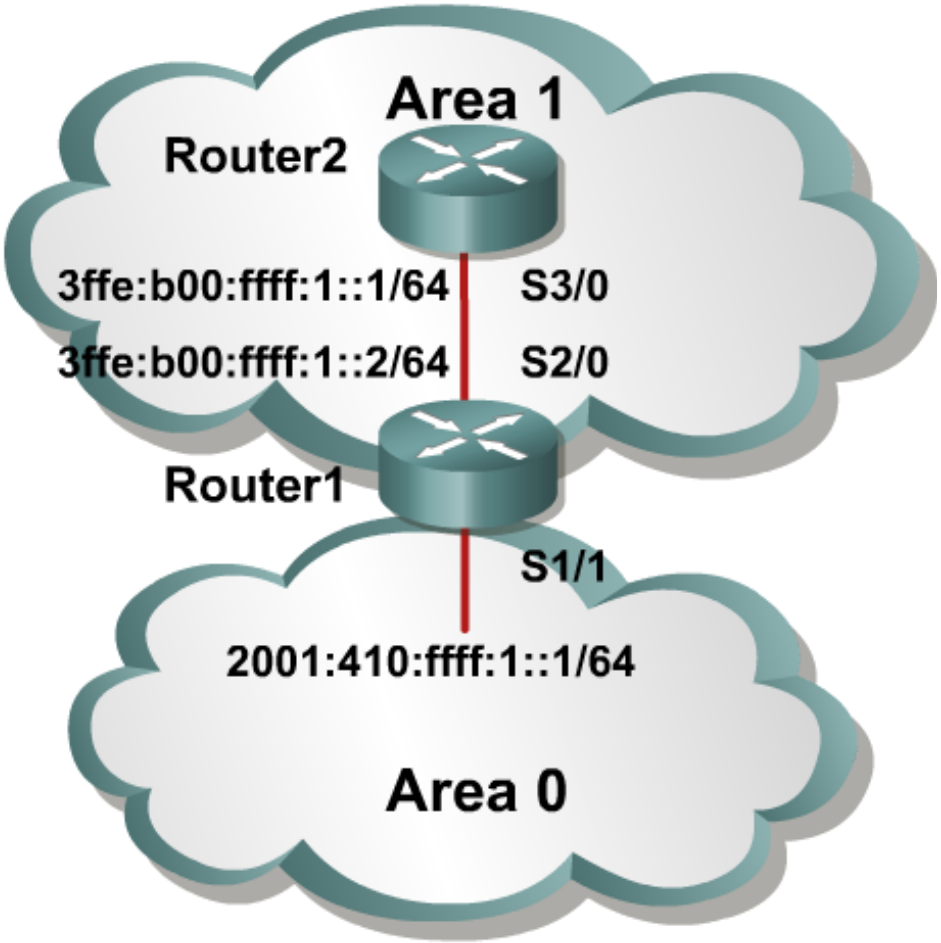
- Configuration is done on interface – similar to OSPFv2
  - **ipv6** is used instead of **ip** command
  - Replaces **network** command
- Native IPv6 router OSPF mode
  - Interface is assigned to OSPFv3 process with command:

```
Router(config)# ipv6 router ospf process-id area X
```

  - Used for area configurations, e.g. **router-id** …
  - The most common commands are same as in OSPFv2 – summarization, area type, redistribution, etc.

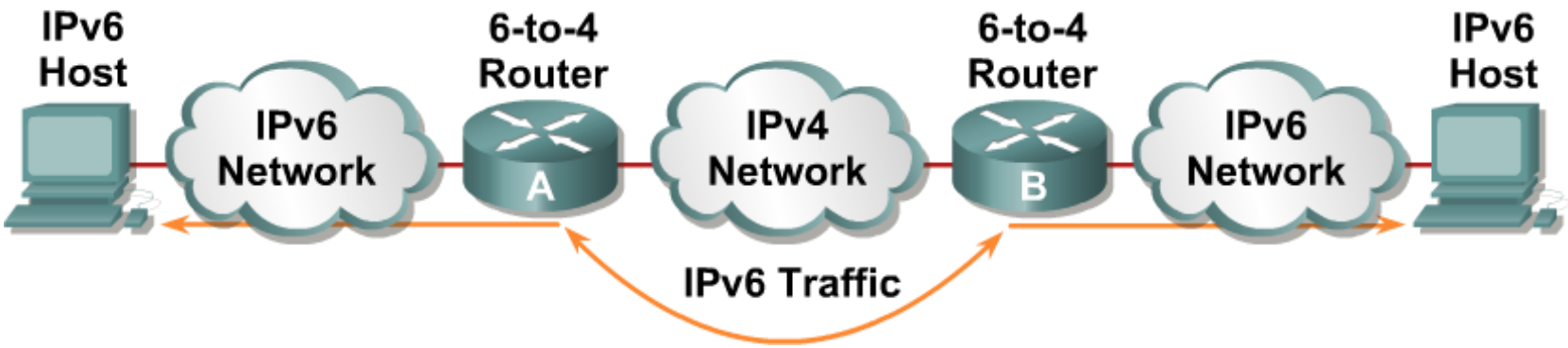
Configuration Example

```
Router1(config)#
ipv6 unicast-routing
interface S1/1
 ipv6 address 2001:410:FFFF:1::1/64
 ipv6 ospf 100 area 0

interface S2/0
 ipv6 address 3FFE:B00:FFFF:1::2/64
 ipv6 ospf 100 area 1

ipv6 router ospf 100
 router-id 10.1.1.3


Router2(config)#
ipv6 unicast-routing
interface S3/0
 ipv6 address 3FFE:B00:FFFF:1::1/64
 ipv6 ospf 100 area 1

ipv6 router ospf 100
 router-id 10.1.1.4
```



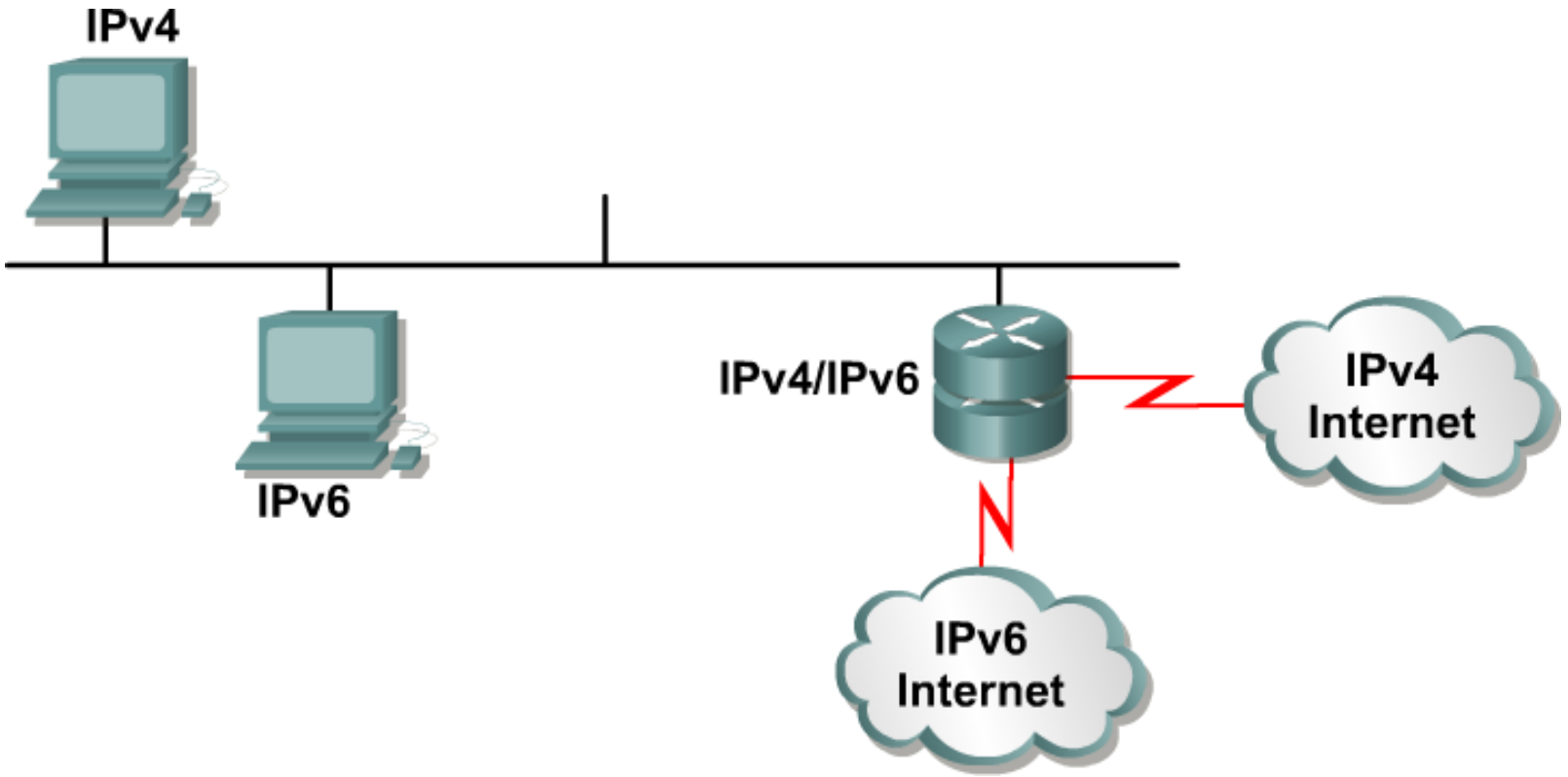
Using IPv6 with IPv4

IPv4-to-IPv6 Transition



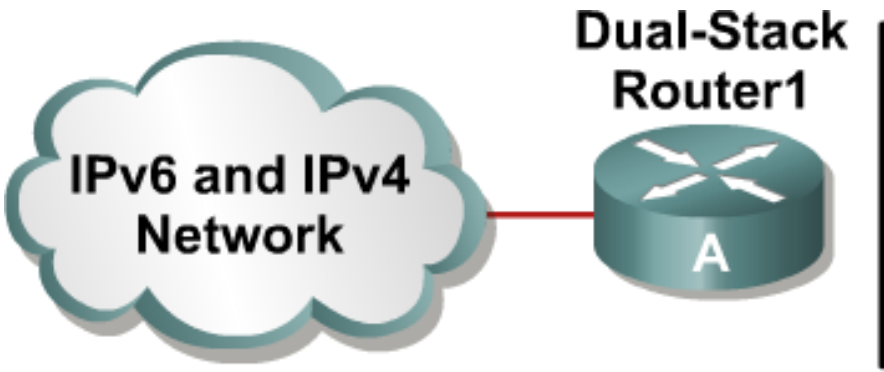
- *No fixed day to convert; no need to convert all at once!*
- Different transition mechanisms are available:
  - Dual-stack
  - Static tunnels, 6over4 tunnels (RFC 2529)
  - 6to4 tunenels (RFC 3056)
  - ISATAP tunnels (RFC 4214)
  - Teredo tunnels (RFC 4380)
  - NPTv6 (Protocol Translation)

Dual Stack



- **Dual stack** is an integration method where a node has "implementation and connectivity" to both an IPv4 and IPv6 network.

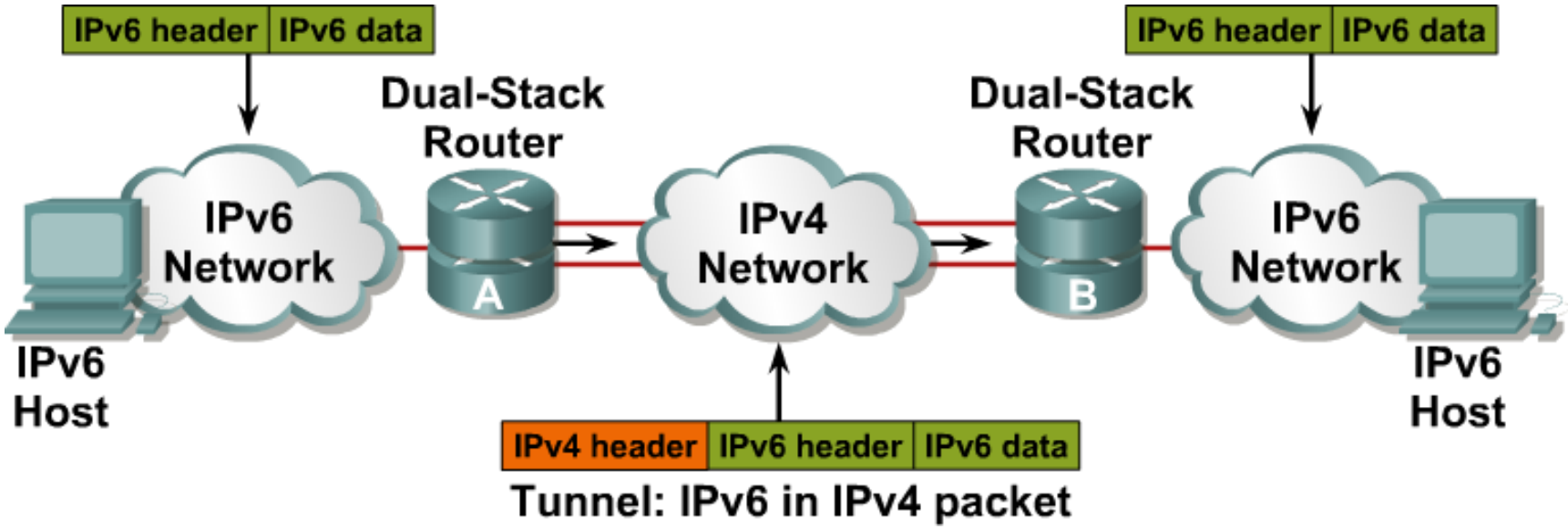
Dual Stack: Configuration




```
conf t
ipv6 unicast-routing

interface ehternet0
 ip address 192.168.99.1 255.255.255.0
 ipv6 address 3ffe:b00:c18:1::3/127
```

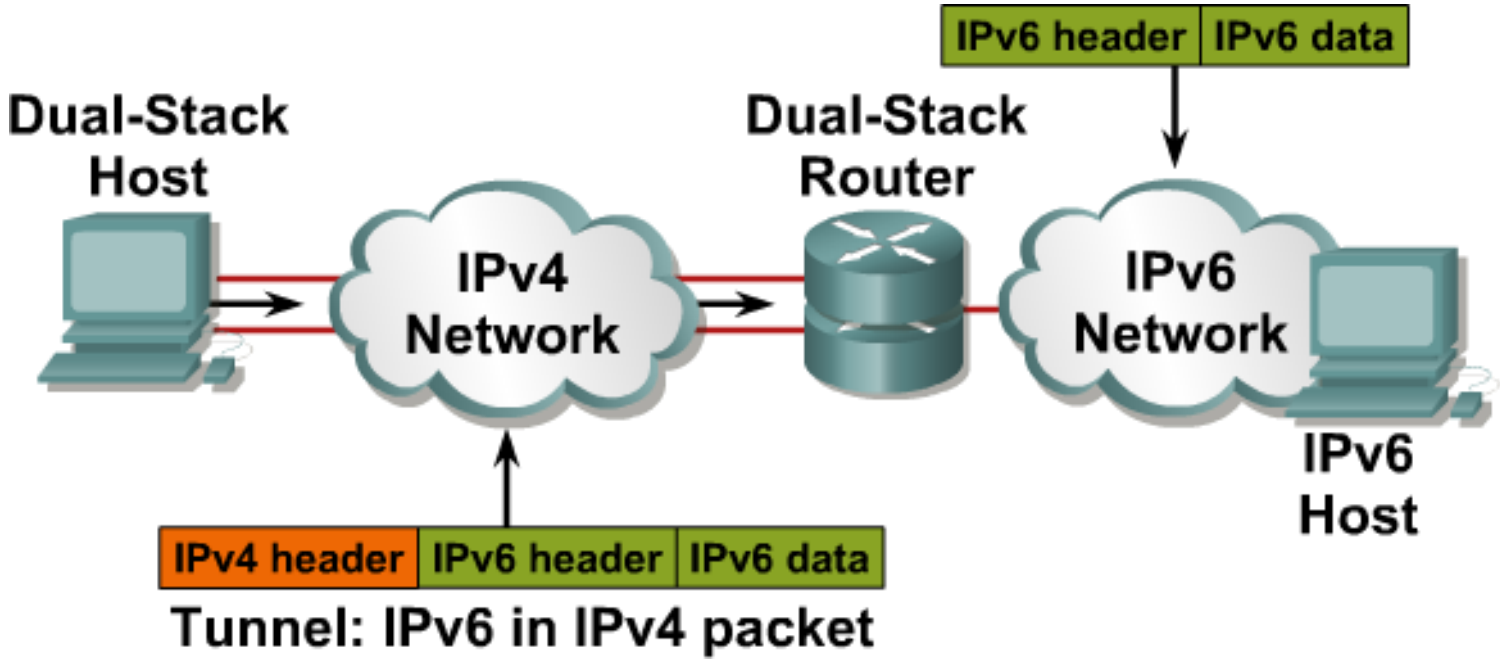
- IF both IPv4 and IPv6 are configured on an interface, THEN this interface is dual-stacked.

Tunneling IPv6 in IPv4 ①



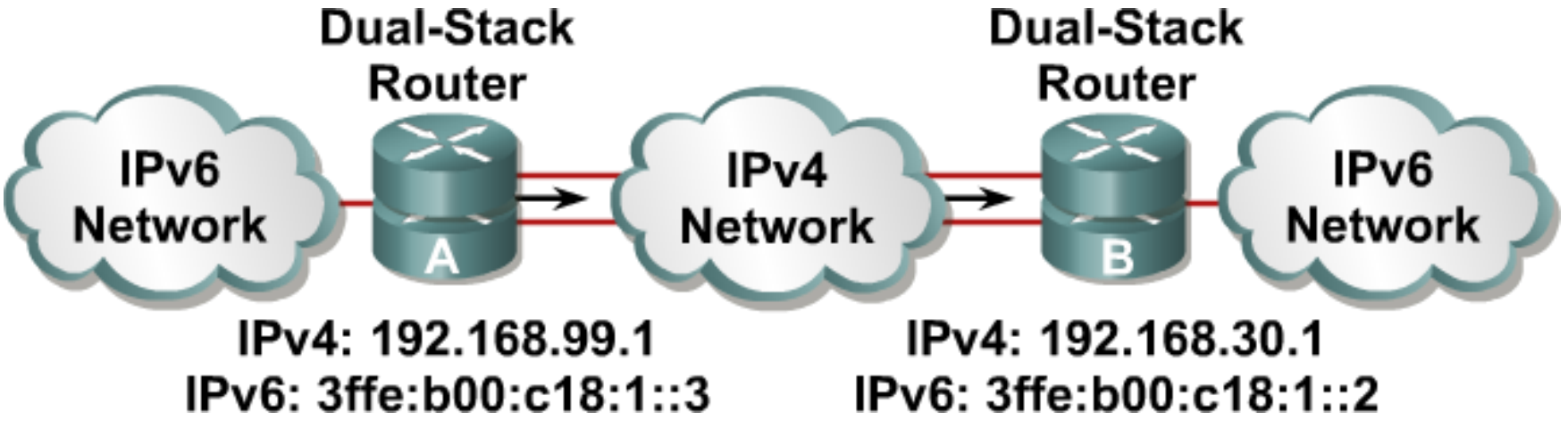
- Tunneling is an integration method where an IPv6 packet is encapsulated within another protocol, such as IPv4. This method of encapsulation is IPv6-in-IPv4 protocol 41:
  - This includes a 20-byte IPv4 header with no options and an IPv6 header and payload
  - Dual stack routers are necessary

Tunneling IPv6 in IPv4 ②



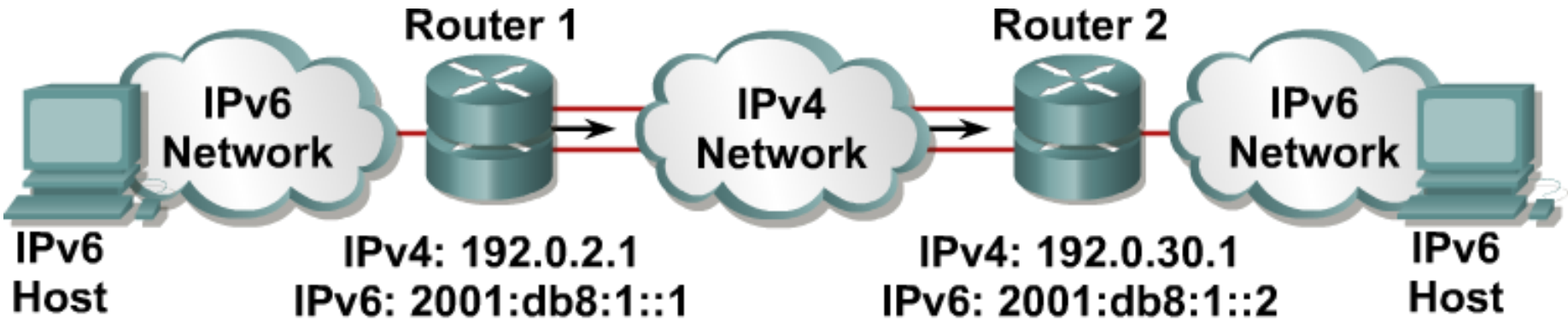
- Encapsulation can be done by edge routers between hosts or between a host and a router

Configuring Tunnel



- Configuring tunnels requires:
  - Dual-stack endpoints
  - IPv4 and IPv6 addresses configured at each end

Example: Tunnel configuration




```
Router1#

 interface Tunnel0
  ipv6 address 2001:db8:1::1/64
  tunnel source 192.0.2.1
  tunnel destination 192.0.30.1
  tunnel mode ipv6ip
```

```
Router2#

 interface Tunnel0
  ipv6 address 2001:db8:1::2/64
  tunnel source 192.0.30.1
  tunnel destination 192.0.2.1
  tunnel mode ipv6ip
```

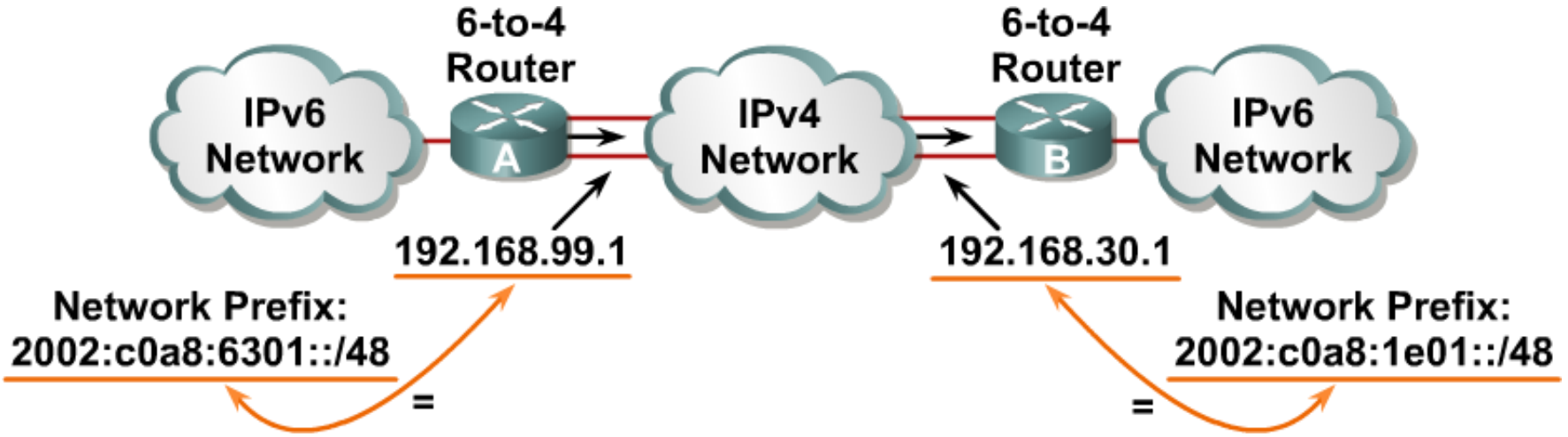
6to4 Tunnels ①

- **6to4** tunnel establishes a transient link between IPv6 domains which are connected by an IPv4 backbone
- Public IPv4 address is necessary – it is hardcoded into the IPv6 address, so the IPv6 address is globally unique
  - 6to4 addresses use prefix 2002::/16
  - The next 32 bits are the original source address in hexadecimal form
  - Subnet ID can be 16 bits long, Interface ID is 64bits as usual

6to4 Tunnels ②

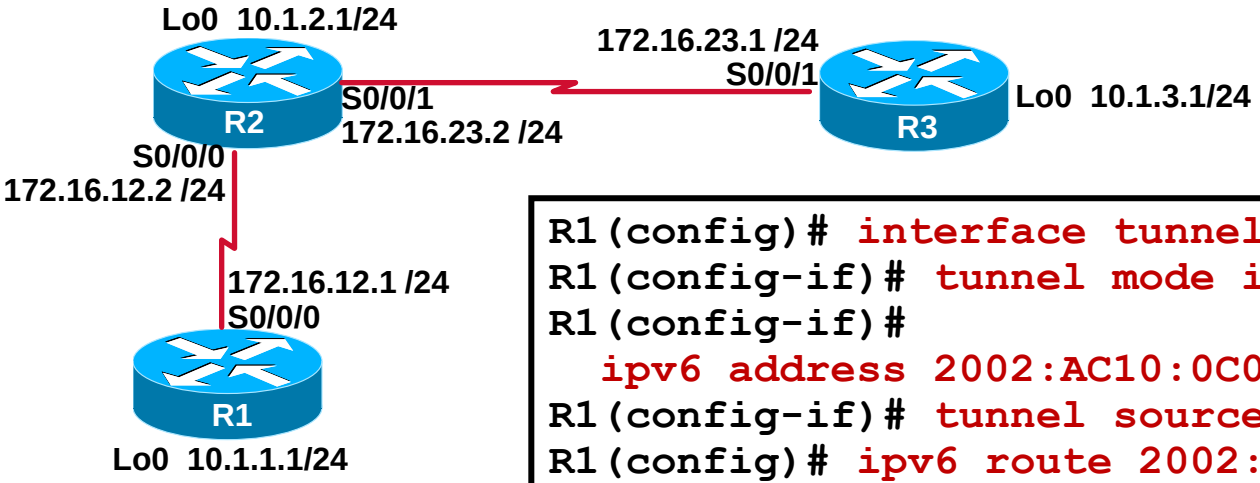
- Example:
  - Public IPv4 address of a router is 192.0.2.36
  - IPv4 address converted to hex: C0:0:2:24
  - IPv6 devices connected in a domain have network prefix 2002:C000:0224/48
  - Other routers have to be configured to use 6to4 tunnel if a device tries to access IPv6 network with 2002::/16 prefix
- Configuration:
  - Create a tunnel interface
  - Set tunnel mode with the **`tunnel mode ipv6ip 6to4`** command
  - Creating IPv6 addressing plan and set IPv6 addresses on interfaces
  - Create an IPv6 specific address for the tunnel
  - Configure IPv6 routing through the tunnel

6to4 Tunnels ③



- 6to4:
  - Anycast IPv4 address 192.88.99.1 is used as the tunnel endpoint
  - Every IPv6 domain cloud has unique global IPv6 prefix

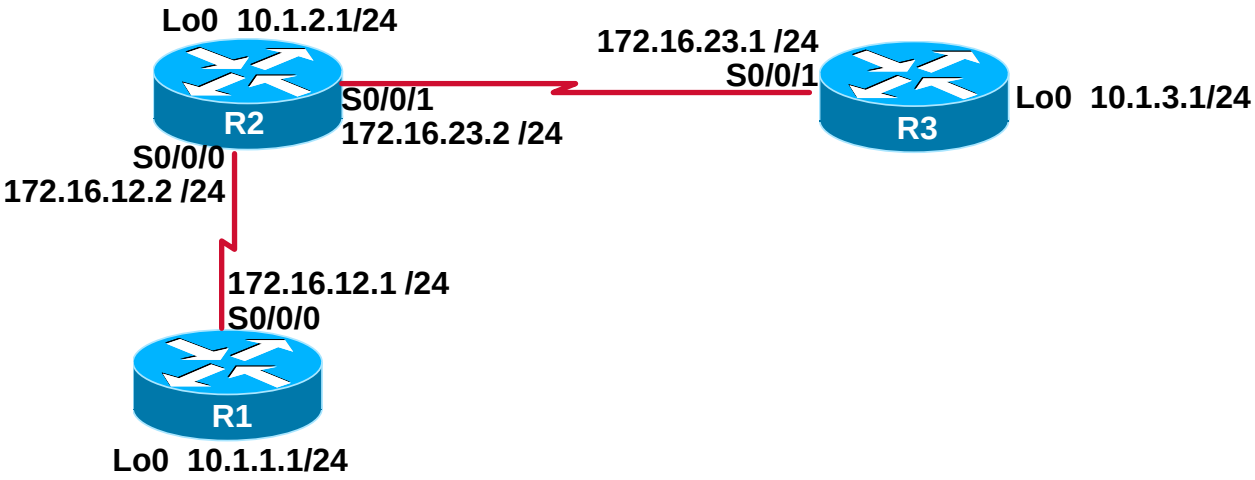
Configuration: 6to4 tunnel




```
R1(config)# interface tunnel 0
R1(config-if)# tunnel mode ipv6ip 6to4
R1(config-if)#
  ipv6 address 2002:AC10:0C01:1::1/64
R1(config-if)# tunnel source s0/0/0
R1(config)# ipv6 route 2002::/16 tunnel 0

Verification
R1# ping 2002:AC10:1701:1::3
```

```
R3(config)# interface tunnel 0
R3(config-if)# tunnel mode ipv6ip 6to4
R3(config-if)#
  ipv6 address 2002:AC10:1701:1::3/64
R3(config-if)# tunnel source s0/0/1
R3(config)# ipv6 route 2002::/16 tunnel 0

Verification
R3# ping 2002:AC10:0C01:1::1
```

Configuration: Static Tunnel




```
Router1(config)# ipv6 unicast-routing
Router1(config)# ipv6 route FEC0::3:0/112 2002:AC10:1701:1::3
```

```
Router3(config)# ipv6 unicast-routing
Router3(config)# ipv6 route FEC0::1:0/112 2002:AC10:C01:1::1
```

- The next hop for both routers is the IPv6 address at the other end of the tunnel

6to4 Relay: Connection to the IPv6 World

- 6to4 can be used not as a static tunnel but as a connection to the IPv6 world
  - Gateway has to accept 6to4 tunnels
  - All IPv6 traffic will be forwarded through the gateway
- Providers can create their own 6to4 relay as specified in the RFC 3068
  - Gateway must have IPv4 address 192.88.99.1
  - Routers use static route to the IPv6 world (only static routes and BGP can be used)

```
Router(config)# ipv6 route 2000::/3 2002:c058:6301::
```

ISATAP Tunnels

- An **Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)** tunnel is very similar to a 6to4 IPv6 tunnel
  - It embeds an IPv4 address within the IPv6 address
  - ISATAP was designed to transport IPv6 packets within a site
  - ISATAP IPv6 address structure:
    - First 64 bits: link-local or global unicast prefix
    - Next 32 bits: 0000:5EFE constant
    - Next 32 bits: IPv4 address of the station
  - Routers used as a gateways can be locate automatically using DHCP and DNS
    - DHCP offers domain name, host than query DNS server for `isatap.domain.tld` record

Conclusion

IPv4 and IPv6 Comparison ①

IPv4	IPv6
Address is 32 bits long	Address is 128 bits long
IPsec support is optional	IPsec support is mandatory (not in real life)
No flow identification for QoS in the header	Flow label field in the IPv6 header – How should be interpreted?
Fragmentation is done by routers and end nodes	Fragmentation is done only by end nodes – routers however have to reassembly the header for packet filtering
Header contains checksum for packet integrity checks	Checksum is removed from the header
Header contains optional options field	Extension headers are used instead of options. No options field in the basic header.
IP to MAC translation is done by ARP protocol	IP to MAC translation is done by ICMPv6 protocol using multicast addresses (not broadcast)

IPv4 and IPv6 Comparison ②

IPv4	IPv6
Multicast groups are managed by IGMP protocol	IGMP is replaced by MLD protocol
Default gateway is configured usually using DHCP	ICMPv6 Router Solicitation and Router Advertisement are used for the default gateway discovery.
Broadcast address is used for sending packets to all nodes.	Broadcast address does not exists. Multicast address link-local scope all-nodes is replacement.
Manual or DHCP address configuration	Autoconfiguration can be used (however together with DHCPv6 or manual configuration of DNS resolvers)

Slides adapted by Matěj Grégr (and extended by Vladimír Veselý)
partially from official course materials
but the most of the credit goes to CCIE#23527 Ing. Peter Palúch, Ph.D.

Last update: 2012-08-19