CS 540
Computer Networks II

Sandy Wang
sandy.w@svuca.edu

4. IPV6

Topics

1. Overview
2. LAN Switching
3. IPv4
4. **IPv6**
5. Tunnels
6. Routing Protocols -- RIP, RIPng
7. Routing Protocols -- OSPF
8. IS-IS
9. Midterm Exam
10. BGP
11. MPLS
12. Transport Layer -- TCP/UDP
13. Congestion Control & Quality of Service (QoS)
14. Access Control List (ACL)
15. Final Exam

Reference Books

- **Routing TCP/IP Volume I, 2nd Edition** by Jeff Doyle and Jennifer Carroll

  ISBN: 1-57870-089-2
- **Routing TCP/IP Volume II** by Jeff Doyle and Jennifer DeHaven
  ISBN: 1-57870-089-2
- **Cisco CCNA Routing and Switching ICND2 200-101 Official Cert Guide, Academic Edition** by Wendel Odom -- July 10, 2013**.**
  ISBN-13: 978-1587144882
- **The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference** by Charles M. Kozierok – October 1, 2005.
  ISBN-13: 978-1593270476
- **CCNA Routing and Switching 200-120 Network Simulator.** By Wendell Odom, Sean Wilkins. Published by Pearson IT Certification.
- http://class.svuca.edu/~sandy/class/CS540/

Topics:

- IPv6 Protocol
- ICMPv6
- Neighbor Discovery Protocol (NDP)

- Understand the shortcomings of IPv4
- Know the IPv6 address format, address types, and abbreviations
- Be familiar with the IPv6 header format
- Know the extension header types
- Know the differences between ICMPv4 and ICMPv6
- Know the strategies for transitioning from IPv4 to IPv6

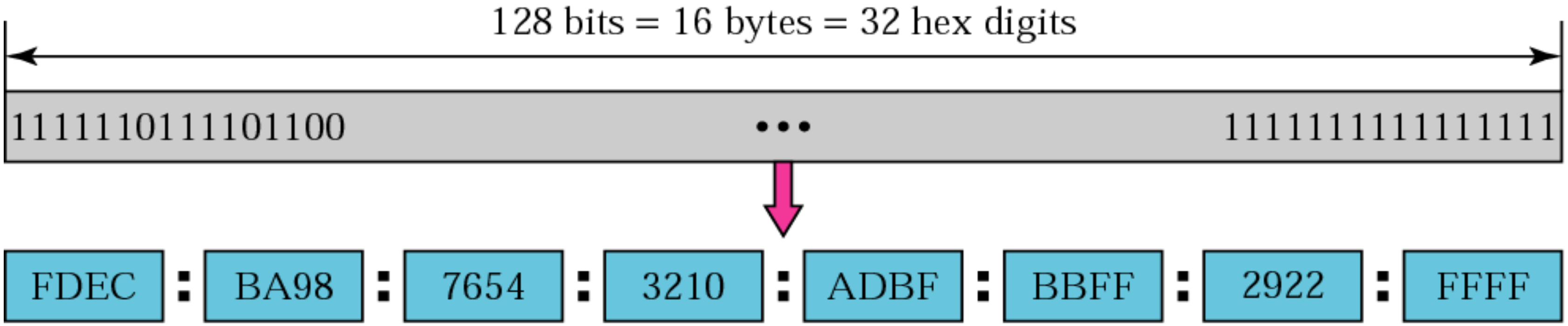
IPv6 Advantages Over IPv4

- Larger address space
- Better header format
- New options
- Allowance for extension
- Support for resource allocation
- Support for more security

Auto-configuration

- Obtain an Interface ID that is unique on the link to which the host is attached -- link layer addresses used.
- Obtain correct prefix -- router periodically advertises.
- Put them together.

IPv6 address



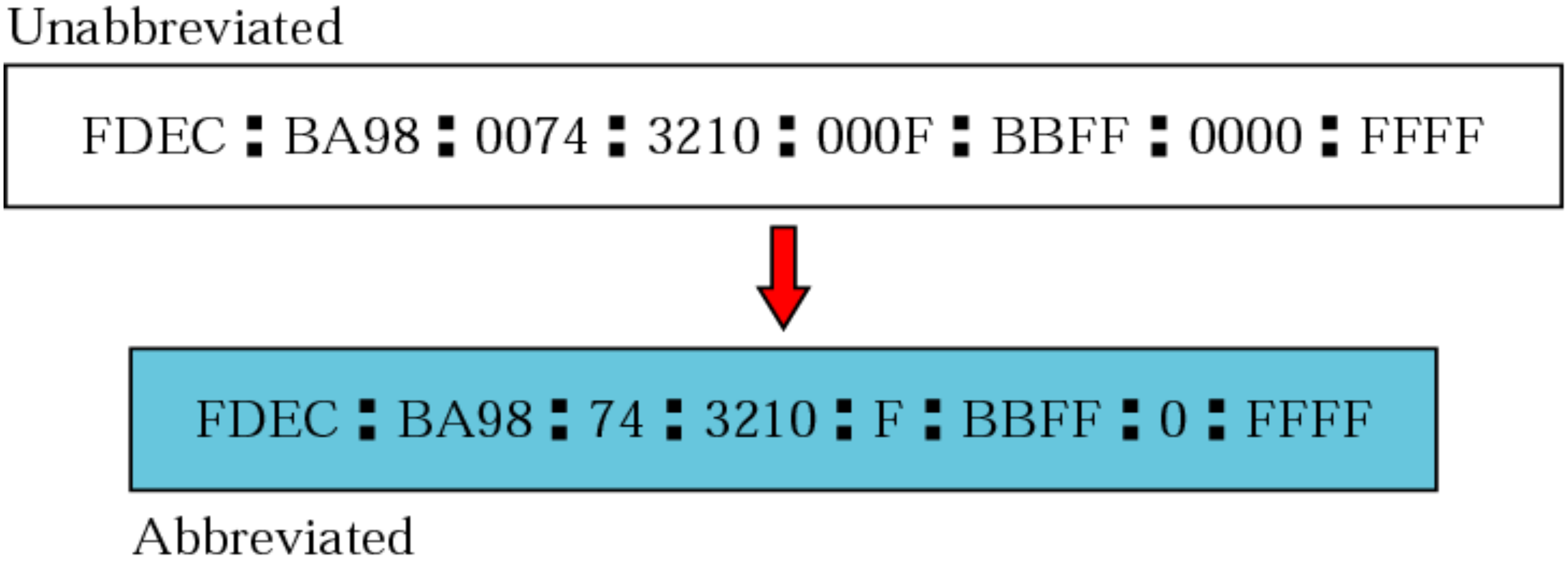
128 bits = 16 bytes = 32 hex digits

1111110111101100 ••• 1111111111111111

FDEC : BA98 : 7654 : 3210 : ADBF : BBFF : 2922 : FFFF

$$2^{128} = 2^{8} \text{ x } (2^{10})^{12} = 256 \text{ x } (1024)^{12} \approx 256\underbrace{,000,000,\ldots\ldots,000}_{\text{36 zeros}}$$

Abbreviated address



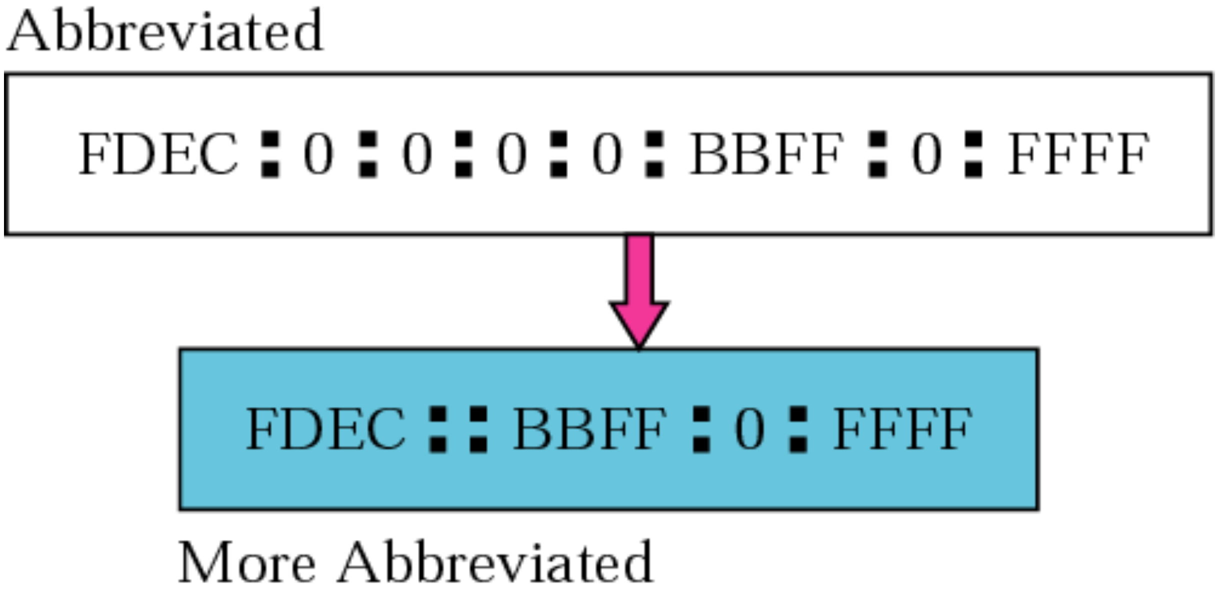
Unabbreviated

FDEC : BA98 : 0074 : 3210 : 000F : BBFF : 0000 : FFFF

FDEC : BA98 : 74 : 3210 : F : BBFF : 0 : FFFF

Abbreviated

Abbreviated address with consecutive zeros



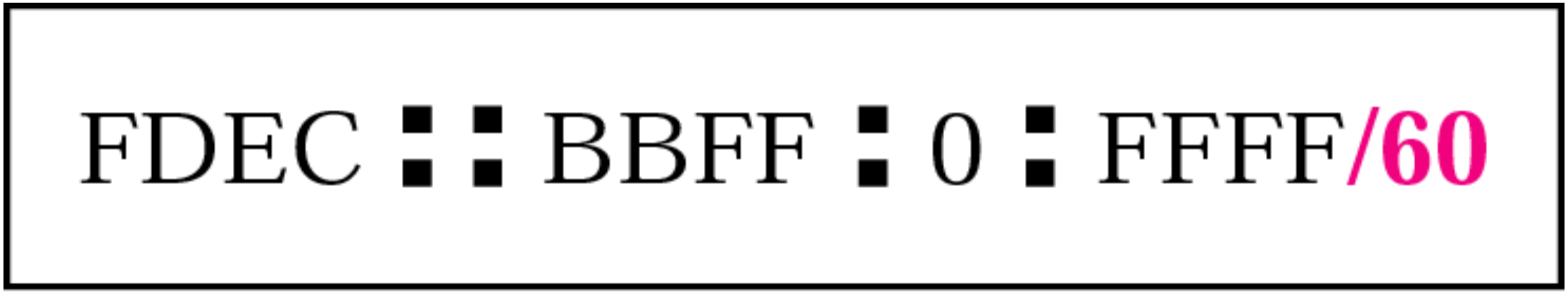
Abbreviated

FDEC:0:0:0:0:BBFF:0:FFFF

FDEC::BBFF:0:FFFF

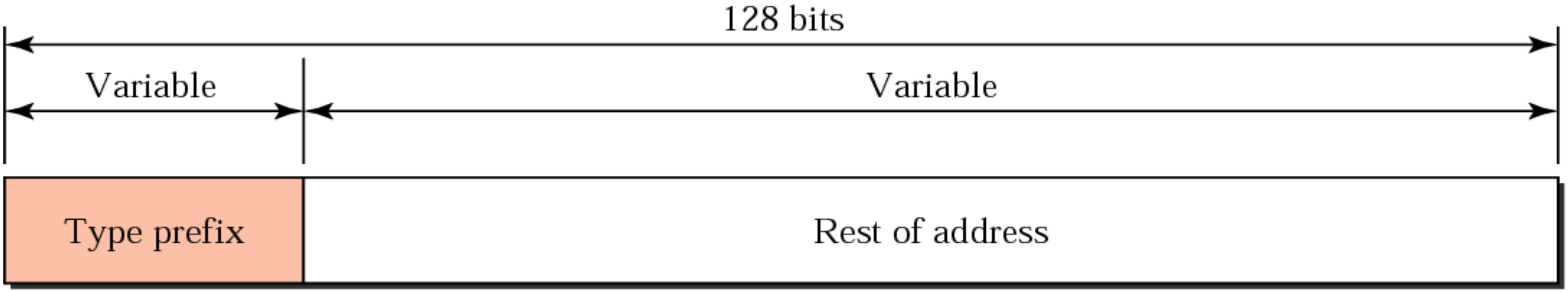
More Abbreviated

CIDR address – Classless Inter-Domain Routing



FDEC::BBFF:0:FFFF/60

Address structure



Address Type Identification – RFC 4291

Address type	Binary Prefix	IPv6 Notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	1111 1111	FF00::/8 224.0.0.0/4
Link Local Unicast	1111 1110 10	FE80::/10
Global Unicast		

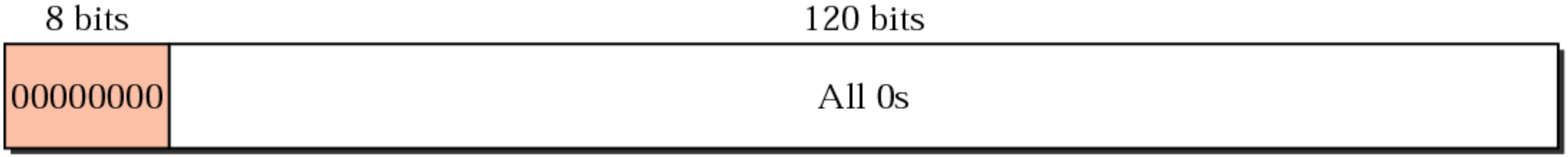
Special Use IPv4 Addresses – RFC 5735

Address type	Address	Description
	0.0.0.0/8	“this network”
Loopback	127.0.0.0/8	127.0.0.0/8
Private Address	10.0.0.0/8	2^{24} → 16 M
	172.16.0.0/12	2^{20} → 1 M
	192.168.0.0/16	2^{16} → 64 K

Type prefixes for IPv6 addresses

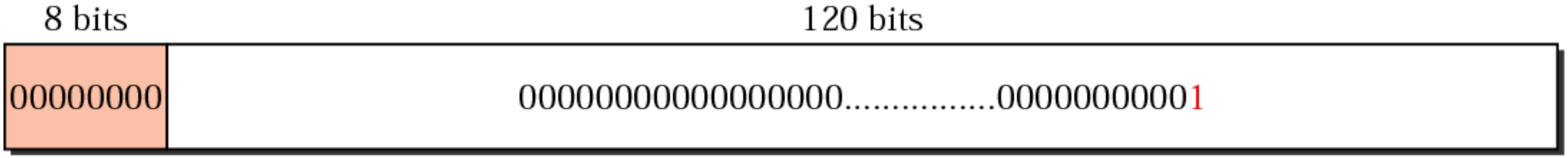
Type Prefix	*Type*	*Fraction*
010	**Provider-based unicast addresses**	**1/8**
011	Reserved	1/8
100	Geographic unicast addresses	1/8
101	Reserved	1/8
110	Reserved	1/8
1110	Reserved	1/16
1111 0	Reserved	1/32
1111 10	Reserved	1/64
1111 110	Reserved	1/128
1111 1110 0	Reserved	1/512
1111 1110 10	Link local addresses	1/1024
1111 1110 11	Site local addresses	1/1024
1111 1111	Multicast addresses	1/256

Unspecified address

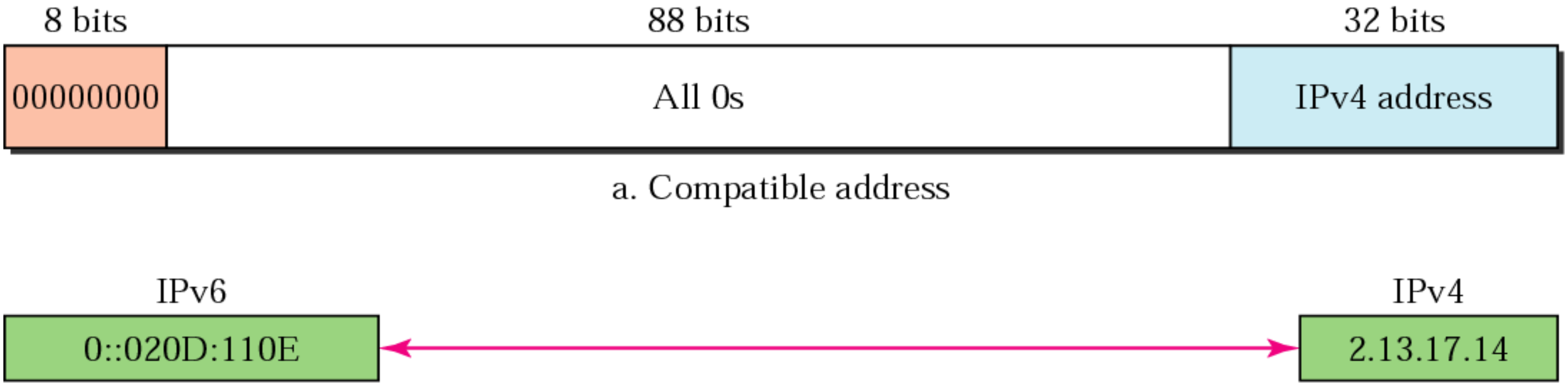


8 bits	120 bits
00000000	All 0s

Loopback address



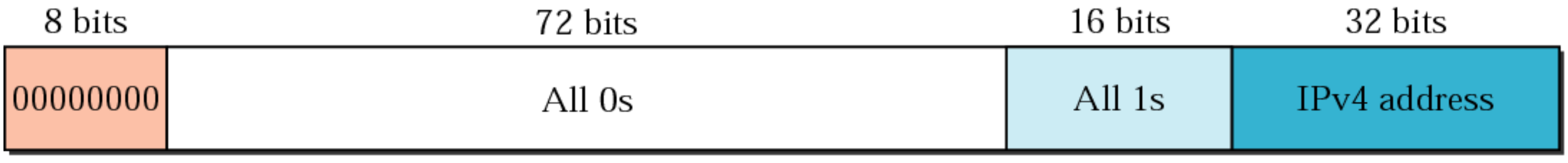
Compatible address



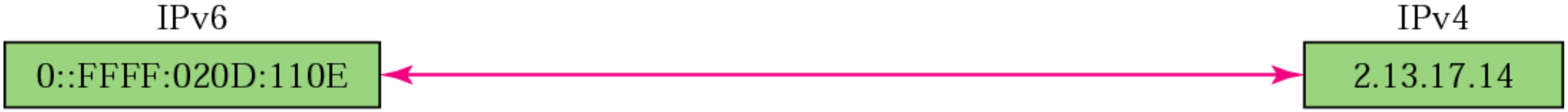
a. Compatible address

b. An example of address transformation

Mapped address

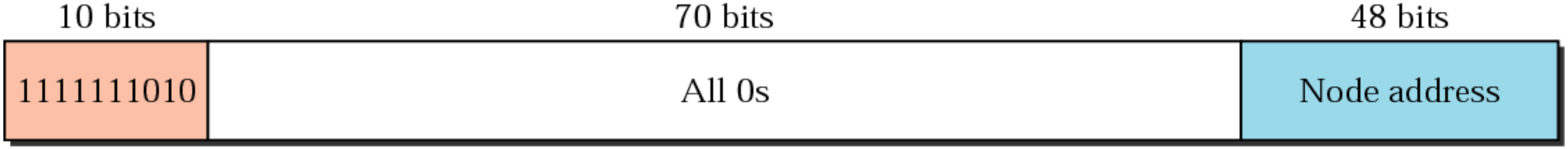


a. Mapped address



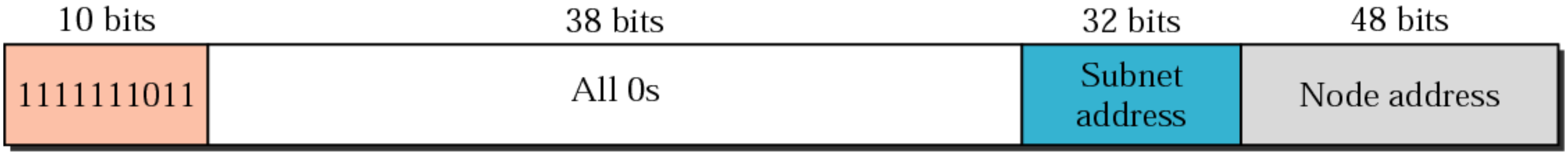
b. An example of address transformation

Link local address



10 bits	70 bits	48 bits
1111111010	All 0s	Node address

Site local address – Deprecated in RFC 3879



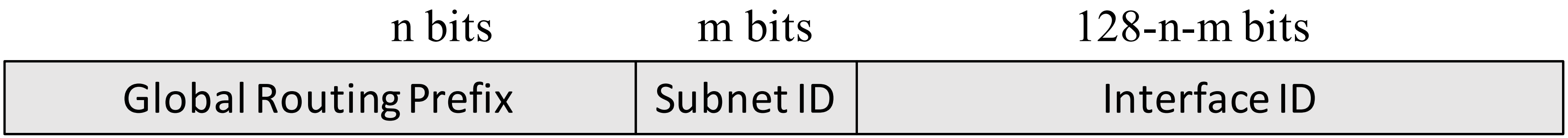
10 bits	38 bits	32 bits	48 bits
1111111011	All 0s	Subnet address	Node address

Unique Local Unicast Address – RFC 4193
FC00::/7

7 bits	1 bit	40 bits	16 bits	64 bits
1111110	L	Global ID	Subnet ID	Interface ID

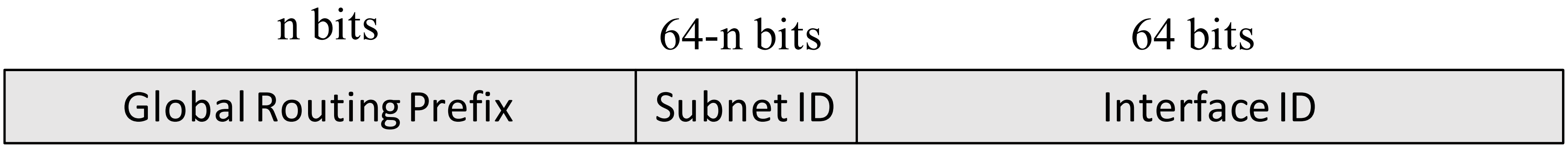
IPv6 Global Unicast Address – RFC 3587

General Format



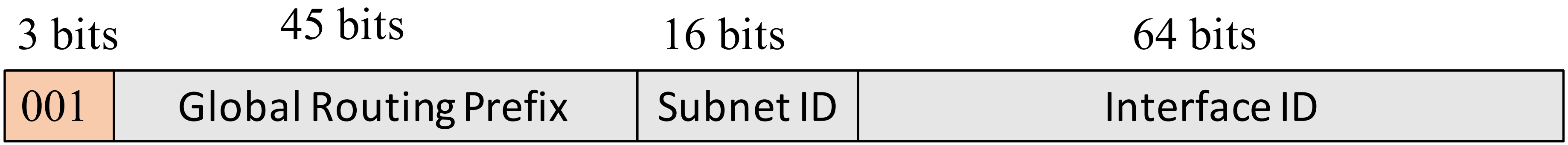
n bits	m bits	128-n-m bits
Global Routing Prefix	Subnet ID	Interface ID

For prefix not starting with binary value 000 (::/3)



n bits	64-n bits	64 bits
Global Routing Prefix	Subnet ID	Interface ID

Example:

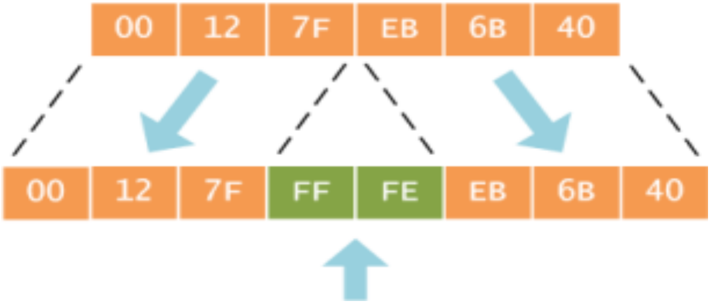


3 bits	45 bits	16 bits	64 bits
001	Global Routing Prefix	Subnet ID	Interface ID

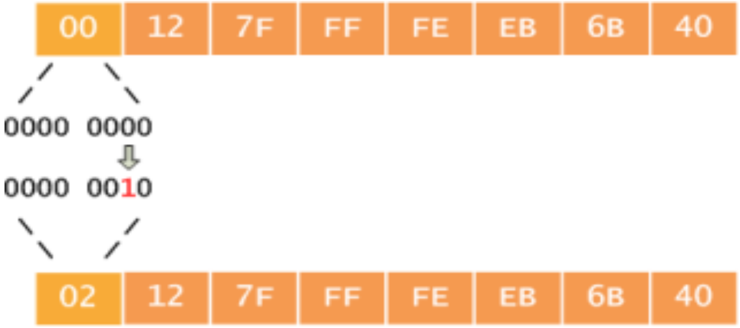
EUI-64 in IPv6

IEEE's 64-bit Extended Unique Identifier (EUI-64) format

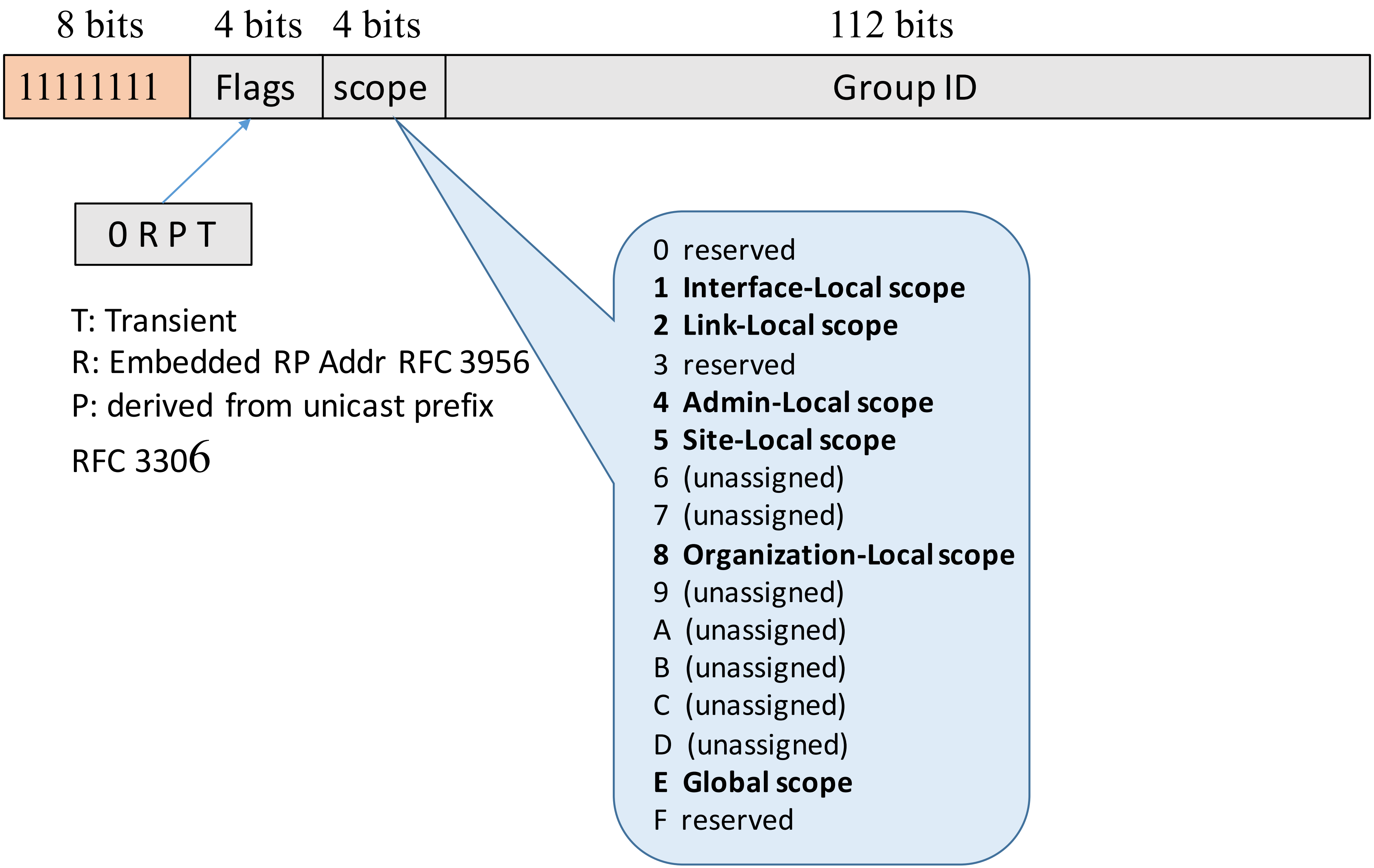
1. Break MAC Organizationally Unique Identifier (OUI) and the NIC specific part.



2. Invert the universal/local (U/L) flag (bit 7) in the OUI



Multicast address



8 bits | 4 bits | 4 bits | 112 bits

11111111 | Flags | scope | Group ID

0 R P T

T: Transient
R: Embedded RP Addr RFC 3956
P: derived from unicast prefix
RFC 3306

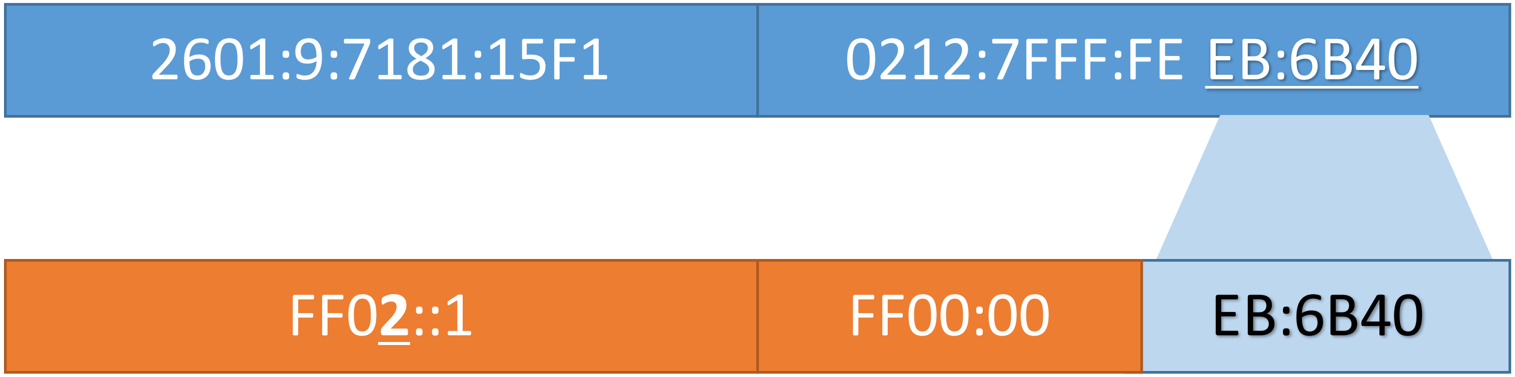
0 reserved
1 Interface-Local scope
2 Link-Local scope
3 reserved
4 Admin-Local scope
5 Site-Local scope
6 (unassigned)
7 (unassigned)
8 Organization-Local scope
9 (unassigned)
A (unassigned)
B (unassigned)
C (unassigned)
D (unassigned)
E Global scope
F reserved

Well-known IPv6 multicast addresses

Address	Description
FF02::1	All nodes on the local network segment
FF02::2	All routers on the local network segment
FF02::5	OSPFv3 All SPF routers
FF02::6	OSPFv3 All DR routers
FF02::8	IS-IS for IPv6 routers
FF02::9	RIP routers
FF02::A	EIGRP routers
FF02::D	PIM routers
FF02::16	MLDv2 reports (defined in RFC 3810)
FF02::1:2	All DHCP servers and relay agents on the local network segment (defined in RFC 3315)
FF05::1:3	All DHCP servers on the local network site (defined in RFC 3315)

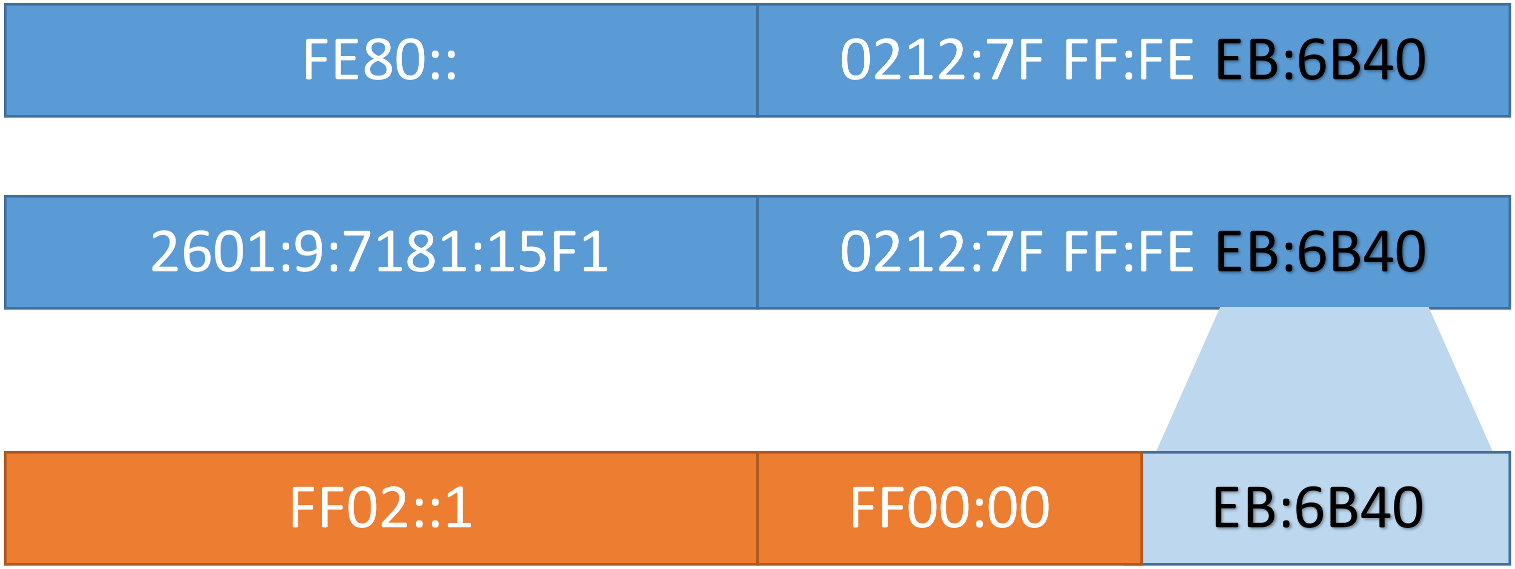
Solicited-Node Multicast Address

Taking the last 24 bits of IPv6 Unicast or Anycast address and append to FF02::1:FF00:0/104



A host is required to join a Solicited-Node multicast group for each of its configured unicast or anycast addresses

Solicited-Node Multicast Group



FE80:: 0212:7F FF:FE EB:6B40

2601:9:7181:15F1 0212:7F FF:FE EB:6B40

FF02::1 FF00:00 EB:6B40

Type of IPv6 Addresses

- ❑ **Unicast**
  - ➢An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.
- ❑ **Anycast**
  - ➢An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the "nearest" one, according to the routing protocols' measure of distance).
- ❑ **Multicast**
  - ➢An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.

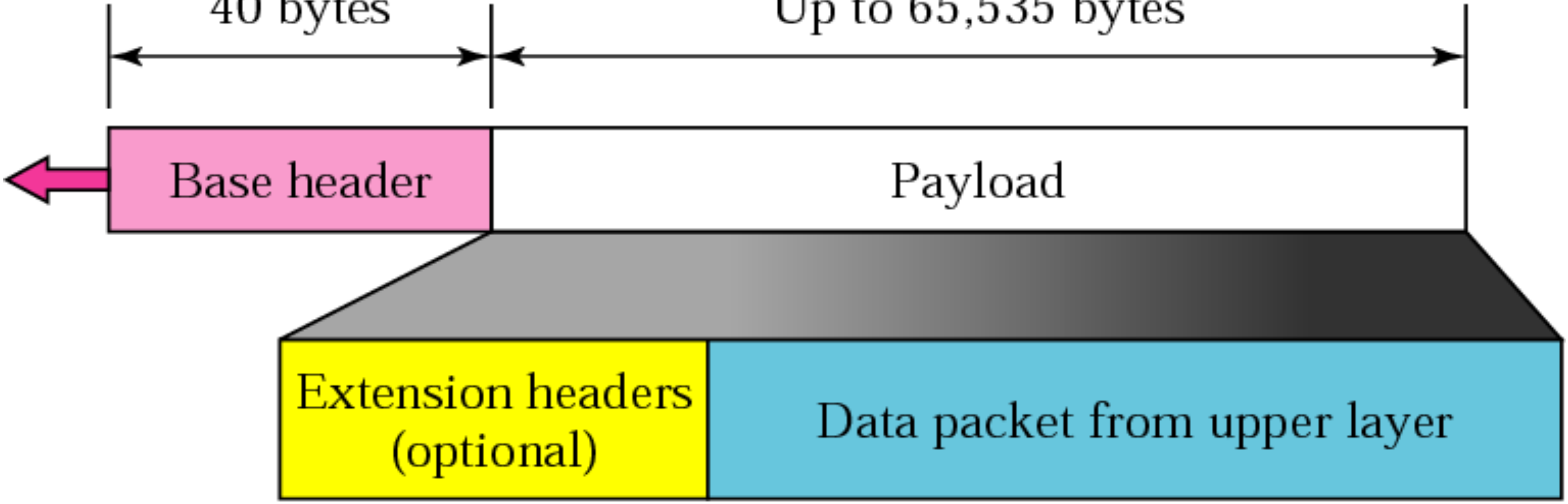
Required Anycast Address

Subnet-Router anycast address

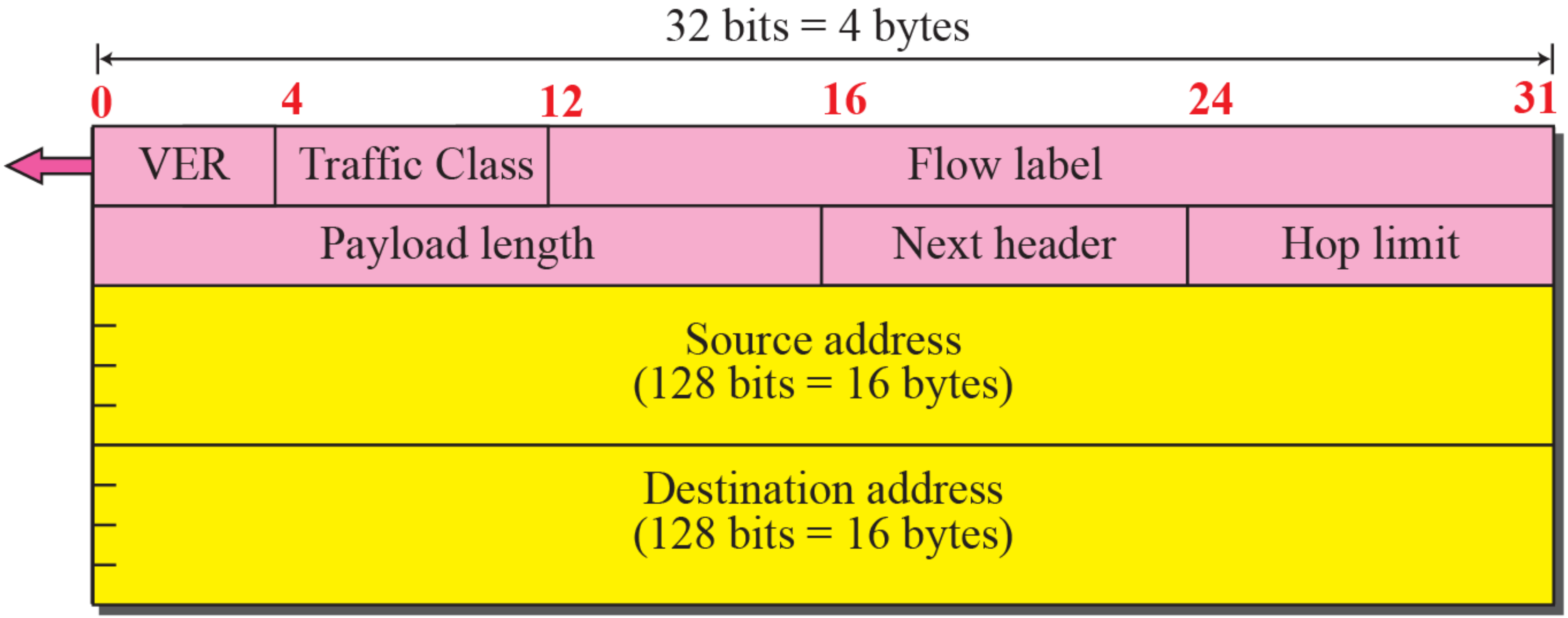
n bits	128-n bits
Subnet Prefix	000000000000

- Packets sent to the Subnet-Router anycast address will be delivered to one router on the subnet.

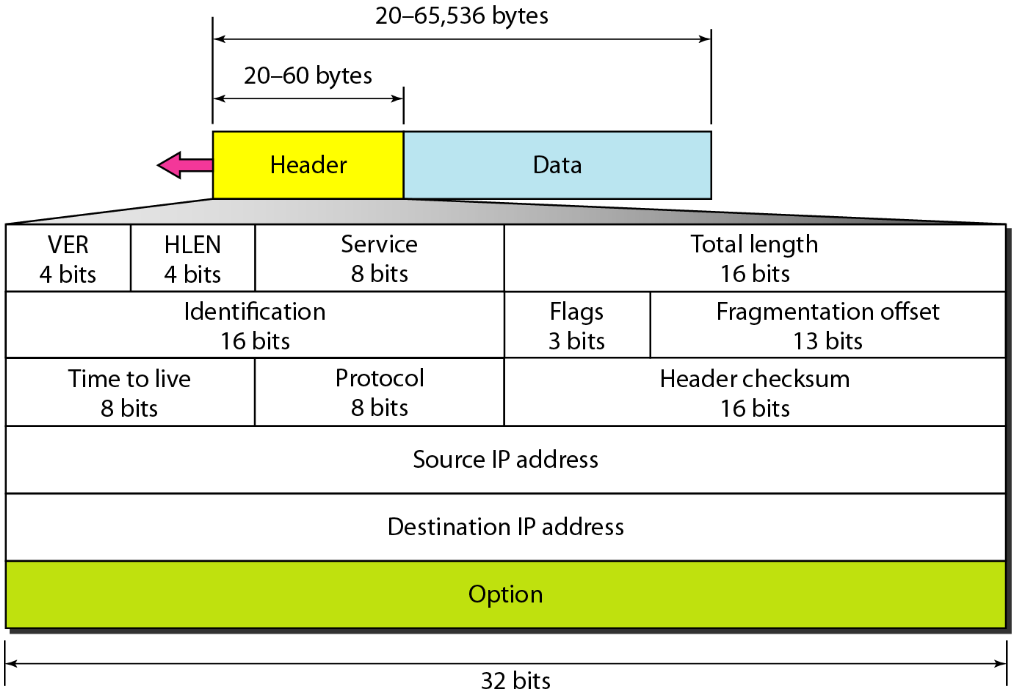
IPv6 datagram



IPv6 Header


32 bits = 4 bytes
0
4
12
16
24
31
VER
Traffic Class
Flow label
Payload length
Next header
Hop limit
Source address
(128 bits = 16 bytes)
Destination address
(128 bits = 16 bytes)

IPv4 and IPv6 Header



20–65,536 bytes

20–60 bytes

Header

Data

VER 4 bits

HLEN 4 bits

Service 8 bits

Total length 16 bits

Identification 16 bits

Flags 3 bits

Fragmentation offset 13 bits

Time to live 8 bits

Protocol 8 bits

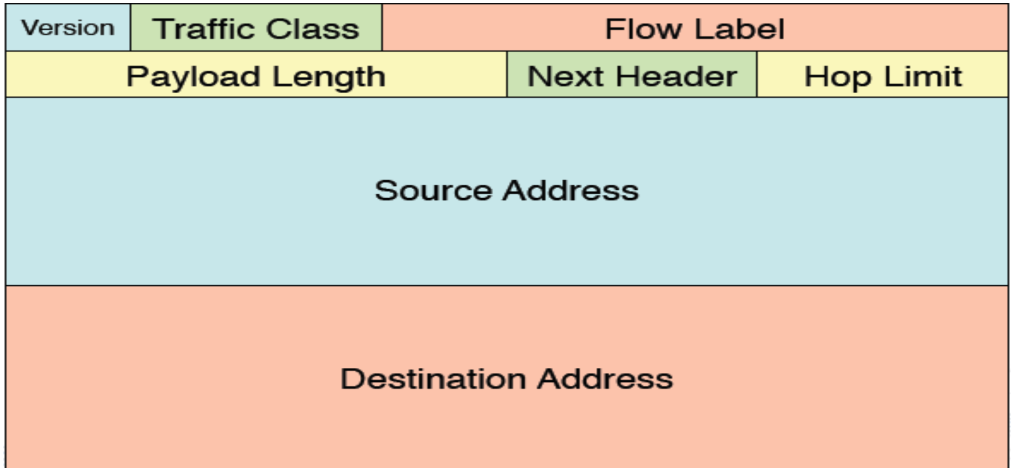
Header checksum 16 bits

Source IP address

Destination IP address

Option

32 bits



Version

Traffic Class

Flow Label

Payload Length

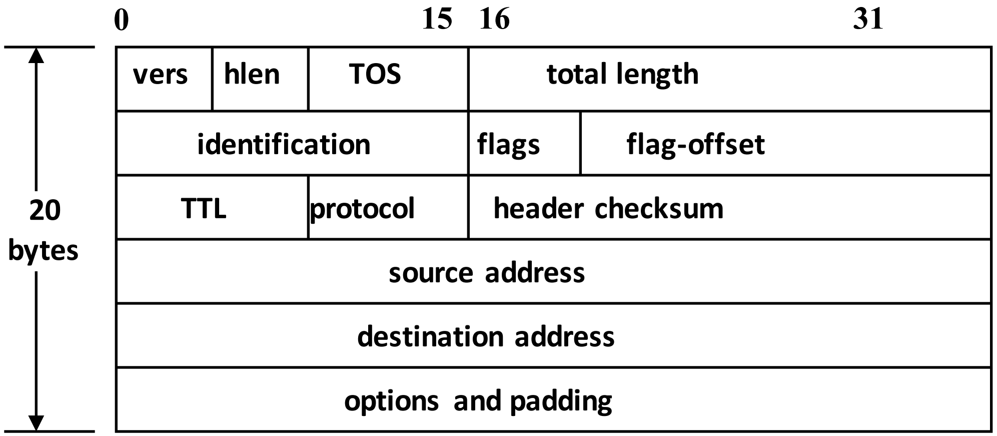
Next Header

Hop Limit

Source Address

Destination Address

Header comparison


0 15 16 31

vers | hlen | TOS | total length

identification | flags | flag-offset

TTL | protocol | header checksum

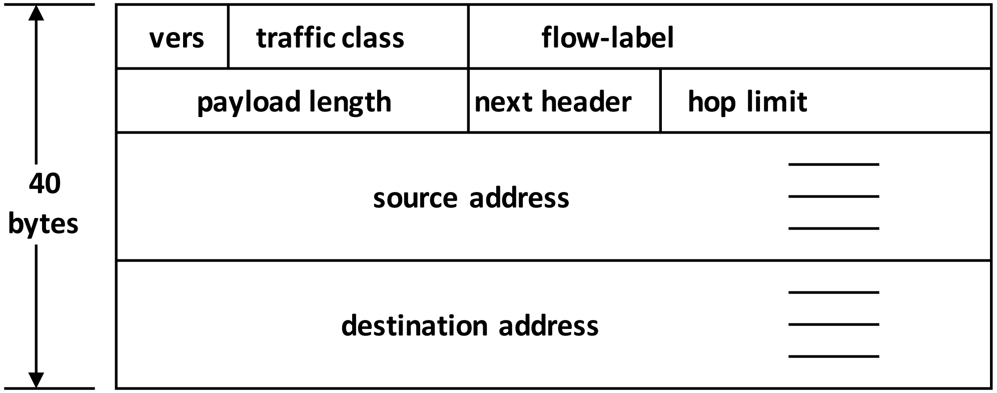
source address

destination address

options and padding

20 bytes

IPv4


vers | traffic class | flow-label

payload length | next header | hop limit

source address

destination address

40 bytes

IPv6

Removed (6)

- ID, flags, flag offset
- TOS, hlen
- header checksum

Changed (3)

- total length => payload
- protocol => next header
- TTL => hop limit

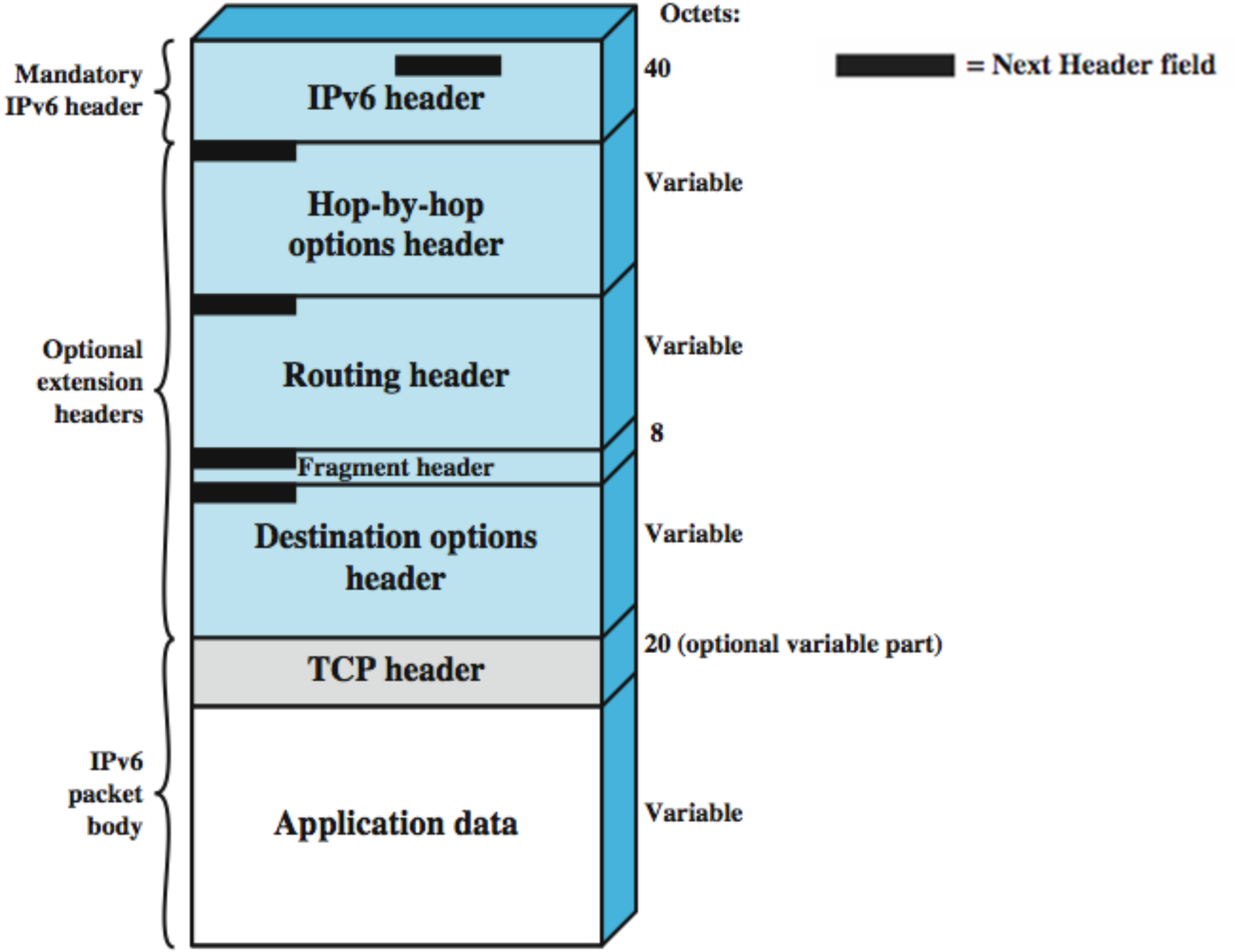
Added (2)

- traffic class
- flow label

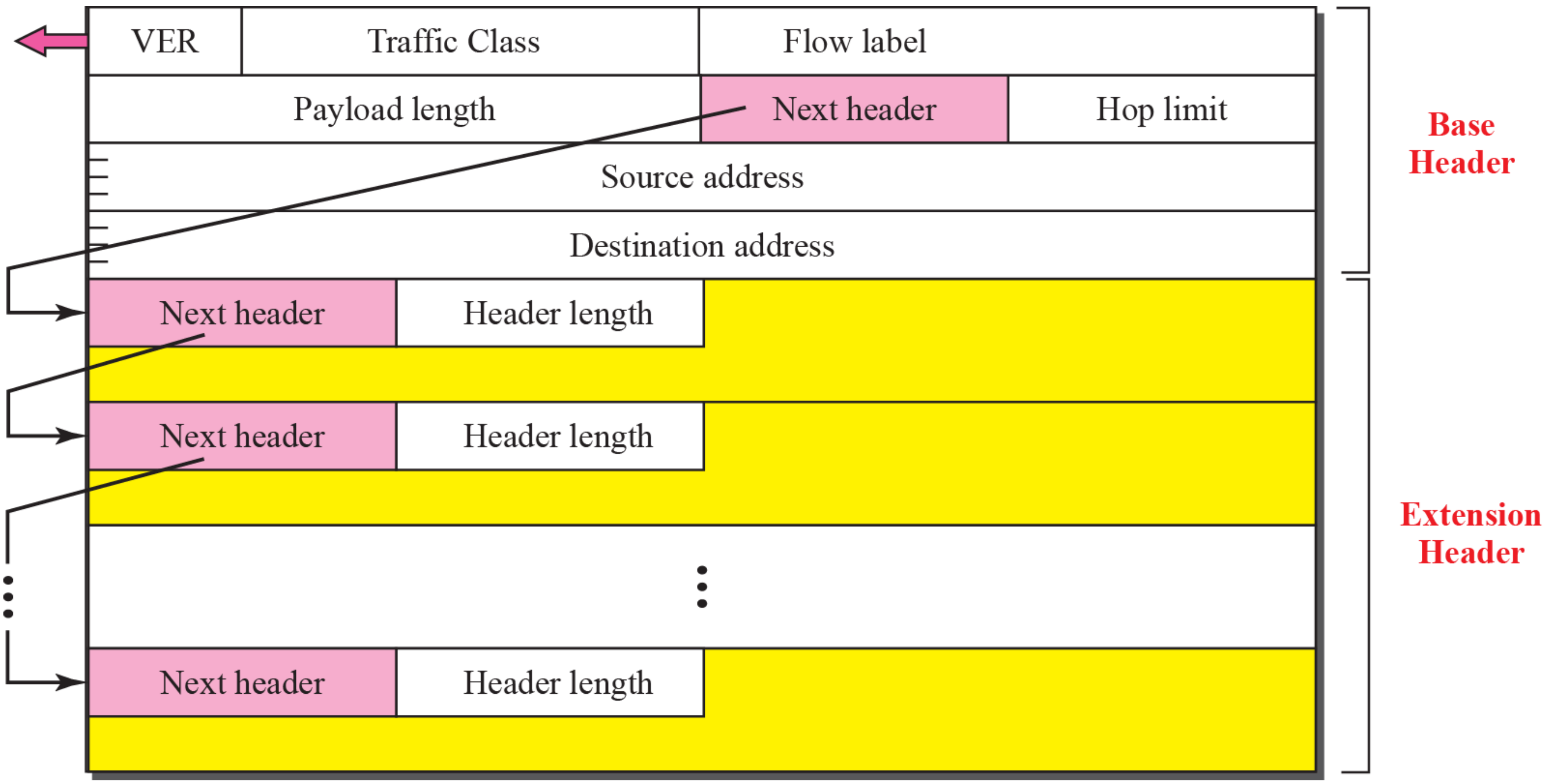
Expanded

- address 32 to 128 bits

IPv6 Packet (PDU) Structure



Extension header format



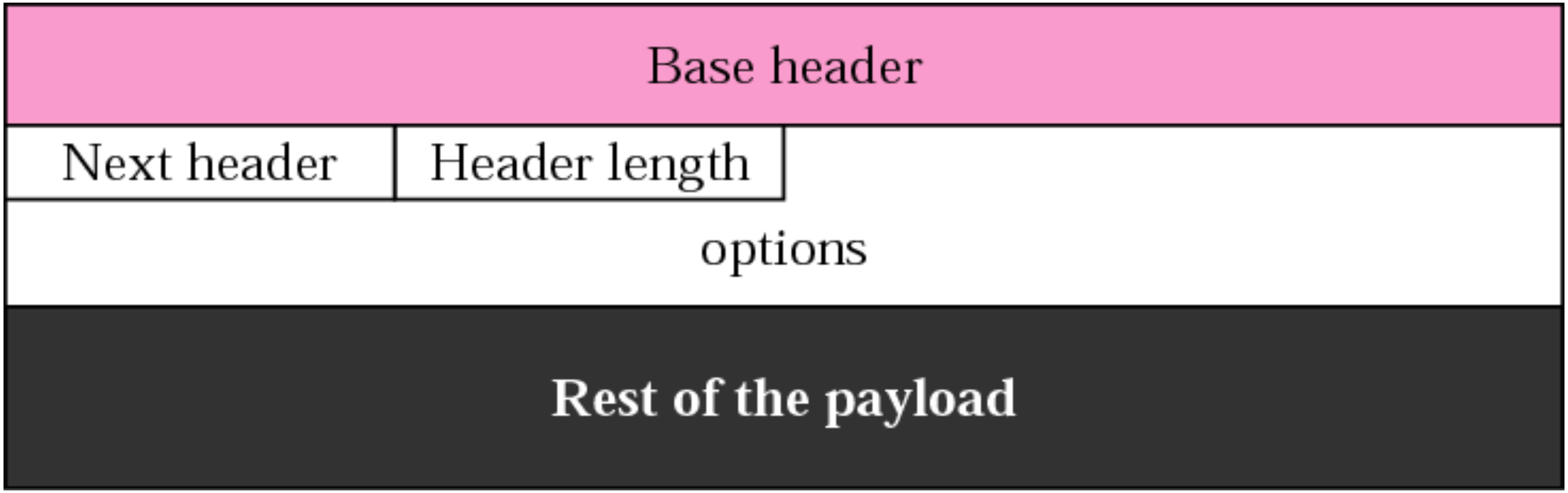
Extension Header

Extension Header	**Type**	**Description**
Hop-by-Hop Options	0	Options that need to be examined by all devices on the path.
Destination Options (before routing header)	60	Options that need to be examined only by the destination of the packet.
Routing	43	Methods to specify the route for a datagram (used with Mobile IPv6).
Fragment	44	Contains parameters for fragmentation of datagrams.
Authentication Header (AH)	51	Contains information used to verify the authenticity of most parts of the packet.
Encapsulating Security Payload (ESP)	50	Carries encrypted data for secure communication.
Destination Options (before upper-layer header)	60	Options that need to be examined only by the destination of the packet.
Mobility (currently without upper-layer header)	135	Parameters used with Mobile IPv6.

Extension Header

Extension Header	Type	Description
Experimental Use	139	Host Identity Protocol RFC5201
Shim6 Protocol	140	RFC5533
Use for experimentation and testing	253	Used for testing protocol extensions or new features
Use for experimentation and testing	254	Used for testing protocol extensions or new features

Hop-by-hop option header format



Base header

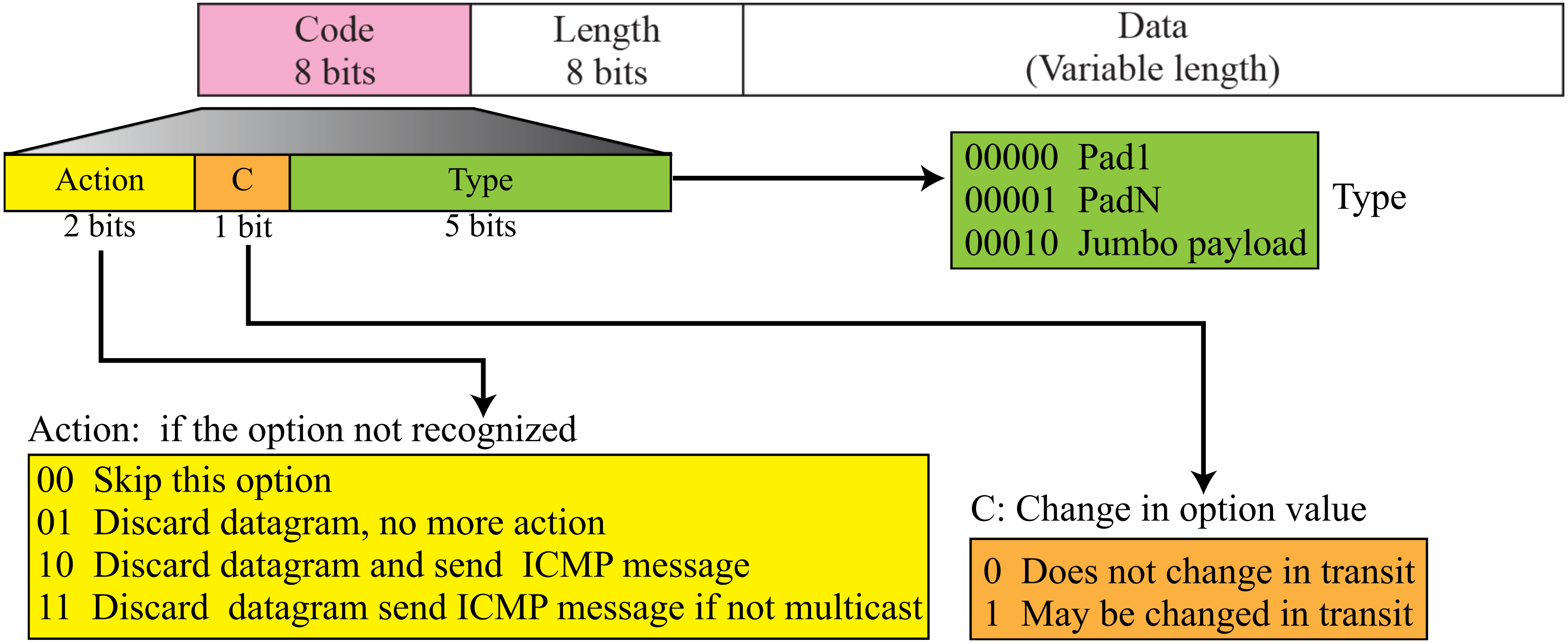
Next header

Header length

options

Rest of the payload

The format of options in a hop-by-hop option header



Code 8 bits

Length 8 bits

Data (Variable length)

Action 2 bits

C 1 bit

Type 5 bits

Type

00000 Pad1
00001 PadN
00010 Jumbo payload

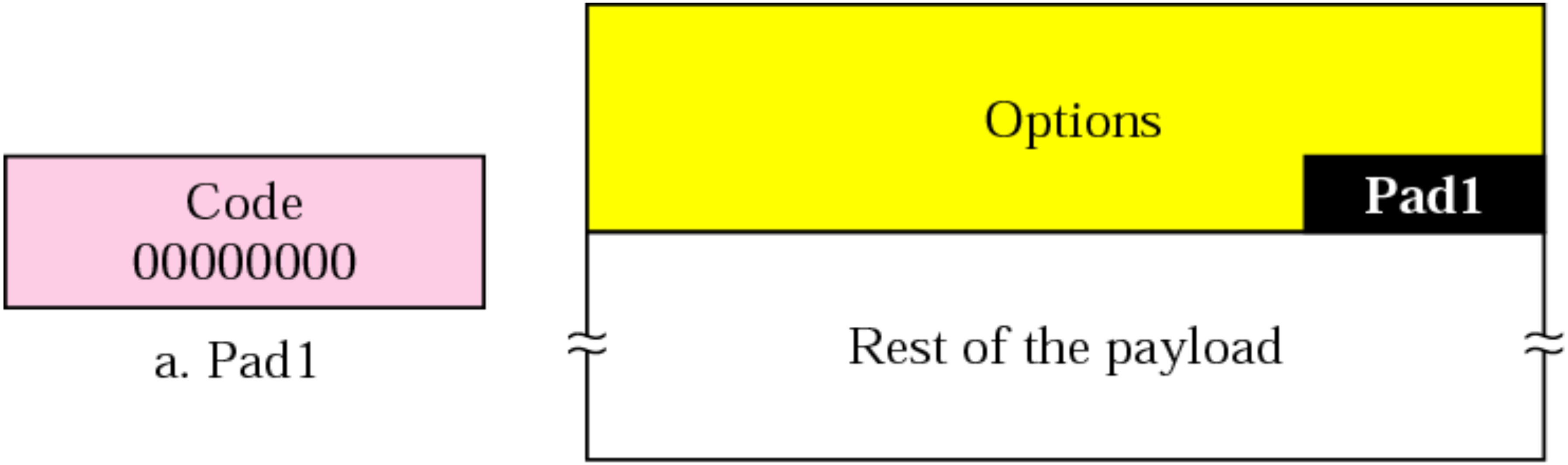
Action: if the option not recognized

00 Skip this option
01 Discard datagram, no more action
10 Discard datagram and send ICMP message
11 Discard datagram send ICMP message if not multicast

C: Change in option value

0 Does not change in transit
1 May be changed in transit

Pad1



Code
00000000

a. Pad1

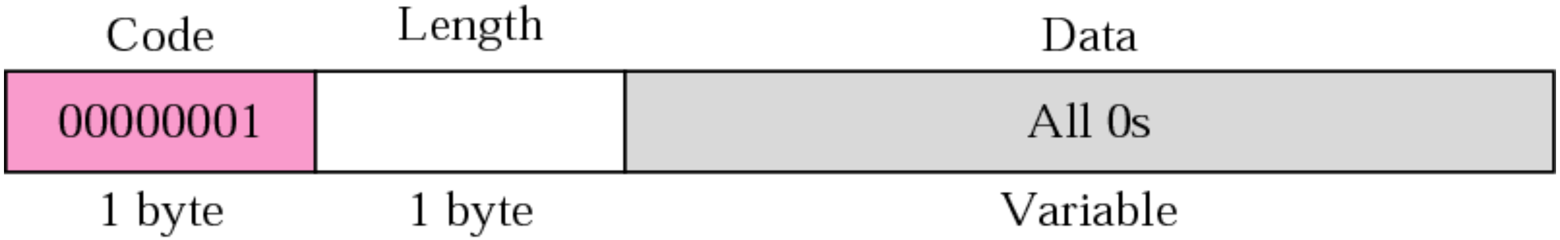
Options

Pad1

Rest of the payload

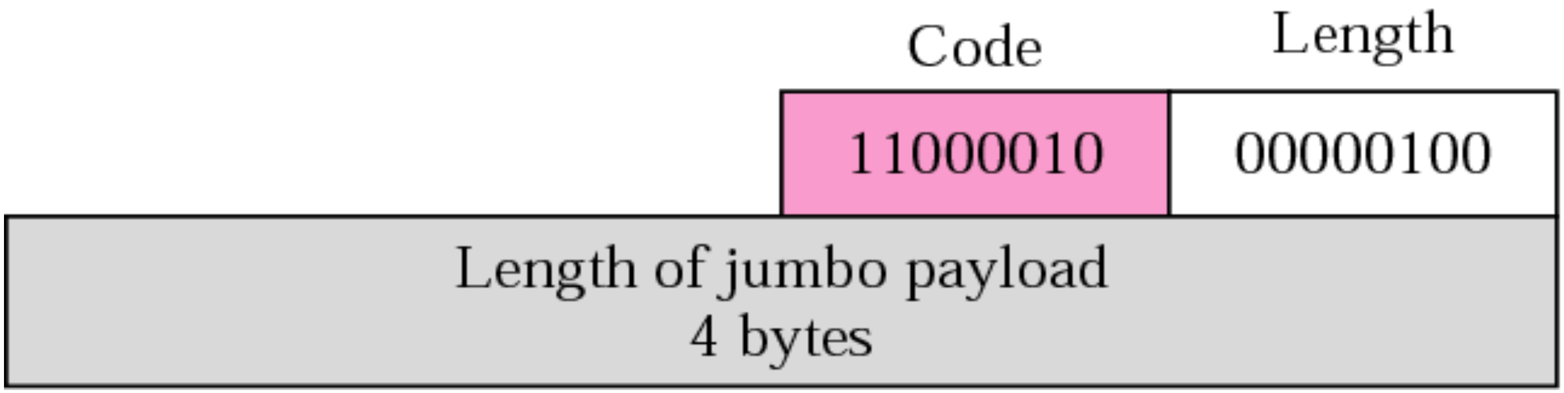
b. Used for padding

PadN



Code	Length	Data
00000001		All 0s
1 byte	1 byte	Variable

Jumbo payload



Code

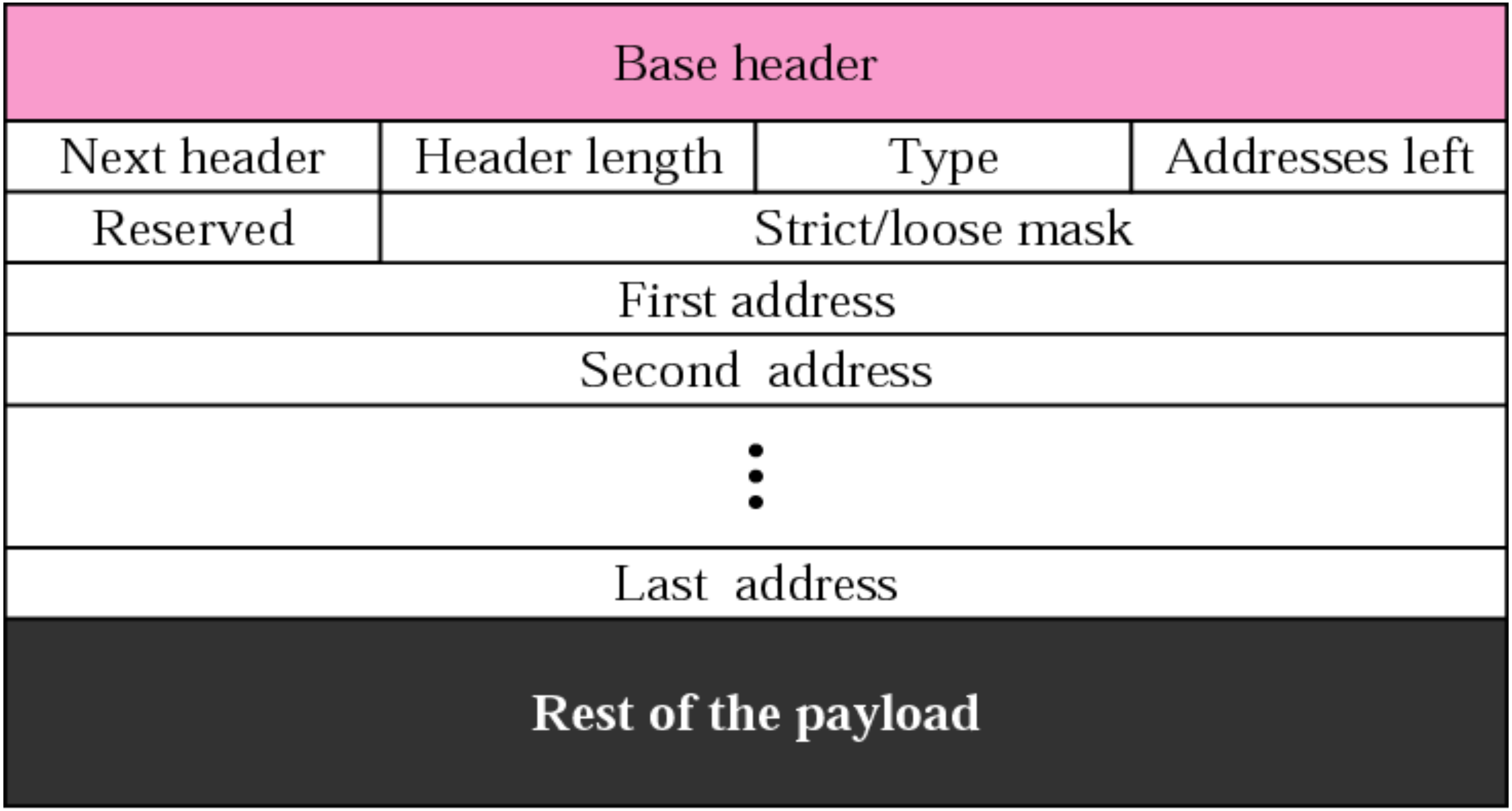
Length

11000010

00000100

Length of jumbo payload
4 bytes

Source routing



Base header

Next header | Header length | Type | Addresses left

Reserved | Strict/loose mask

First address

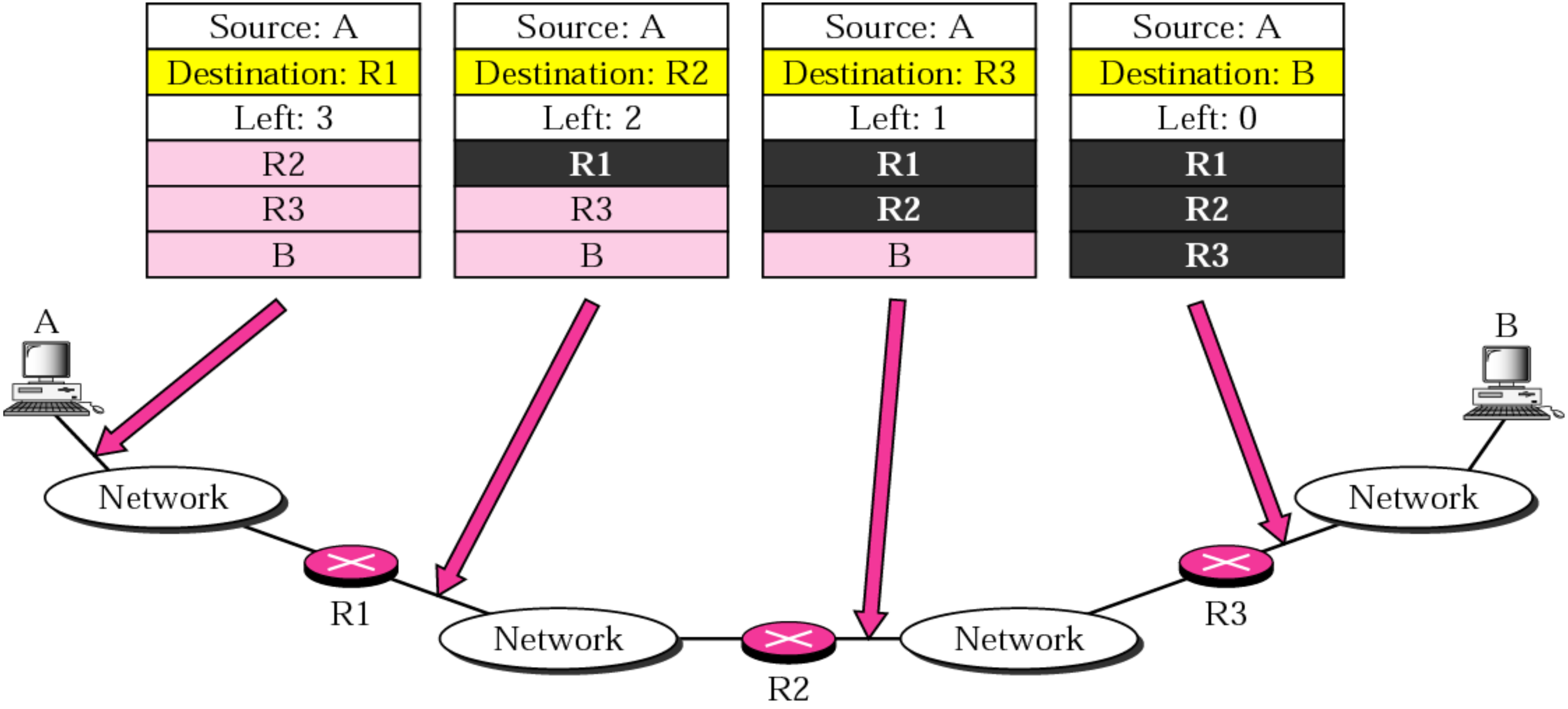
Second address

⋮

Last address

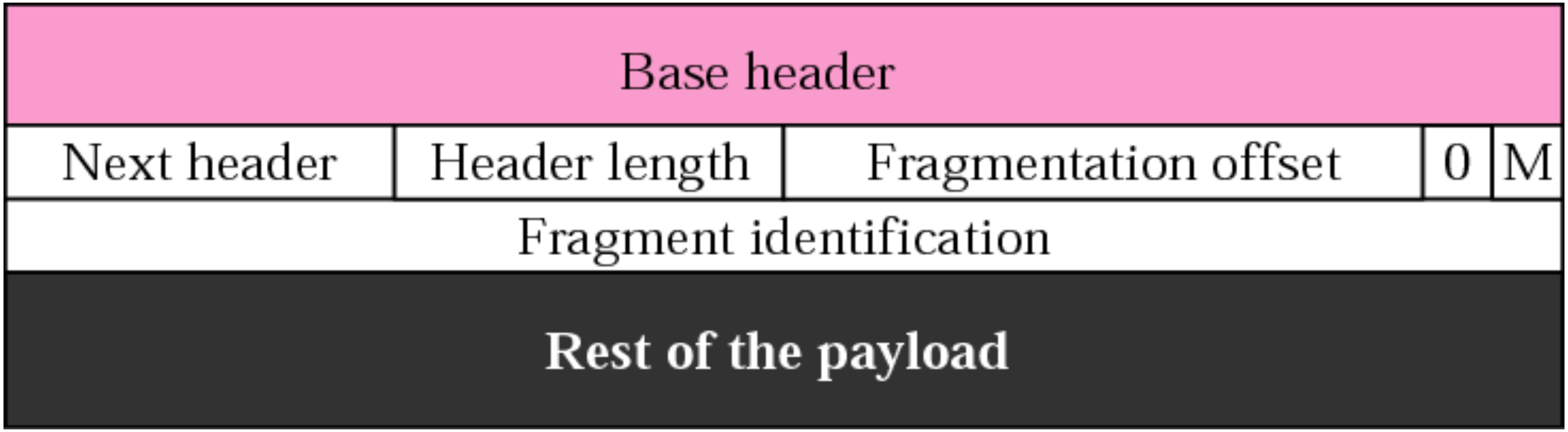
Rest of the payload

Source routing example



Source: A	Source: A	Source: A	Source: A
Destination: R1	Destination: R2	Destination: R3	Destination: B
Left: 3	Left: 2	Left: 1	Left: 0
R2	R1	R1	R1
R3	R3	R2	R2
B	B	B	R3

Fragmentation



Base header

Next header

Header length

Fragmentation offset

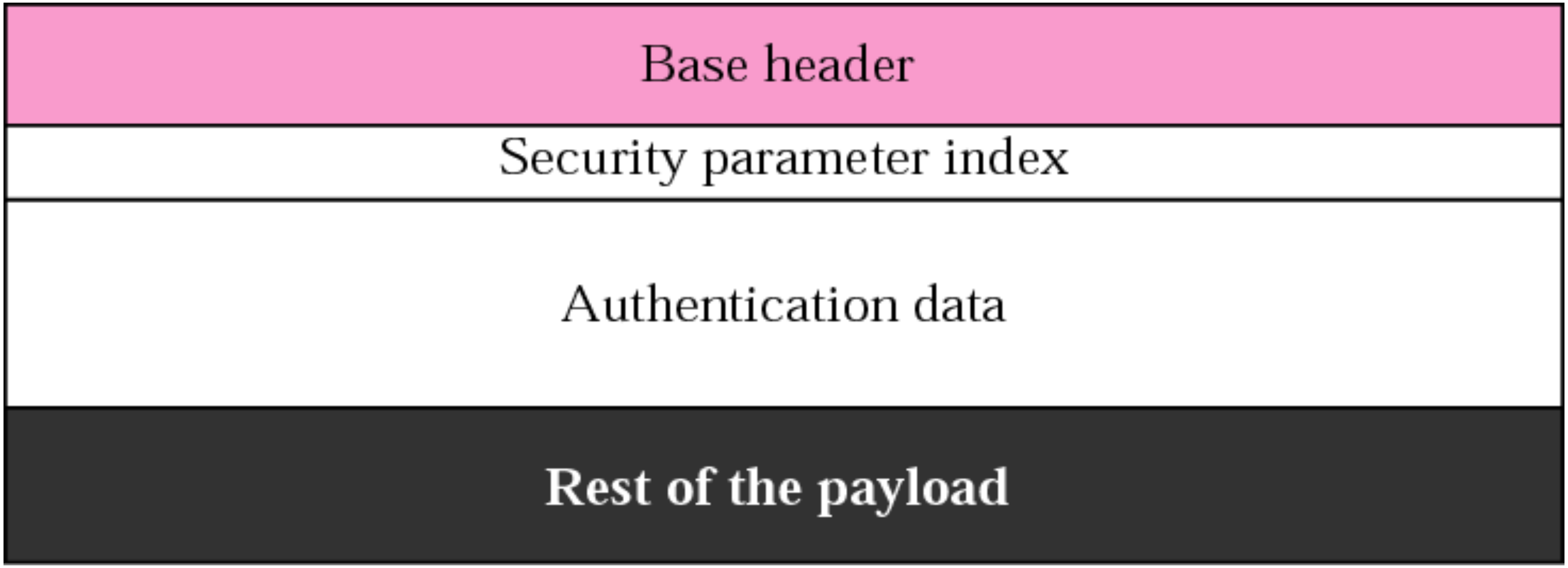
0

M

Fragment identification

Rest of the payload

Authentication



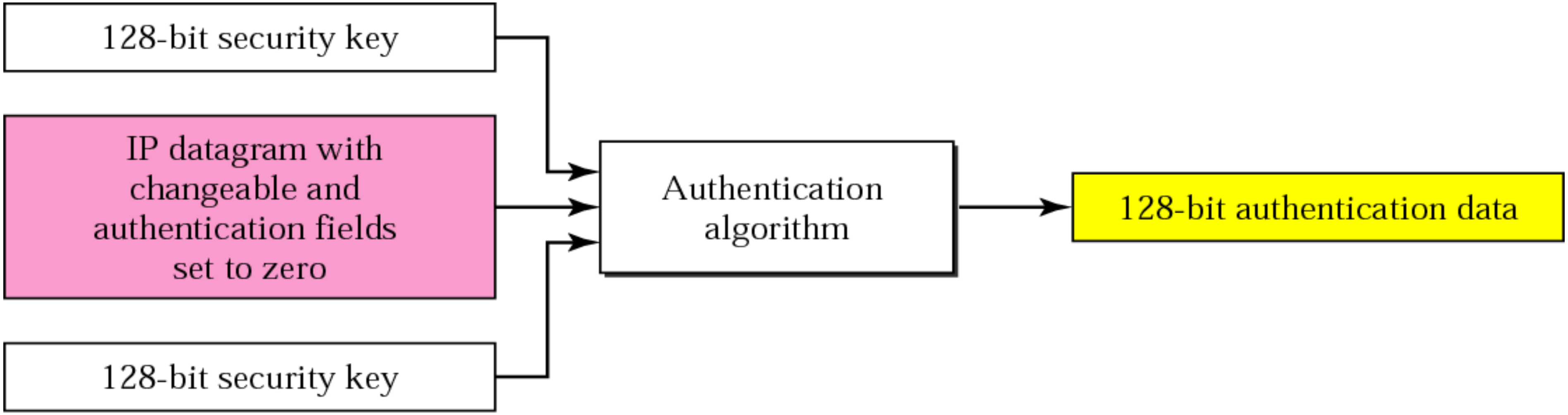
Base header

Security parameter index

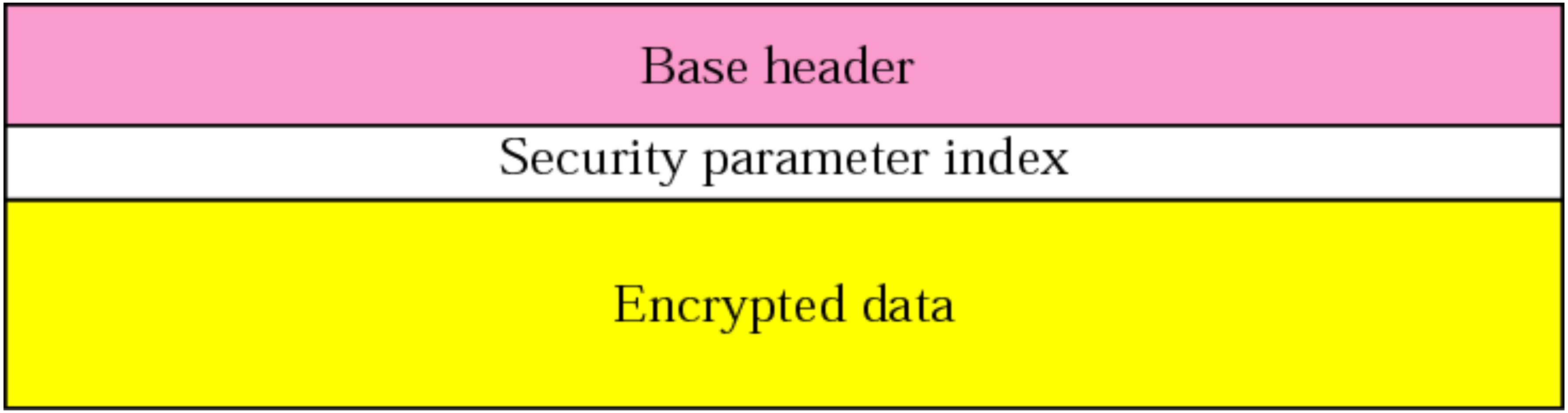
Authentication data

Rest of the payload

Calculation of authentication data


128-bit security key
IP datagram with changeable and authentication fields set to zero
128-bit security key
Authentication algorithm
128-bit authentication data

Encrypted security payload

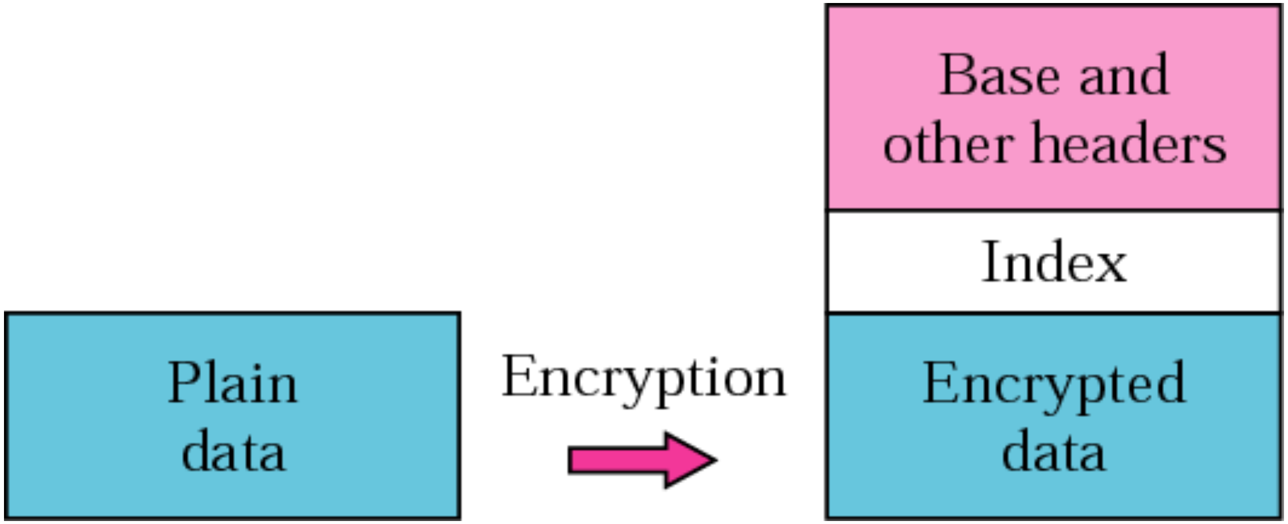


Base header

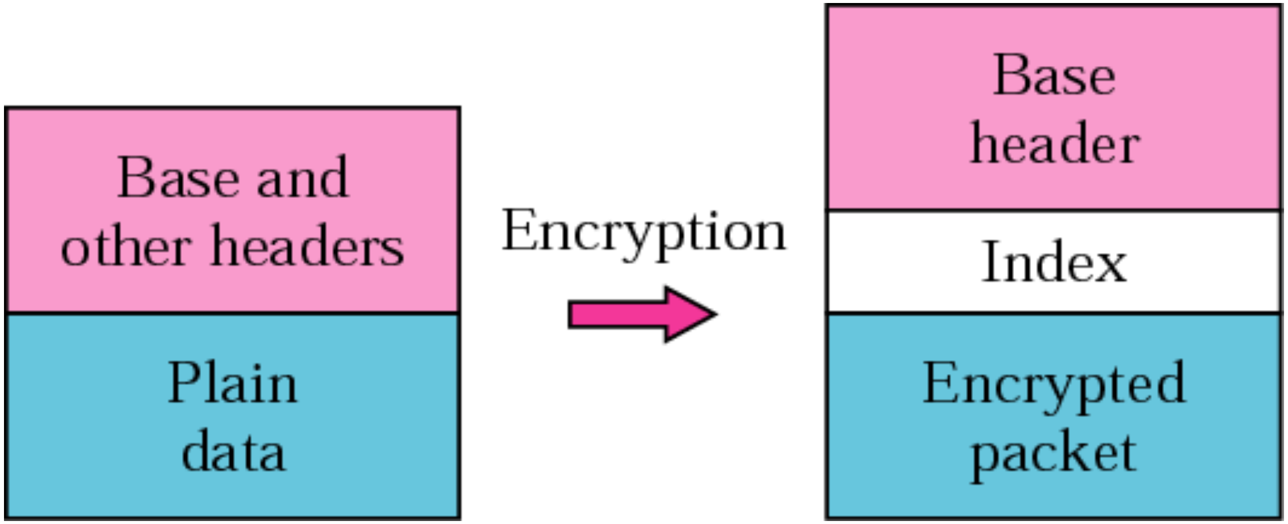
Security parameter index

Encrypted data

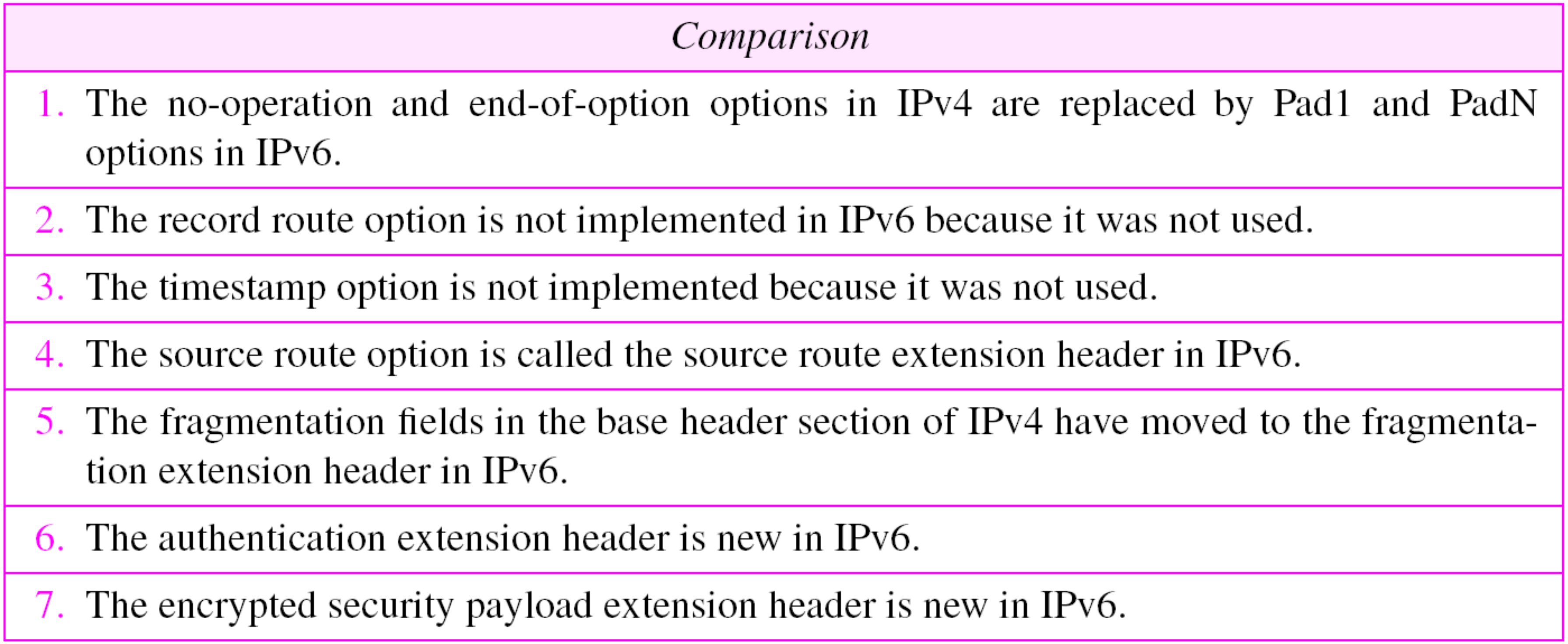
Transport mode encryption



Tunnel-mode encryption



Comparison between IPv4 options and IPv6 extension headers



Comparison
1. The no-operation and end-of-option options in IPv4 are replaced by Pad1 and PadN options in IPv6.
2. The record route option is not implemented in IPv6 because it was not used.
3. The timestamp option is not implemented because it was not used.
4. The source route option is called the source route extension header in IPv6.
5. The fragmentation fields in the base header section of IPv4 have moved to the fragmentation extension header in IPv6.
6. The authentication extension header is new in IPv6.
7. The encrypted security payload extension header is new in IPv6.

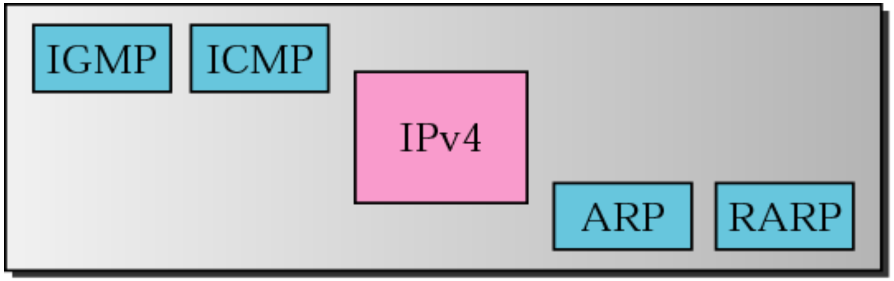
ICMPv6

ICMPv6, while similar in strategy to ICMPv4, has changes that makes it more suitable for IPv6. ICMPv6 has absorbed some protocols that were independent in version 4.

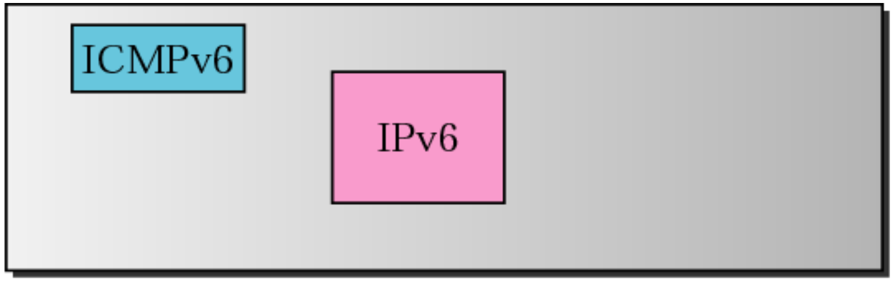
The topics discussed in this section include:

Error Reporting
Query

Comparison of network layers in version 4 and version 6

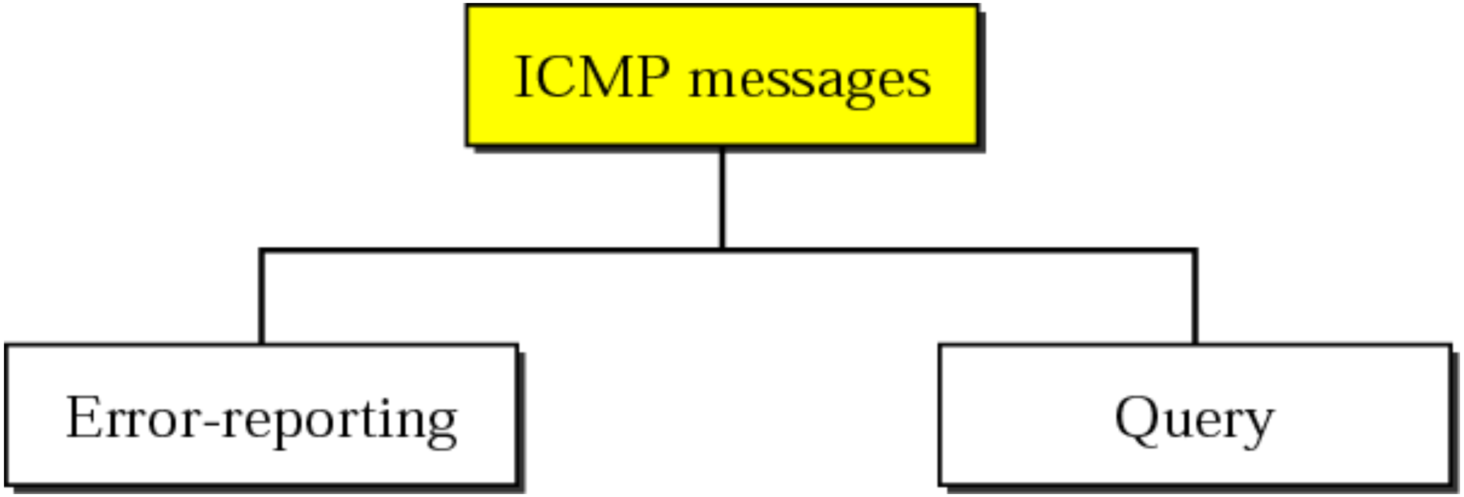


Network layer in version 4



Network layer in version 6

Categories of ICMPv6 messages

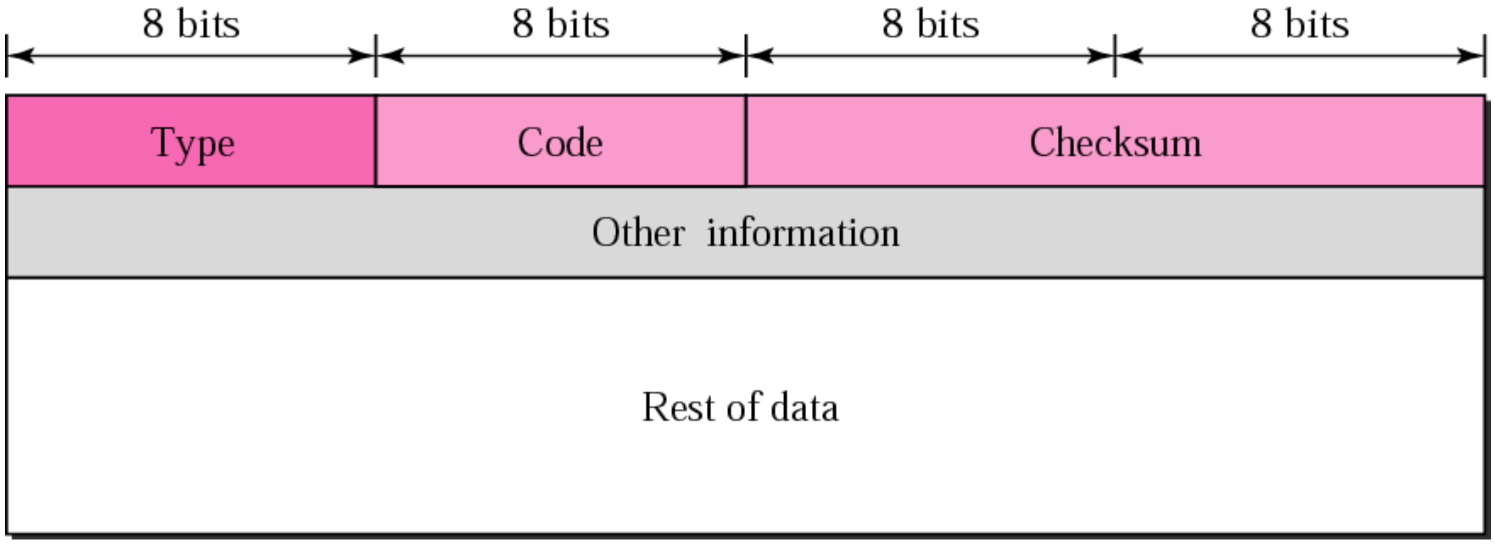


ICMP messages

Error-reporting

Query

General Format of ICMP messages



8 bits

8 bits

8 bits

8 bits

Type

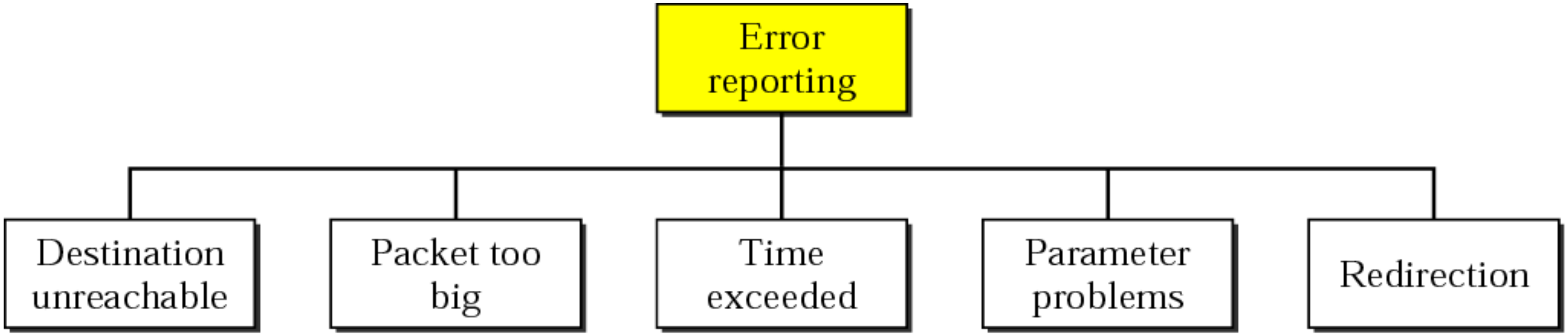
Code

Checksum

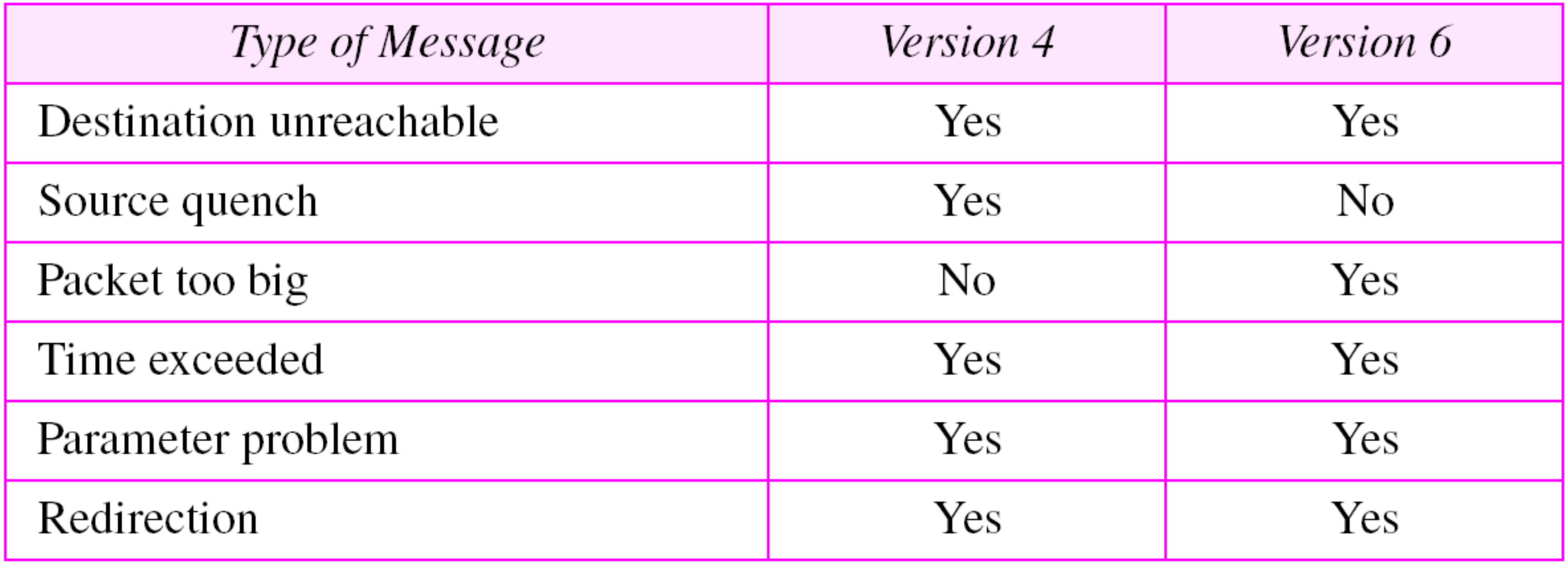
Other information

Rest of data

Error-reporting messages

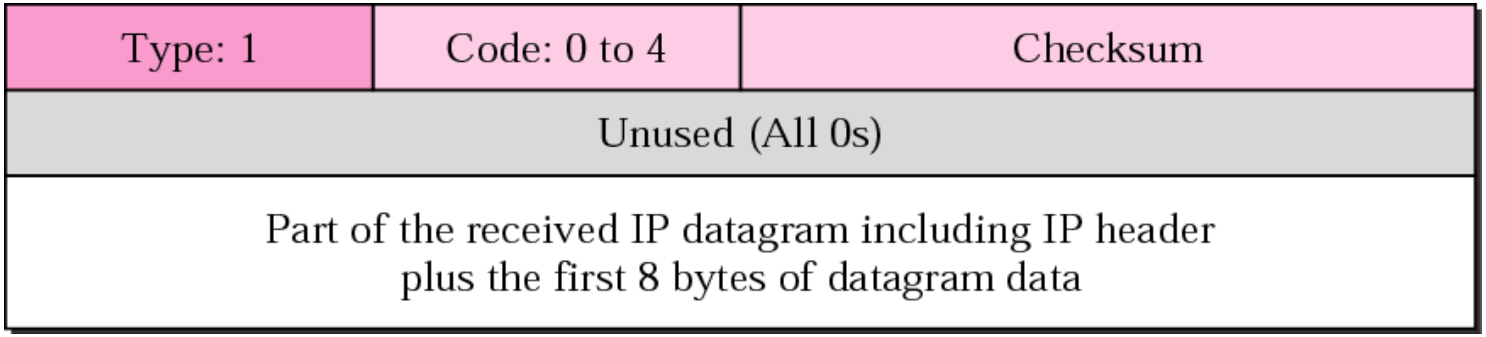


Comparison of error-reporting messages in ICMPv4 and ICMPv6



Type of Message	*Version 4*	*Version 6*
Destination unreachable	Yes	Yes
Source quench	Yes	No
Packet too big	No	Yes
Time exceeded	Yes	Yes
Parameter problem	Yes	Yes
Redirection	Yes	Yes

Destination-unreachable message format



Type: 1

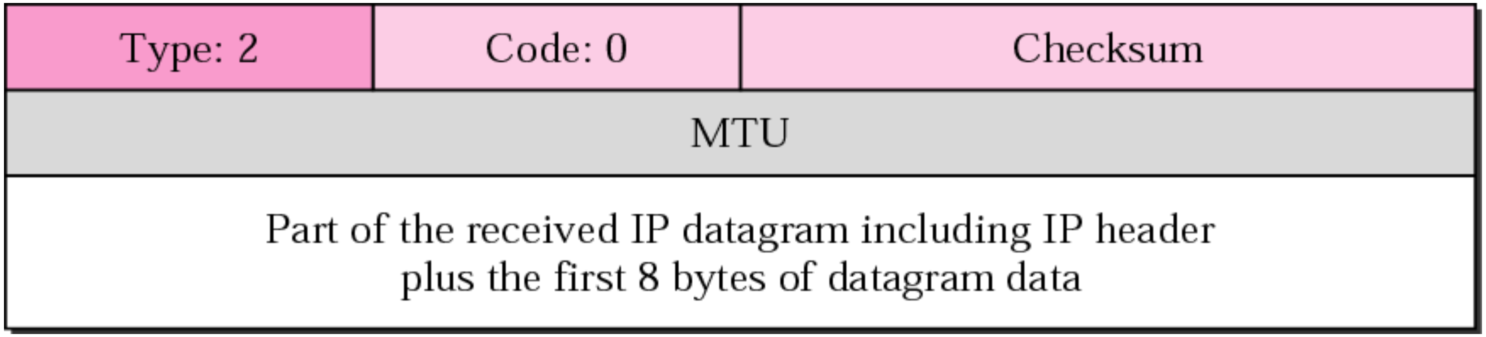
Code: 0 to 4

Checksum

Unused (All 0s)

Part of the received IP datagram including IP header
plus the first 8 bytes of datagram data

Packet-too-big message format



Type: 2

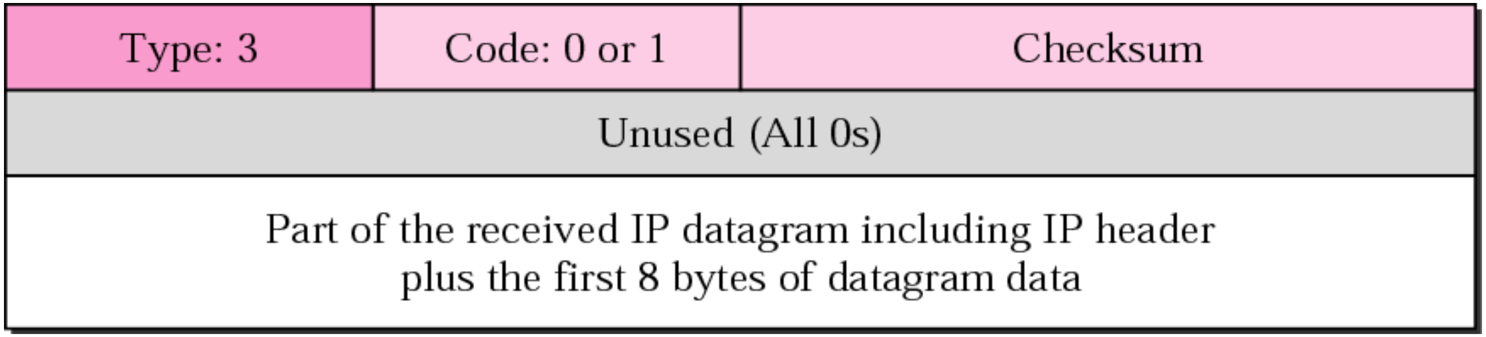
Code: 0

Checksum

MTU

Part of the received IP datagram including IP header
plus the first 8 bytes of datagram data

Time-exceeded message format



Type: 3

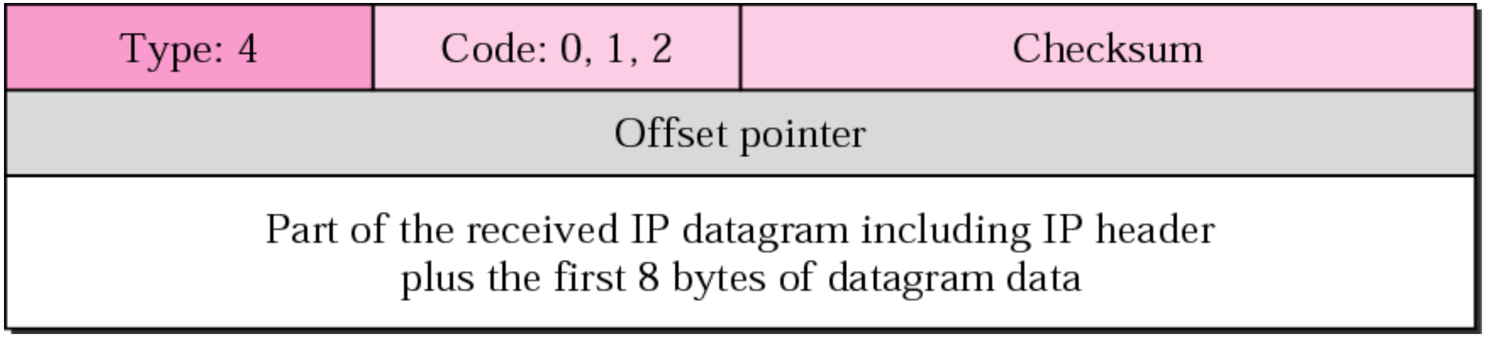
Code: 0 or 1

Checksum

Unused (All 0s)

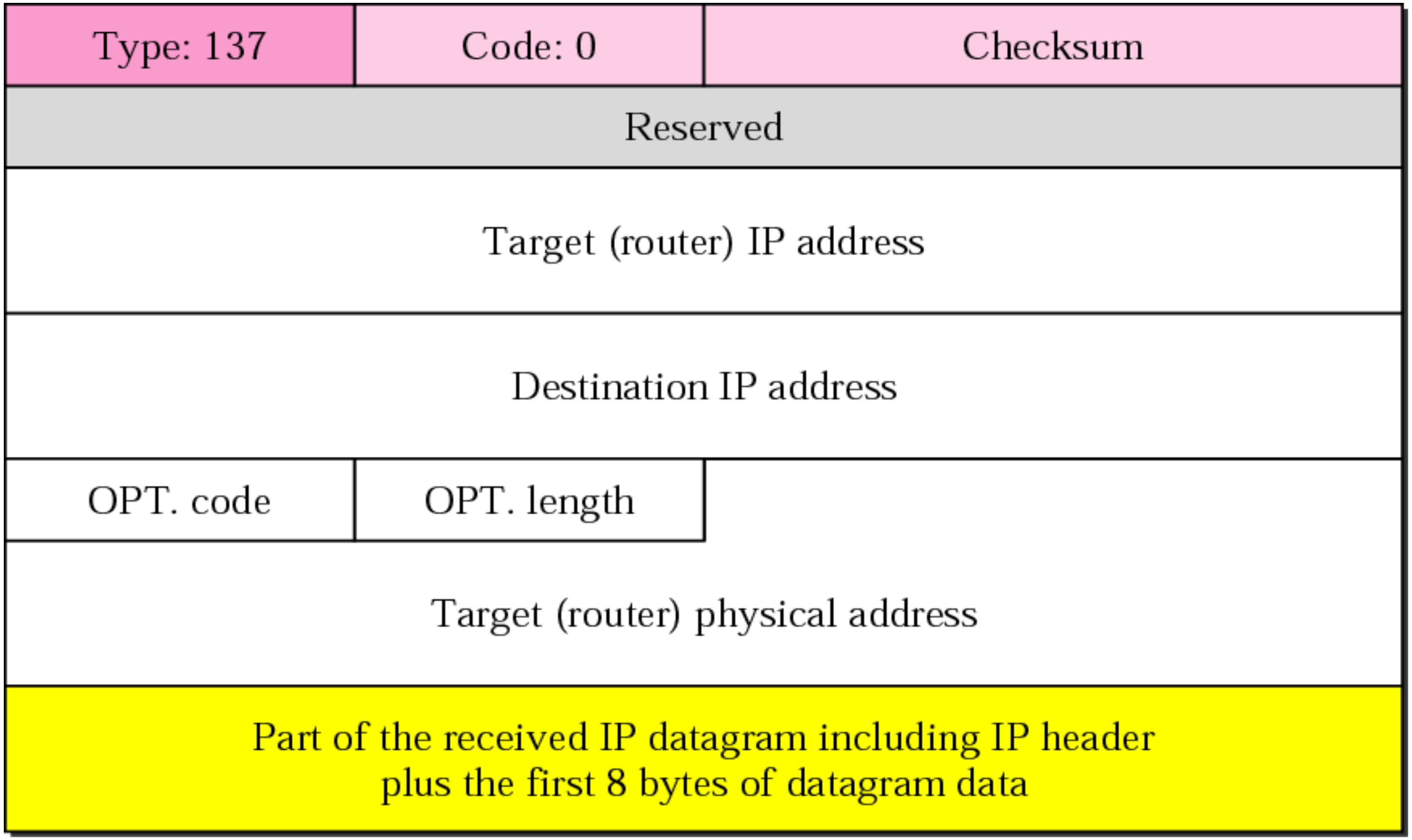
Part of the received IP datagram including IP header
plus the first 8 bytes of datagram data

Parameter-problem Message Format



Type: 4	Code: 0, 1, 2	Checksum
Offset pointer		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Redirection Message Format



Type: 137

Code: 0

Checksum

Reserved

Target (router) IP address

Destination IP address

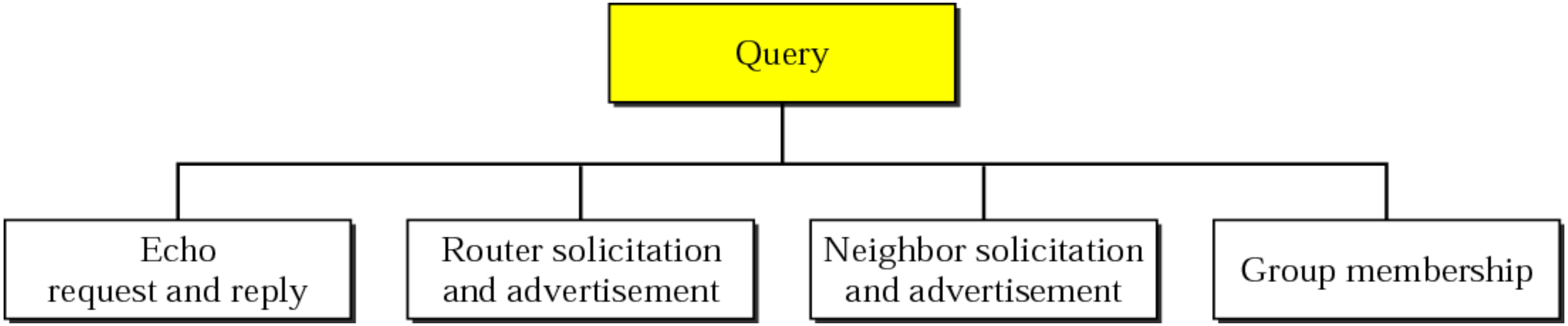
OPT. code

OPT. length

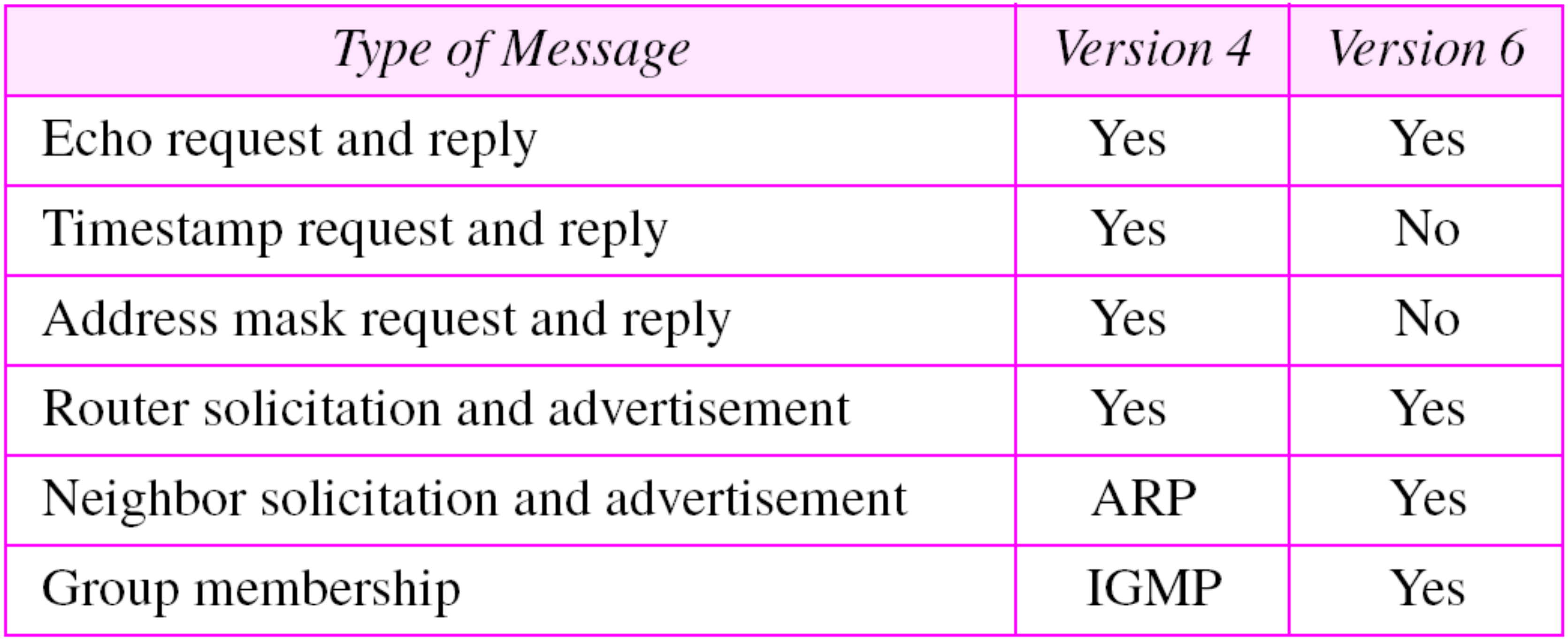
Target (router) physical address

Part of the received IP datagram including IP header
plus the first 8 bytes of datagram data

Query Messages

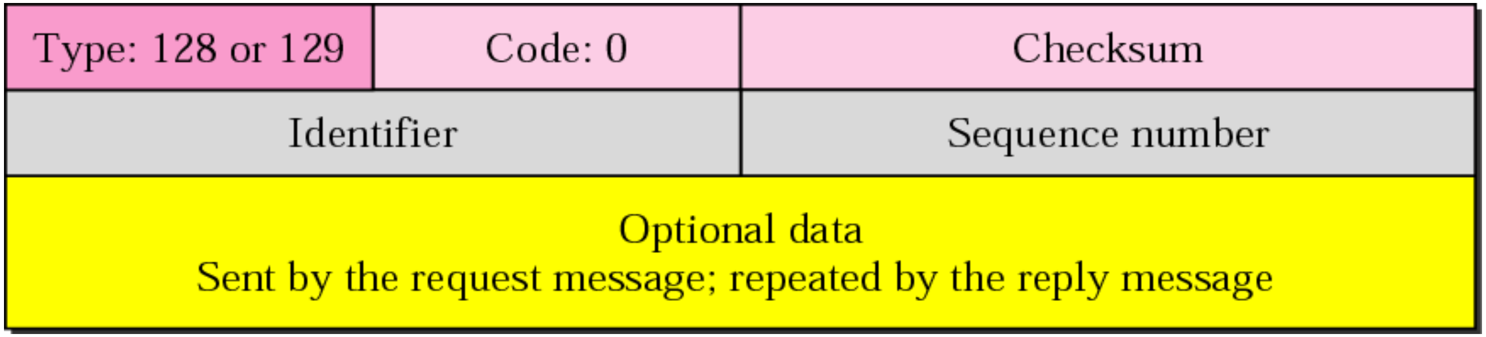

Query
Echo request and reply
Router solicitation and advertisement
Neighbor solicitation and advertisement
Group membership

Comparison Of Query Messages In ICMPv4 and ICMPv6



Type of Message	*Version 4*	*Version 6*
Echo request and reply	Yes	Yes
Timestamp request and reply	Yes	No
Address mask request and reply	Yes	No
Router solicitation and advertisement	Yes	Yes
Neighbor solicitation and advertisement	ARP	Yes
Group membership	IGMP	Yes

Echo Request And Reply Messages

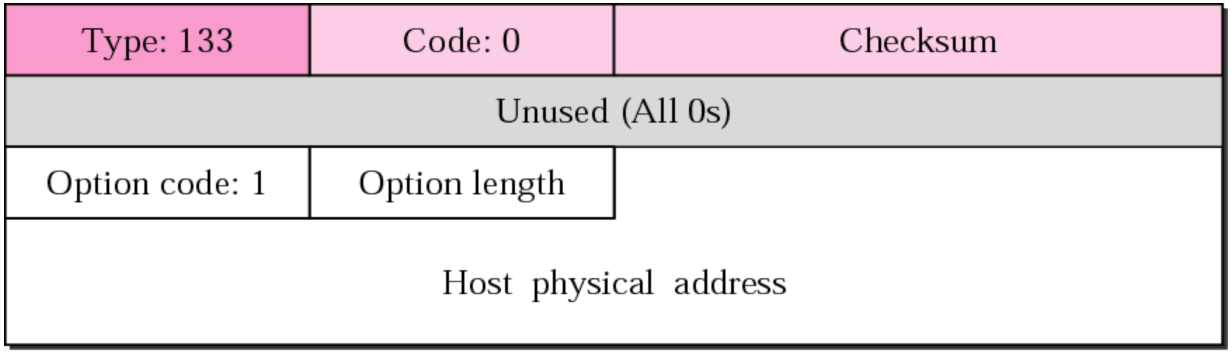


Type: 128 or 129 | Code: 0 | Checksum

Identifier | Sequence number

Optional data
Sent by the request message; repeated by the reply message

Router-solicitation And Advertisement Message Formats



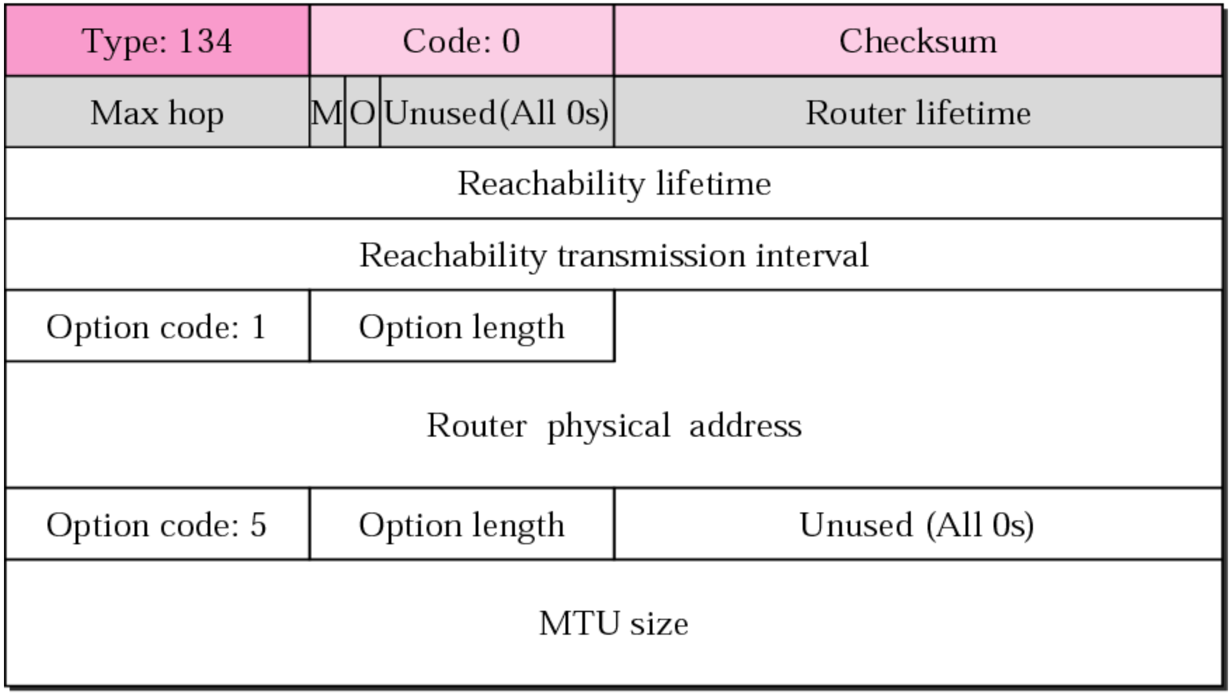
Type: 133 | Code: 0 | Checksum

Unused (All 0s)

Option code: 1 | Option length

Host physical address

a. Router solicitation format



Type: 134 | Code: 0 | Checksum

Max hop | M | O | Unused(All 0s) | Router lifetime

Reachability lifetime

Reachability transmission interval

Option code: 1 | Option length

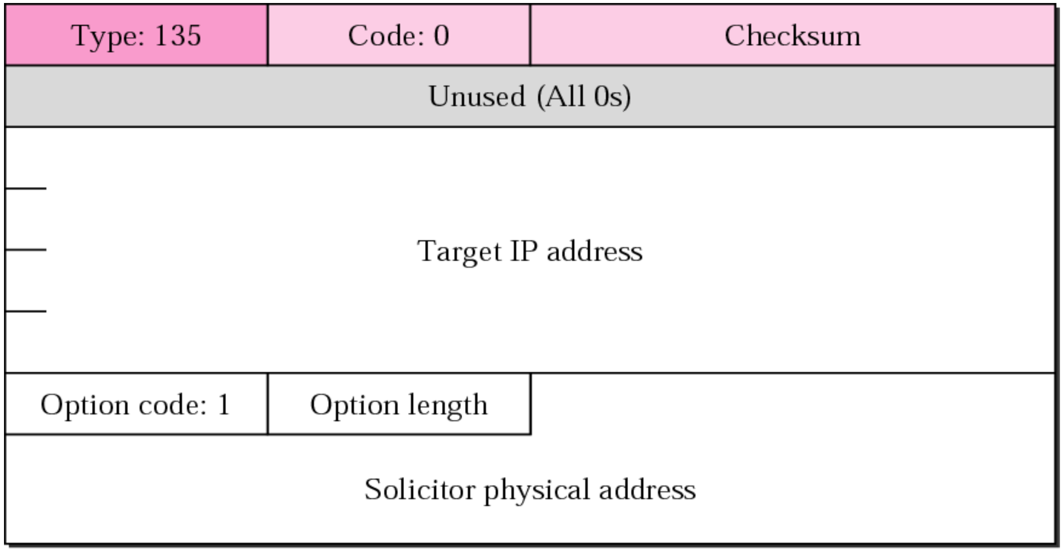
Router physical address

Option code: 5 | Option length | Unused (All 0s)

MTU size

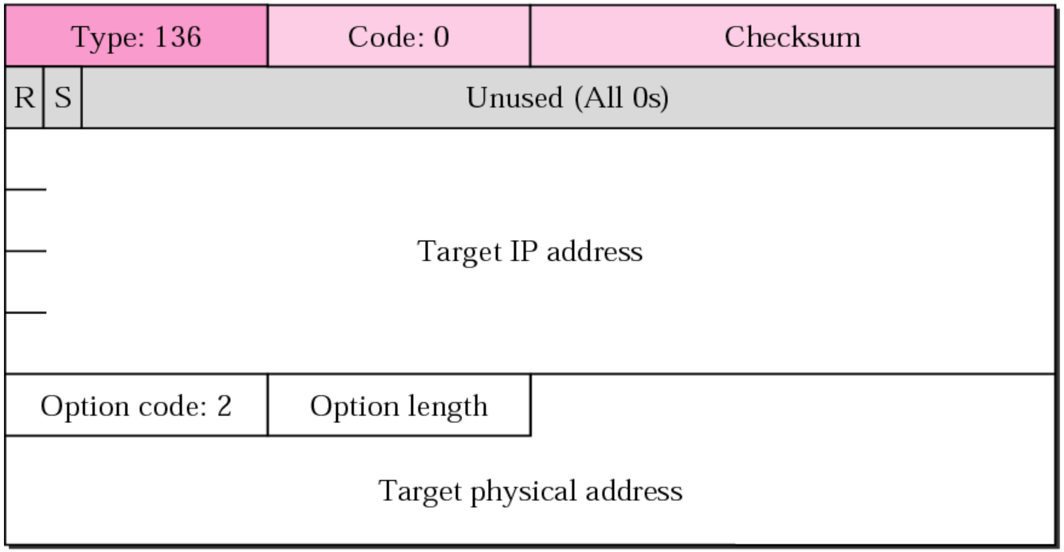
b. Router advertisement format

Neighbor-solicitation And Advertisement Message Formats



Type: 135
Code: 0
Checksum
Unused (All 0s)
Target IP address
Option code: 1
Option length
Solicitor physical address

a. Neighbor solicitation



Type: 136
Code: 0
Checksum
R
S
Unused (All 0s)
Target IP address
Option code: 2
Option length
Target physical address

b. Neighbor advertisement

Internet Control Message Protocol version 6 (ICMPv6)

Protocol Overview

- ICMPv6 is a multipurpose protocol used for
  - Reporting errors encountered in processing packets
  - Performing diagnostics
  - Performing Neighbor Discovery
  - Reporting multicast memberships.
- ICMP messages are transported within an IPv6 packet in which extension headers can also be present.
- An ICMP message is identified by a value of **58 in the Next Header** field of the IPv6 header or of the preceding Header.

ICMPv6: Introduction

- IPv6 uses the ICMP as defined for IPv4 with a number of changes.
- The resulting protocol is called ICMPv6.
- ICMP messages, delivered in IP packets, are used for out-of-band messages related to network operation or mis-operation.
- ICMP uses IP, ICMP packet delivery is unreliable, so hosts can't count on receiving ICMP packets for any network problem.

ICMPv6: Introduction

- **The ICMPv6** is an integral part of the IPv6 architecture and must be completely supported by all IPv6 implementations.
- ICMPv6 combines functions previously subdivided among different protocols, such as
  - ICMP (Internet Control Message Protocol version 4)
  - IGMP (Internet Group Membership Protocol)
  - ARP (Address Resolution Protocol)
- It introduces some simplifications by eliminating obsolete types of messages no longer in use.

ICMP: Functions

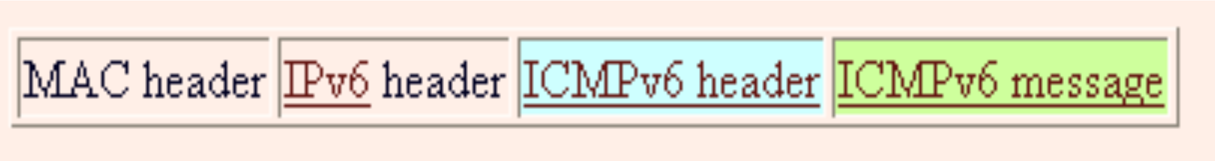
- **Announce network errors**
  - A host or entire portion of the network being unreachable, due to some type of failure.
  - A TCP or UDP packet directed at a port number with no receiver attached is also reported via ICMP.
- **Announce network congestion**
  - When a router begins buffering too many packets, due to an inability to transmit them as fast as they are being received, it will generate **ICMP *Source Quench* messages**.
  - Directed at the sender, these messages should cause the rate of packet transmission to be slowed.
  - Generating too many Source Quench messages would cause even more network congestion, so they are used sparingly.

ICMP: Functions

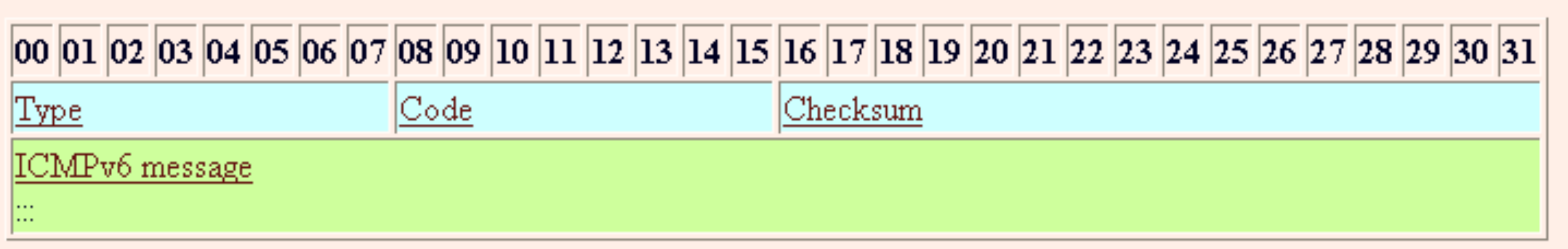
- **Assist Troubleshooting**
  - ICMP supports an ***Echo*** function, which just sends a packet on a round--trip between two hosts.
  - Ping will transmit a series of packets, measuring average round--trip times and computing loss percentages.
- **Announce Timeouts**
  - If an IP packet's TTL field drops to zero, the router discarding the packet will often generate an ICMP packet announcing this fact.
  - **TraceRoute** is a tool which maps network routes by sending packets with small TTL values and watching the ICMP timeout announcements.

ICMPv6 Header

- **Three Fields**
  1. **Type (8 bits)**
     - Indicates the type of the message.
     - If the high order bit = 0 (0- 127)→ error message
     - if the high-order bit = 1 (128 – 255) → information message.
  2. **Code (8 bits)**
     - content depends on the message type, and it is used to create an additional level of message granularity.
  3. **Checksum (16 bits)**
     - Used to detect errors in the ICMP message and in part of the IPv6 message.



MAC header	IPv6 header	ICMPv6 header	ICMPv6 message



00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type								Code								Checksum															
ICMPv6 message ...																															

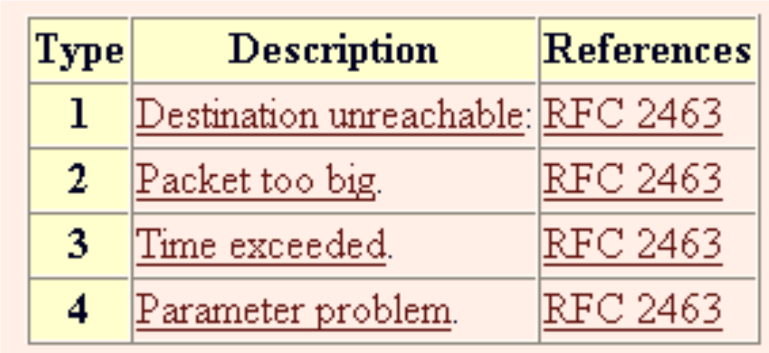
Types of ICMPv6 Messages

- ICMPv6 messages are grouped into two classes:
- **Error messages**
  - To provide feedback to a source device about an error that has occurred.
  - Generated specifically in response to some sort of action, usually the transmission of a datagram
  - Identified as such by having a zero in the high-order bit of their message
  - Type field values 0 to 127.

Informational messages

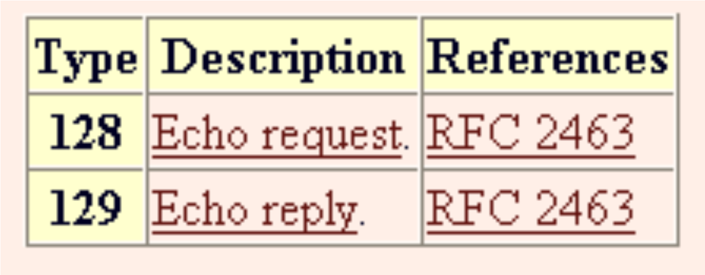
  - Used to let devices exchange information, implement certain IP-related features, and perform testing.
  - Message Types from 128 to 255.
- Many of these ICMP types have a "code" field.

Error messages



Type	Description	References
1	Destination unreachable:	RFC 2463
2	Packet too big.	RFC 2463
3	Time exceeded.	RFC 2463
4	Parameter problem.	RFC 2463

Informational messages

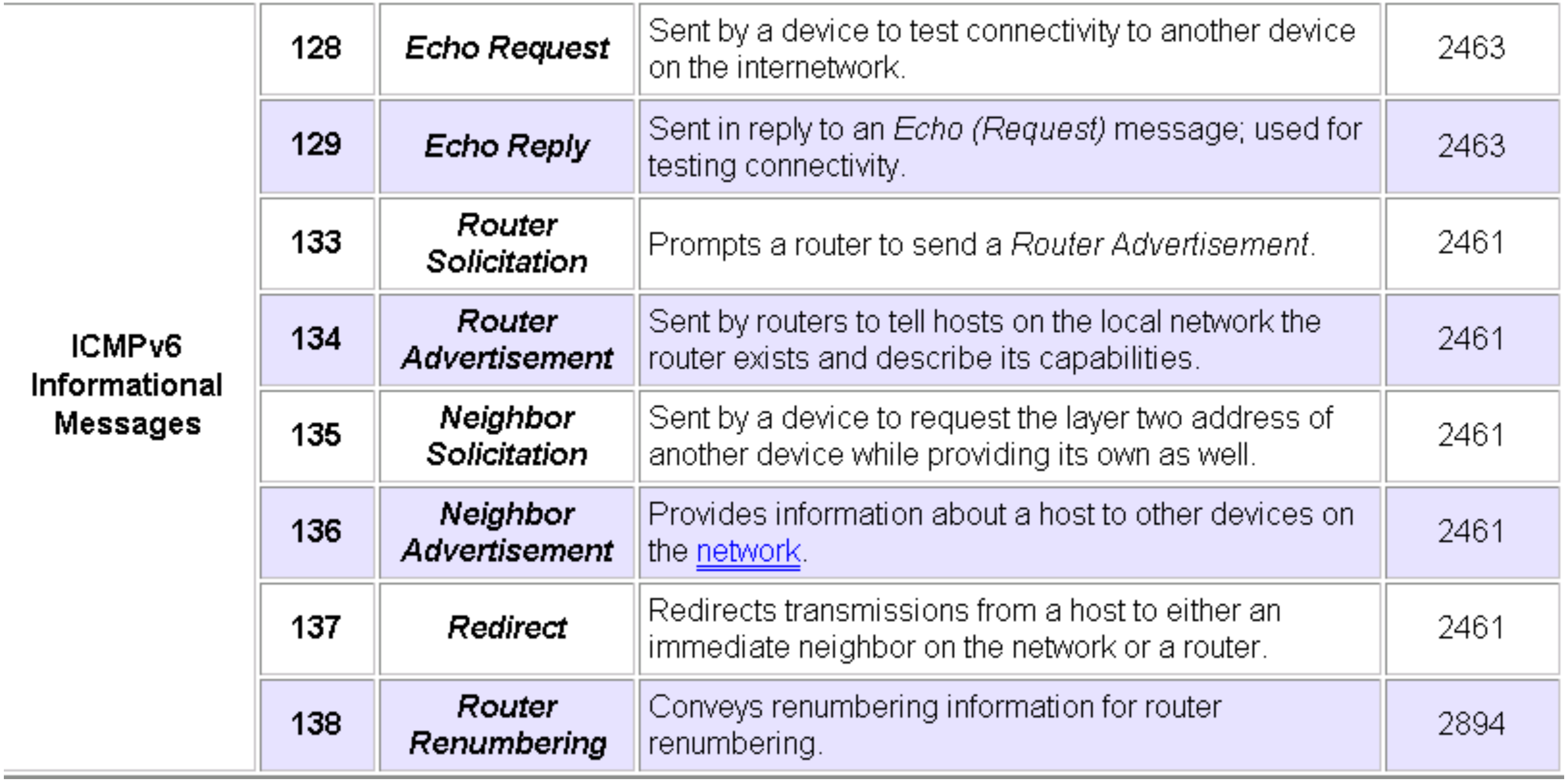


Type	Description	References
128	Echo request.	RFC 2463
129	Echo reply.	RFC 2463

ICMPv6 Error Messages

Type Value	**Message Name**	**Summary Description of Message Type**
1	Destination Unreachable	Indicates that a datagram could not be delivered to its destination. *Code* value provides more information on the nature of the error.
2	Packet Too Big	Sent when a datagram cannot be forwarded because it is too big for the MTU of the next hop in the route. This message is needed in IPv6 and not IPv4 because in IPv4, routers can fragment oversized messages, while in IPv6 they cannot.
3	Time Exceeded	Sent when a datagram has been discarded prior to delivery due to the *Hop Limit* field being reduced to zero.
4	Parameter Problem	Indicates a miscellaneous problem (specified by the *Code* value) in delivering a datagram.

ICMP Information Messages



ICMPv6 Informational Messages	**128**	***Echo Request***	Sent by a device to test connectivity to another device on the internetwork.	2463
	129	***Echo Reply***	Sent in reply to an *Echo (Request)* message; used for testing connectivity.	2463
	133	***Router Solicitation***	Prompts a router to send a *Router Advertisement*.	2461
	134	***Router Advertisement***	Sent by routers to tell hosts on the local network the router exists and describe its capabilities.	2461
	135	***Neighbor Solicitation***	Sent by a device to request the layer two address of another device while providing its own as well.	2461
	136	***Neighbor Advertisement***	Provides information about a host to other devices on the network.	2461
	137	***Redirect***	Redirects transmissions from a host to either an immediate neighbor on the network or a router.	2461
	138	***Router Renumbering***	Conveys renumbering information for router renumbering.	2894

Path MTU Discovery (PMTUD) for IPv6

PMTUDv6 - Overview

- To enable hosts to discover the min. MTU on a path to a particular destination.
- Fragmentation in IPv6 is not performed by intermediary routers.
- The source node may fragment packets by itself only when the path MTU is smaller than the packets to deliver
- PMTUD for IPv6 uses ICMPv6 error message
  - **Type 2 Packet Too Big**
- For detail info - **http://www.ietf.org/rfc/rfc1981.txt**

Differences between IPv4 & IPv6 MTU

- **Increased Default MTU**
  - In IPv4, the minimum MTU that routers and physical links were required to handle = **576 bytes**.
  - In IPv6, all links must handle a datagram size of at least **1280 bytes**.
  - improves efficiency by increasing the ratio of maximum payload to header length, and reduces the frequency with which fragmentation is required.
- **Elimination of En Route Fragmentation**
  - In IPv4, datagrams may be fragmented by either the source device, or by routers during delivery.
  - In IPv6, only the source node can fragment; **routers do not**.
  - The source must therefore fragment to the size of the smallest MTU on the route before transmission.

Differences between IPv4 & IPv6 MTU

- **MTU Size Error Feedback**
  - Since routers cannot fragment datagrams, they must drop them if they are forced to try to send a too-large datagram over a physical link.
  - A feedback process has been defined using ICMPv6 that lets routers tell source devices that they are using datagrams that are too large for the route.
- **Movement of Fragmentation Header Fields**
  - To reflect the decreased importance of fragmentation in IPv4, the permanent fields related to the process that were in the IPv4 header have been farmed out to a *Fragment* extension header, included only when needed.

Determining the Appropriate Datagram Size

How does the source know what size to use?

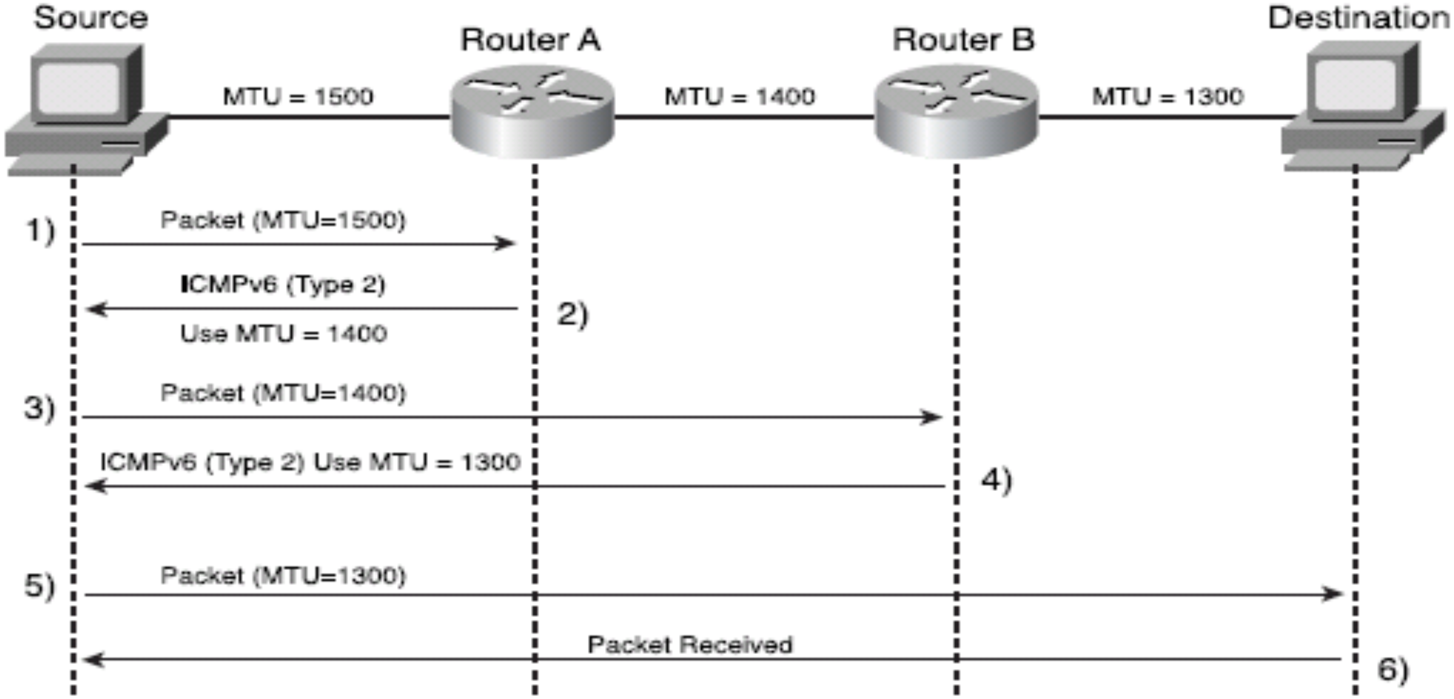
It has two choices:

1. **Use Default MTU**
   - ➢ Use the default MTU of **1280**, which all physical networks must be able to handle.
   - ➢ Good choice especially for short communications or for sending small amounts of data.
2. **Use Path MTU Discovery feature**
   - ➢ A node sends messages over a route to determine what the overall minimum MTU for the path is, in a technique very similar to how it is done in IPv4.

Path MTU Discovery Process

1. The sending node assumes that the path MTU is the link MTU of the interface on which the traffic is being forwarded.
2. The sending node sends IPv6 packets at the **path MTU size**.
3. If a router on the path is unable to forward the packet over a link with a link MTU that is smaller than the size of the packet, it discards the IPv6 packet and sends an **ICMPV6 Packet Too Big message back to the sending node**. The ICMPV6 Packet Too Big message contains the link MTU of the link on which the forwarding failed.
4. The sending node sets the path MTU for packets being sent to the destination to the value of the MTU field in the ICMPv6 Packet Too Big message.
5. The sending node starts again at step 2 and repeats steps 2 through 4 for as many times as are necessary to discover the path MTU.

PMTUD uses ICMPv6 Type 2 Message



IPv6 Fragmentation

IPv6 Fragmentation

For purposes of fragmentation, IPv6 datagrams are broken into two pieces:

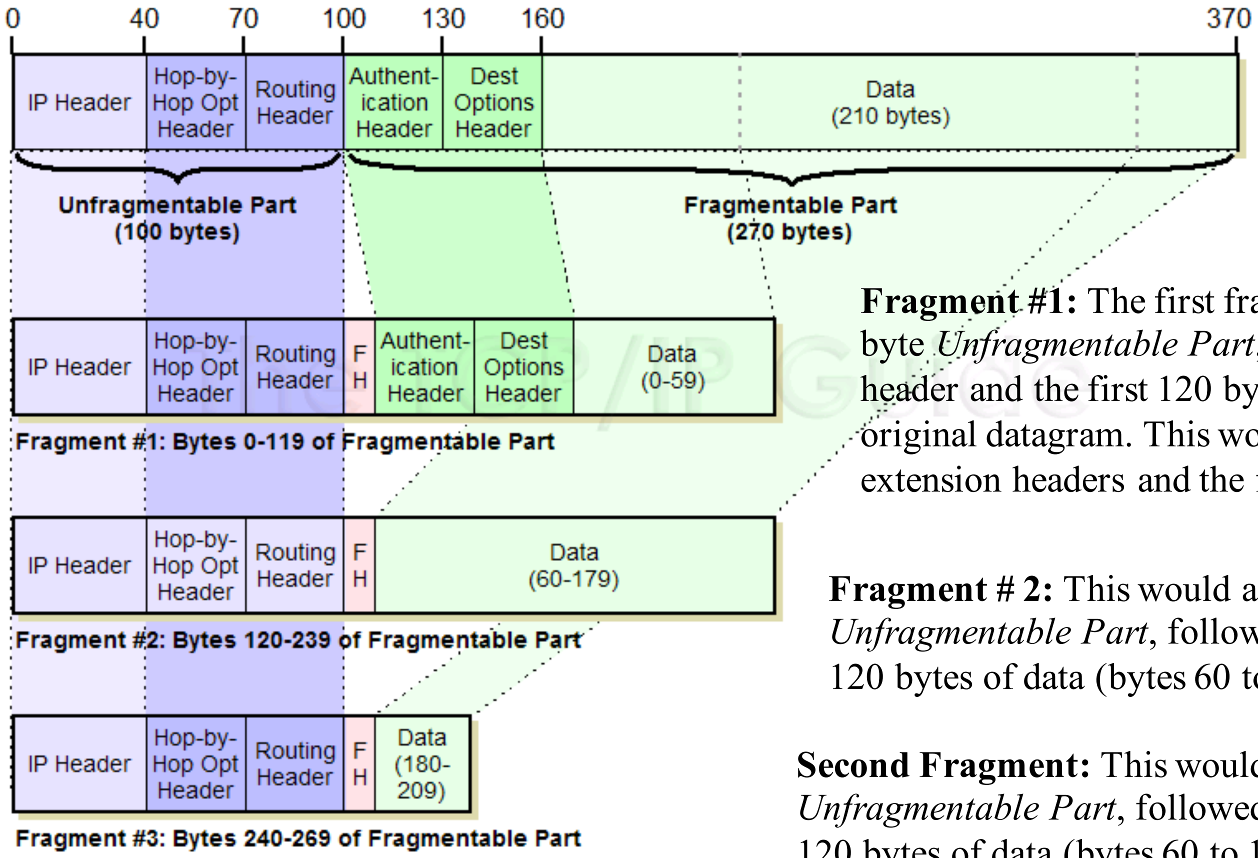
- **Unfragmentable Part**
  - ➢ Includes the main header of the original datagram + any extension headers that need to be present in each fragment - ***Hop-By-Hop Options***, ***Destination Options*** (for those options to be processed by devices along a route) and ***Routing***.
- **Fragmentable Part**
  - ➢ Data portion of the datagram + other extension headers if present - authentication Header, Encapsulating Security Payload and/or Destination Options (for options to be processed only by the final destination).
- **Unfragmentable Part** must be present in each fragment, while the **fragmentable part** is split up amongst the fragments.

IPv6 Fragment Sets

- So to fragment a datagram, a device creates a set of fragment datagrams, each of which contains the following, in order:
- **Unfragmentable Part**
  - The full *Unfragmentable Part* of the original datagram, with its *Payload Length* changed to the length of the fragment datagram.
- **Fragment Header**
  - A *Fragment* header with the *Fragment Offset*, *Identification* and *M* flags set in the same way they are used in IPv4.
- **Fragment**
  - A fragment of the *Fragmentable Part* of the original datagram. Note that each fragment must have a length that is a multiple of 8 bytes, because the value in the *Fragment Offset* field is specified in multiples of 8 bytes.

IPv6 Fragmentation

Suppose we need to send this over a link with an MTU of only 230 bytes.
We would actually require three fragments, not the two, because of the need to put the two 30-byte unfragmentable extension headers in each fragment, and the requirement that each fragment be a length that is a multiple of 8.



Fragment #1: The first fragment would consist of the 100-byte *Unfragmentable Part*, followed by an 8-byte *Fragment* header and the first 120 bytes of the *Fragmentable Part* of the original datagram. This would contain the two fragmentable extension headers and the first 60 bytes of data.

Fragment # 2: This would also contain the 100-byte *Unfragmentable Part*, followed by a *Fragment* header and 120 bytes of data (bytes 60 to 179).

Second Fragment: This would also contain the 100-byte *Unfragmentable Part*, followed by a *Fragment* header and 120 bytes of data (bytes 60 to 179).

IPv6 Neighbour Discovery Protocol (NDP)

IPv6 ND - Overview

- IPv6 ND is a set of messages and processes that determine relationships between neighboring nodes.
- ND replaces ARP, ICMP Router Discovery, and ICMP Redirect used in IPv4 and provides additional functionality.

IPv6 ND - Overview

- **ND is used by hosts to:**
  - Discover neighbouring routers.
  - Discover addresses, address prefixes, and other configuration parameters.
- **ND is used by routers to:**
  - Advertise their presence, host configuration parameters, and on-link prefixes.
  - Inform hosts of a better next-hop address to forward packets for a specific destination.
- **ND is used by nodes to:**
  - Resolve the link-layer address of a neighbouring node to which an IPv6 packet is being forwarded and determine when the link-layer address of a neighbouring node has changed.
  - Determine whether a neighbour is still reachable.

IPv6 ND – What is Neighbour?

- **Neighbour** is one that has been used for years in various networking standards and technologies to refer to devices that are local to each other.
- Two devices are **neighbours** if they are on the same local network, meaning that they can send information to each other directly.
- Most of the functions of the ND protocol are implemented using a set of **five special ICMPv6 control messages**.
- ND is a **messaging protocol.**
  - It doesn't implement a single specific function but rather a group of activities that are performed through the exchange of messages.
  - ND standard describes **nine specific functions** performed by the protocol.

ICMPv6 Control Messages Used By NDP

talk to router

1. ***RS- Router Solicitation*** **Messages (ICMPv6 Type 133)**
   - Sent by hosts to request that any local routers send a *Router Advertisement* message so they don't have to wait for the next regular advertisement message.
2. ***RA - Router Advertisement Messages (ICMPv6 Type 134)***
   - Sent regularly by routers to tell hosts that they exist and provide important prefix and parameter information to them.

similar to ARP: ask ur neighbor who has this IP.

3. ***NS - Neighbor Solicitation Messages (ICMPv6 Type 135)***
   - Sent to verify the existence of another host and to ask it to transmit a Neighbor Advertisement.

then ur neighbor replies.

talk to neighbor

4. ***NA - Neighbor Advertisement*** **Messages (ICMPv6 Type 136)**
   - Sent by hosts to indicate the existence of the host and provide information about it.
5. ***Redirect*** **Messages (ICMPv6 Type 137)**
   - Sent by a router to tell a host of a better method to route data to a particular destination.

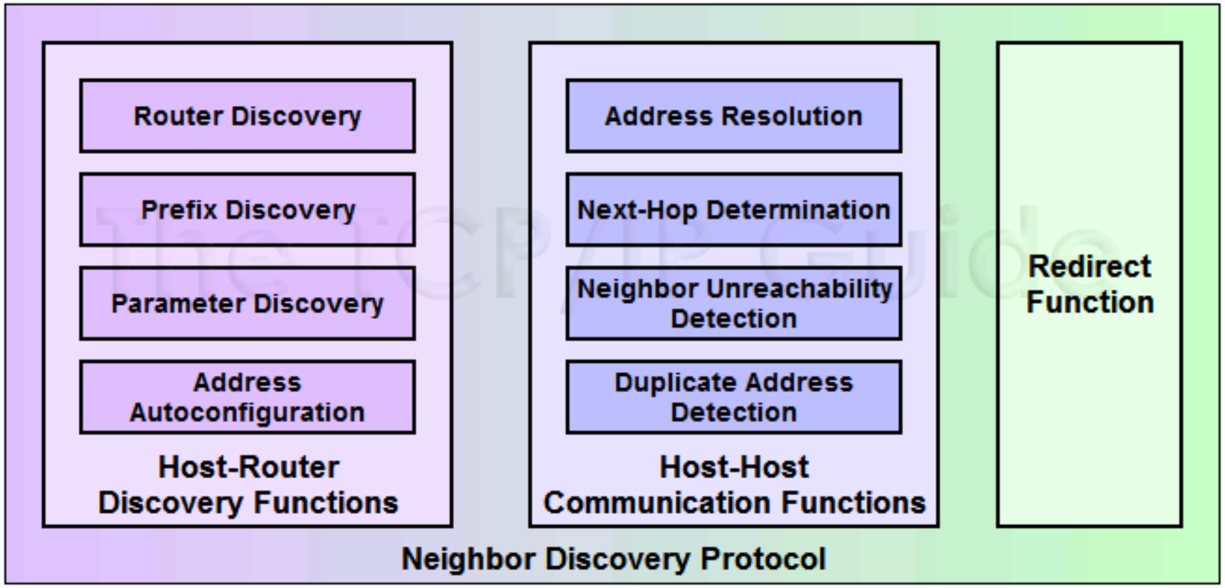
ICMPv6 Messages Used by NDP

Mechanism	Type 133 (RS)	Type 134 (RA)	Type 135 (NS)	Type 136 (NA)	Type 137 (Redirect)
Replacement of ARP			X	X	
Prefix advertisement	X	X			
Prefix renumbering	X	X			
DAD ← duplicate address detection.			X		
Router redirection					X

NDP Functional Groups and Functions

Mainly three functions

1. Host-Router Functions
2. Host-Host Communication Functions
3. Redirect Function



Router Discovery

Prefix Discovery

Parameter Discovery

Address Autoconfiguration

Host-Router Discovery Functions

Address Resolution

Next-Hop Determination

Neighbor Unreachability Detection

Duplicate Address Detection

Host-Host Communication Functions

Redirect Function

Neighbor Discovery Protocol

1. Host-Router Discovery Functions

One of the two main groups of functions in ND are those that facilitate the discovery of local routers and the exchange of information between them and hosts.

- **Router Discovery**
  - Core function of this group: the method by which hosts locate routers on their local network.
- **Prefix Discovery**
  - Closely related to the process of router discovery is prefix discovery.
  - To determine what network they are on, which in turn tells them how to differentiate between local and distant destinations and whether to attempt direct or indirect delivery of datagrams.
- **Parameter Discovery**
  - A host learns important parameters about the local network and/or routers, such as the MTU of the local link.
- **Address Auto-configuration**
  - Hosts in IPv6 are designed to be able to **automatically configure themselves**, but this requires information that is normally provided by a router.

2. Host-Host Communication Functions

- **Address Resolution**

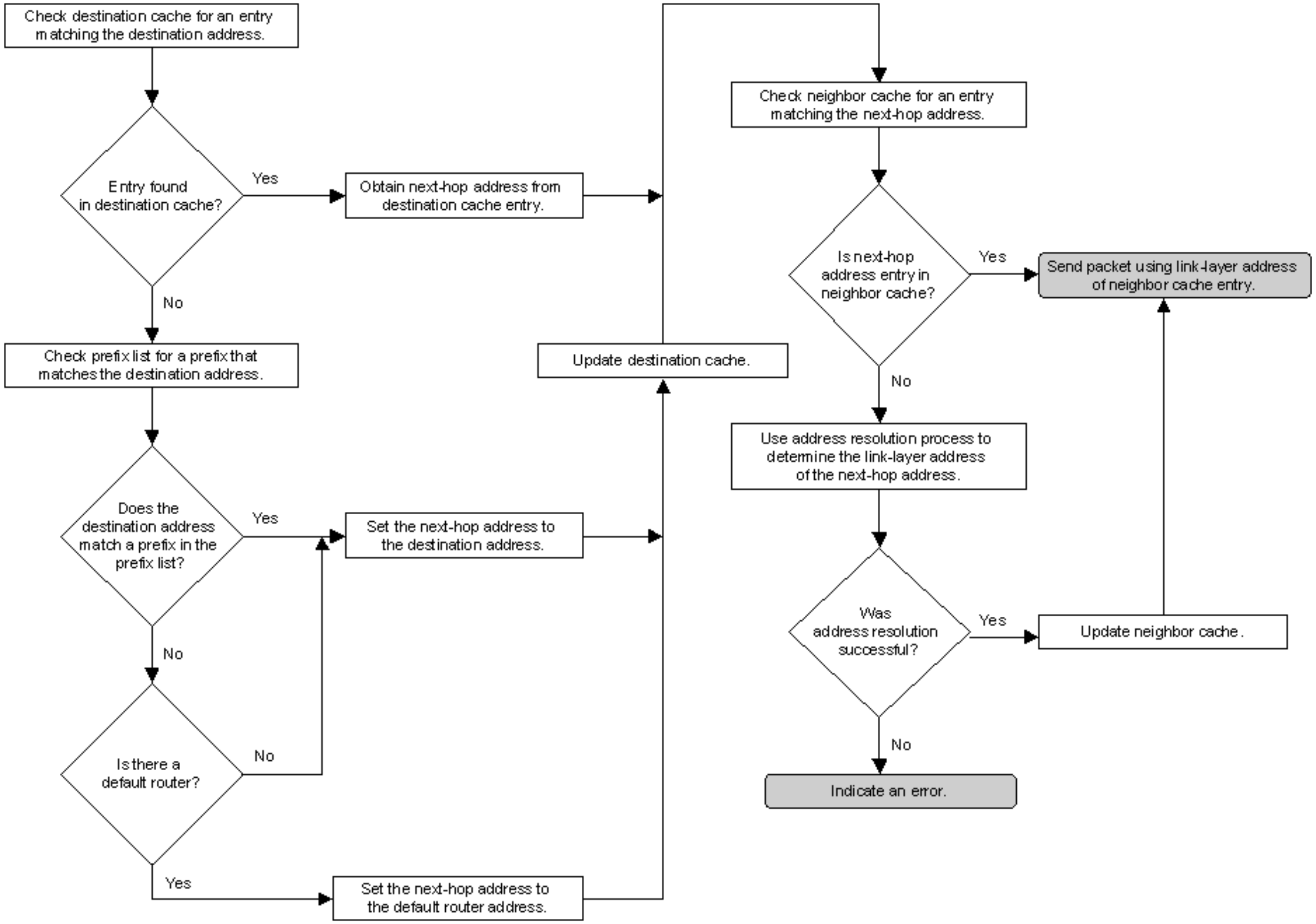
  IP → MAC translation.

  - The process by which a device determines the layer two address of another device on the local network from that device's layer three (IP) address.
  - Performed by ARP in IP version 4.
- **Next-Hop Determination**
  - Looking at an IP datagram's destination address and determining where it should next be sent.
- **Neighbor Unreachability Detection**
  - Determining whether or not a neighbor device can be directly contacted.
- **Duplicate Address Detection (DAD)**
  - Determining if an address that a device wishes to use already exists on the network.

3. Redirect Function

- The last functional group contains just one function: *Redirect*.
- The technique whereby a router informs a host of a better next-hop node to use for a particular destination.

The Host Sending Algorithm



Check destination cache for an entry matching the destination address.

Entry found in destination cache?

Yes

Obtain next-hop address from destination cache entry.

No

Check prefix list for a prefix that matches the destination address.

Does the destination address match a prefix in the prefix list?

Yes

Set the next-hop address to the destination address.

No

Is there a default router?

No

Yes

Set the next-hop address to the default router address.

Update destination cache.

Check neighbor cache for an entry matching the next-hop address.

Is next-hop address entry in neighbor cache?

Yes

Send packet using link-layer address of neighbor cache entry.

No

Use address resolution process to determine the link-layer address of the next-hop address.

Was address resolution successful?

Yes

Update neighbor cache.

No

Indicate an error.

How Neighbour Solicitation and Neighbour Advertisement Works

- A node can use following special addresses:
  - All-node multicast address (FF02::1, destination)
  - All-routers multicast address (FF02::2, destination)
  - Solicited-mode multicast address (destination)
  - Link-local address (sources or destination)
  - Unspecified address (::, source)

Address Resolution

- The address resolution process for IPv6 nodes consists of an exchange of Neighbor Solicitation and Neighbor Advertisement messages to resolve the link-layer address of the on-link next-hop address for a given destination.
- The sending host sends a **multicast Neighbor Solicitation message** on the appropriate interface.
- The multicast address of the Neighbor Solicitation message is the solicited-node multicast address derived from the target IP address.
- The Neighbor Solicitation message includes the link-layer address of the sending host in the Source Link-Layer Address option.
- When the target host receives the Neighbor Solicitation message, it updates its own neighbor cache based on the source address of the Neighbor Solicitation message and the link-layer address in the Source Link-Layer Address option.

Address Resolution

- Next, the target node sends a **unicast Neighbor Advertisement** to the Neighbor Solicitation sender.
- The Neighbor Advertisement includes the Target Link-Layer Address option.
- After receiving the Neighbor Advertisement from the target, the sending host updates its neighbor cache with an entry for the target based upon the information in the Target Link-Layer Address option.
- At this point, unicast IPv6 traffic between the sending host and the target of the Neighbor Solicitation can be sent.

Mapping IPv4/IPv6 Multicast Address into Multicast MAC Addresses

- IPv4 01:00:5E:00:00:00 - 01:00:5E:7F:FF:FF
    - Insert the lower 23 bits of a IPv4 multicast address into the MAC address
- IPv6 33:33:XX:XX:XX:XX
    - Insert the lower 32 bits of a IPv6 multicast address into the MAC address

Address Resolution: Example

the idea is similar to IPv4 ARP but differen

The multicast Neighbor Solicitation for address resolution

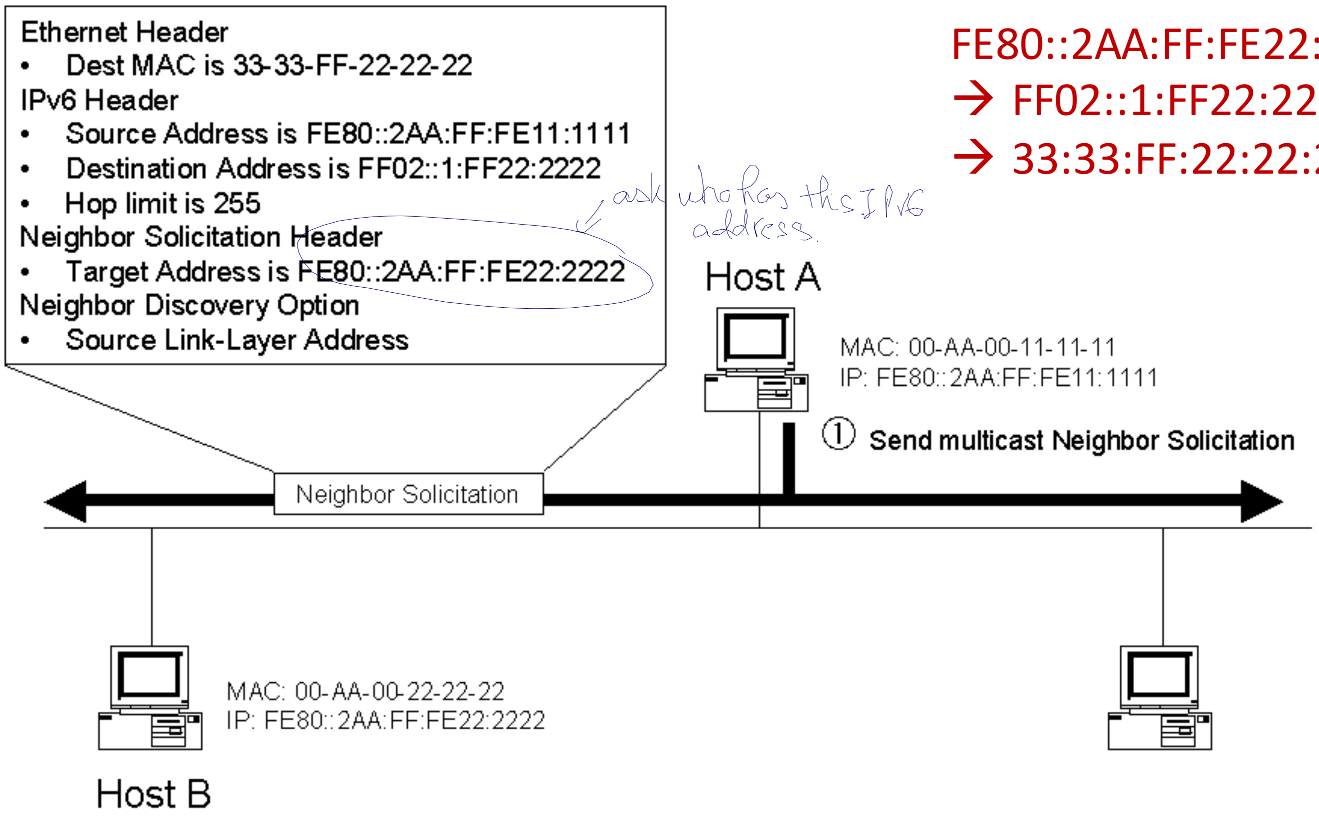
is IPv4 ARP we broadcast

& IPv6 we multicast

FE80::2AA:FF:FE22:2222

→ FF02::1:FF22:2222

→ 33:33:FF:22:22:22



Ethernet Header
- Dest MAC is 33-33-FF-22-22-22

IPv6 Header
- Source Address is FE80::2AA:FF:FE11:1111
- Destination Address is FF02::1:FF22:2222
- Hop limit is 255

Neighbor Solicitation Header
- Target Address is FE80::2AA:FF:FE22:2222

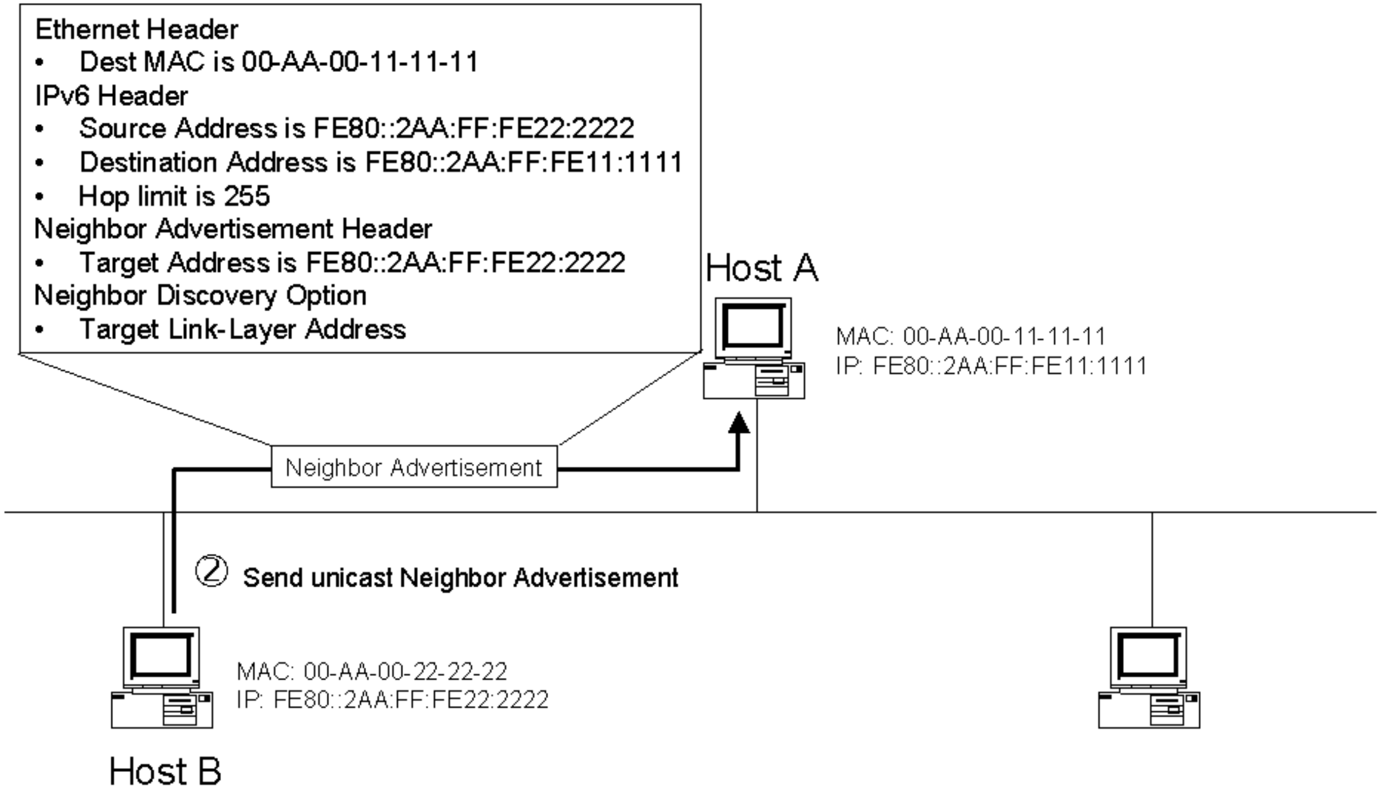
Neighbor Discovery Option
- Source Link-Layer Address

ask who has this IPv6 address.

① Send multicast Neighbor Solicitation

Address Resolution: Example continue

The unicast Neighbor Advertisement for address resolution



Ethernet Header
- Dest MAC is 00-AA-00-11-11-11

IPv6 Header
- Source Address is FE80::2AA:FF:FE22:2222
- Destination Address is FE80::2AA:FF:FE11:1111
- Hop limit is 255

Neighbor Advertisement Header
- Target Address is FE80::2AA:FF:FE22:2222

Neighbor Discovery Option
- Target Link-Layer Address

Neighbor Advertisement

② Send unicast Neighbor Advertisement

Host A
MAC: 00-AA-00-11-11-11
IP: FE80::2AA:FF:FE11:1111

Host B
MAC: 00-AA-00-22-22-22
IP: FE80::2AA:FF:FE22:2222

Prefix advertisement

It uses

- Router Advertisement (RA) message &
- All node multicast address (FF02::1)
- RA sent periodically on the local link to all node-multicast address

Router Advertisement Message Parameters

- IPv6 prefix
  - Multiple ipv6 prefixes can be advertised per local link
  - By default – prefix length = 64 bits
  - Nodes gets IPv6 address, they append their link-layer in EUI-format to the prefix received = 128 bit IPv6 node address.
- Life-time
  - Lifetime may vary from 0 to infinite.
  - Two types of lifetime value per prefix:
    - Valid Lifetime: how long the node's address remains in valid state
    - Preferred Lifetime: how long the address configured by a node remains preferred. It must be <= valid lifetime

Router Advertisement Message Parameters

- Default router information
- Information about the existence and lifetime of the default router’s ipv6 address
- Default router’s address = router’s link local address
- Flags/options
- Use flags to instruct nodes to use stateful configuration than stateless

Router Discovery

- Router discovery is the process through which nodes attempt to discover the set of routers on the local link.
- Router discovery in IPv6 is similar to ICMP Router Discovery for IPv4 described in RFC 1256.
- An important difference between ICMPv4 Router Discovery and IPv6 Router Discovery is the mechanism through which a new default router is selected when the current one becomes unavailable.
- **In ICMPv4 Router Discovery**, the Router Advertisement message includes an **Advertisement Lifetime field.**
- It is the time after which the router, upon receiving its last Router Advertisement message, can be considered unavailable.
- In the worst case, a router can become unavailable and hosts will not attempt to discover a new default router until the Router Advertisement time has elapsed.

IPv6 Router Discovery

- IPv6 has a Router Lifetime field in the Router Advertisement message.
- It indicates the length of time that the router can be considered a default router.
- If the current default router becomes unavailable, the condition is detected through **neighbor unreachability detection** instead of the Router Lifetime field in the Router Advertisement message.
- Because neighbor unreachability detection determines that the router is no longer reachable, a new router is chosen immediately from the default router list.

Router Discovery - parameters

- In addition to configuring a default router, IPv6 router discovery also configures the following:
- The default setting for the **Hop Limit field** in the IPv6 header.

use DHCP

- A determination of whether the node should use a **stateful address protocol, such as DHCPv6**, for addresses and other configuration parameters.

stateless uses router solicitation & advertement

- The **timers** used in reachability detection and the retransmission of Neighbor Solicitations.
- The **list of network prefixes** defined for the link. Each network prefix contains both the IPv6 network prefix and its valid and preferred lifetimes.
- If indicated, a network prefix combined with the interface identifier creates a stateless IP address configuration for the receiving interface. A network prefix also defines the range of addresses for nodes on the local link.
- The **MTU** of the local link.

IPv6 Router Discovery Processes

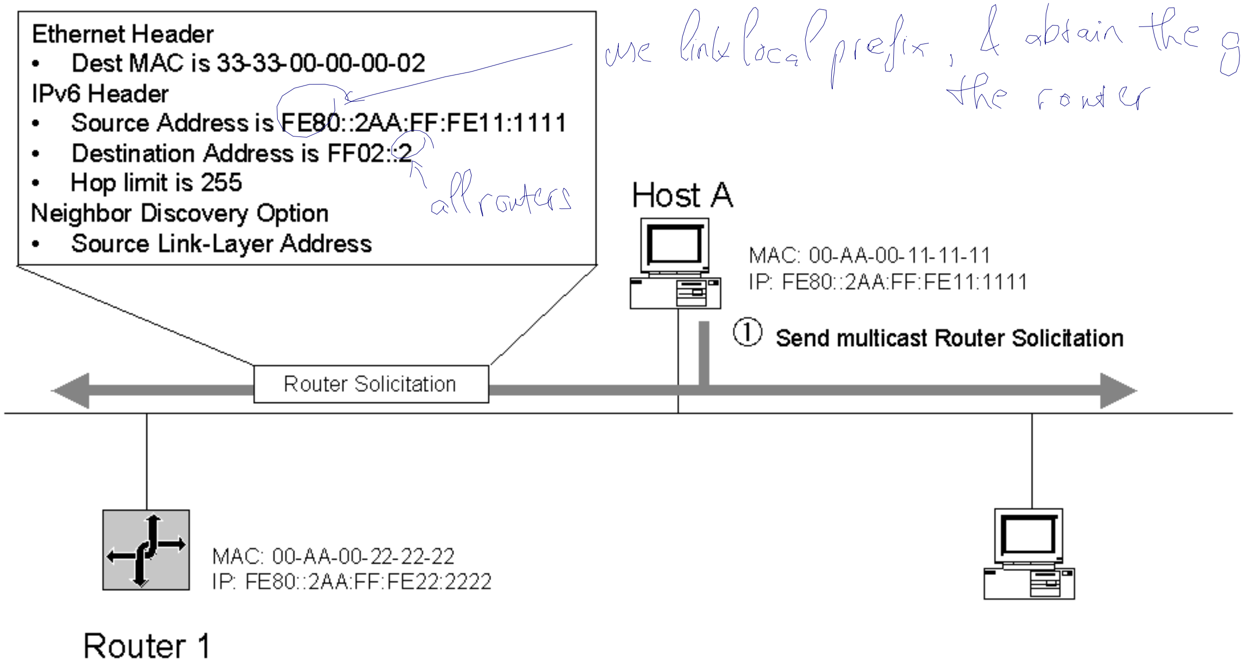
- IPv6 routers periodically send a Router Advertisement message on the local link advertising their existence as routers.
- They also provide configuration parameters such as default hop limit, MTU, and prefixes.
- Active IPv6 hosts on the local link receive the Router Advertisement messages and use the contents to maintain the default router list, the prefix list, and other configuration parameters.
- A host that is starting up sends a Router Solicitation message to the link-local scope all-routers multicast address (FF02::2).
- Upon receipt of a Router Solicitation message, all routers on the local link send a unicast Router Advertisement message to the node that sent the Router Solicitation.
- The node receives the Router Advertisement messages and uses their contents to build the default router and prefix lists and set other configuration parameters.

IPv6 Router Discovery Processes

- Any node can send RS to all-routers multicast address FF02::2 on the local link
- When RS is received, a router responds with RA using all-node multicast FF02::1
- To avoid flooding of RS on the link, each node can send only three RS at boot time.

IPv6 Router Discovery Processes -

The multicast Router Solicitation for router and prefix discovery



Ethernet Header
- Dest MAC is 33-33-00-00-00-02

IPv6 Header
- Source Address is FE80::2AA:FF:FE11:1111
- Destination Address is FF02::2
- Hop limit is 255

Neighbor Discovery Option
- Source Link-Layer Address

use link local prefix, & obtain the global prefix from the router

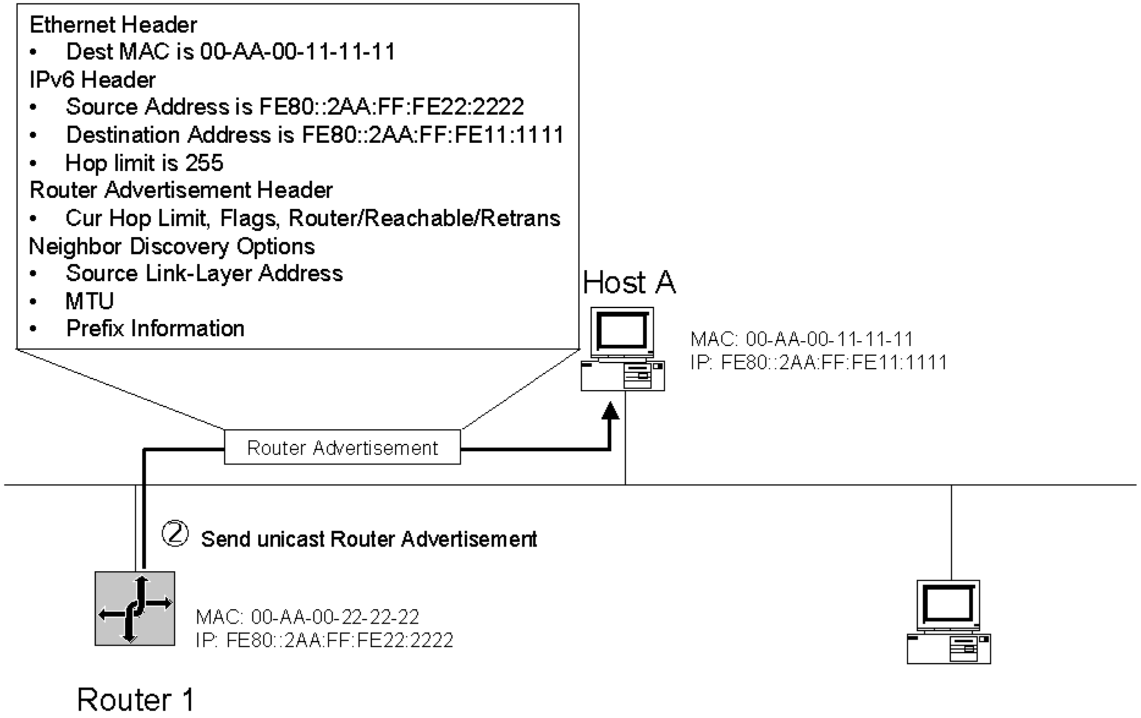
all routers

① Send multicast Router Solicitation

- To forward packets to off-link destinations, Host A must discover the presence of Router 1.
- Host A sends a multicast Router Solicitation to the address FF02::2

IPv6 Router Discovery Processes –

The unicast Router Advertisement for router and prefix discovery



Ethernet Header
- Dest MAC is 00-AA-00-11-11-11

IPv6 Header
- Source Address is FE80::2AA:FF:FE22:2222
- Destination Address is FE80::2AA:FF:FE11:1111
- Hop limit is 255

Router Advertisement Header
- Cur Hop Limit, Flags, Router/Reachable/Retrans

Neighbor Discovery Options
- Source Link-Layer Address
- MTU
- Prefix Information

Host A

MAC: 00-AA-00-11-11-11
IP: FE80::2AA:FF:FE11:1111

Router Advertisement

② Send unicast Router Advertisement

MAC: 00-AA-00-22-22-22
IP: FE80::2AA:FF:FE22:2222

Router 1

- Router 1, having registered the multicast address of 33-33-00-00-00-02 with its Ethernet adapter, receives and processes the Router Solicitation.
- Router 1 responds with a unicast Router Advertisement message containing configuration parameters and local link prefixes

Duplicate Address Detection (DAD)

everybody has a link local. FE80:: , so it needs to use DAD. to detect if its IP is duplicated → use neighbor solicitator

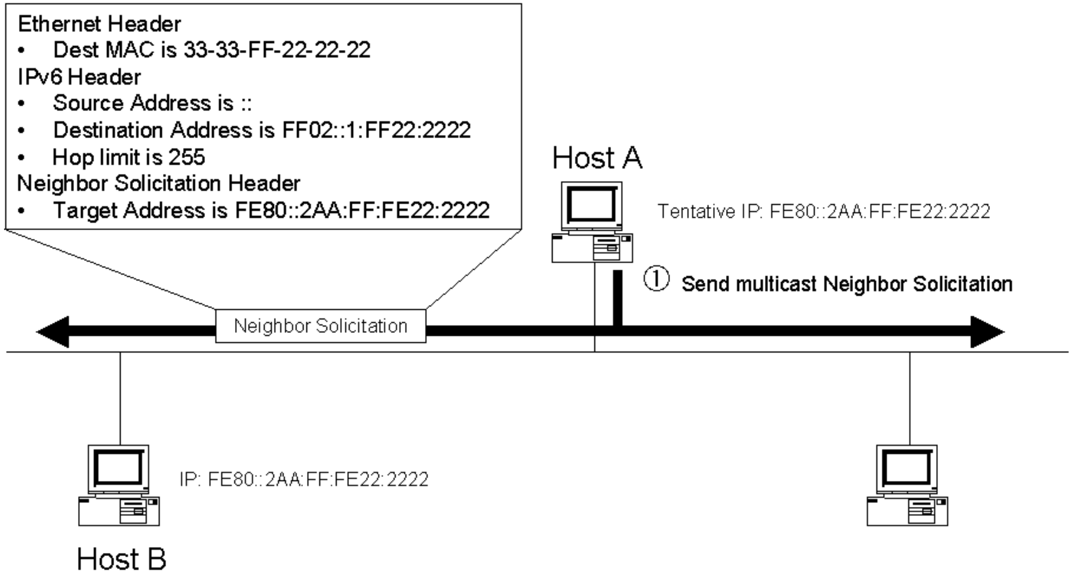
- IPv4 nodes use ARP Request messages and a method called gratuitous ARP to detect a duplicate IP address on the local link.
- Similarly, IPv6 nodes use the **Neighbor Solicitation message** to detect duplicate address use on the local link.
- Before a node can configure its IPv6 address using stateless autoconfiguration, it must verify on the local link that the tentative address it wants to use is unique and not already in use by another mode.
- Node sending a Neighbour Solicitation (NS) on the local link using **unspecified address (::)** as its source address and **solicited-node multicast** of the tentative unicast address as the destination address.
- If a duplicate address – no assignment of this unicast address

DAD – Example

The multicast Neighbor Solicitation for duplicate address detection

- Host B has a link-local address of FE80::2AA:FF:FE22:2222.
- Host A is attempting to use the link-local address of FE80::2AA:FF:FE22:2222.
- Before Host A can use this link-local address, it must verify its uniqueness through duplicate address detection.

Host A sends a solicited-node multicast Neighbor Solicitation to the address FF02::1:FF22:2222



Ethernet Header
- Dest MAC is 33-33-FF-22-22-22

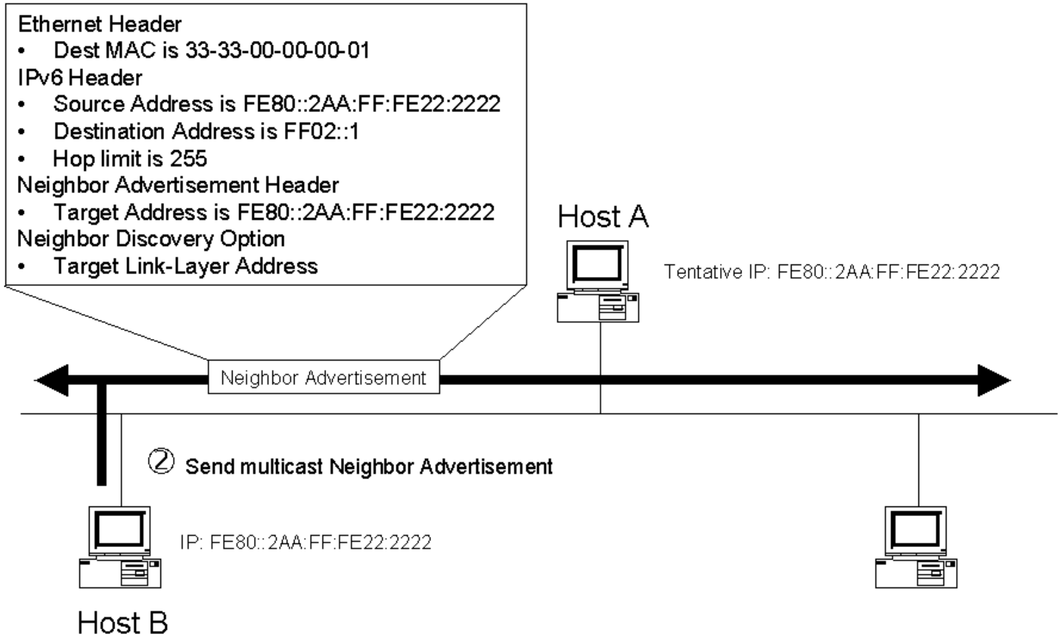
IPv6 Header
- Source Address is ::
- Destination Address is FF02::1:FF22:2222
- Hop limit is 255

Neighbor Solicitation Header
- Target Address is FE80::2AA:FF:FE22:2222

Host A
Tentative IP: FE80::2AA:FF:FE22:2222
① Send multicast Neighbor Solicitation
Neighbor Solicitation
IP: FE80::2AA:FF:FE22:2222
Host B

DAD – Example

The multicast Neighbor Advertisement for duplicate address detection



Ethernet Header
- Dest MAC is 33-33-00-00-00-01

IPv6 Header
- Source Address is FE80::2AA:FF:FE22:2222
- Destination Address is FF02::1
- Hop limit is 255

Neighbor Advertisement Header
- Target Address is FE80::2AA:FF:FE22:2222

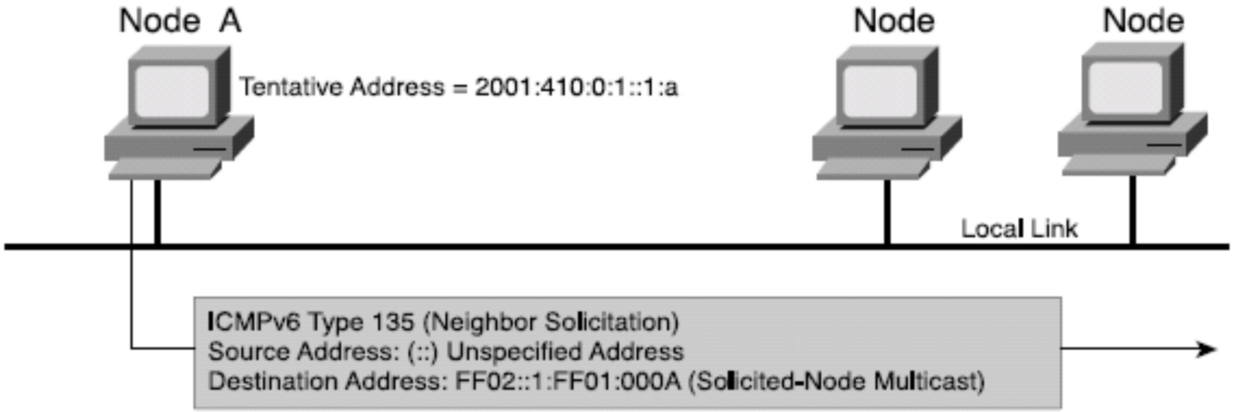
Neighbor Discovery Option
- Target Link-Layer Address

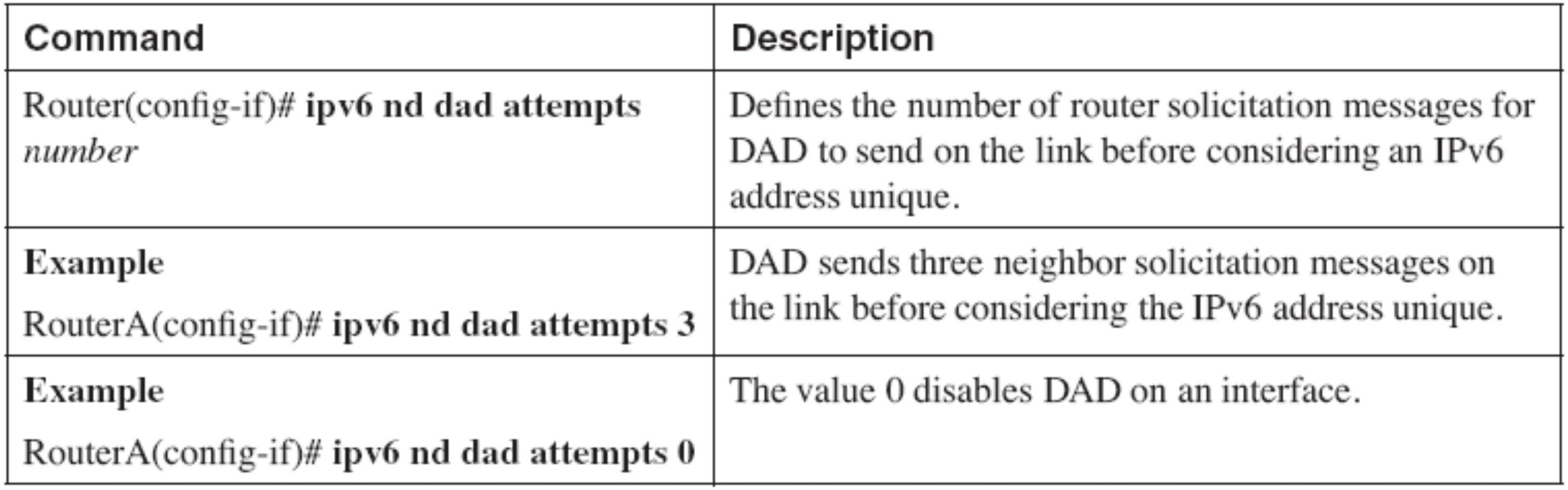
Host A
Tentative IP: FE80::2AA:FF:FE22:2222
Neighbor Advertisement
② Send multicast Neighbor Advertisement
IP: FE80::2AA:FF:FE22:2222
Host B

- Host B, having registered the solicited-node multicast address of 33-33-FF-22-22-22 with its Ethernet adapter, receives and processes the Neighbor Solicitation.
- Host B notes that the source address is the unspecified address.
- Host B then responds with a multicast Neighbor Advertisement message

Duplicate Address Detection

- By default, DAD is enabled on Cisco routers


Node A
Tentative Address = 2001:410:0:1::1:a
Node
Node
Local Link
ICMPv6 Type 135 (Neighbor Solicitation)
Source Address: (::) Unspecified Address
Destination Address: FF02::1:FF01:000A (Solicited-Node Multicast)

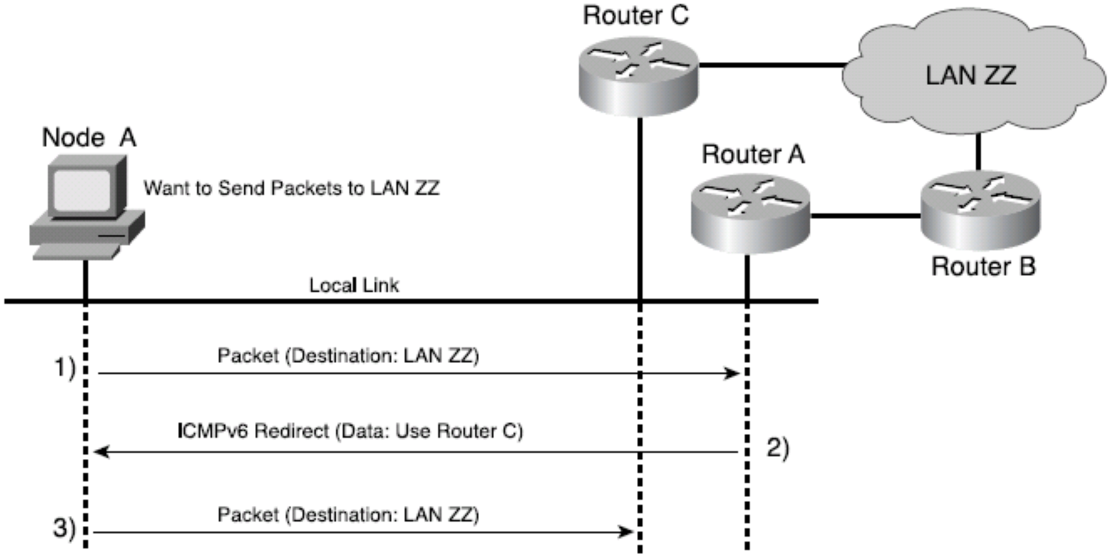


Command	Description
Router(config-if)# **ipv6 nd dad attempts** *number*	Defines the number of router solicitation messages for DAD to send on the link before considering an IPv6 address unique.
Example RouterA(config-if)# **ipv6 nd dad attempts 3**	DAD sends three neighbor solicitation messages on the link before considering the IPv6 address unique.
Example RouterA(config-if)# **ipv6 nd dad attempts 0**	The value 0 disables DAD on an interface.

Router Redirect

finished.

- Routers use the redirect function to inform originating hosts of a better first-hop neighbor to which traffic should be forwarded for a specific destination.
- Nodes receiving it may modify its routing table according to the new router address.



Router C
LAN ZZ
Node A
Router A
Want to Send Packets to LAN ZZ
Router B
Local Link
1)
Packet (Destination: LAN ZZ)
ICMPv6 Redirect (Data: Use Router C)
2)
3)
Packet (Destination: LAN ZZ)

✓

Router Redirect

- There are two instances where redirect is used:

1. **A router informs an originating host of the IP address of a router available on the local link that is “closer” to the destination.**
   - “Closer” is routing metric function used to reach the destination network segment.
   - This condition can occur when there are multiple routers on a network segment and the originating host chooses a default router and it is not the best one to use to reach the destination.
2. **A router informs an originating host that the destination is a neighbor (it is on the same link as the originating host).**
   - This condition can occur when the prefix list of a host does not include the prefix of the destination.
   - Because the destination does not match a prefix in the list, the originating host forwards the packet to its default router.

Router Redirect Process

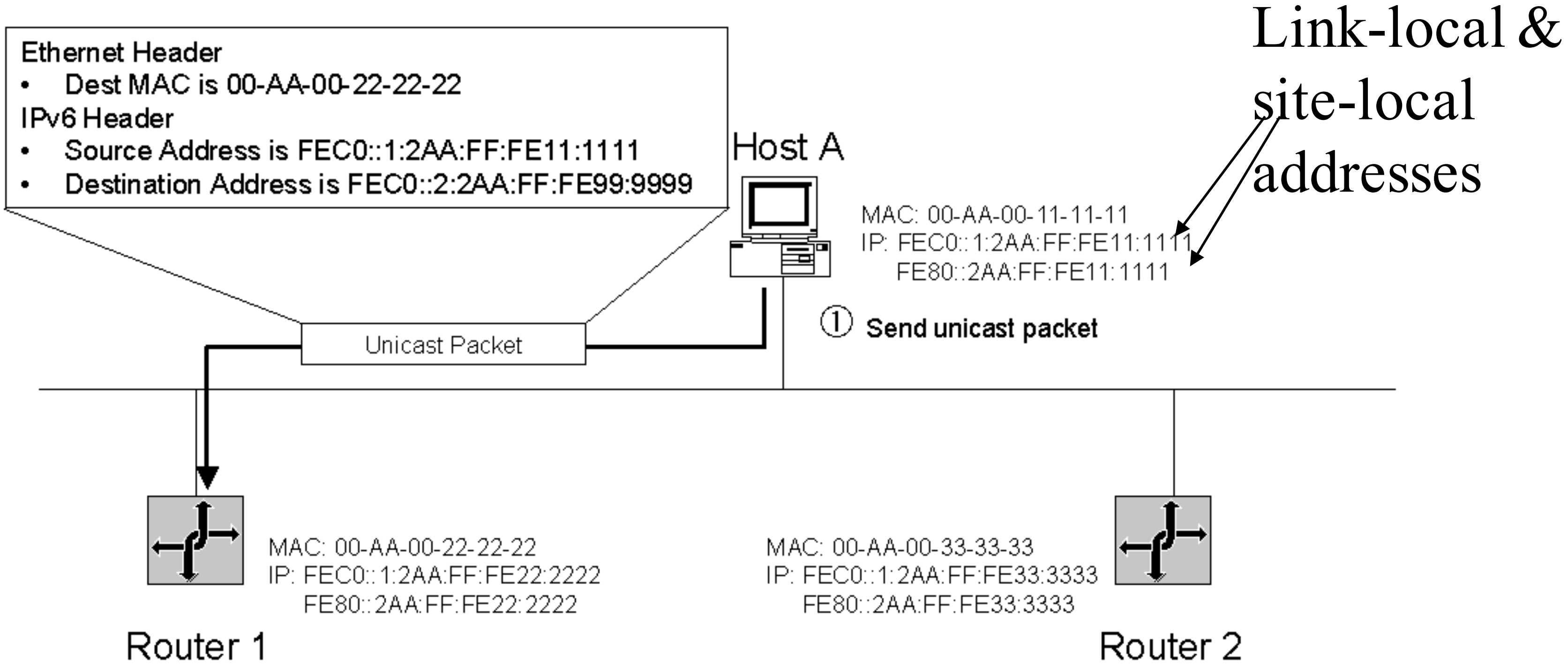
1. The originating host forwards a unicast packet to its default router.
2. The router processes the packet and notes that the address of the originating host is a neighbor.
   - Additionally, it notes that the addresses of both the originating host and the next-hop are on the same link.
3. The router forwards the packet to the appropriate next-hop address.
4. The router sends the originating host a Redirect message.
   - In the Target Address field of the Redirect message is the next-hop address of the node to which the originating host should send packets addressed to the destination.

Router Redirect Process

- **For packets redirected to a router**, the Target Address field is set to the link-local address of the router.
- **For packets redirected to a host**, the Target Address field is set to the destination address of the packet originally sent.
- The Redirect message includes the Redirected Header option. It might also include the Target Link-Layer Address option.
- Upon receipt of the Redirect message, the originating host updates the destination address entry in the destination cache with the address in the Target Address field.
  - If the Target Link-Layer Address option is included in the Redirect message, its contents are used to create or update the corresponding neighbor cache entry.

Router Redirect Process - Example

The unicast packet forwarded by the originating node



Ethernet Header
- Dest MAC is 00-AA-00-22-22-22

IPv6 Header
- Source Address is FEC0::1:2AA:FF:FE11:1111
- Destination Address is FEC0::2:2AA:FF:FE99:9999

Host A

MAC: 00-AA-00-11-11-11
IP: FEC0::1:2AA:FF:FE11:1111
FE80::2AA:FF:FE11:1111

Link-local & site-local addresses

① Send unicast packet

Unicast Packet

Router 1

MAC: 00-AA-00-22-22-22
IP: FEC0::1:2AA:FF:FE22:2222
FE80::2AA:FF:FE22:2222

Router 2

MAC: 00-AA-00-33-33-33
IP: FEC0::1:2AA:FF:FE33:3333
FE80::2AA:FF:FE33:3333

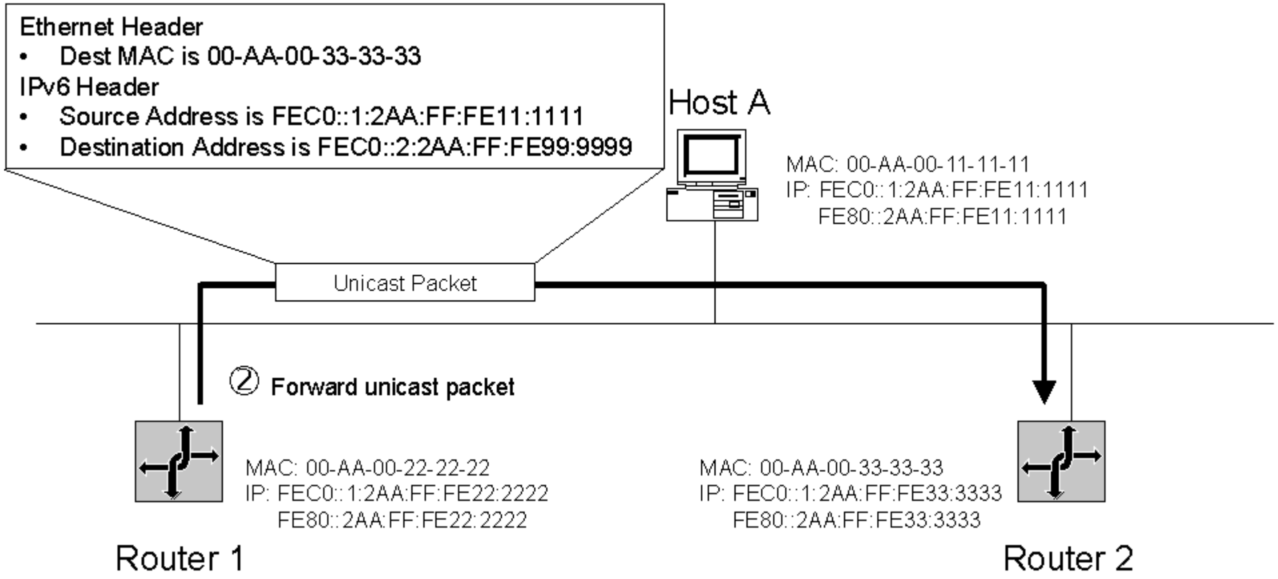
Host A is sending a packet to an off-link host at FEC0::2:2AA:FF:FE99:9999 (not shown) and is using Router 1 as its current default router.

However, Router 2 is the better router to use to reach this destination.

Host A sends the packet destined to FEC0::2:2AA:FF:FE99:9999 to Router 1

Router Redirect Process - Example

The unicast packet forwarded by the router



Ethernet Header

- Dest MAC is 00-AA-00-33-33-33

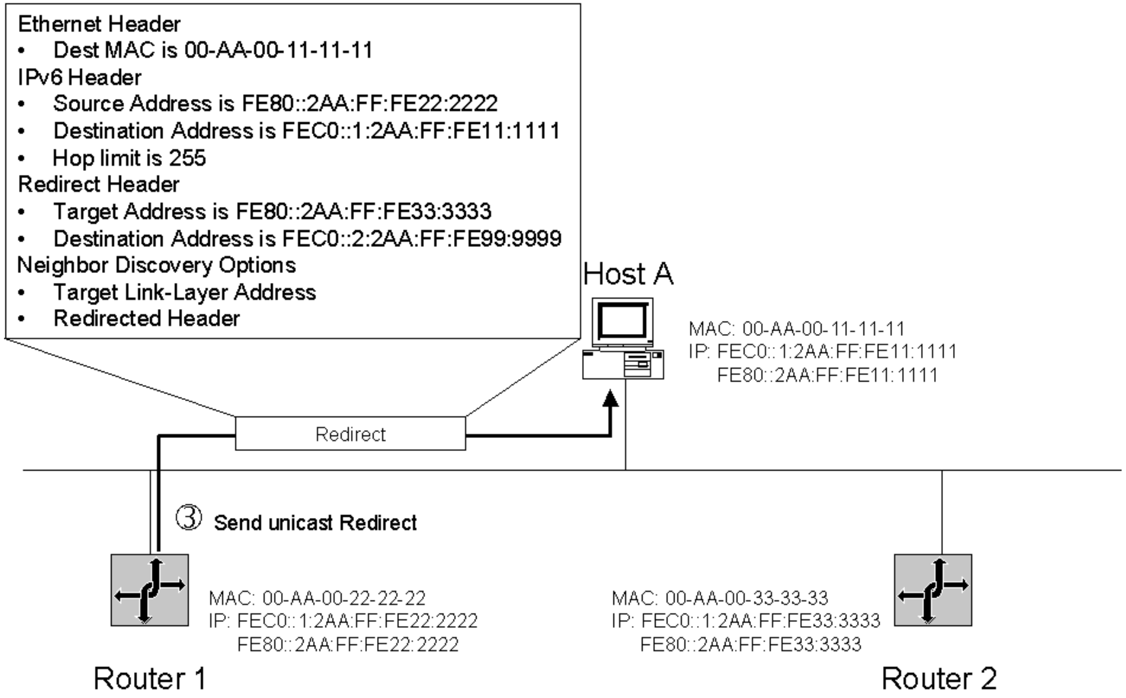
IPv6 Header

- Source Address is FEC0::1:2AA:FF:FE11:1111
- Destination Address is FEC0::2:2AA:FF:FE99:9999

- Router 1 receives the packet from Host A and notes that Host A is a neighbor.
- It also notes that Host A and the next-hop address for the destination are on the same link. Based on the contents of its local routing table, Router 1 forwards the unicast packet received from Host A to Router 2

Router Redirect Process - Example

The Redirect message sent by the router



Ethernet Header
- Dest MAC is 00-AA-00-11-11-11

IPv6 Header
- Source Address is FE80::2AA:FF:FE22:2222
- Destination Address is FEC0::1:2AA:FF:FE11:1111
- Hop limit is 255

Redirect Header
- Target Address is FE80::2AA:FF:FE33:3333
- Destination Address is FEC0::2:2AA:FF:FE99:9999

Neighbor Discovery Options
- Target Link-Layer Address
- Redirected Header

- To inform Host A that subsequent packets to the destination of FEC0::2:2AA:EE:FE99:9999 should be sent to Router 2, Router 1 sends a Redirect message to Host A

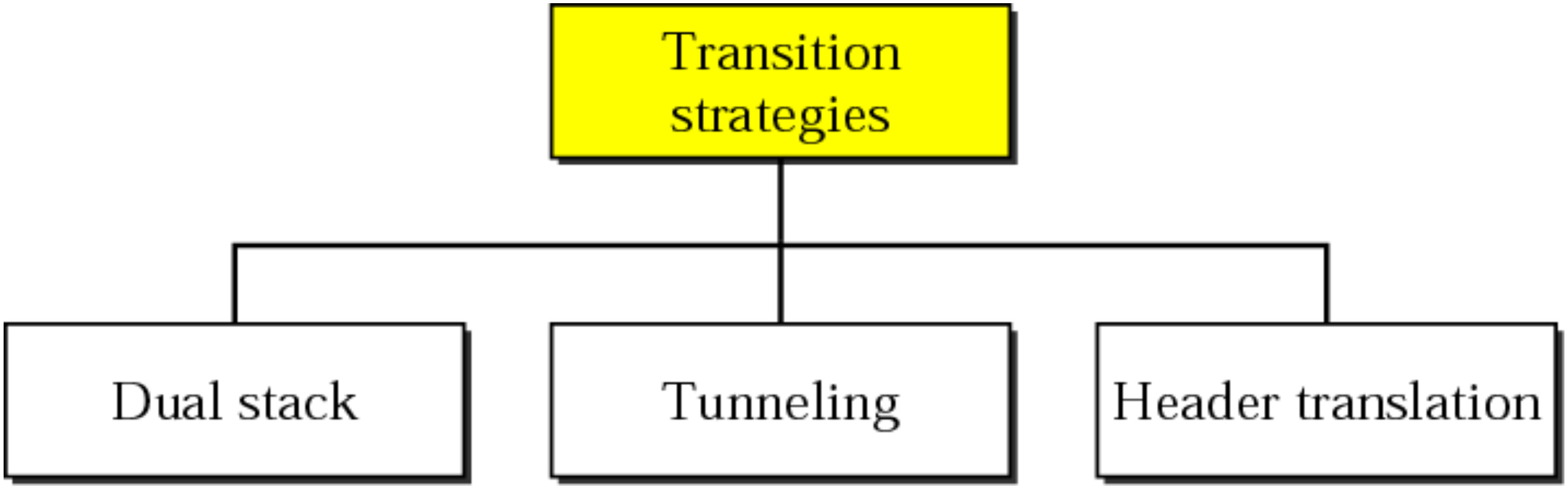
Transformation From IPv4 to IPv6

Three strategies have been devised by the IETF to provide for a smooth transition from IPv4 to IPv6.

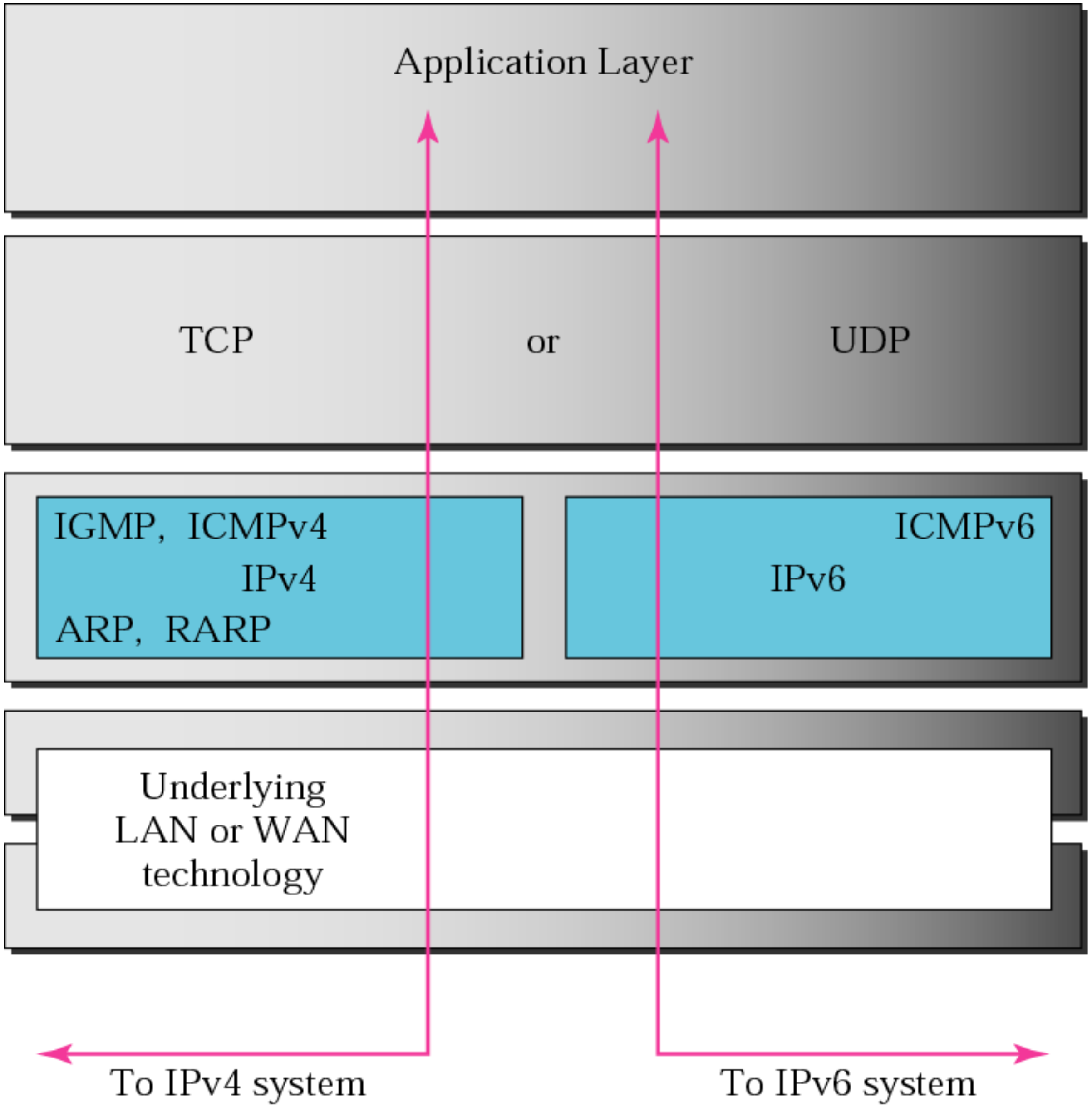
The topics discussed in this section include:

Dual Stack
Tunneling
Header Translation

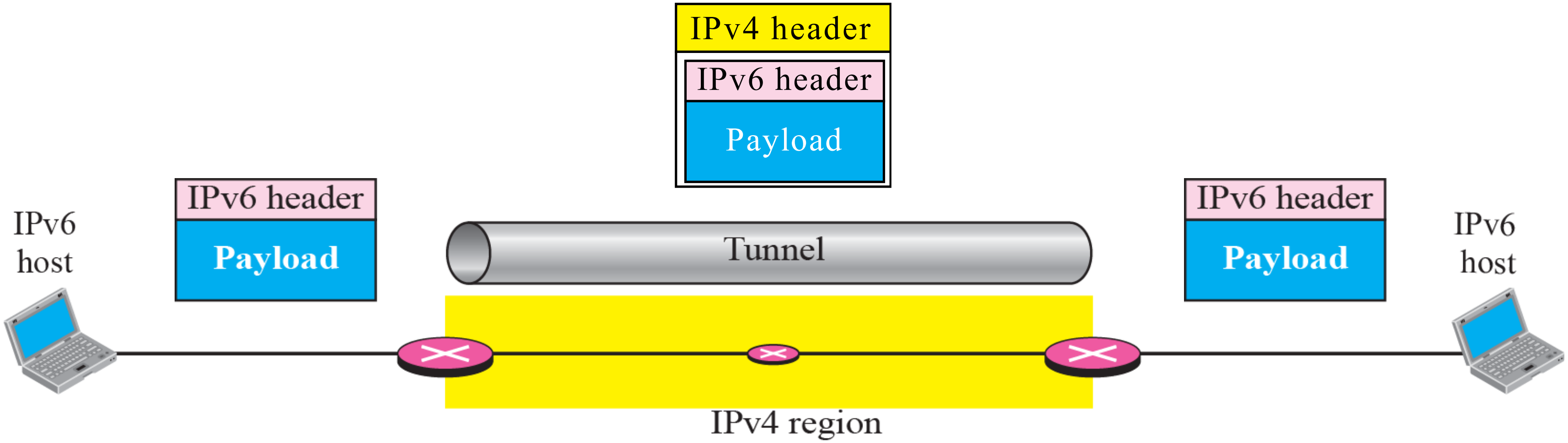
Three transition strategies



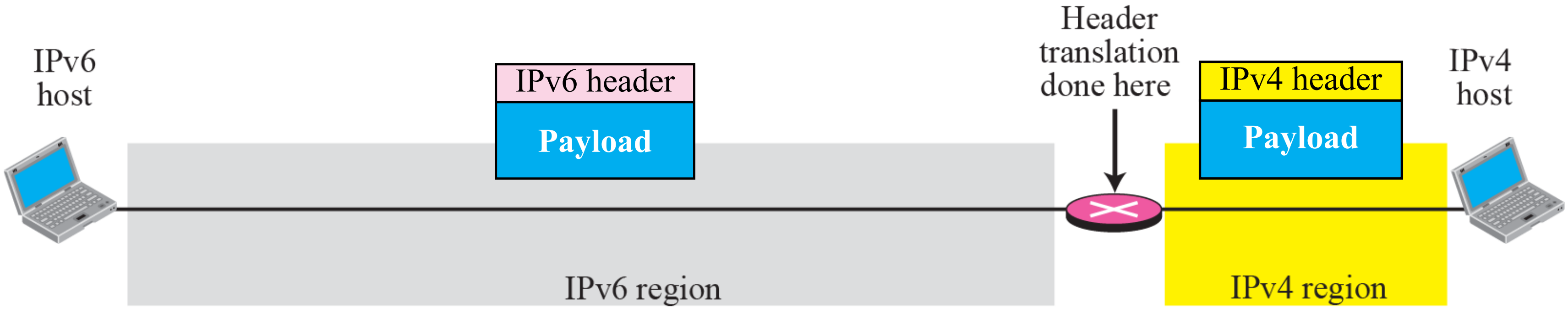
Dual stack


Application Layer
TCP
or
UDP
IGMP, ICMPv4
IPv4
ARP, RARP
ICMPv6
IPv6
Underlying
LAN or WAN
technology
To IPv4 system
To IPv6 system

Tunneling strategy



Header translation strategy

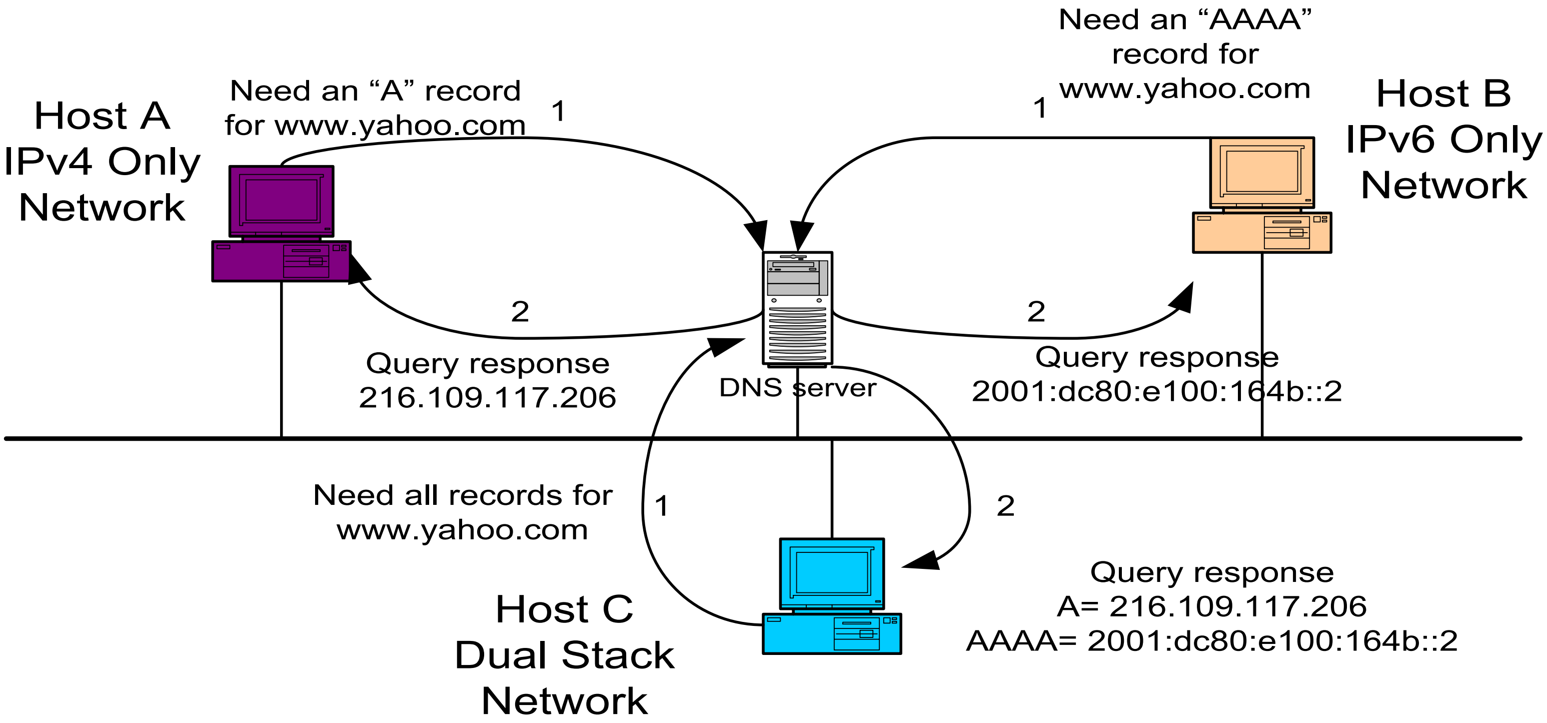


Naming Services

- DNS must be included in transition strategy
- Resolving Names:
  - IPv4 specifies “A” records
  - IPv6 specifies “AAAA” records
- Applications should be aware of both records
- Will require development update and thorough testing
- Tools like “Scrubber” by Sun make it easy

Naming Services

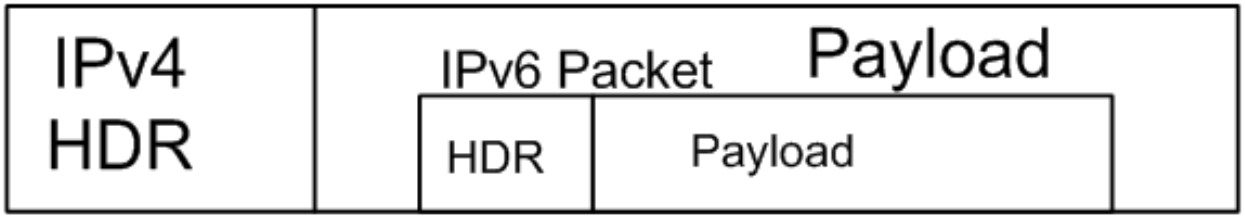
Querying DNS server



Manually Configured Tunnels

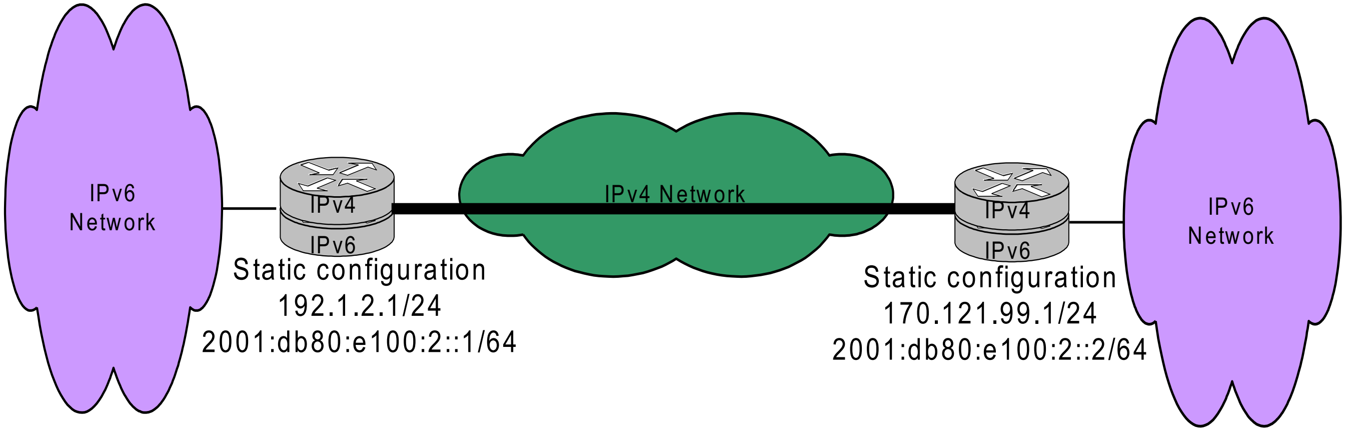
- Manually configured tunnels are logical tunnels formed when one protocol version packet is encapsulated in the payload of another version packet
- e.g. IPv4 encapsulated in IPv6 or IPv6 encapsulated in IPv4

IPv4 Packet with tunneling



Configured Tunnel-building

- Configured tunnels require static IPv4 addresses
- Configured tunnels are generally setup and maintained by a network administrator
- Configured tunnels are a proven IPv6 deployment technique and provide stable links



Static configuration
192.1.2.1/24
2001:db80:e100:2::1/64

Static configuration
170.121.99.1/24
2001:db80:e100:2::2/64

Potential Tunnel Issues

- MTU fragmentation
- ICMPv4 error handling
- Filtering protocol 41
- NAT (Network Address Translation)

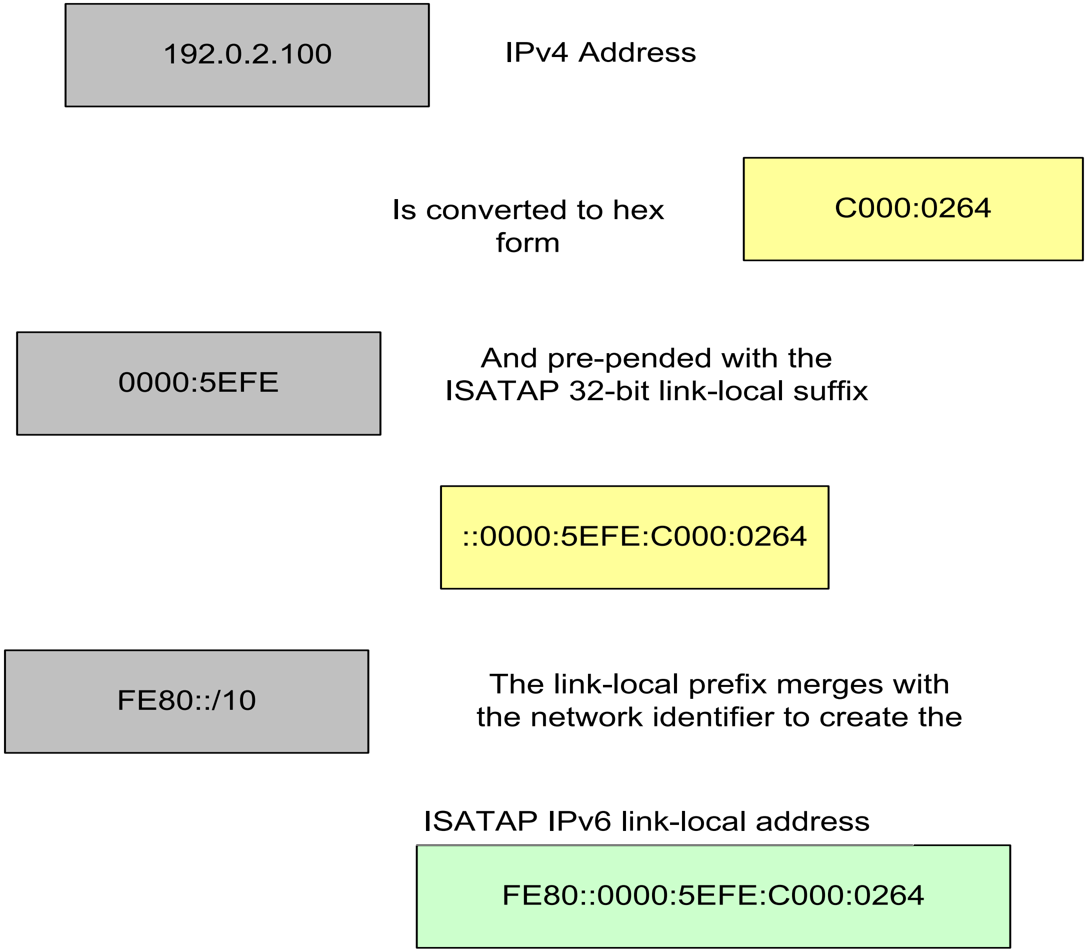
ISATAP

- **ISATAP** (Intra-Site Automatic Tunneling Addressing Protocol) an automatic tunneling mechanism used inside an organization that has an IPv4-dominant backbone, but has selected users that need IPv6 capability

ISATAP Functions

- ISATAP connects dual-stack nodes, isolated within an IPv4-only network
    - To exchange IPv6 traffic with each other (host ISATAP)
    - To exchange traffic with the global IPv6 Internet
- ISATAP is a mechanism with minimal configuration required
- ISATAP is ideal when there are relatively few, relatively scattered individual nodes that need service

Link-Local ISATAP



192.0.2.100

IPv4 Address

Is converted to hex form

C000:0264

0000:5EFE

And pre-pended with the ISATAP 32-bit link-local suffix

::0000:5EFE:C000:0264

FE80::/10

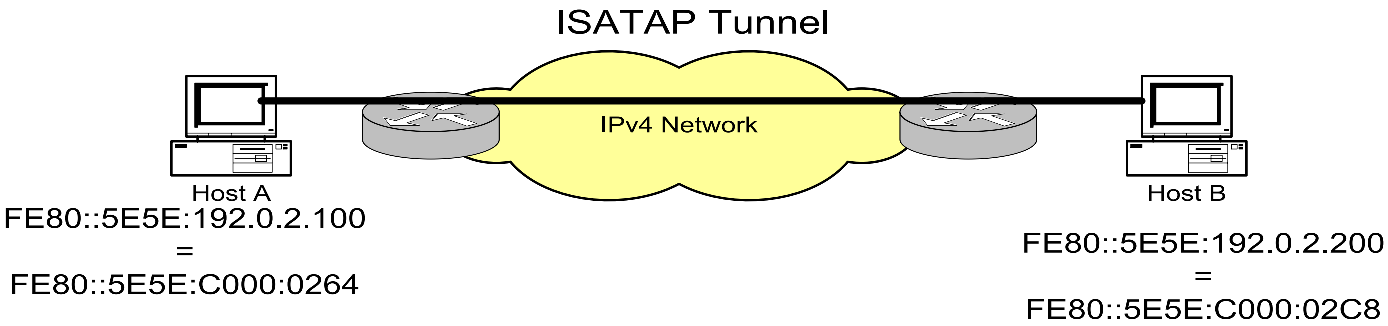
The link-local prefix merges with the network identifier to create the

ISATAP IPv6 link-local address

FE80::0000:5EFE:C000:0264

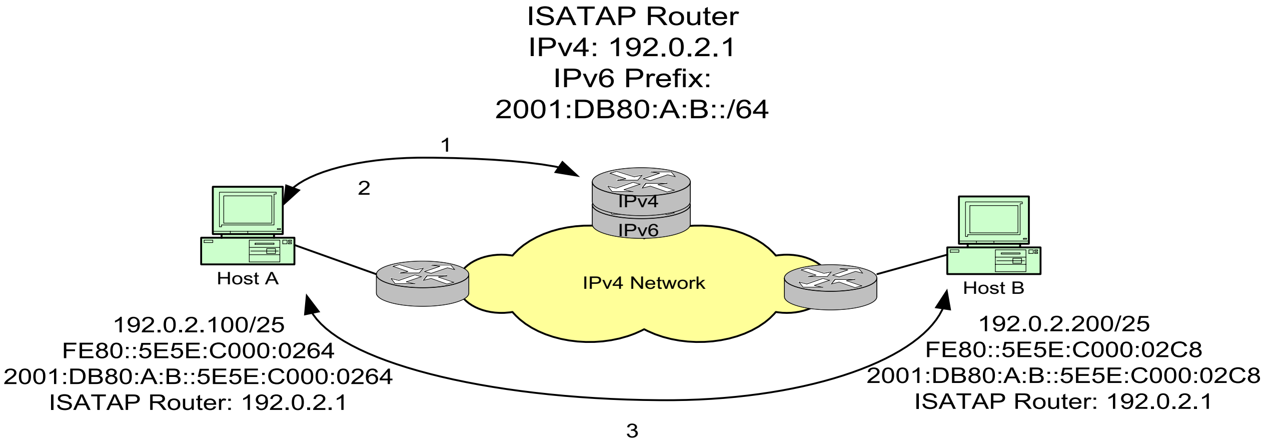
Link-local ISATAP example

- Two ISATAP hosts exchanging packets using link-local addresses
- Only route on ISATAP hosts is “send all IPv6 traffic via ISATAP pseudo-IF”
- Hosts are many IPv4 hops away which appear link-local to IPv6



Globally-routable ISATAP

- ISATAP more flexible when using an ISATAP router
- ISATAP hosts are configured with ISATAP router IPv4 address
- Hosts sends router solicitation, inside tunnel, and ISATAP router responds



ISATAP Summary

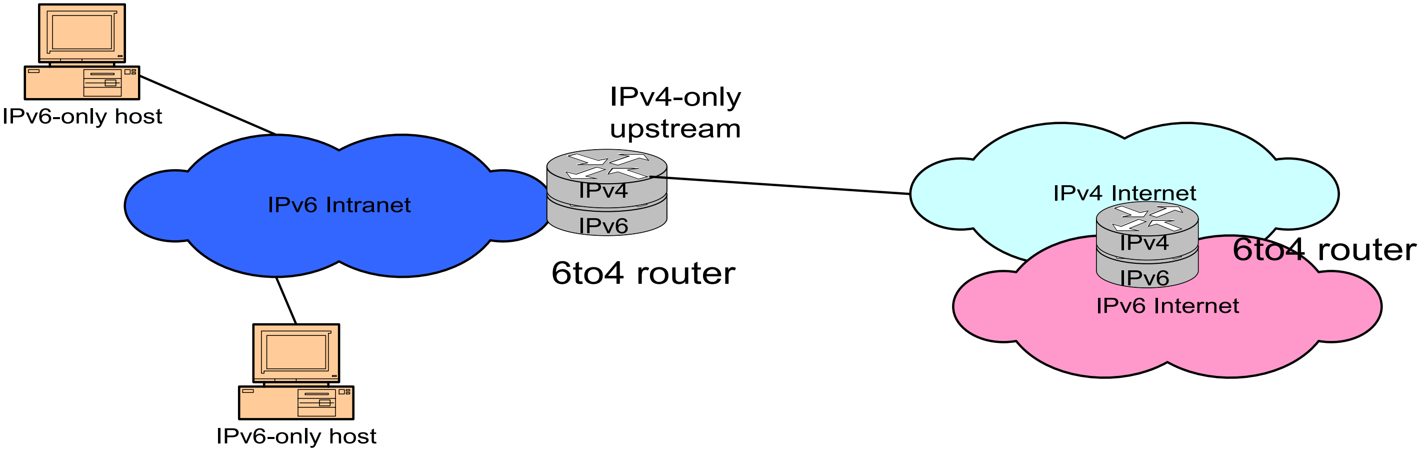
- ISATAP scales better than manually configured tunnels inside the enterprise
- Decapsulate-from-anywhere issues (like 6to4) mitigated by internal deployment
- No authentication provided – any dual stack node that knows ISATAP router address can obtain services
- May need to look at other alternatives if security is required

IPv6 6to4 Transition Mechanism

- 6to4 is an automatic tunneling mechanism that provides v6 capability to a dual-stack node or v6-capable site that has only IPv4 connectivity to the site

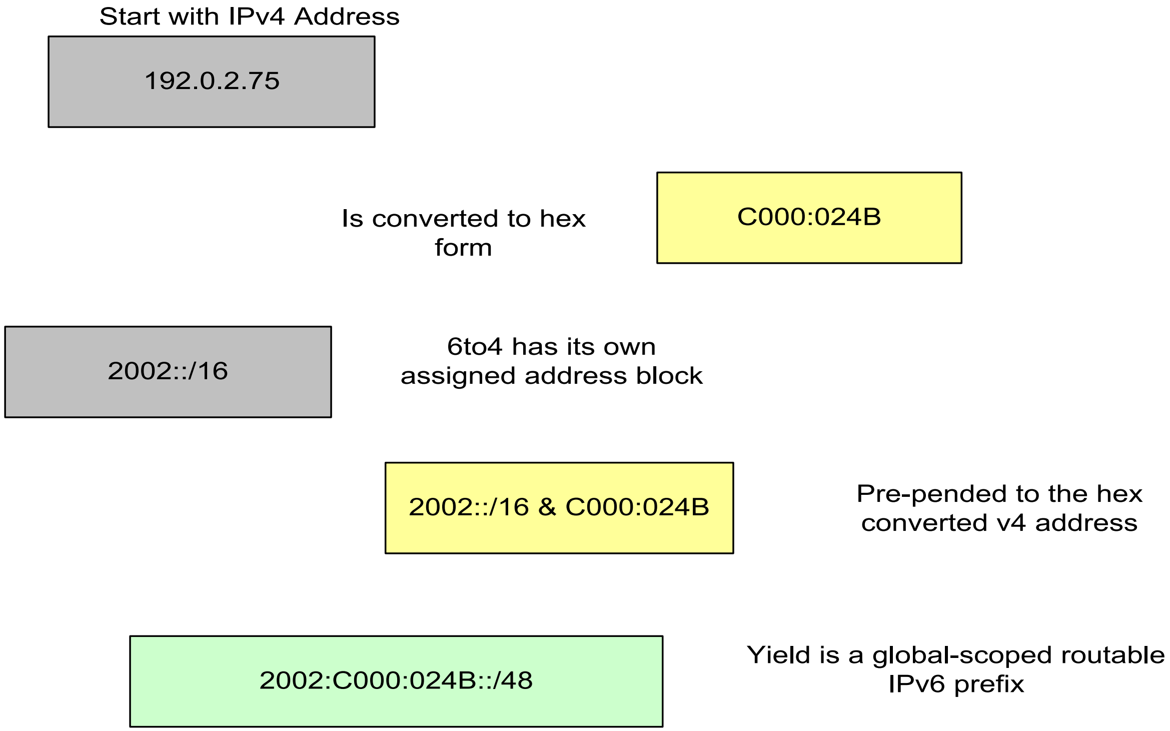
6to4 Basics

- 6to4 is an automatic tunnel mechanism
- Provides v6 upstream for v6-capable site over v4-only Internet connection
- Uses embedded addressing (v4addr embedded in v6addr) as do other automatic mechanisms



6to4 Address Construction

- 6to4 setups a valid, unique /48 IPv6 prefix from the outside IPv4 address of the site router



Start with IPv4 Address

192.0.2.75

Is converted to hex form

C000:024B

2002::/16

6to4 has its own assigned address block

2002::/16 & C000:024B

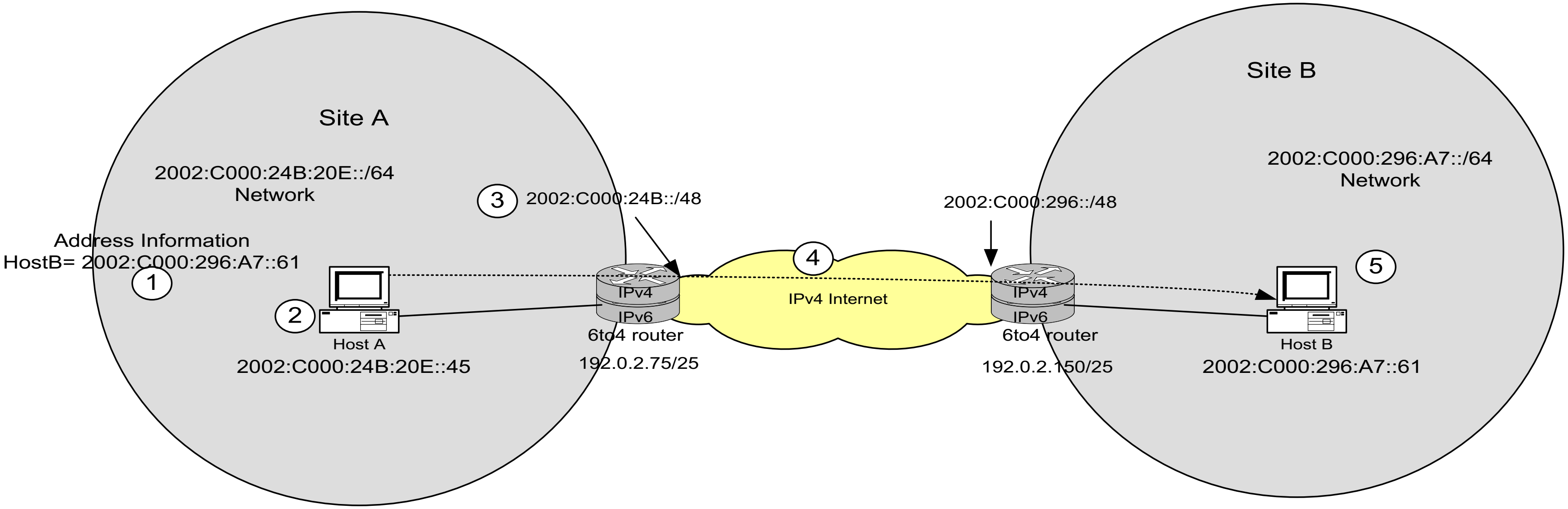
Pre-pended to the hex converted v4 address

2002:C000:024B::/48

Yield is a global-scoped routable IPv6 prefix

6to4 Site-to-Site Example

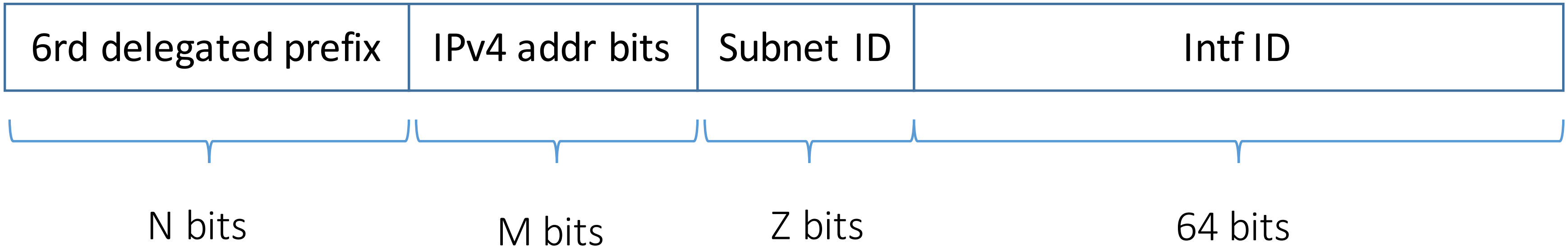
- 6to4 edge devices are called “6to4 site routers”
- IPv4-only between sites, full IPv6 within sites
- Host A packet tunneled through IPv4 network to destination 6to4 site



6RD Tunnel

- IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) – RFC 5969
- Utilize an SP's own IPv6 address prefix rather than a well-known prefix (2002::/16)

6RD Address Format



6rd delegated prefix

IPv4 addr bits

Subnet ID

Intf ID

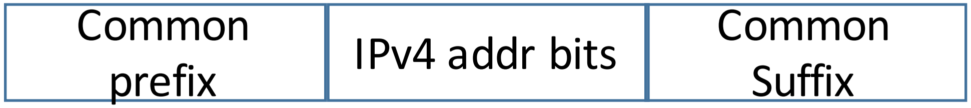
N bits

M bits

Z bits

64 bits

N + M + Z = 64 bits



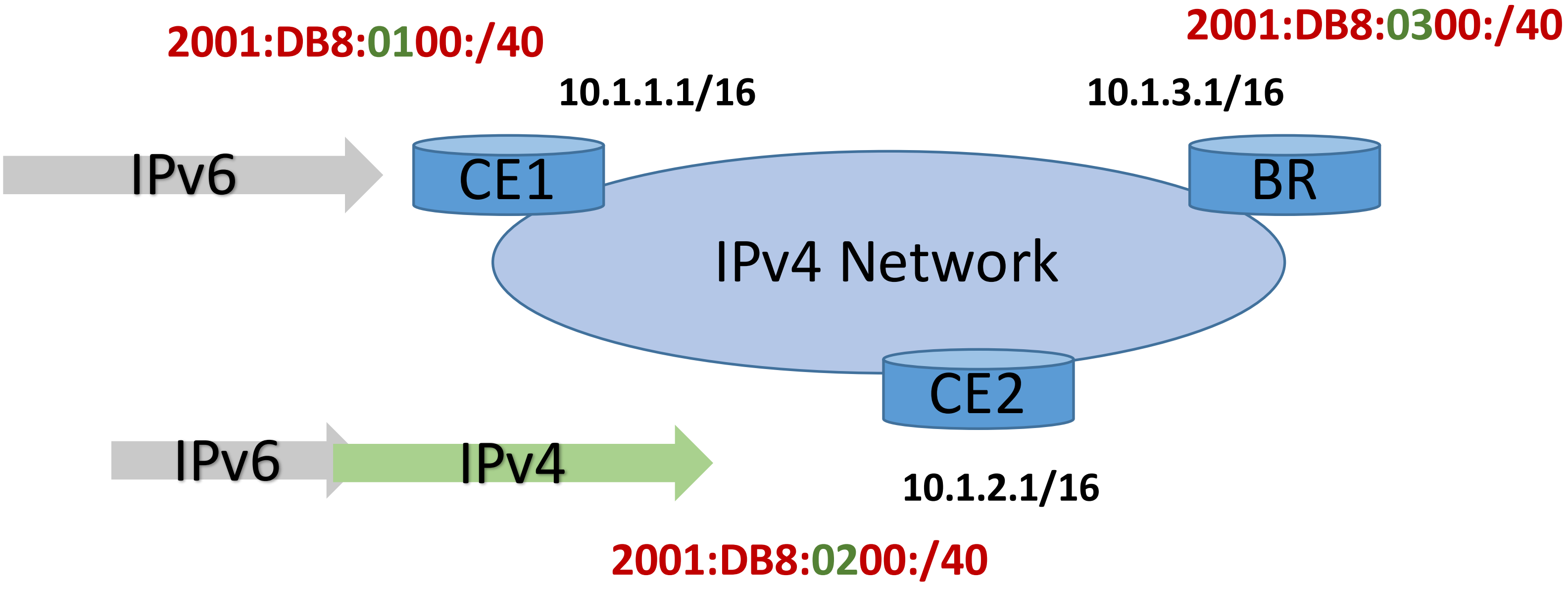
Common prefix

IPv4 addr bits

Common Suffix

Parameter	Value
6rd Prefix/length	2001:DB8::/32
IPv4 Common prefix/length	10.1.0.0/16
IPv4 Common suffix/length	0.0.0.1/8

6rd Prefix 2001:DB8::/32
Ipv4 common prefix: 10.1.0.0/16
Ipv4 common suffix: 0.0.0.1/8



2001:DB8:0100:/40

10.1.1.1/16

2001:DB8:0300:/40

10.1.3.1/16

IPv6

CE1

BR

IPv4 Network

CE2

IPv6

IPv4

10.1.2.1/16

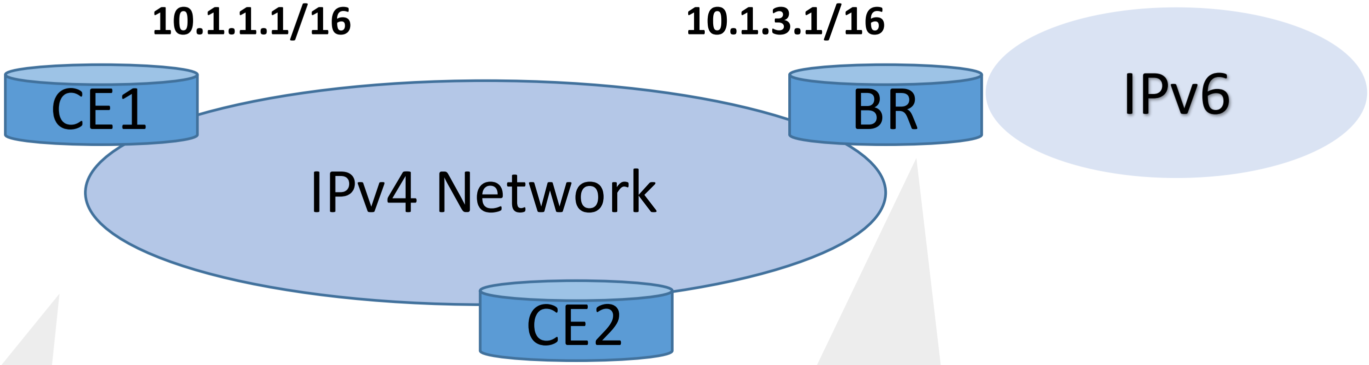
2001:DB8:0200:/40

IPv6: 2001:DB8:0100::C15C:0 → 2001:DB8:0200::C26B:0
IPv4: 10.1.1.1 → 10.2.1.1

6rd Prefix 2001:DB8::/32

Ipv4 common prefix: 10.1.0.0/16

Ipv4 common suffix: 0.0.0.1/8



2001:DB8:0100:/40

10.1.1.1/16

2001:DB8:0300:/40

10.1.3.1/16

10.1.2.1/16

2001:DB8:0200:/40

2001:DB8::/32	Tunnel0
::/0	2001:BABE::1

2001:DB8::/32	Tunnel0
::/0	Tunnel0 or
::/0	2001:DB8:0300::D55C:3