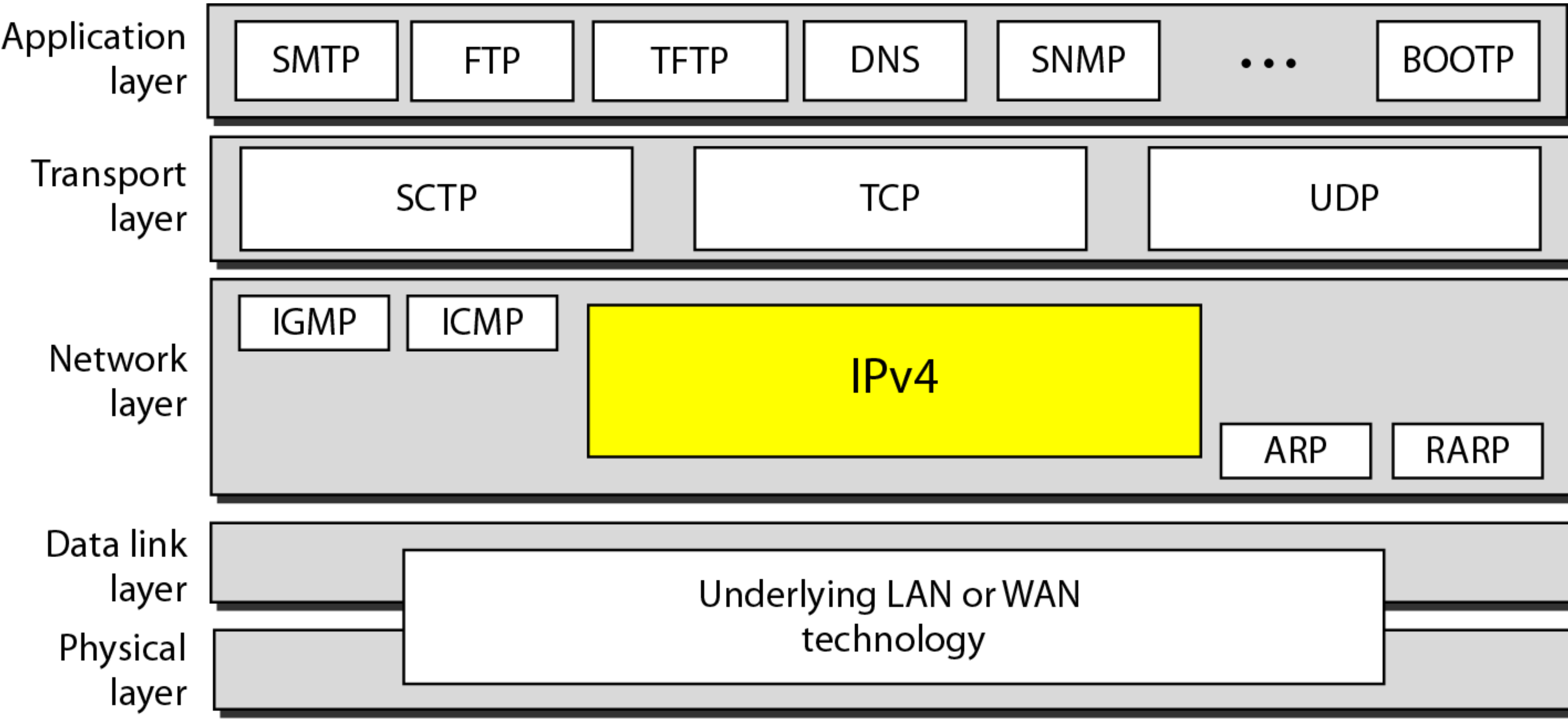
Lecture 10

Network Layer: Internet Protocol

IPv4

The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols.

Position of IPv4 in TCP/IP protocol suite



SMTP: simple mail transfer protocol

FTP: file transfer protocol

TFTP: trivial file transfer protocol

DNS: domain name system

SNMP: simple network management protocol

BOOTP: bootstrap protocol

SCTP: stream control transmission protocol

TCP: transmission control protocol

UDP: user datagram protocol

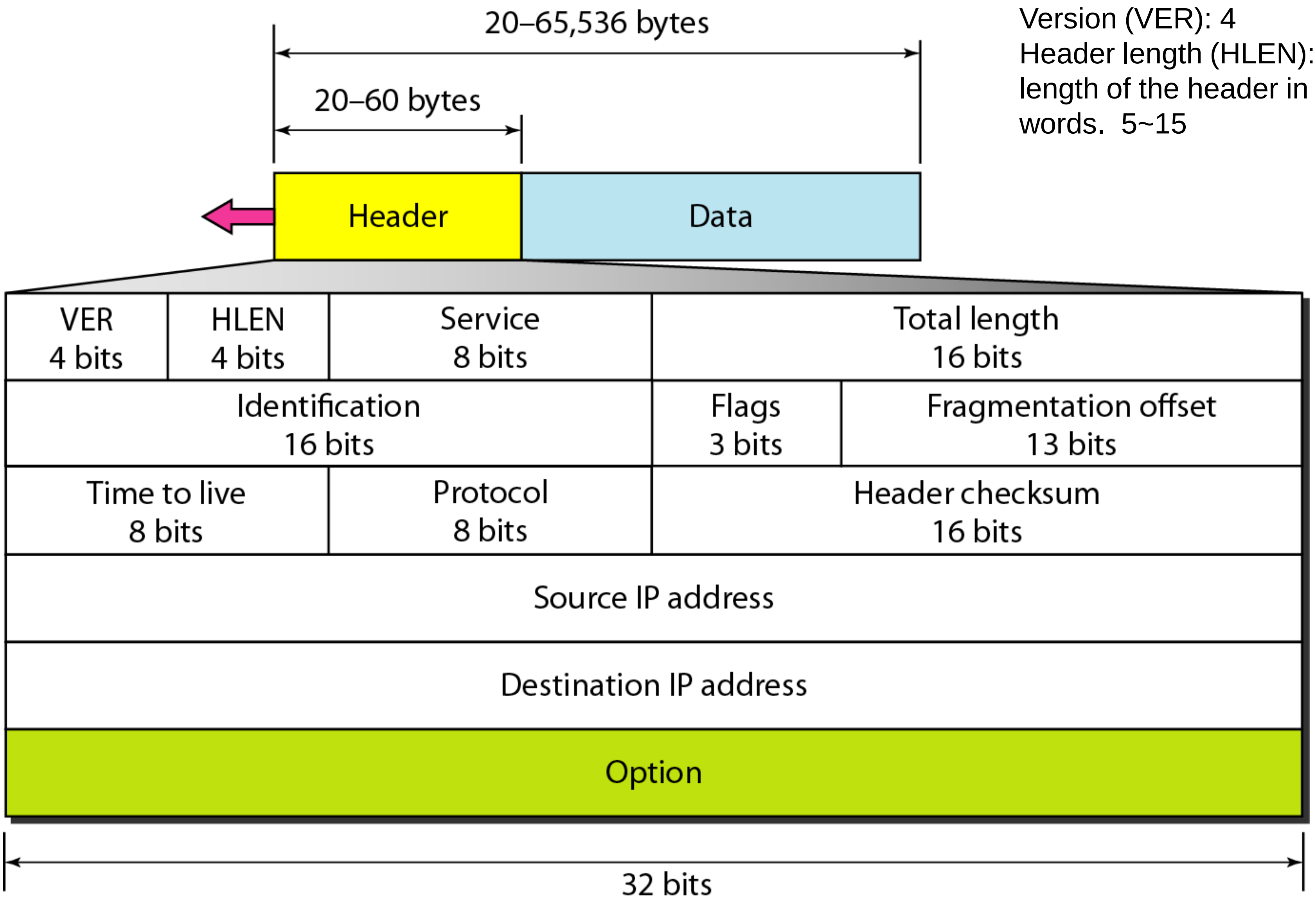
IGMP: Internet group management protocol

ICMP: Internet control message protocol

ARP: address resolution protocol

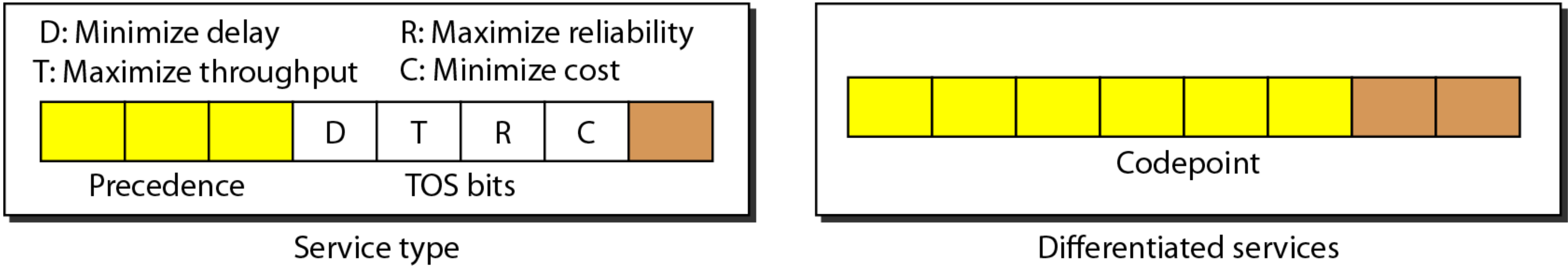
RARP: reverse address resolution protocol

IPv4 datagram format



Version (VER): 4
Header length (HLEN): total length of the header in 4-byte words. 5~15

Service type or differentiated services



Note

The precedence subfield was part of version 4, but never used.

Precedence: priority of the datagram in issues such as congestion (e.g., lowest precedence datagrams may be discarded first).

Types of service

TOS Bits DTRC	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

At most one bit is one

Default types of service

Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

interior gateway protocol

Simple network management protocol

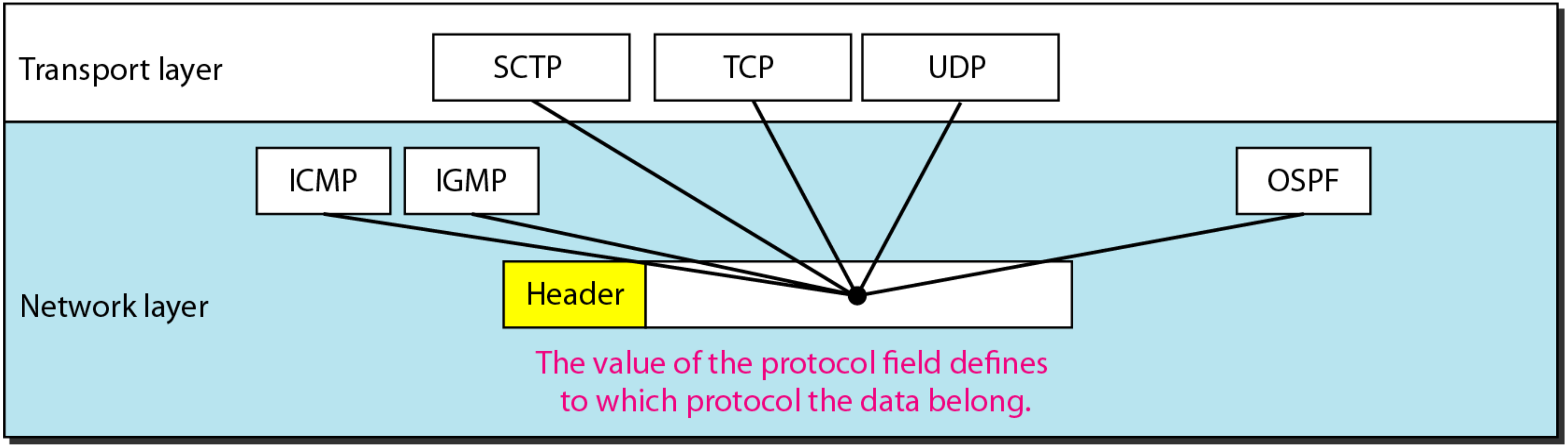
Note

The total length field defines the total length (in bytes) of the datagram including the header.

Identification, flags, and fragment offset are used in fragmentation

Time to live: A datagram has a limited lifetime in its travel through the Internet. It is used mostly to control the maximal number of hops (routers) visited by the datagram. Each router that processes the datagram decrements this number by 1. If the value, after be decremented, is zero, the router discards the datagram.

Protocol field and encapsulated data



Protocol values

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Example 1

An IPv4 packet has arrived with the first 8 bits as shown:

01000010

The receiver discards the packet. Why?

Solution

There is an error in this packet. The 4 leftmost bits (0100) show the version, which is correct. The next 4 bits (0010) show an invalid header length ($2 \times 4 = 8$). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example 2

In an IPv4 packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

Solution

The HLEN value is 8, which means the total number of bytes in the header is 8 × 4, or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

Example 3

In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is 0x0028. How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5×4, or 20 bytes (no options). The total length is 40 bytes, which means the packet is carrying 20 bytes of data $(40 - 20)$.

Example 4

An IPv4 packet has arrived with the first few hexadecimal digits as shown.

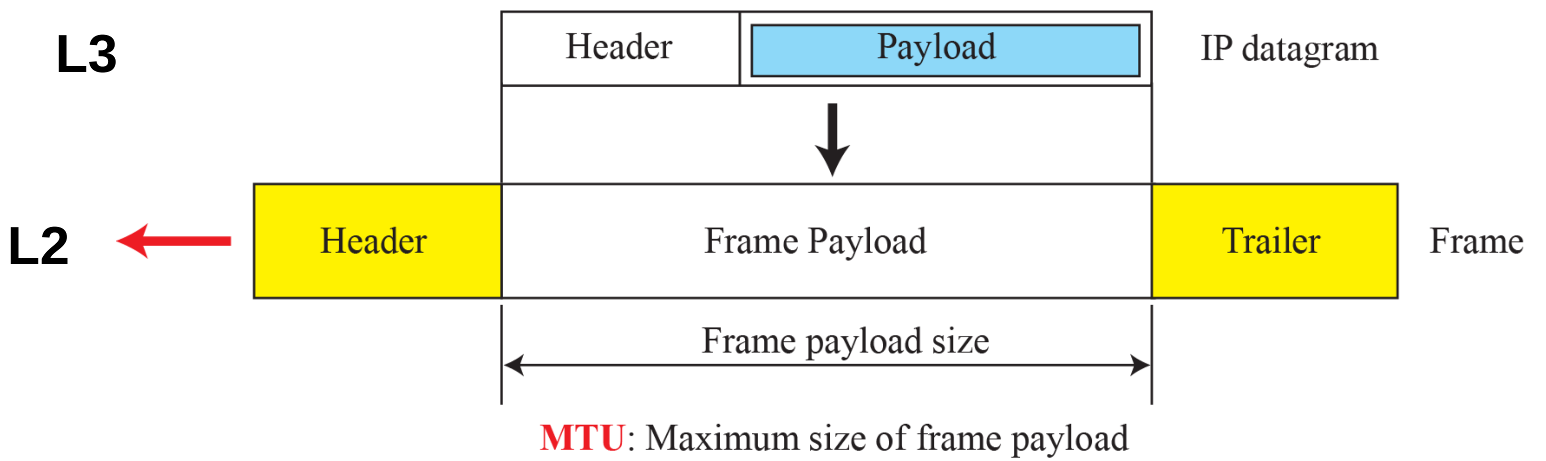
0x45000028000100000102 . . .

How many hops can this packet travel before being dropped? The data belong to what upper-layer protocol?

Solution

To find the time-to-live field, we skip 8 bytes. The time-to-live field is the ninth byte, which is 01. This means the packet can travel only one hop. The protocol field is the next byte (02), which means that the upper-layer protocol is IGMP.

Maximum transfer unit (MTU)



Each data link layer protocol has its own frame format in most protocols. One of the fields in the format defines the maximum size of the data field in an L2 frame.

MTUs for some networks

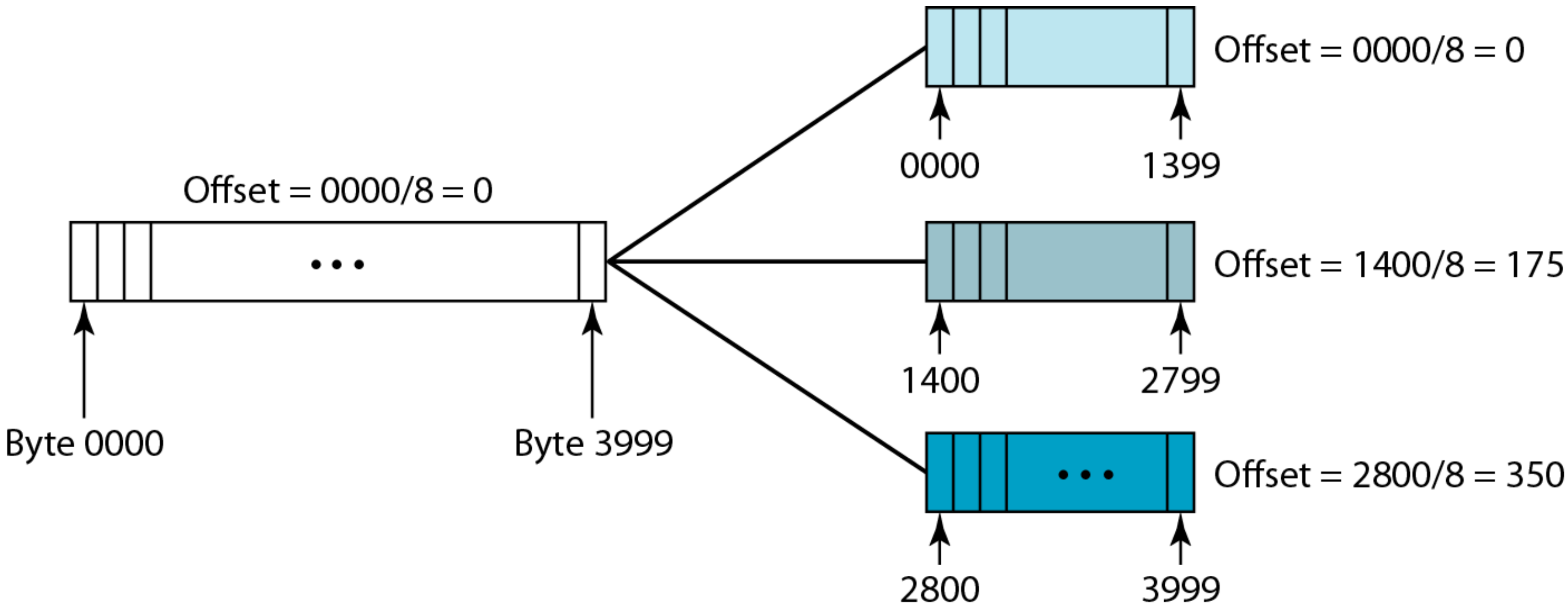
Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

Flags used in fragmentation

If D=1: the machine must not fragment the datagram. If it cannot pass the datagram through any physical network, its discards the datagram and sends an error message to the source host.

If D=0: the datagram can be fragmented if necessary.

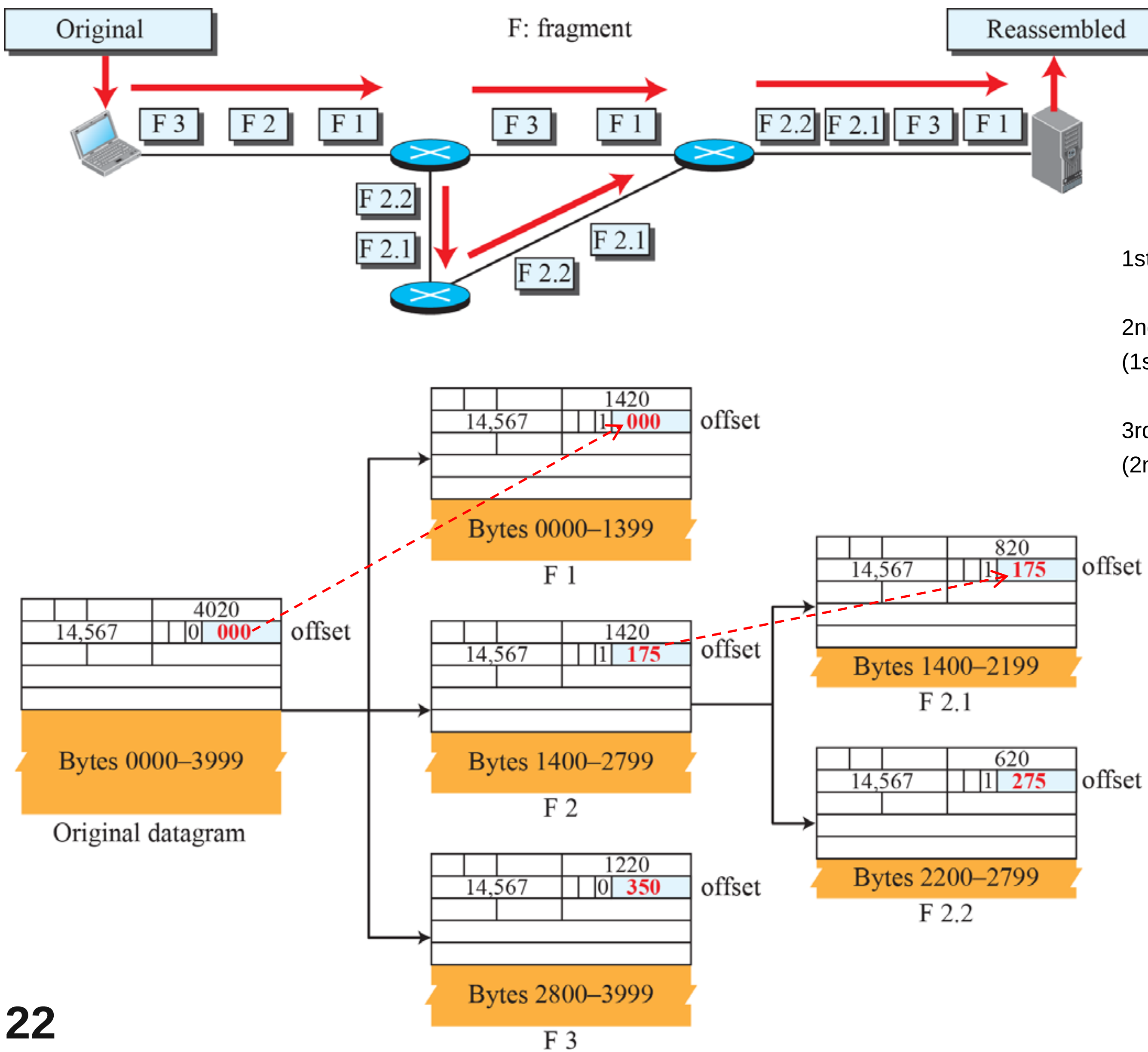
Fragmentation example



Offset is measured in the units of 8 bytes. Why?

The byte index may range 0 ~ (2^{16}-20). But we only have 13 bits in "offset" field.
The size of each fragment should be a multiple of 8.

Detailed fragmentation example



1st fragment offset = offset of initial packet.

2nd fragment offset = 1st fragment offset + (1st fragment size/8)

3rd fragment offset = 2nd fragment offset + (2nd fragment size/8)

The identification field identifies a datagram originating from the source host.
When fragmented, the value is copied to all fragments.
Further fragment may happen at a router. If performing fragmentation, from the offset, the router knows index of the first byte.

Example 5

A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 0, it means that there are no more fragments; the fragment is the last one. However, we cannot say if the original packet was fragmented or not. A non-fragmented packet is considered the last fragment.

Example 6

A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 1, it means that there is at least one more fragment. This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset).

Example 7

A packet has arrived with an M bit value of 1 and a fragmentation offset value of 0. Is this the first fragment, the last fragment, or a middle fragment?

Solution

Because the M bit is 1, it is either the first fragment or a middle one. Because the offset value is 0, it is the first fragment.

Example 8

A packet has arrived in which the offset value is 100. What is the index of the first byte? Do we know the index of the last byte?

Solution

To find the index of the first byte, we multiply the offset value by 8. This means that the first byte index is 800. We cannot determine the index of the last byte unless we know the length.

Example 9

A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the indices of the first byte and the last byte?

Solution

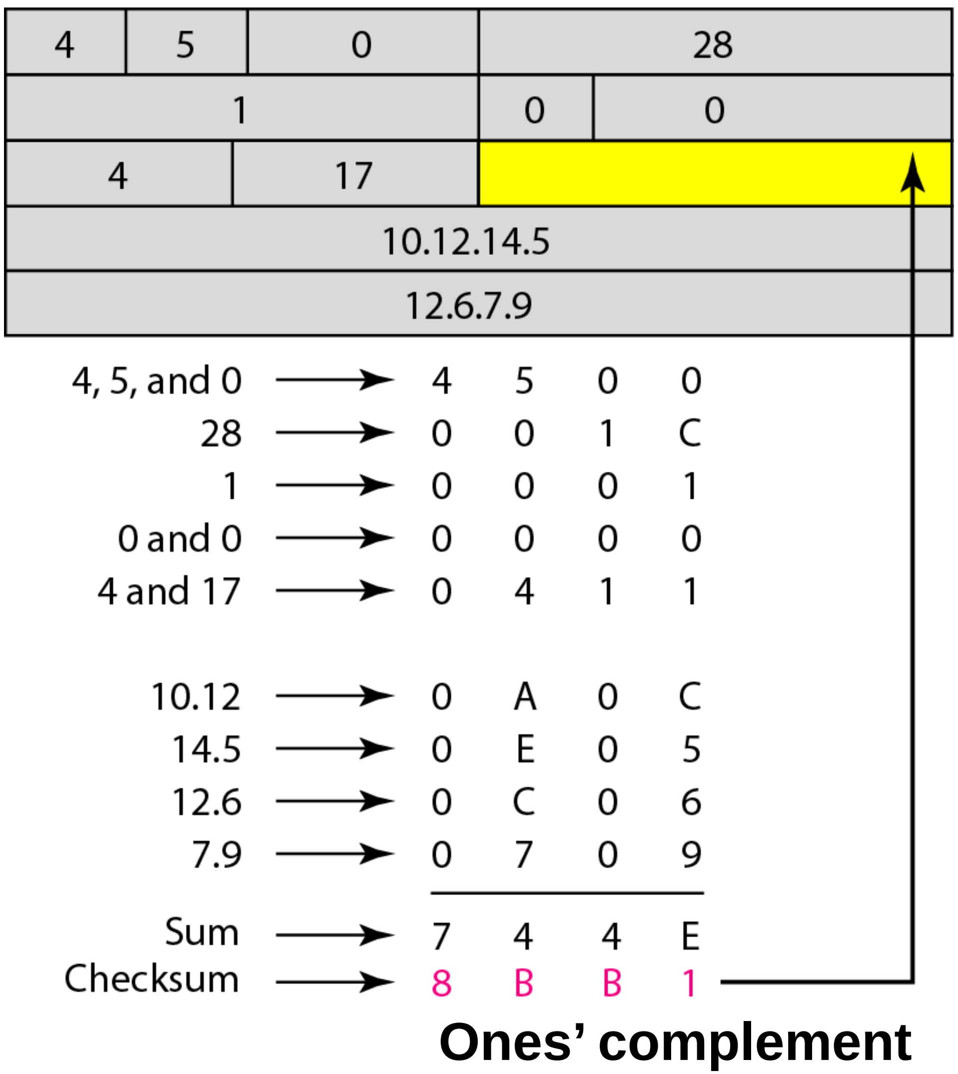
The first byte's index is $100 \times 8 = 800$. The total length is 100 bytes, and the header length is 20 bytes (5×4), which means that there are 80 bytes in this datagram. If the first byte's index is 800, the last byte's index must be 879.

Example 10

Figure 20.13 shows an example of a checksum calculation for an IPv4 header without options. The header is divided into 16-bit sections. All the sections are added and the sum is complemented. The result is inserted in the checksum field.

The checksum covers only the header, not the data. One reason is because all higher-level protocols that encapsulate data in the IPv4 datagram have a checksum field that covers the whole data.

Example of checksum calculation in IPv4



At the sender,
For all other fields except checksum: group every 16 bits together, and get 9 numbers.
Add the 9 numbers and get the sum.
Checksum is one's complement of the sum.

At a router:
Add the 10 numbers (including the checksum).
If the result is not all bit 1's, the router discards the packet.
If the result is all bit 1's, accept the packet. When the router forwards the packet, since some fields of the header are changed (such as Time to Live), it calculates a new checksum.

If header has options, they should also be included in the checksum calculation.

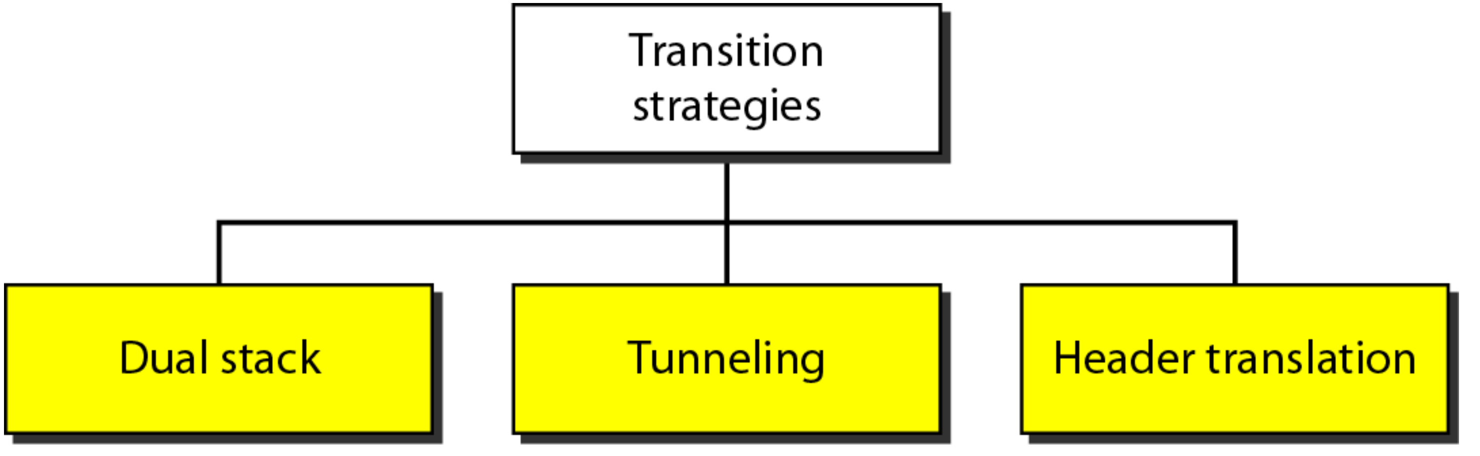
Checksum in IPv4 is not for data.
1.We have checksum in Layer 4, to protect the IPv4 data portion

2.Since the IPv4 checksum should be recalculated at each router, protecting data portion will increase processing time.

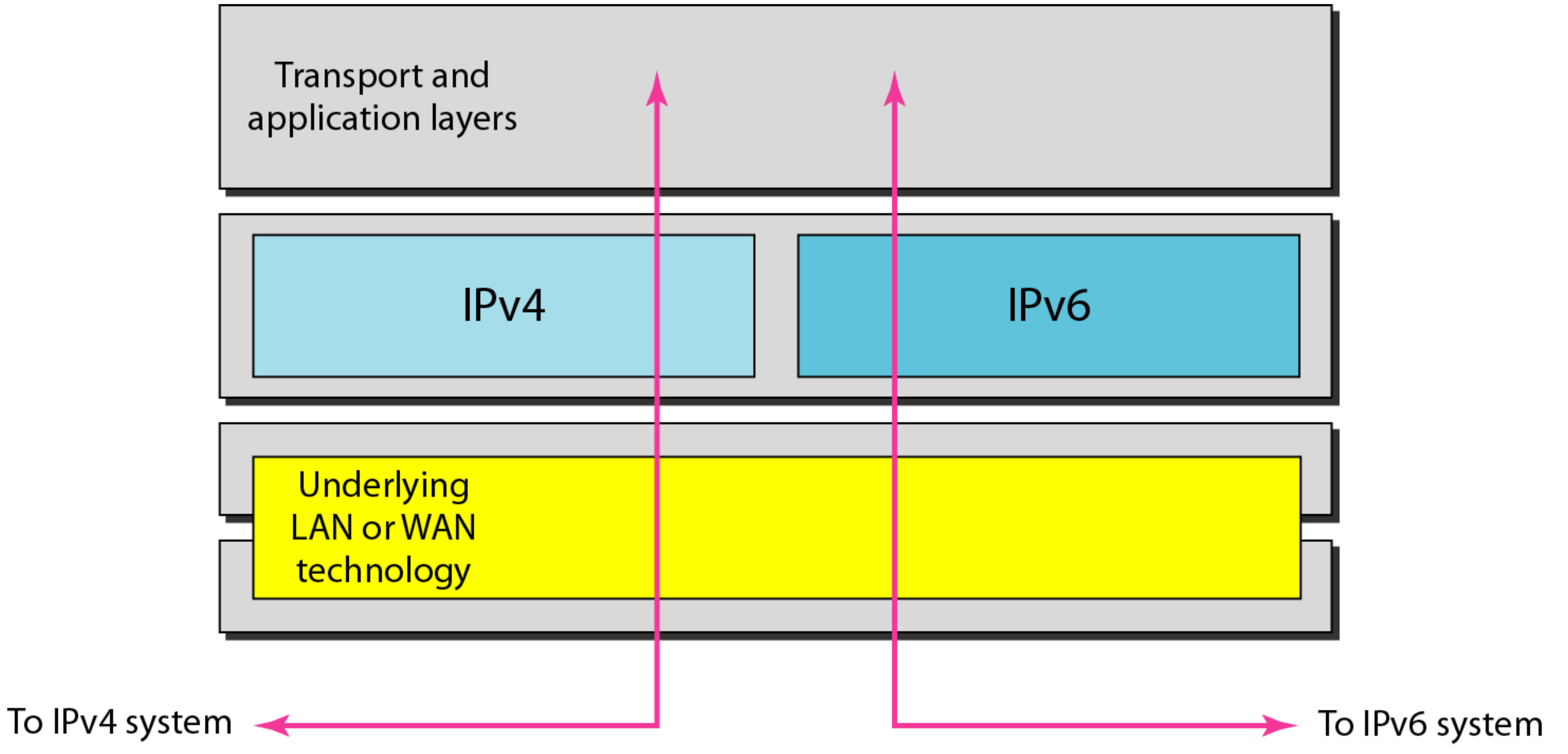
TRANSITION FROM IPv4 TO IPv6

Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly. It takes a considerable amount of time before every system in the Internet can move from IPv4 to IPv6. The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.

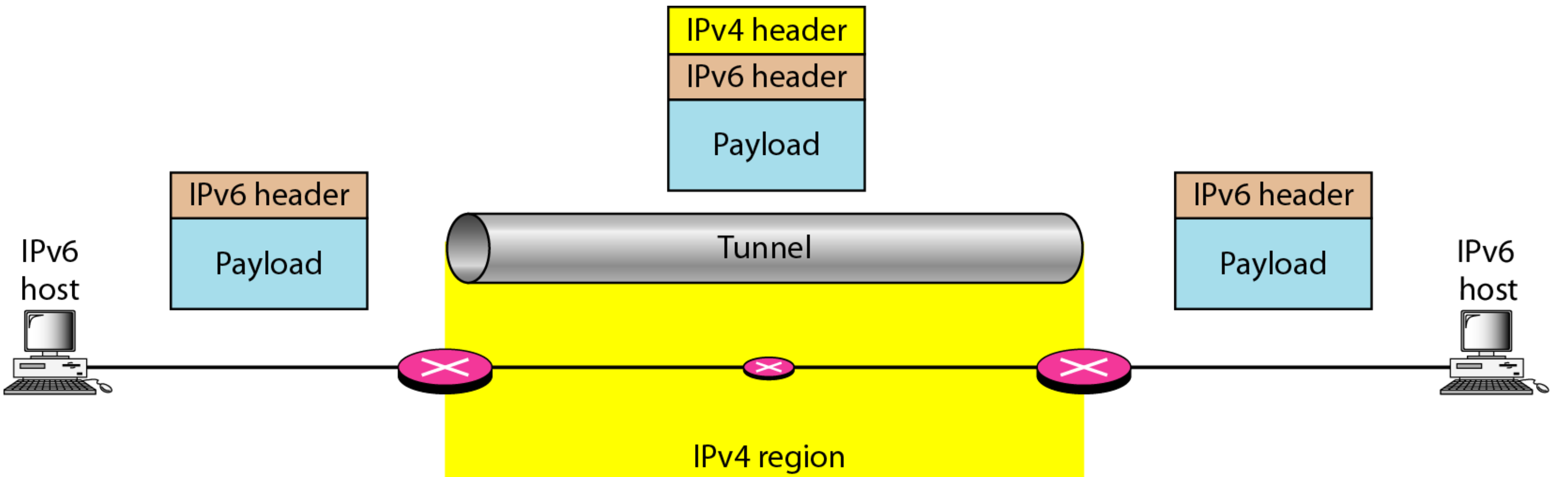
Three transition strategies



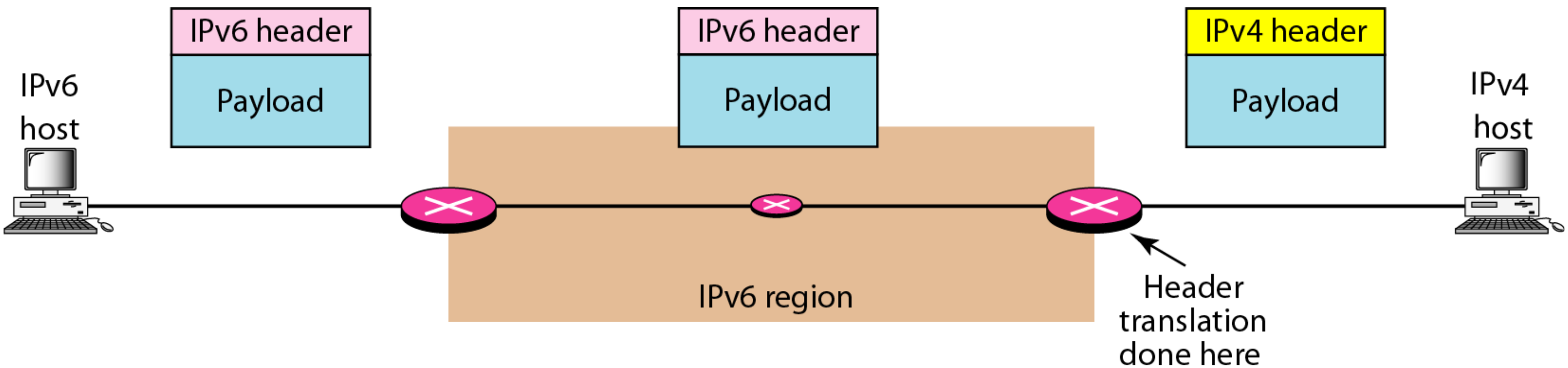
Dual stack


Transport and application layers
IPv4
IPv6
Underlying LAN or WAN technology
To IPv4 system
To IPv6 system

Tunneling strategy



Header translation strategy



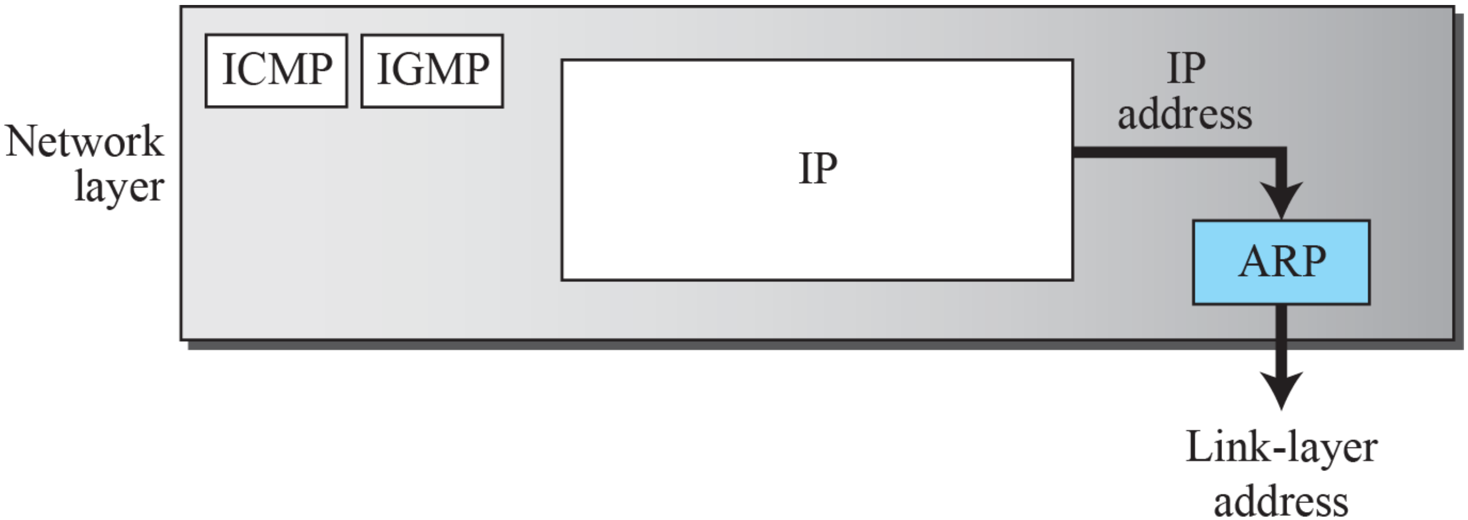
Header translation

Header Translation Procedure
1. The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.
2. The value of the IPv6 priority field is discarded.
3. The type of service field in IPv4 is set to zero.
4. The checksum for IPv4 is calculated and inserted in the corresponding field.
5. The IPv6 flow label is ignored.
6. Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped.
7. The length of IPv4 header is calculated and inserted into the corresponding field.
8. The total length of the IPv4 packet is calculated and inserted in the corresponding field.

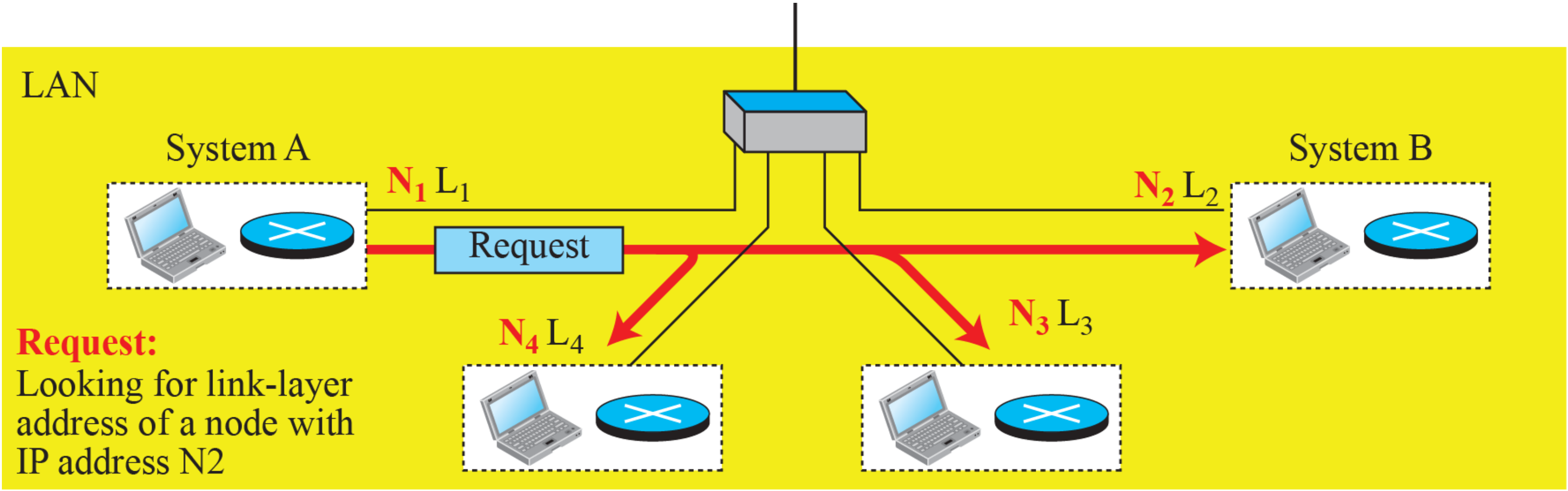
ARP

Anytime a node has an IP datagram to send to another node in a link, it has the IP address of the receiving node. However, the IP address of the next node is not helpful in moving a frame through a link; we need the link-layer address of the next node. This is the time when the Address Resolution Protocol (ARP) becomes helpful.

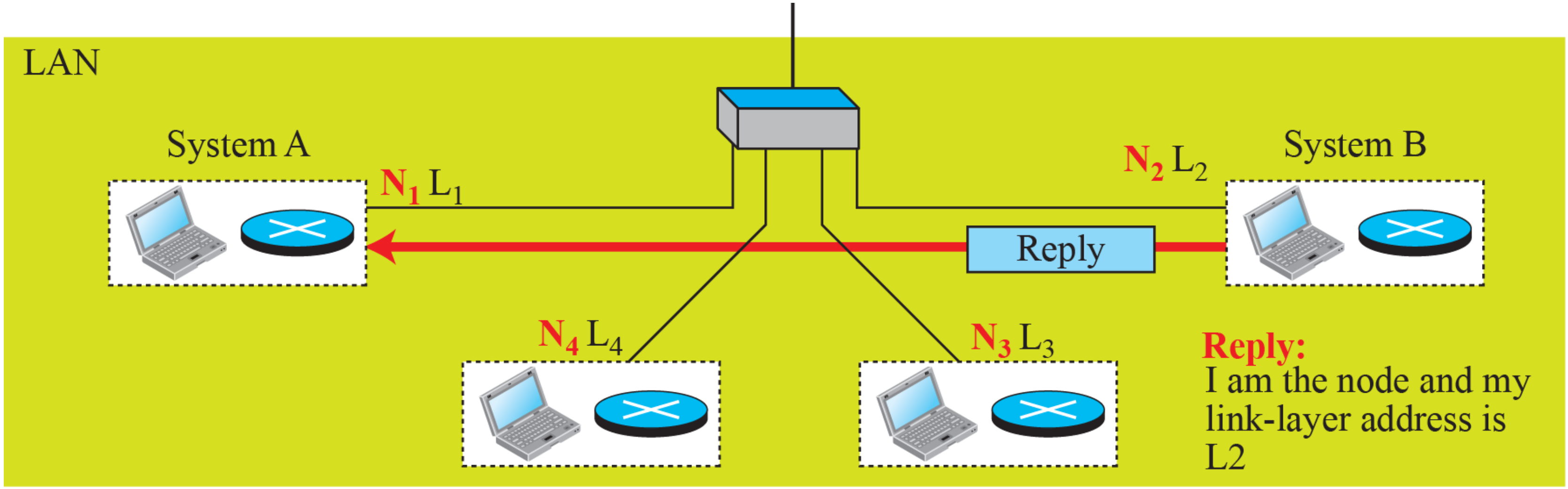
Position of ARP in TCP/IP protocol suite



ARP operation



a. ARP request is broadcast



b. ARP reply is unicast

ARP packet

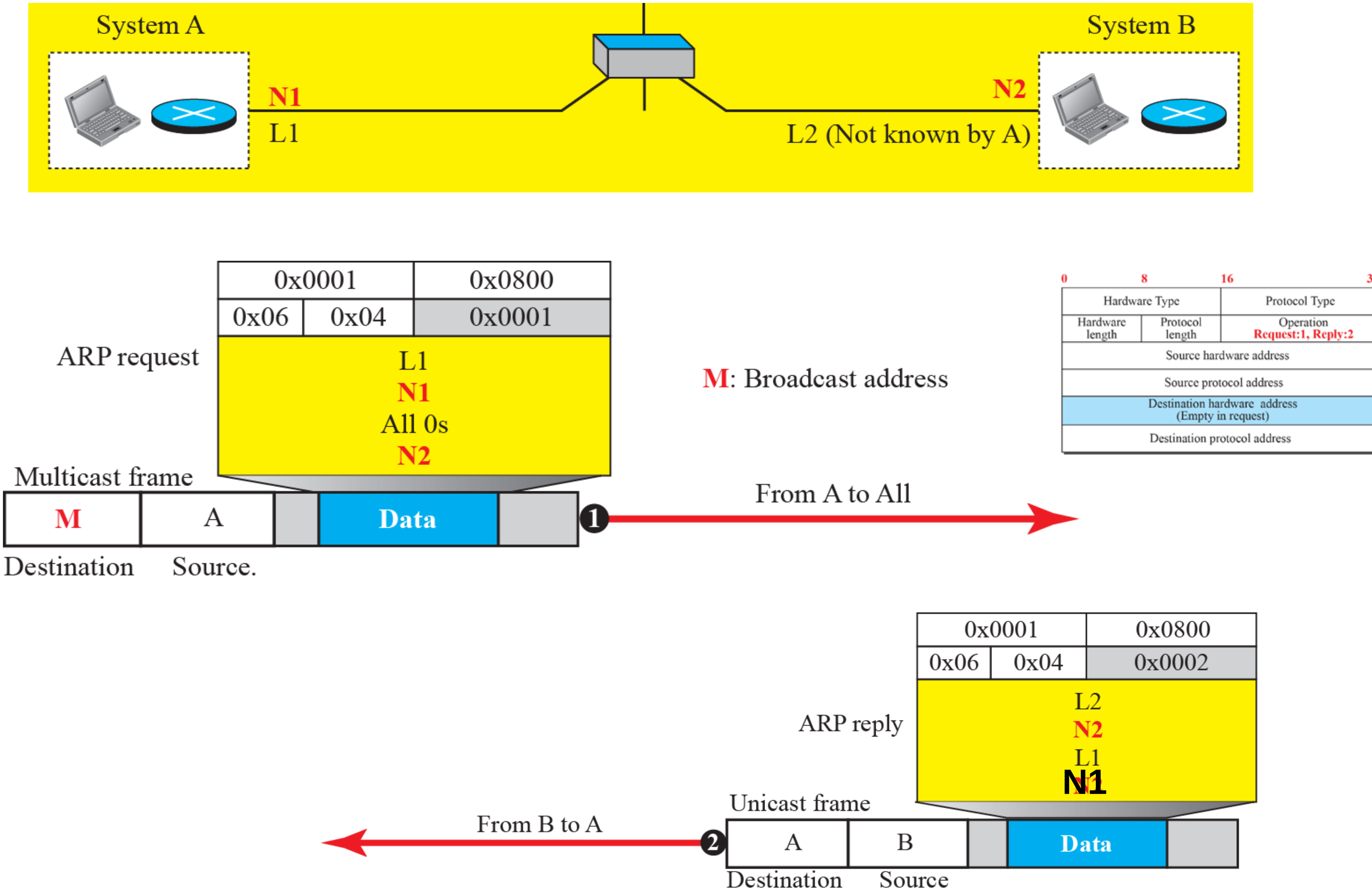
Hardware: LAN or WAN protocol

Protocol: Network-layer protocol

<table>
<tr><th>0</th><th>8</th><th>16 … 31</th></tr>
<tr><td colspan="2">Hardware Type</td><td>Protocol Type</td></tr>
<tr><td>Hardware length</td><td>Protocol length</td><td>Operation
Request:1, Reply:2</td></tr>
<tr><td colspan="3">Source hardware address</td></tr>
<tr><td colspan="3">Source protocol address</td></tr>
<tr><td colspan="3">Destination hardware address
(Empty in request)</td></tr>
<tr><td colspan="3">Destination protocol address</td></tr>
</table>

Example

A host with IP address N1 and MAC address L1 has a packet to send to another host with IP address N2 and physical address L2 (which is unknown to the first host). The two hosts are on the same network. Figure 9.9 shows the ARP request and response messages.


System A
System B
N1
L1
N2
L2 (Not known by A)
ARP request
0x0001
0x0800
0x06
0x04
0x0001
L1
N1
All 0s
N2
Multicast frame
M
A
Data
Destination
Source.
1
From A to All
M: Broadcast address
0
8
16
31
Hardware Type
Protocol Type
Hardware length
Protocol length
Operation
Request:1, Reply:2
Source hardware address
Source protocol address
Destination hardware address
(Empty in request)
Destination protocol address
ARP reply
0x0001
0x0800
0x06
0x04
0x0002
L2
N2
L1
N1
Unicast frame
From B to A
2
A
B
Data
Destination
Source