COMPUTER NETWORKS

Module – 3: Network Layer: Internet Protocol

Network Layer: Logical Addressing: IPv4 Addresses: Address Space, Notation, Classfull Addressing, Classless Addressing, IPv6 Addresses: Structure, Internet Protocol: IPv4 Datagram, IPv6, Transition from IPv4 to IPv6. [19.1,19.2, 20.1, 20.2,20.3,20.4]
Network Address Mapping: Address Mapping, Error Reporting: ARP, RARP, BOOTP and DHCP. [21.1]

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

IPv4 ADDRESSES

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet

Topics discussed in this section :

Address Space
Notations
Classful Addressing
Classless Addressing

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

IPv4 ADDRESSES

- *An IPv4 address is 32 bits long*
- *The IPv4 addresses are unique and universal*
- **The address space of IPv4 is 2^{32} or 4,294,967,296.**
- **Two devices in the Internet can never have the same address at the same time.**
- **An address may be assigned to a device for a time period and then taken away and assigned to another device.**
- **If a device operating at the network layer (e.g. router) has m connections to the Internet, it needs to have m IP address**

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

IPv4 Address Space

- *IPV4 has an address space: is the total number of addresses used by the protocol.*
- *If a protocol uses N bits to define an address, the address space is* 2^n
- *IPv4 uses 32-bit addresses: The address space=* 2^{32} *=4,294,967,296 (more than 4 billion)*
- *This means, if there were no restrictions, more than 4 billion devices could be connected to the Internet.*
- *IPv6 uses 128 bit-addresses*

IPv4 Notations

There are two prevalent notations to show an IPv4 address:

1. Binary notation:

Address is displayed as 32 bits.

Each octet is often referred to as byte.

IPv4 address referred to as 32-bit address or 4-byte address

Example:

01110101 10010101 00011101 00000010

IPv4 Notations

Dotted-decimal notation:

✓ More compact and easier to read

✓ Written in decimal form with a decimal point(dot)

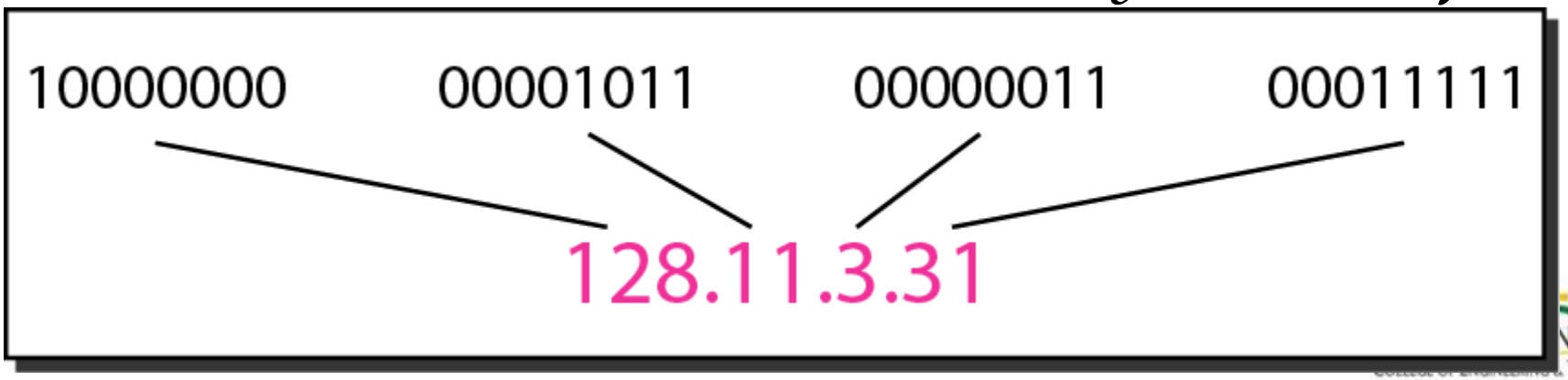
separating the bytes.

Example: 117.149.29.2

Each decimal value range from 0 to 255

Example:

Dotted-decimal notation and binary notation for



Example 19.1

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

Solution

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation.

a. 129.11.11.239

b. 193.131.27.255

Example 19.2

Change the following IPv4 addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78

b. 221.34.7.82

Solution

We replace each decimal number with its binary equivalent (see Appendix B).

a. 01101111 00111000 00101101 01001110

b. 11011101 00100010 00000111 01010010

Example 19.3

Find the error, if any, in the following IPv4 addresses.

a. 111.56.045.78

b. 221.34.7.8.20

c. 75.45.301.14

d. 11100010.23.14.67

Solution

a. There must be no leading zero (045).

b. There can be no more than four numbers.

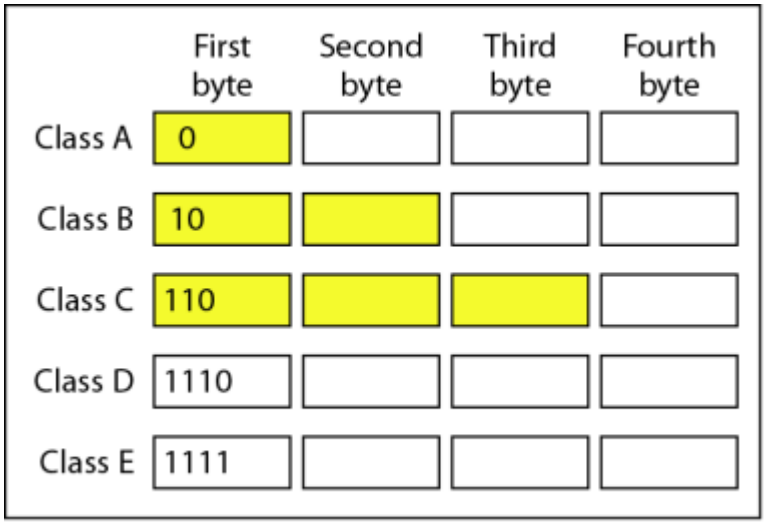
c. Each number needs to be less than or equal to 255.

d. A mixture of binary notation and dotted-decimal notation is not allowed.

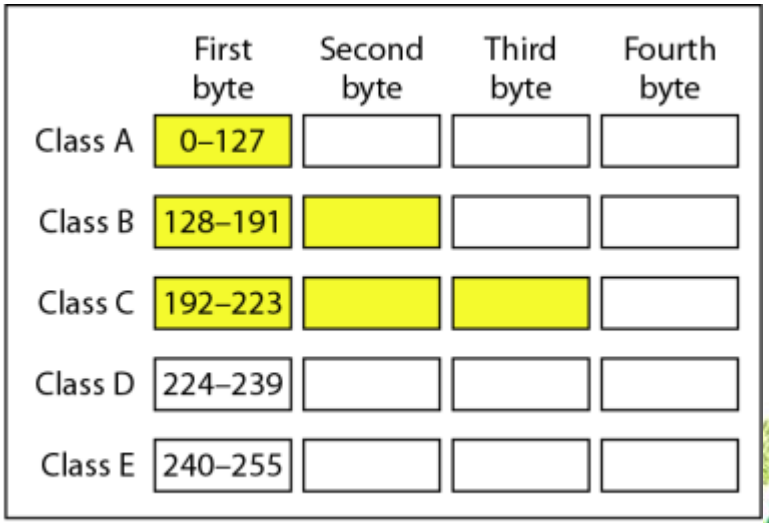
IPv4 Classful Addressing

In classful addressing, the address space is divided into five classes: A, B, C, D, and E.

- *We can find the class of an address in:*
- *Binary notation: the first few bits define the class*
- *Decimal-dotted notation: the first byte define the class*



a. Binary notation



b. Dotted-decimal notation

- *In classful addressing, the address space is divided into five classes: A, B, C, D, and E.*
- *Addresses in Classes A, B and C are uniast addresses*
- *A host needs to have at least one unicast address to be able to send packet (Source).*
- *Addresses in Class D are for multicast address: used only for destination.*
- *Addresses in class E are reserved*

Find the class of each address and Change the following IPv4 addresses from dotted-decimal notation to binary notation and visa versa.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

Example 4

Solution

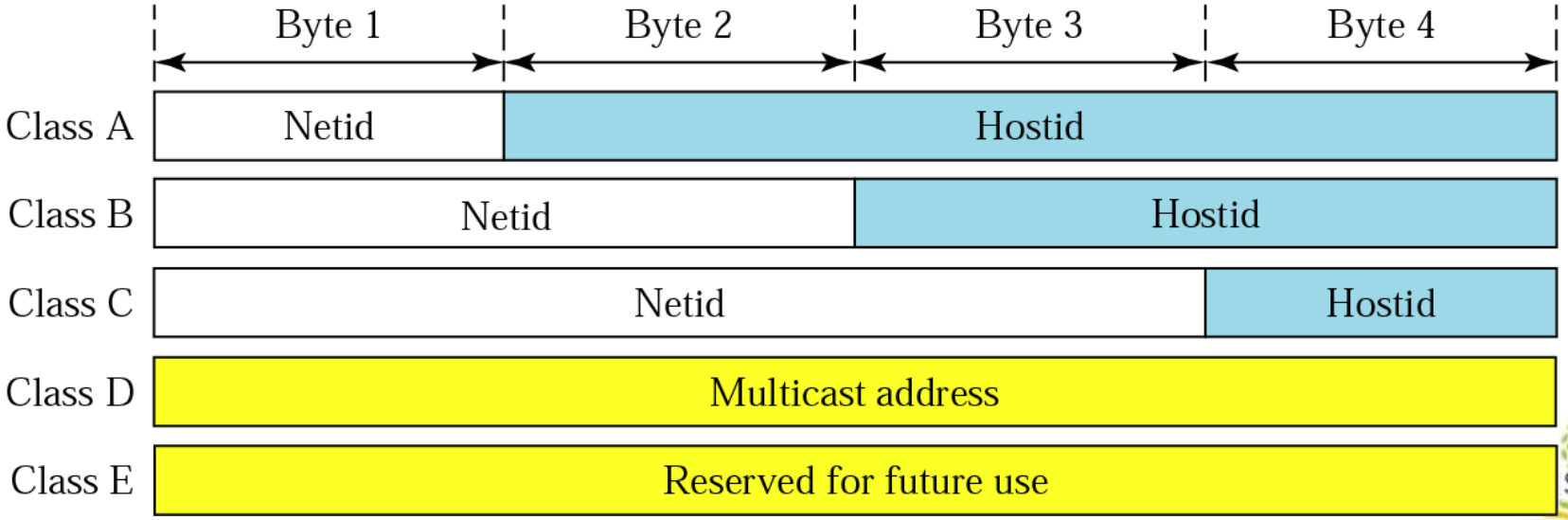
a. The first bit is 0. This is a class A address.

b. The first 2 bits are 1; the third bit is 0. This is a class C address.

NetId and HostId

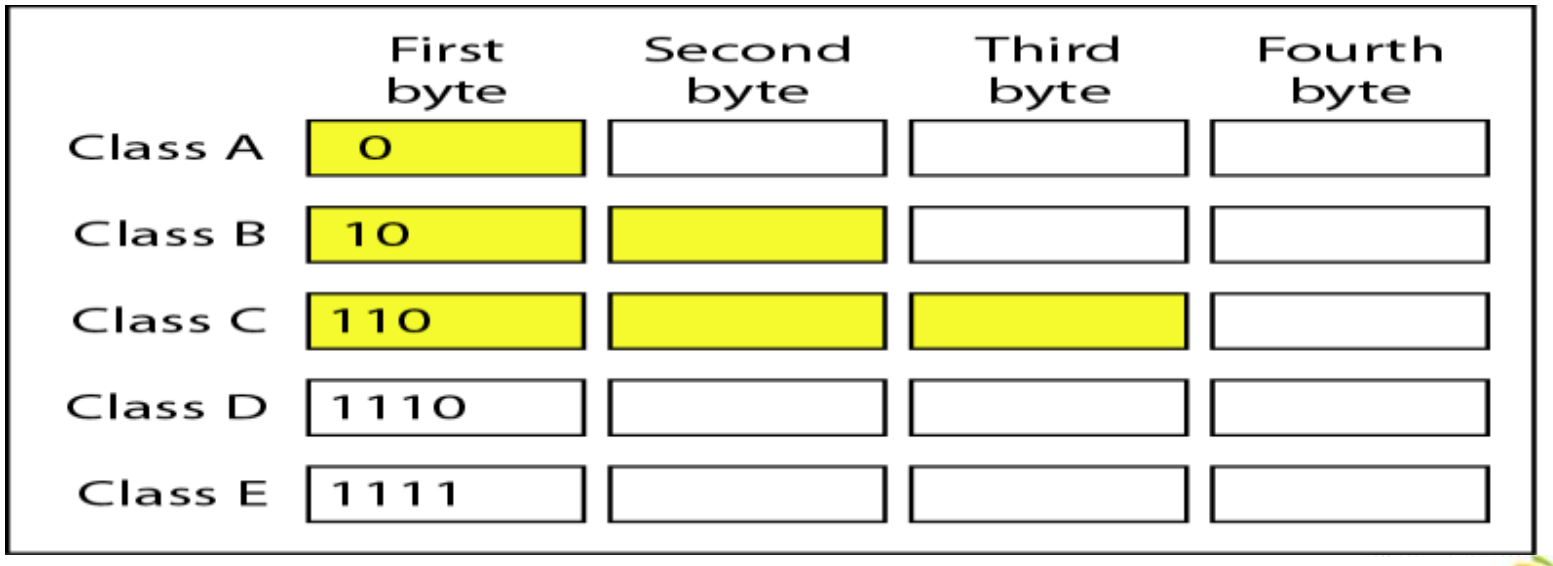
- *The address is divided into Netid and Hostid.*
- *These part are of varying lengths, depending on the class.*

Dose not apply to classes D and E



IPv4 ADDRESSES

Class	Number of Blocks	Block Size	Application
A	$2^7 = 128$	$2^{24} = 16,777,216$	Unicast
B	$2^{14} = 16,384$	$2^{16} = 65,536$	Unicast
C	$2^{21} = 2,097,152$	$2^8 = 256$	Unicast
D	1	$2^{28} = 268,435,456$	Multicast
E	1	$2^{28} = 268,435,456$	Reserved



❖*Class A address: designed for large organizations with a large number of attached hosts or routers. (most of the addresses were wasted and not used)*

❖*Class B address: designed for midsize organizations with ten of thousands of attached hosts or routers(too large for many organizations)*

❖*Class C address: designed for small organizations with a small number of attached hosts or routers (too small for many organizations)*

❖*Class D address: designed for multicasting. (waste of addresses)*

❖ *Class E address: reserved for future use (waste of addresses)*

One problem is that each class is divided into fixed number of blocks with each block having a fixed size

In classful addressing, a large part of the available addresses were wasted.

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

Classful Addressing: Classes and Blocks

Mask (default mask)

- **Help us to find the NetId and HostId**
- **Mask: 32-bit made of 1s followed by 0s.**
- **Dose not apply to classes D and E.**
- **CIDR(Classless Interdomain Routing): used to show the mask in the form /n (n=8,16,24)**

Class	Binary	Dotted-decimal	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

Classful Addressing: Network address

***The network address** is an address that define the network itself to the reset of the internet*

The network address has the following properties:

1. All hostid bytes are 0's

2. It is the first address in the block

3. It cannot be assigned to a host

4. Given the network address, we can find the class of the address

Classful Addressing: Network address

Find the network address for the following:

a. 132.6.17.85

b. 23.56.7.91

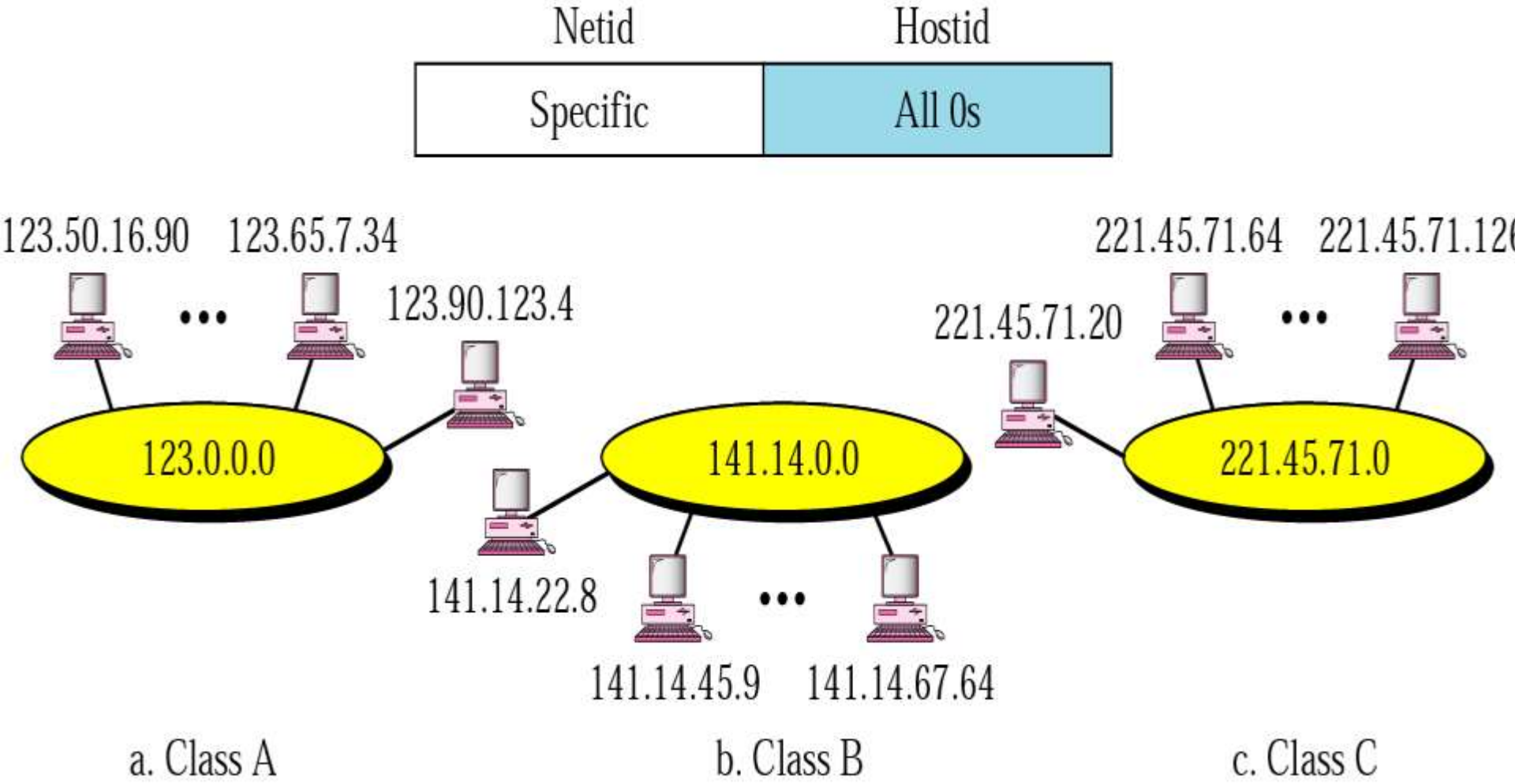
a. The class is B. The first 2 bytes defines the Netid. We can find the network address by replacing the hostid bytes (17.85) with 0s.

Therefore, the network address is 132.6.0.0

b. **The class is A. Only the first byte defines the Netid. We can find the network address by replacing the hostid bytes (56.7.91) with 0s.**

Therefore, the network address is 23.0.0.0

Classful Addressing: Network address

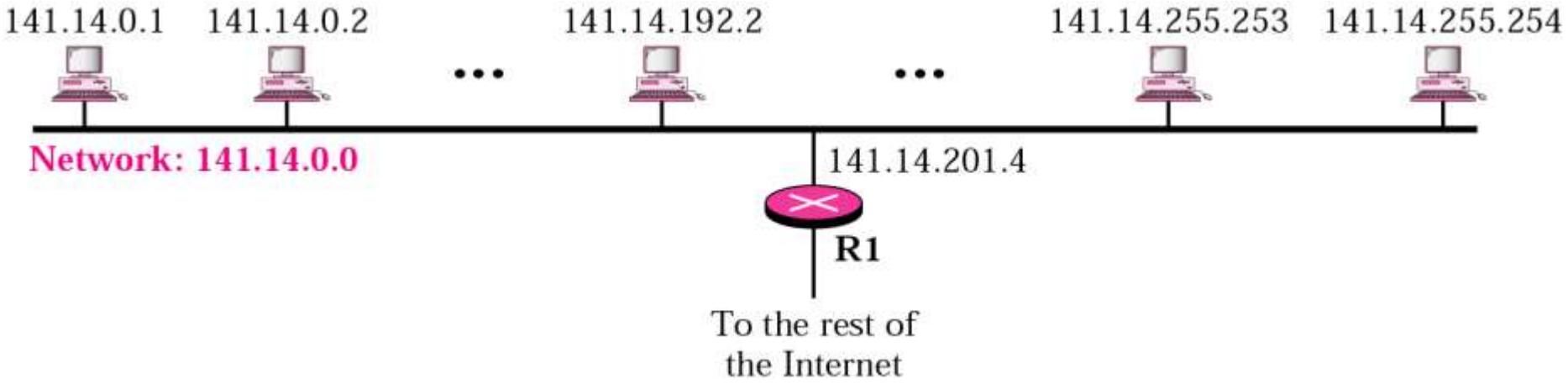


Classful addressing : subnetting

IP addresses are designed with two levels of hierarchy-IP addresses are divided into two parts — the network part and the host part.

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

A network with two levels of hierarchy



Subnetting

If an organization is granted a large address block (in Class A or ClassB) it can divide the block into several contiguous groups and assign each group to smaller networks called subnets.

Subnetting is the process of dividing a large network (given IP block) into smaller networks (called *subnets*) by borrowing bits from the host part of the IP address.

Subnet Mask

- **Subnetting increases the number of 1s in the subnet mask.**
- **The number of 1s in a subnet mask is greater than the number of 1s in the corresponding classful mask.**
- **To create a subnet mask, some of the leftmost 0s in the host portion of the mask are changed to 1s.**

The number of subnets is determine by the number of extra1s.

- **If the number of extra 1 is n, the number of subnets is 2^n.**
- **If the number of subnets is N, the number of extra 1s is $\log_2 N$**

Classful Addressing: Subnet Mask

Class B address
mask : 255.255.0.0 or /16

11111111 11111111	00000000 00000000

For 4 subnets : ($\log_2 4 = 2$; need 2-extra bits)
Subnet mask: 255.255. 192.0 or /18

11111111 11111111	11	000000 00000000

For 8 subnets: ($\log_2 8 = 3$; need 3-extra bits)
subnet mask : 255.255.224.0 or /19

11111111 11111111	111	00000 00000000

Subnetting

Example:

A router receives a packet with destination address 190.240.33.91. Show how it finds the network and the subnetwork address to route the packet. Assume the subnet mask is /19

The router follows steps:

- **Given: destination IP = 190.240.33.91**
  Subnet mask (CIDR) = /19 → mask = 255.255.224.0
- **First byte 190 ⇒ Class B (default mask /16 = 255.255.0.0).**
- **If you AND /16 with the IP you get the classful network:**
- **190.240.33.91 AND 255.255.0.0 → 190.240.0.0 (classful network)**
- **So IP = 10111110 . 11110000 . 00100001 . 01011011**
- **Mask /19 = 11111111 . 11111111 . 11100000 . 00000000 (which is 255.255.224.0)**
- **Octet-wise AND:**
  1st octet: 10111110 AND 11111111 = 10111110 → 190
  2nd octet: 11110000 AND 11111111 = 11110000 → 240
  3rd octet: 00100001 AND 11100000 = 00100000 → 32
  4th octet: 01011011 AND 00000000 = 00000000 → 0

- **So subnets begin at 0, 32, 64, 96, ... in the third octet.**
- **The subnet starting at 190.240.32.0 covers third-octet values 32 through 63 (because next subnet starts at 64).**
- **Broadcast for this subnet = 190.240.63.255.**
- **Usable host range = 190.240.32.1 → 190.240.63.254.**
- **Router checks its routing table for the most specific match.**
- **If it has a route for 190.240.32.0/19, it forwards according to that entry.**

Classful Addressing *supernetting*

◆ **Need for Supernetting**

- **There was a huge demand for midsize address blocks.**
- **Class A and Class B addresses are almost exhausted, while Class C addresses are still available.**
- **However, a single Class C block (with 256 addresses) is often too small for many organizations.**

◆ **Concept of Supernetting**

- **Supernetting allows an organization to combine multiple contiguous Class C networks to form a larger address block.**
- **This combination of several networks creates a supernetwork (or supernet).**
- **This concept is the opposite of subnetting:**
  - **Subnetting adds 1s to the mask (makes it longer).**
  - **Supernetting removes 1s from the mask (makes it shorter).**

Classful Addressing: supernetting

(Classless Inter-Domain Routing – CIDR)

Address Depletion

- **Near depletion of the available address because of the fast growth of the Internet.**
- **Run out of classes A and B address.**
- **Classes C block is too small for most mid size organizations.**

Solution: Classless addressing

Classless Addressing

Address Blocks: In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block (the number of addresses) varies based on the nature and size of the entity.

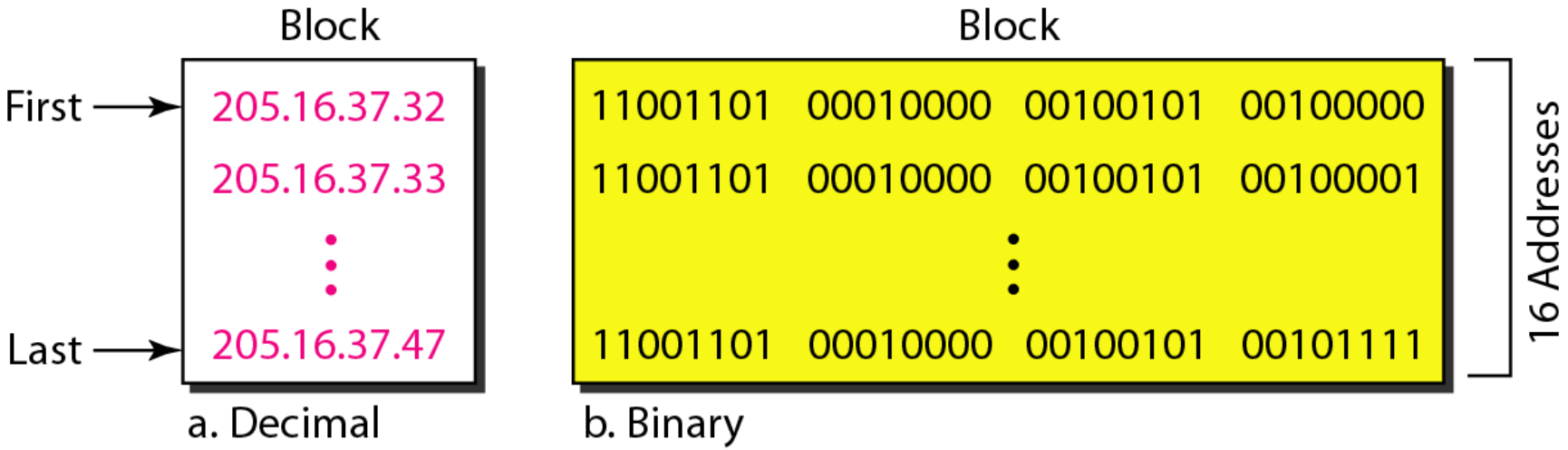
Restriction To simplify the handling of addresses, the Internet authorities impose three restrictions on classless address blocks:

1. The addresses in a block must be contiguous, one after another.

2. The number of addresses in a block must be a power of 2 (I, 2, 4, 8, ...).

3. The first address must be evenly divisible by the number of addresses.

Figure 19.3 *A block of 16 addresses granted to a small organization*



Example 19.5

Figure 19.3 shows a block of addresses, in both binary and dotted-decimal notation, granted to a small business that needs 16 addresses.

We can see that the restrictions are applied to this block. The addresses are contiguous. The number of addresses is a power of 2 ($16 = 2^4$), and the first address is divisible by 16. The first address, when converted to a decimal number, is 3,440,387,360, which when divided by 16 results in 215,024,210.

Decimal Value=(205×2563)+(16×2562)+(37×256)+32= 3,440,387,360

Note

In IPv4 addressing, a block of addresses can be defined as x.y.z.t */n* in which x.y.z.t defines one of the addresses and the */n* defines the mask.

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

Note

The first address in the block can be found by setting the rightmost 32 − *n* bits to 0s.

Example 19.6

A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the first address in the block?

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

If we set 32−28 rightmost bits to 0, we get

*11001101 00010000 00100101 0010**0000***

or

205.16.37.32.

This is actually the block shown in Figure 19.3.

Note

The last address in the block can be found by setting the rightmost 32 − n bits to 1s.

Example 19.7

Find the last address for the block in Example 19.6.

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

If we set 32 − 28 rightmost bits to 1, we get

11001101 00010000 00100101 00101111

or

205.16.37.47

This is actually the block shown in Figure 19.3.

Note

The number of addresses in the block can be found by using the formula 2^{32-n}.

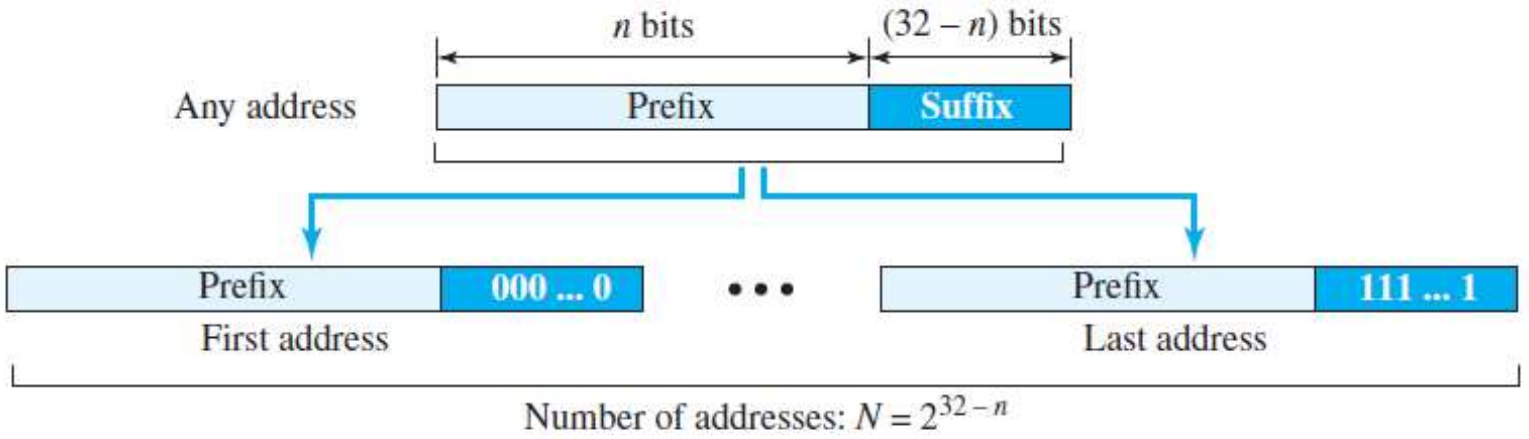
Dr. Chandrappa S
NAGARJUNA
COLLEGE OF ENGINEERING & TECHNOLOGY

Example 19.8

Find the number of addresses in Example 19.6.

Solution

The value of n is 28, which means that number of addresses is 2^{32-28} or 16.



The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s.

Address: 167.199.170.82/27	10100111	11000111	10101010	01010010
First address: 167.199.170.64/27	10100111	11000111	10101010	01000000

The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s.

Address: 167.199.170.82/27	10100111	11000111	10101010	01011111
Last address: 167.199.170.95/27	10100111	11000111	10101010	01011111

Another way to find the first and last addresses in the block is to use the address mask.
The address mask is a 32-bit number in which the *n* leftmost bits are set to 1s and the rest of the bits (32 − *n*) are set to 0s.

1. The number of addresses in the block N = **NOT** (mask) + 1.
2. The first address in the block = (Any address in the block) **AND** (mask).
3. The last address in the block = (Any address in the block) **OR** [(**NOT** (mask)].

Example 19.9

Another way to find the first address, the last address, and the number of addresses is to represent the mask as a 32-bit binary (or 8-digit hexadecimal) number. This is particularly useful when we are writing a program to find these pieces of information. In Example 19.5 the /28 can be represented as

11111111 11111111 11111111 11110000

(twenty-eight 1s and four 0s).

Find

a. The first address

b. The last address

c. The number of addresses.

Example 19.9 (continued)

Solution

a. *The first address can be found by ANDing the given addresses with the mask. ANDing here is done bit by bit. The result of ANDing 2 bits is 1 if both bits are 1s; the result is 0 otherwise.*

Address:	11001101	00010000	00100101	00100111
Mask:	11111111	11111111	11111111	11110000
First address:	11001101	00010000	00100101	00100000

Example 19.9 (continued)

b. ***The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit. The result of ORing 2 bits is 0 if both bits are 0s; the result is 1 otherwise. The complement of a number is found by changing each 1 to 0 and each 0 to 1.***

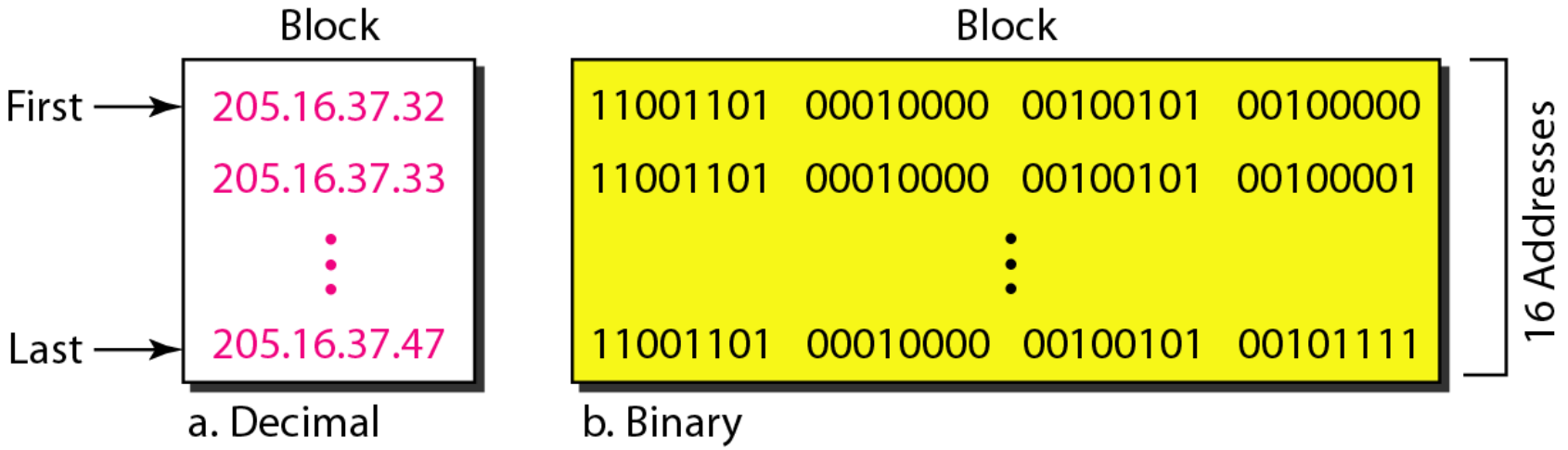
Address:	11001101	00010000	00100101	00100111
Mask complement:	**00000000**	**00000000**	**00000000**	**00001111**
Last address:	11001101	00010000	00100101	00101111

Example 19.9 (continued)

c. ***The number of addresses can be found by complementing the mask, interpreting it as a decimal number, and adding 1 to it.***

Mask complement:	00000000 00000000 00000000 00001111
Number of addresses:	15 + 1 = 16

Figure 19.4 *A network configuration for the block 205.16.37.32/28*



Note

The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.

Figure 19.6 *A frame in a character-oriented protocol*

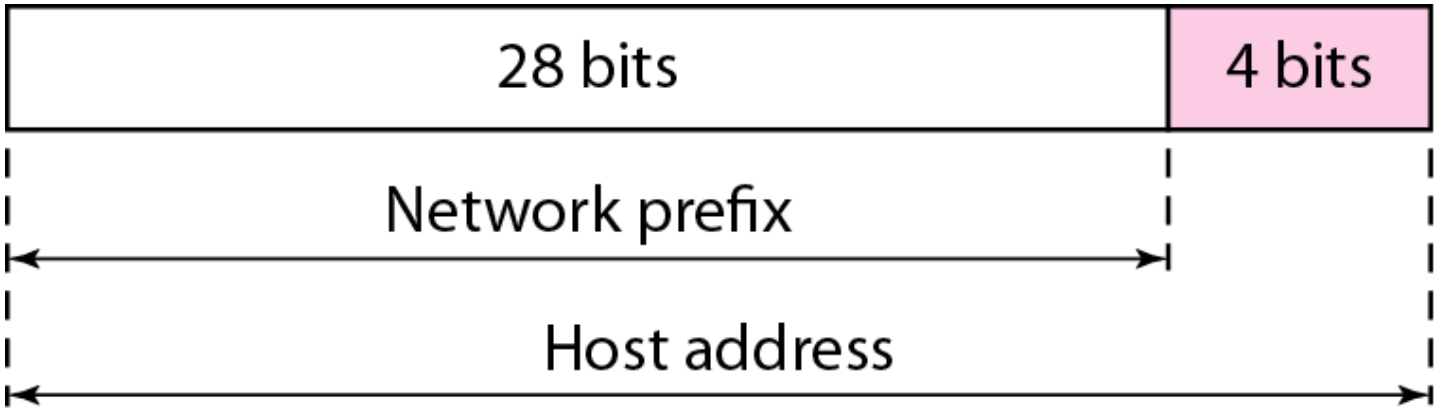


Figure 19.7 *Configuration and addresses in a subnetted network*

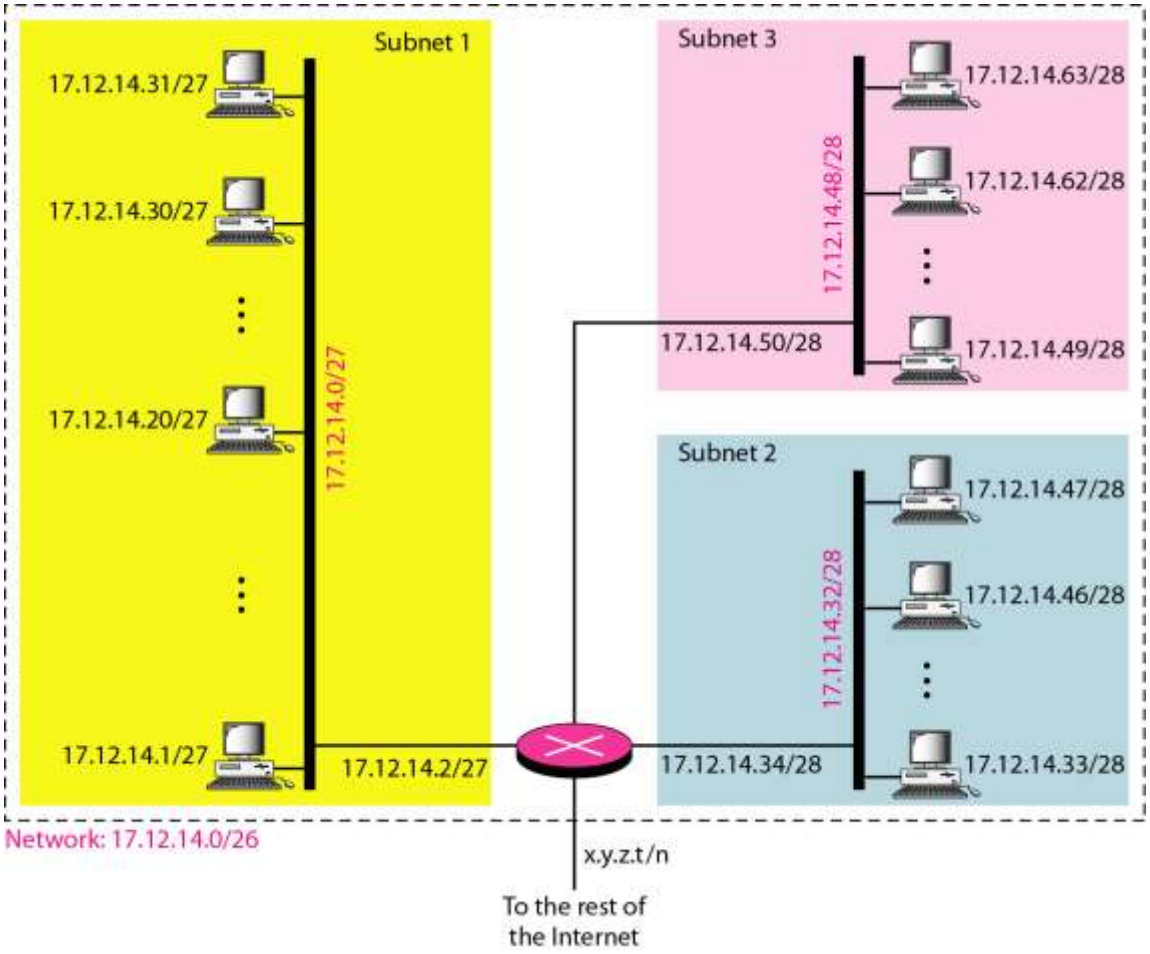
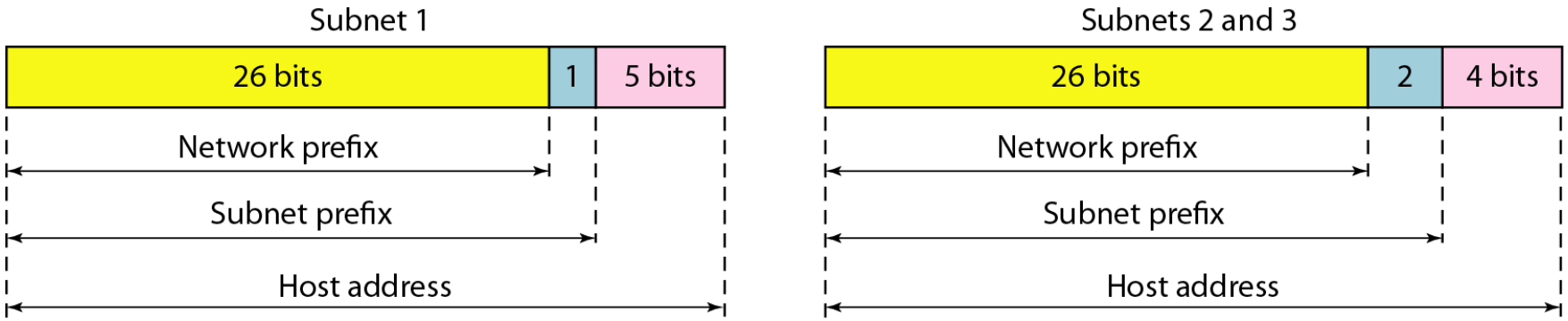


Figure 19.8 *Three-level hierarchy in an IPv4 address*



Example 19.10

An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

a. The first group has 64 customers; each needs 256 addresses.

b. The second group has 128 customers; each needs 128 addresses.

c. The third group has 128 customers; each needs 64 addresses.

Design the subblocks and find out how many addresses are still available after these allocations.

Example 19.10 (continued)

Solution

Figure 19.9 shows the situation.

Group 1

For this group, each customer needs 256 addresses. This means that 8 (log2 256) bits are needed to define each host. The prefix length is then 32 − 8 = 24. The addresses are

1st Customer:	190.100.0.0/24	190.100.0.255/24
2nd Customer:	190.100.1.0/24	190.100.1.255/24
. . .		
64th Customer:	190.100.63.0/24	190.100.63.255/24

Total = 64 × 256 = 16,384

Example 19.10 (continued)

Group 2

For this group, each customer needs 128 addresses. This means that 7 (log2 128) bits are needed to define each host. The prefix length is then 32 − 7 = 25. The addresses are

1st Customer:	*190.100.64.0/25*	*190.100.64.127/25*
2nd Customer:	*190.100.64.128/25*	*190.100.64.255/25*
. . .		
128th Customer:	*190.100.127.128/25*	*190.100.127.255/25*

Total = 128 × 128 = 16,384

Example 19.10 (continued)

Group 3

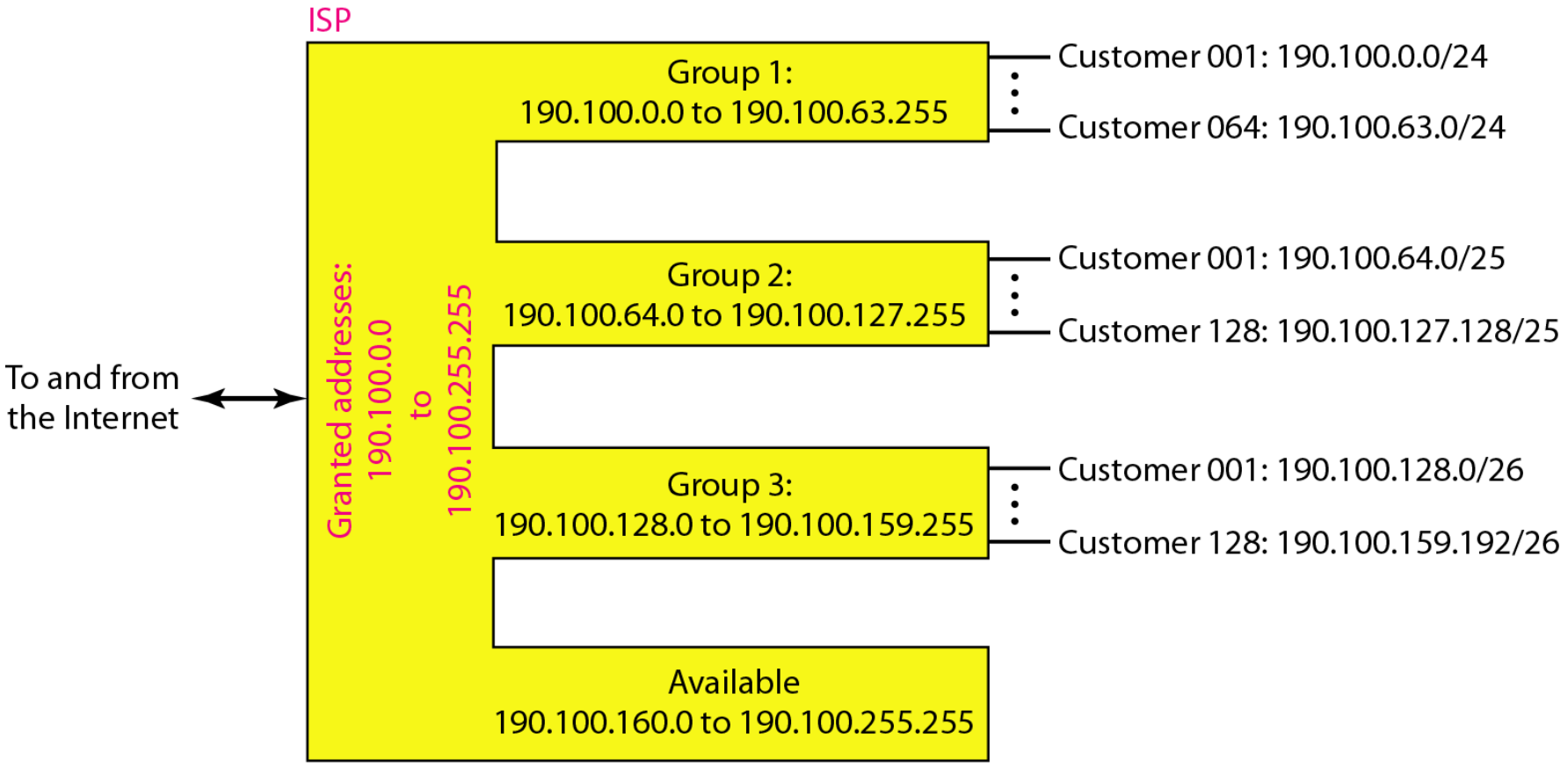
For this group, each customer needs 64 addresses. This means that 6 ($log_2 64$) bits are needed to each host. The prefix length is then 32 − 6 = 26. The addresses are

1st Customer:	*190.100.128.0/26*	*190.100.128.63/26*
2nd Customer:	*190.100.128.64/26*	*190.100.128.127/26*
. . .		
128th Customer:	*190.100.159.192/26*	*190.100.159.255/26*

Total = 128 × 64 = 8192

Number of granted addresses to the ISP: 65,536
Number of allocated addresses by the ISP: 40,960
Number of available addresses: 24,576

Figure 19.9 *An example of address allocation and distribution by an ISP*



19-2 IPv6 ADDRESSES

Despite all short-term solutions, address depletion is still a long-term problem for the Internet. This and other problems in the IP protocol itself have been the motivation for IPv6.

Topics discussed in this section:

Structure
Address Space

Figure 19.14 *IPv6 address in binary and hexadecimal colon notation*

Structure An IPv6 address consists of 16 bytes (octets); it is 128 bits long.

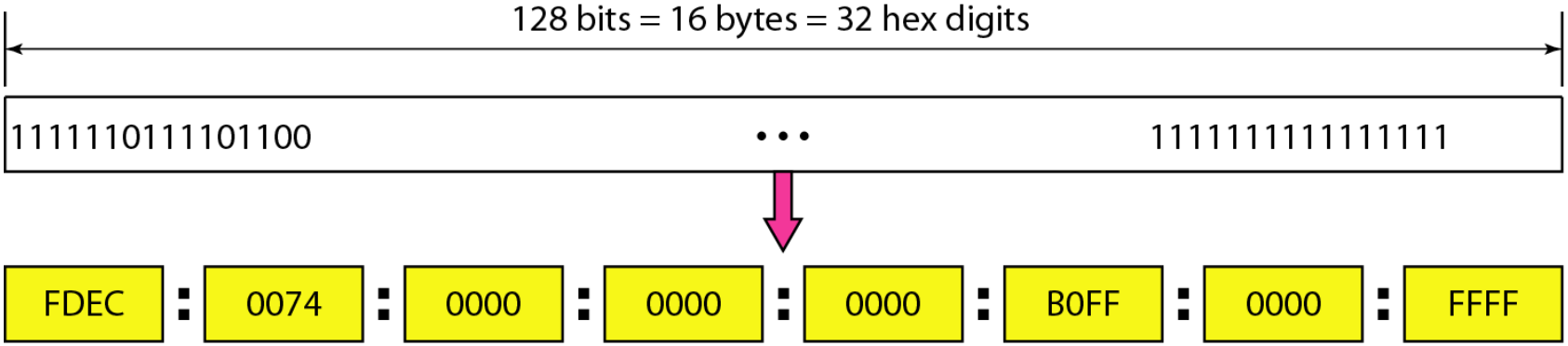
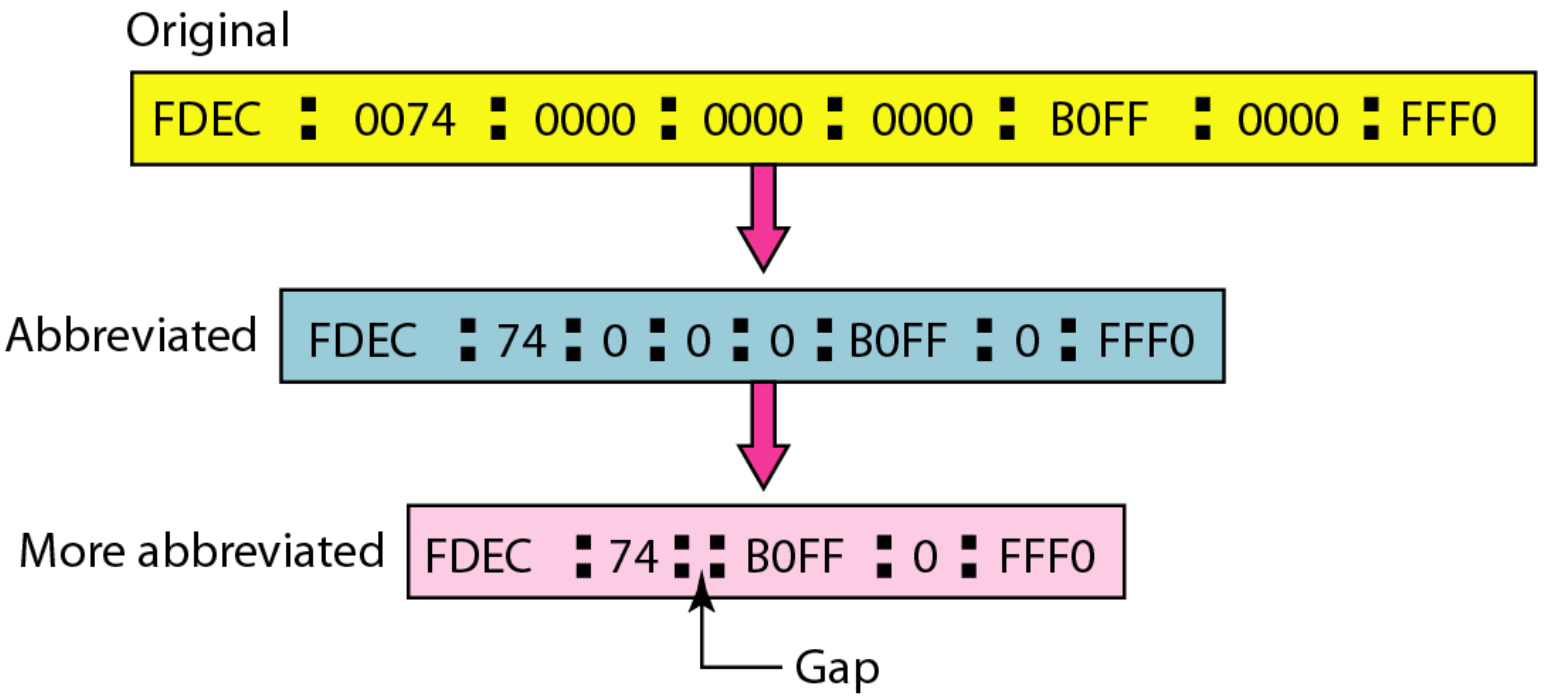


Figure 19.15 *Abbreviated IPv6 addresses*



Example 19.11

Expand the address 0:15::1:12:1213 to its original.

Solution

We first need to align the left side of the double colon to the left of the original pattern and the right side of the double colon to the right of the original pattern to find how many 0s we need to replace the double colon.

```
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
  0:   15:                  :    1:   12:1213
```

This means that the original address is.

```
0000:0015:0000:0000:0000:0001:0012:1213
```

IPv4

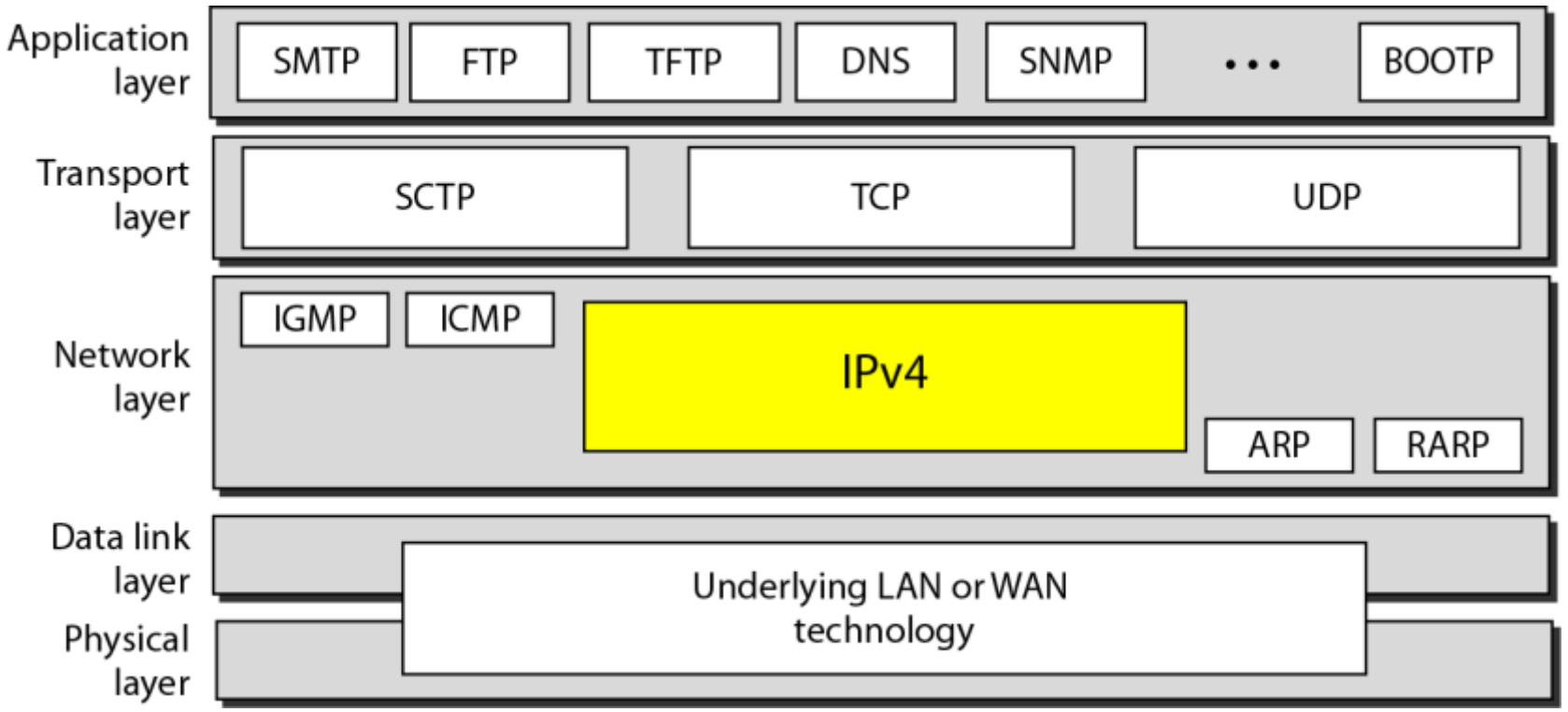
The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols.

Topics discussed in this section:

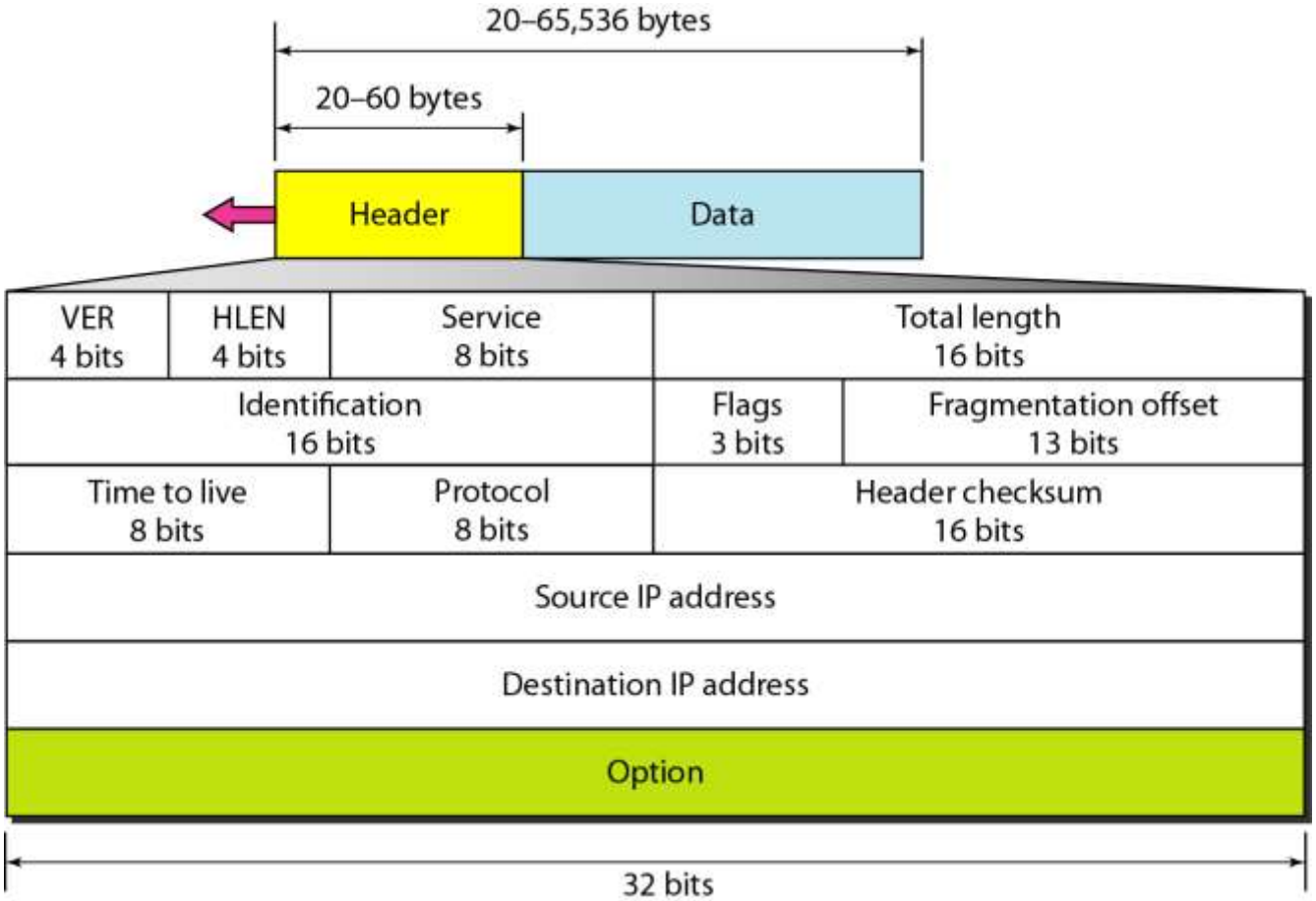
Datagram
Fragmentation
Checksum
Options

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

Figure 20.4 *Position of IPv4 in TCP/IP protocol suite*



IPv4 datagram format



IPv4 Header Fields (20–60 bytes)

1. **Version (4 bits)**
   Indicates the IP version (for IPv4, the value is 4).
2. **Header Length (IHL) (4 bits)**
   Specifies the length of the header in 32-bit words (minimum 5 → 20 bytes, maximum 15 → 60 bytes).
3. **Type of Service (ToS) / Differentiated Services (8 bits)**
   Used for Quality of Service (QoS), prioritizing packets.
4. **Total Length (16 bits)**
   Entire packet size (header + data), maximum = 65,535 bytes.
5. **Identification (16 bits)**
   Unique ID for fragments of a packet.
6. **Flags (3 bits)**
   Control fragmentation:
   1. **Bit 0: Reserved (always 0)**
   2. **Bit 1: Don't Fragment (DF)**
   3. **Bit 2: More Fragments (MF)**
7. **Fragment Offset (13 bits)**
   Position of this fragment in the original packet.
8. **Time to Live (TTL) (8 bits)**
   Limits packet lifetime (decremented by each router; if 0, packet is discarded).
9. **Protocol (8 bits)**
   Indicates upper-layer protocol:
   1. **1 = ICMP**
   2. **6 = TCP**
   3. **17 = UDP, etc.**
10. **Header Checksum (16 bits)**
    Error checking for the header only.
11. **Source IP Address (32 bits)**
    Address of the sender.
12. **Destination IP Address (32 bits)**
    Address of the receiver.
13. **Options (variable, optional)**
    Extra fields for testing, security, or routing (rarely used).
14. **Padding**
    Ensures the header length is a multiple of 32 bits.

1. Version (VER):

This 4-bit field defines the version of the IP protocol

2. Header length (HLEN):

When there are no options, the header length is 20 bytes, and the value of this field is 5 (5 x 4 = 20).

When the option field is at its maximum size, the value of this field is 15 (15 x 4 = 60).

3. Services-*Service type or differentiated services*

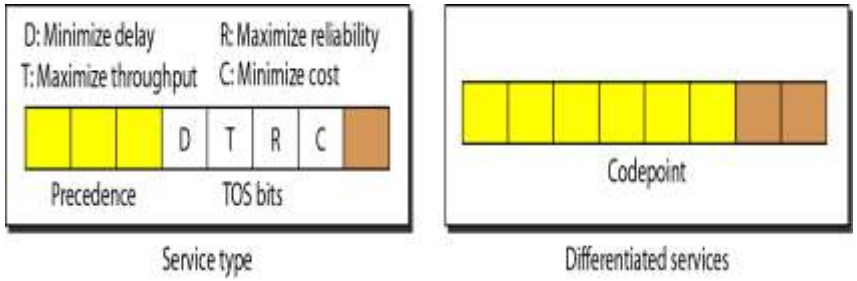


Table : ***Types of service***

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

Table 20.3 *Values for codepoints*

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

Open Shortest Path First

Figure 20.7 *Encapsulation of a small datagram in an Ethernet frame*

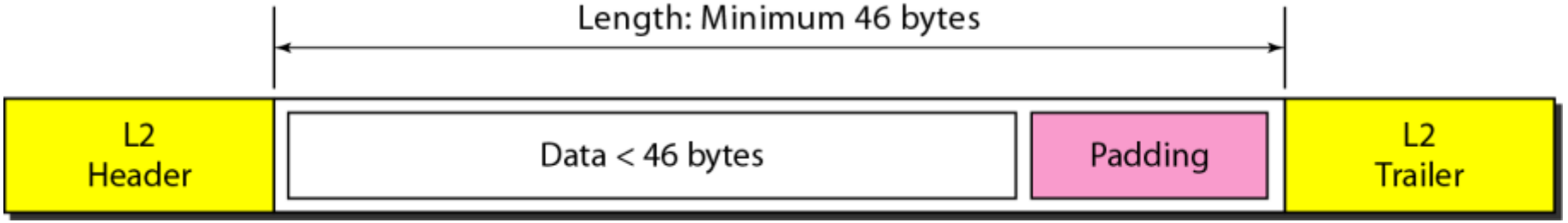


Figure 20.9 *Maximum transfer unit (MTU)*

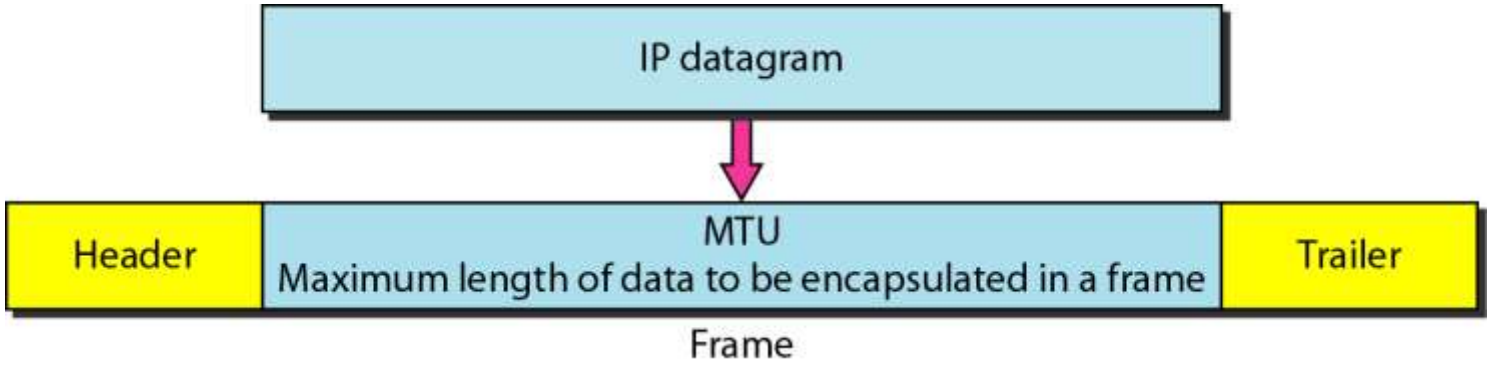


Table 20.5 *MTUs for some networks*

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

Fiber Distributed Data Interface

Fragmentation:

Identification: This 16-bit field identifies a datagram originating from the source host.

Flags

Figure 20.10 ***Flags used in fragmentation***

Figure 20.11 *Fragmentation example*

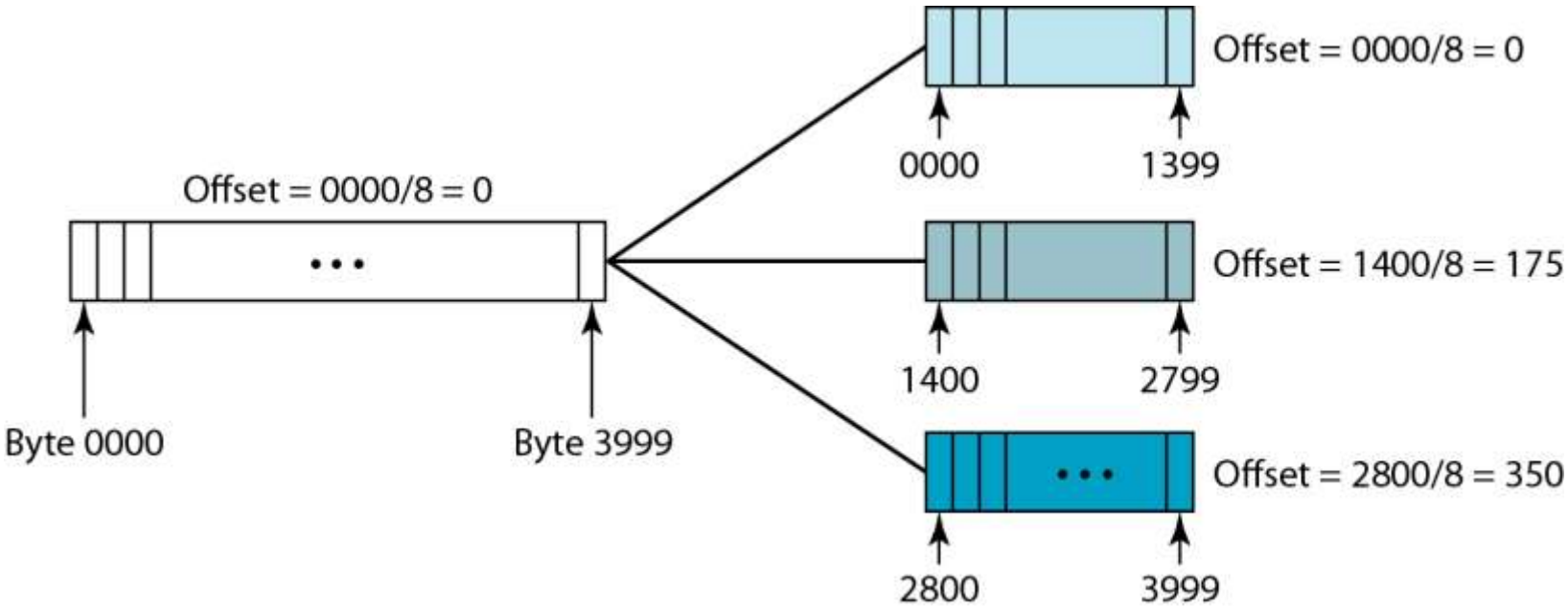


Figure 20.12 *Detailed fragmentation example*

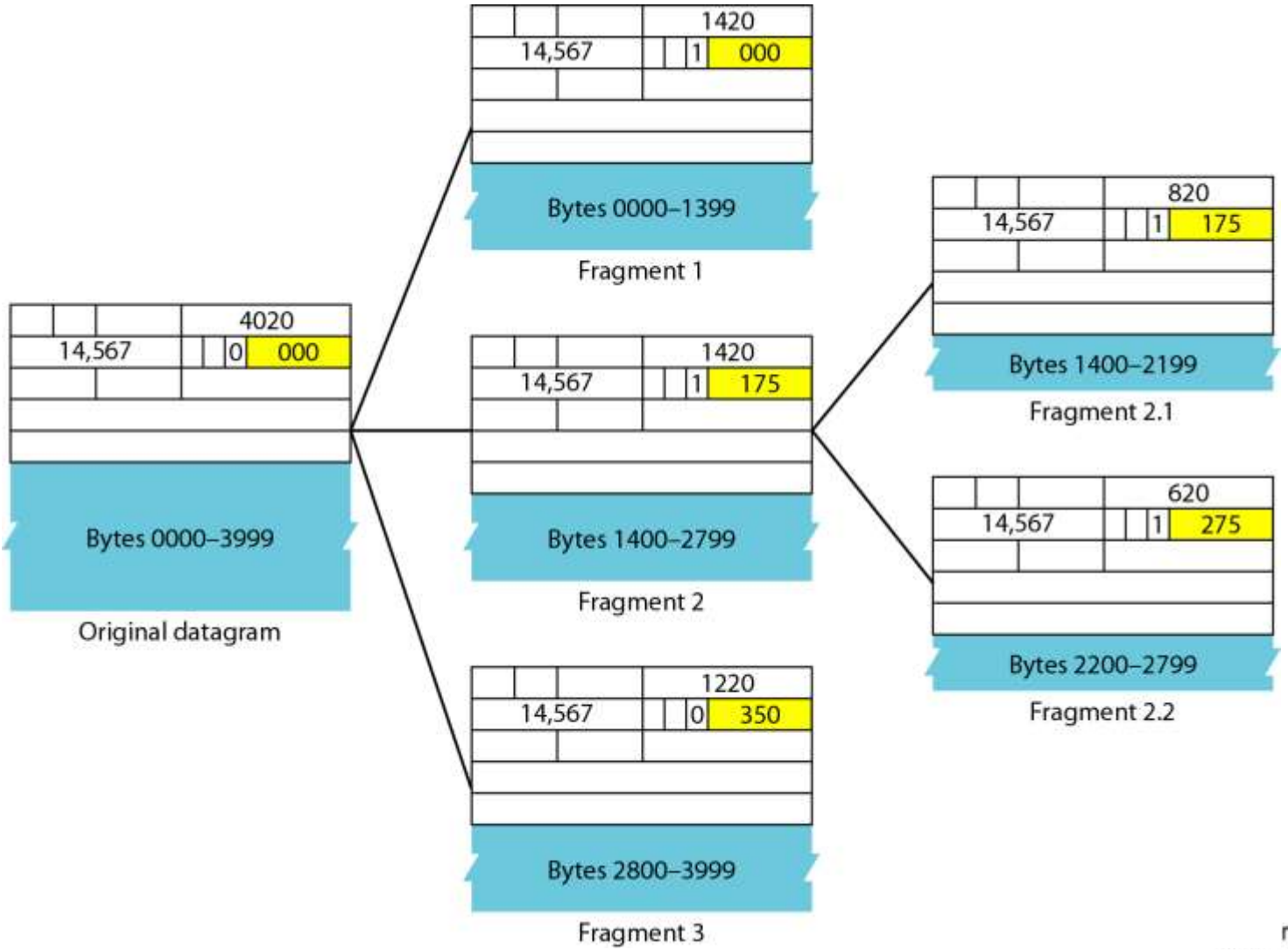
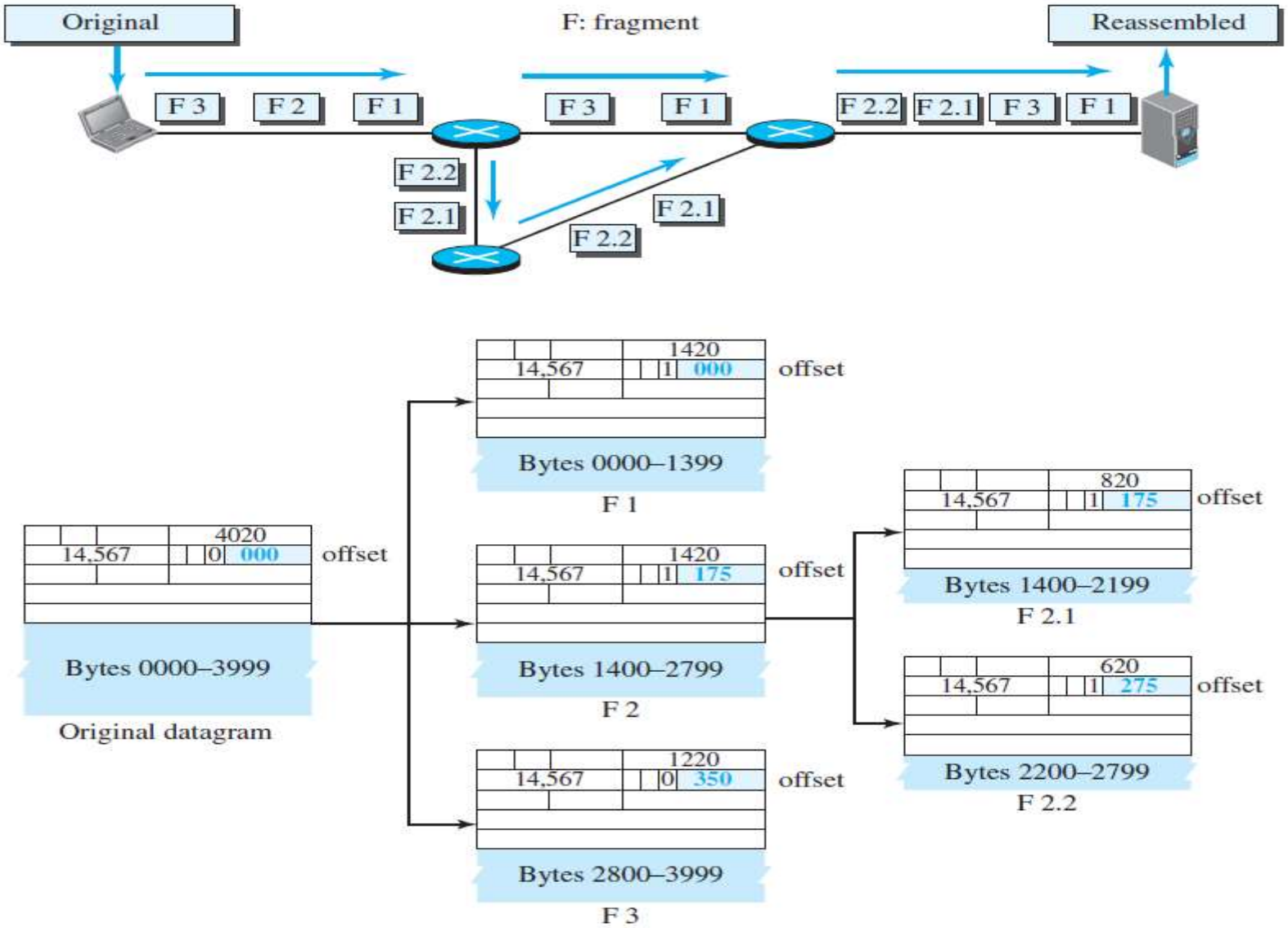


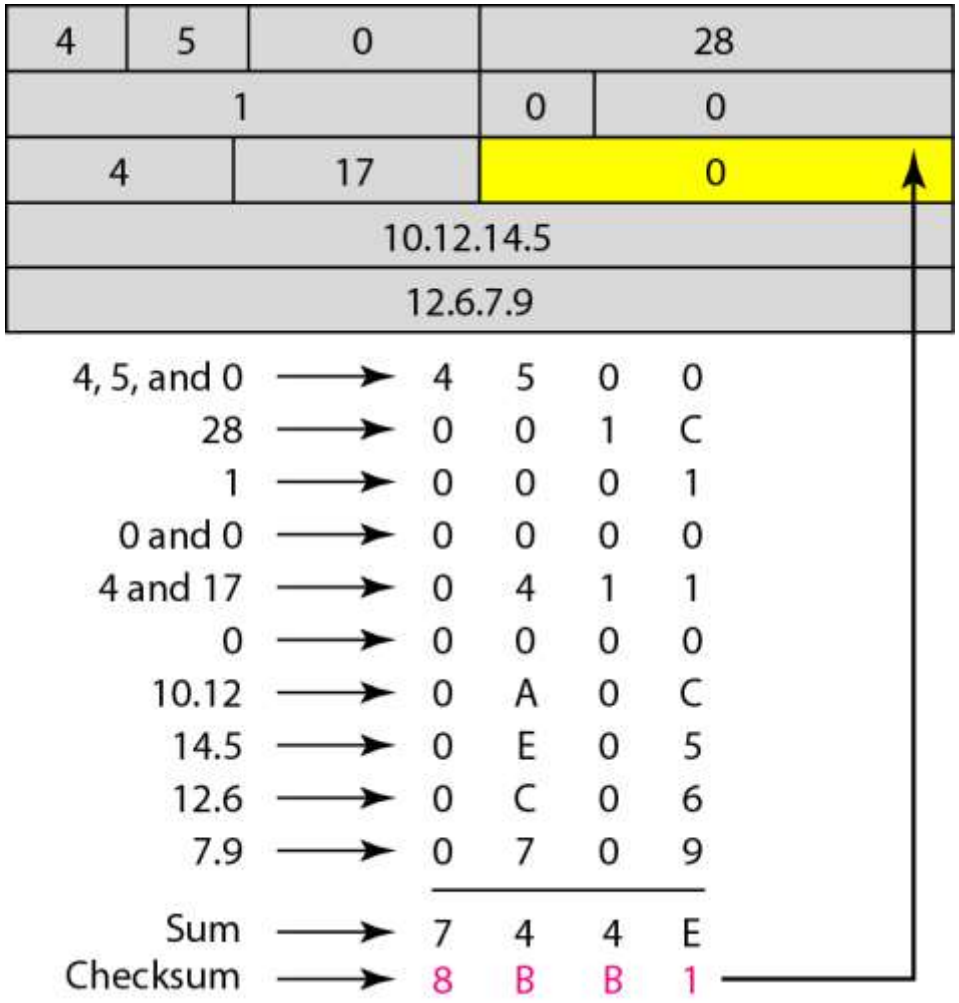
Figure 19.7 *Detailed fragmentation example*



Example 20.10

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.

Figure 20.13 *Example of checksum calculation in IPv4*



options

The header of the IPv4 datagram is made of two parts: a fixed part and a variable part. The fixed part is 20 bytes long. The variable part comprises the options that can be a maximum of 40 bytes (in multiples of 4-bytes) to preserve the boundary of the header. They can be used for network testing and debugging

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

IPv6

Topics discussed in this section:

Advantages
Packet Format
Extension Headers

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

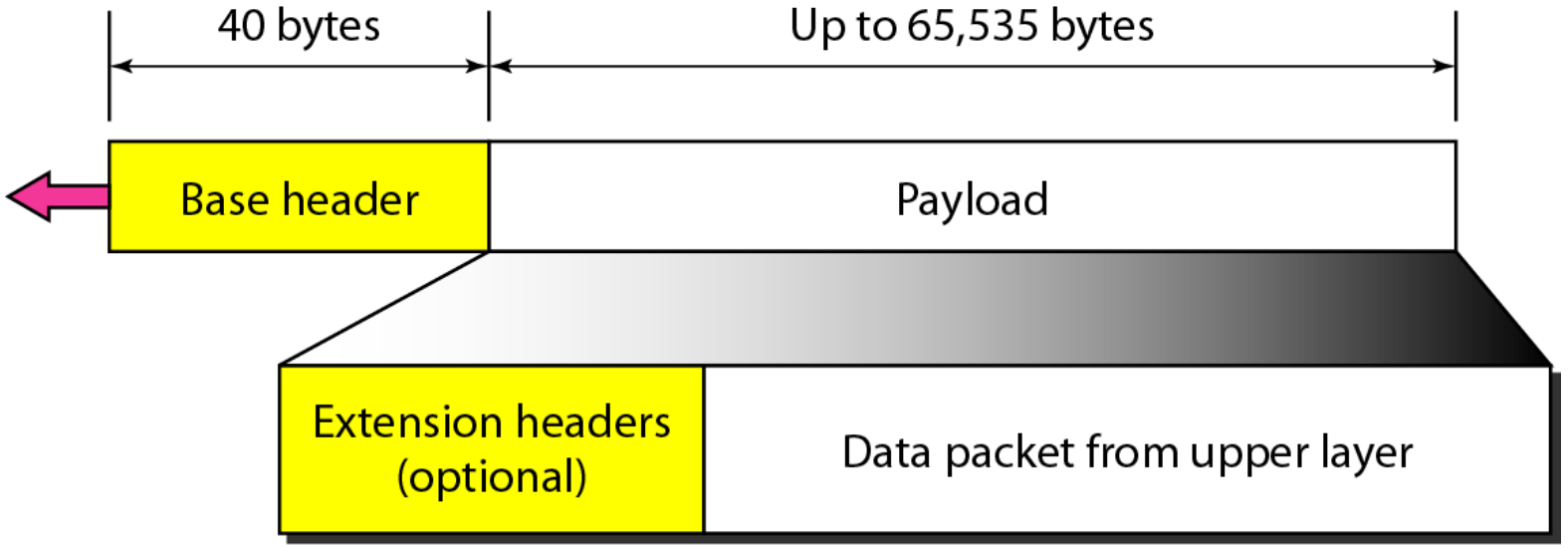
changes implemented in the protocol- Advantages

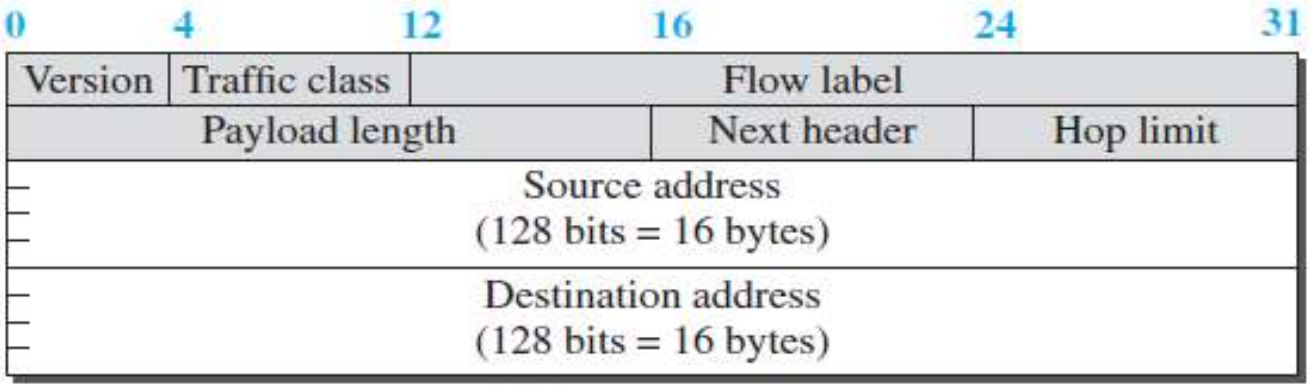
- ***Better header format*.** IPv6 uses a new header format in which **options are separated from the base header** and inserted, when needed, between the base header and the data. This simplifies and speeds up the routing process
- ***New options*.** IPv6 has new options to allow for additional functionalities.
- ***Allowance for extension*.** IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- ***Support for resource allocation*.** In IPv6, the type-of-service field has been removed, but two new fields, **traffic class and flow label**, have been added to enable the source to request special handling of the packet.
- ***Support for more security*. The encryption and authentication options** in IPv6 provide confidentiality and integrity of the packet.

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

Packet Format

Figure 20.15 *IPv6 datagram header and payload*





b. Base header

- ***Version*.** The 4-bit version field defines the version number of the IP. For IPv6, the value is 6.
- ***Traffic class*.** The 8-bit traffic class field is used to **distinguish different payloads with different delivery requirements**. It replaces the *type-of-service* field in IPv4.
- ***Flow label*.** The flow label is a 20-bit field that is designed to **provide special handling for a particular flow of data.**
- ***Payload length*.** The 2-byte payload length field defines the length of the IP datagram excluding the header.
- ***Next header*.** The **next header** is an 8-bit field defining the type of the first extension header (if present) or the type of the data that follows the base header in the datagram.
- ***Hop limit*.** The 8-bit hop limit field serves the same purpose as the TTL field in IPv4.
- ***Source and destination addresses*.** The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram. The destination address field is a 16-byte (128-bit) Internet address that identifies the destination of the datagram.
- ***Payload*.** Compared to IPv4, the payload field in IPv6 has a different format and meaning

Figure 20.16 *Format of an IPv6 datagram*

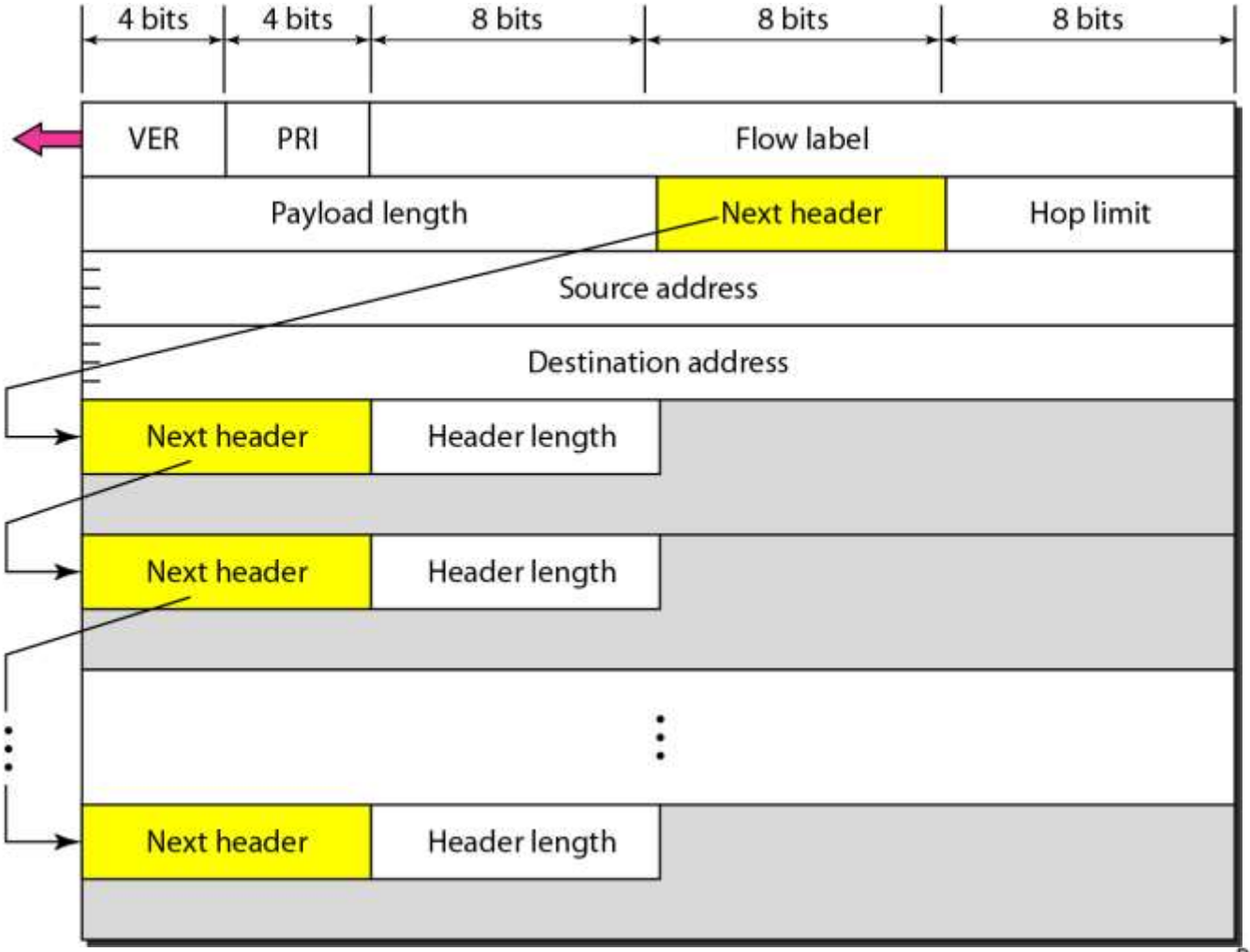
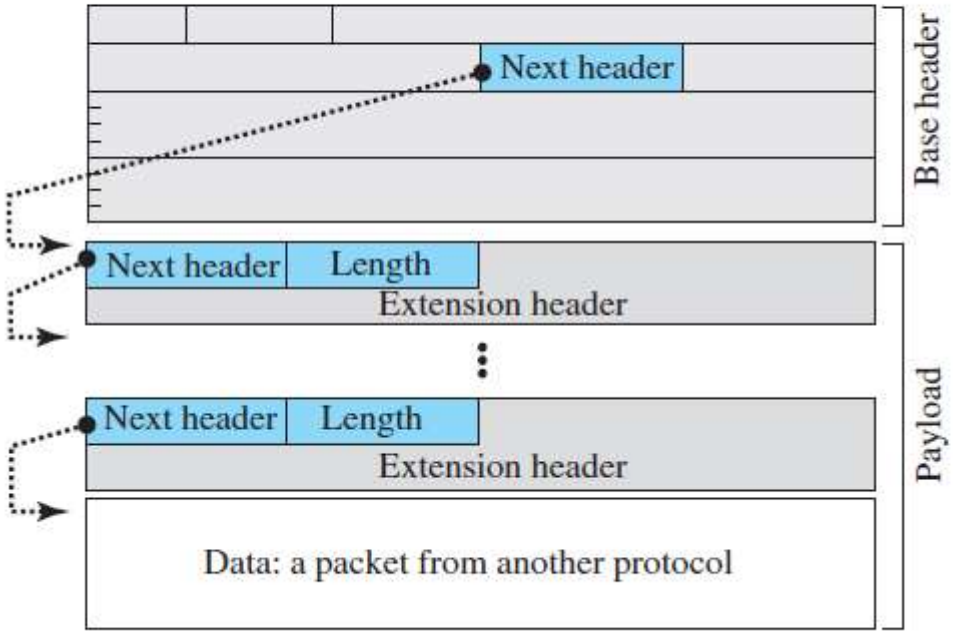


Figure 22.7 *Payload in an IPv6 datagram*



Some next-header codes

00: Hop-by-hop option
02: ICMPv6
06: TCP
17: UDP
43: Source-routing option
44: Fragmentation option
50: Encrypted security payload
51:Authentication header
59: Null (no next header)
60: Destination option

Table 20.9 *Comparison between IPv4 and IPv6 packet headers*

Comparison
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.

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.

Topics discussed in this section:

Dual Stack
Tunneling
Header Translation

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

Figure 20.18 *Three transition strategies*

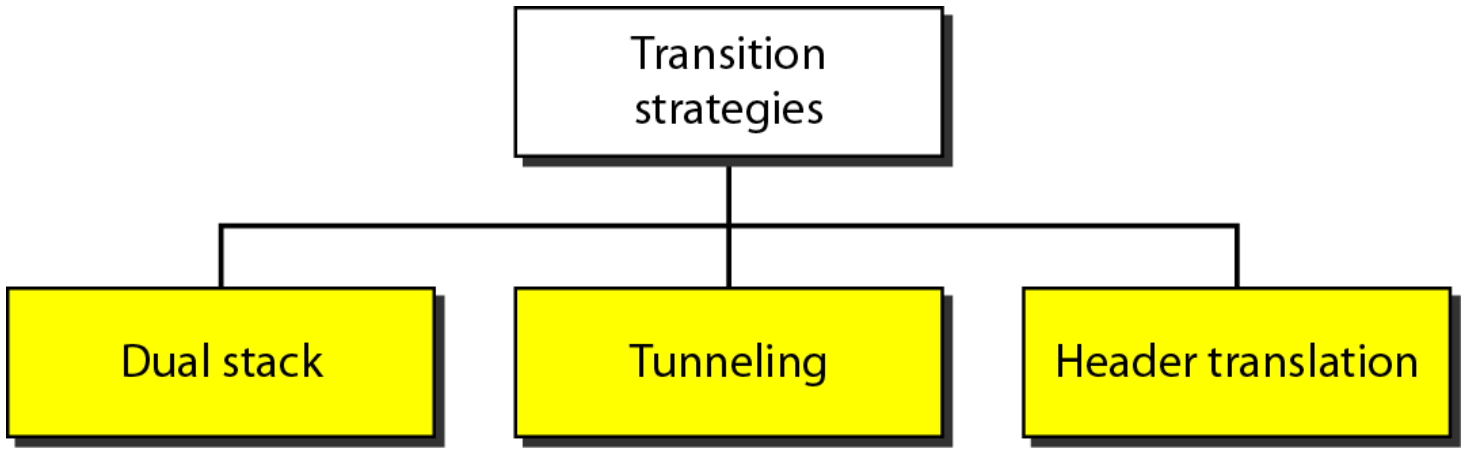
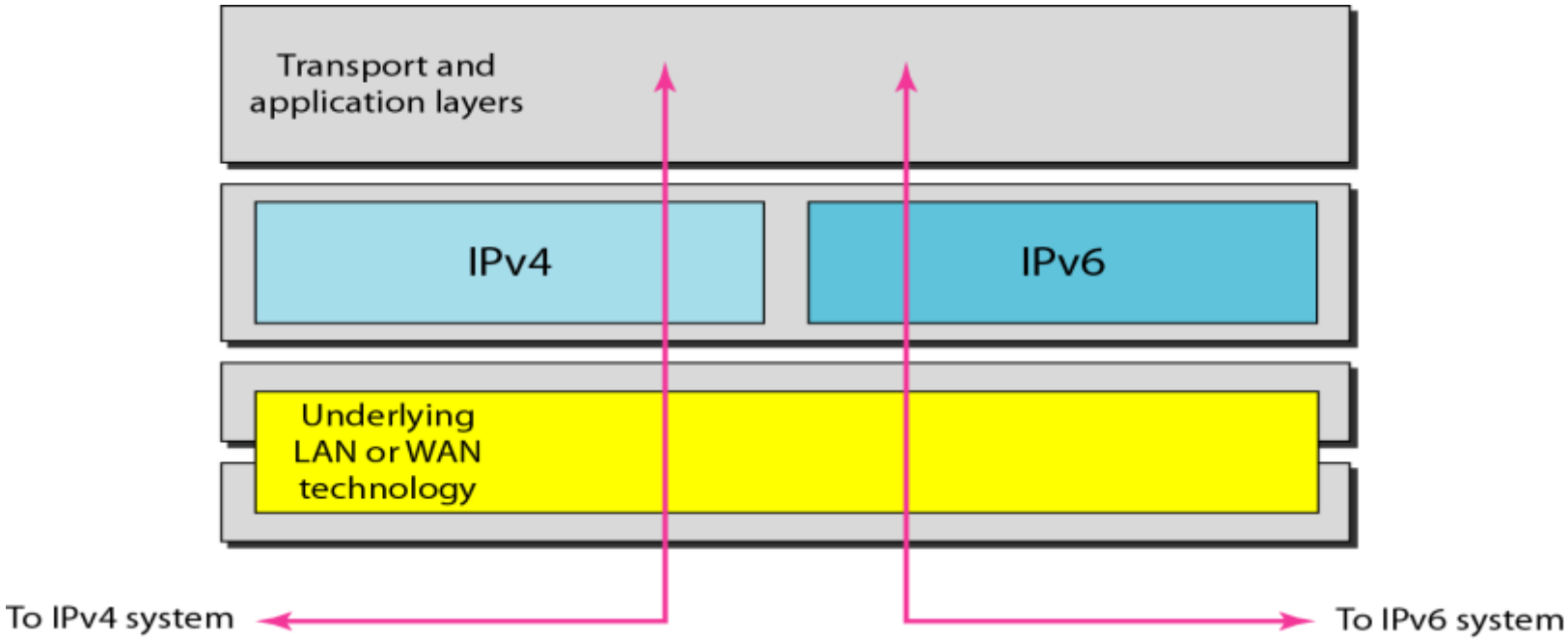
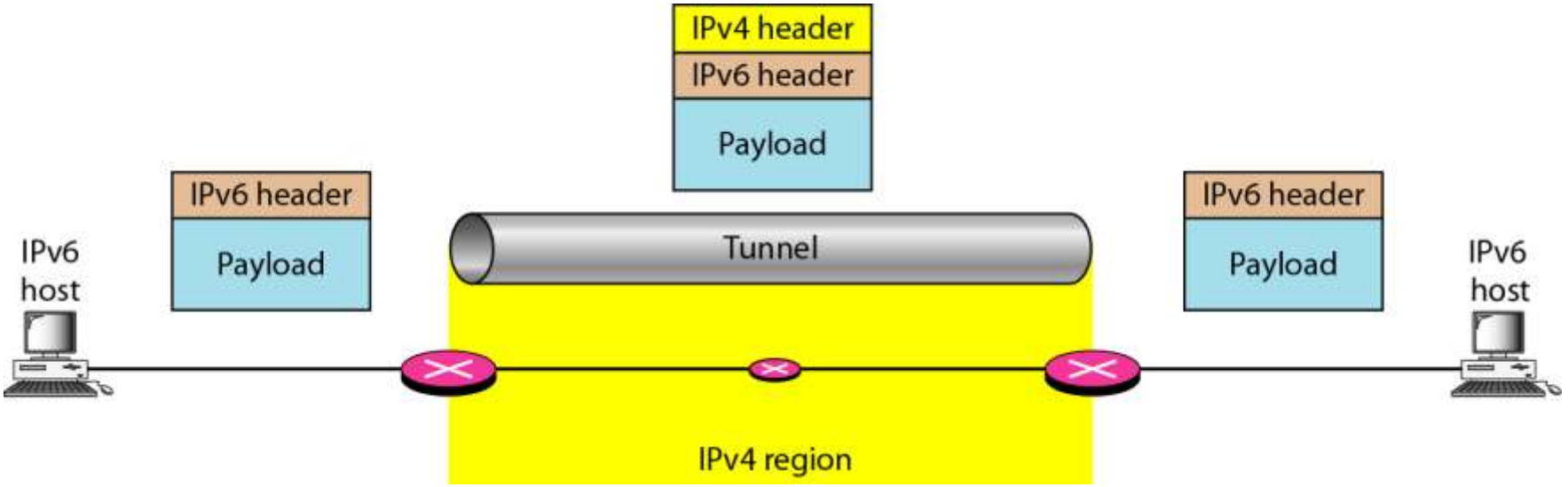


Figure 20.19 *Dual stack*



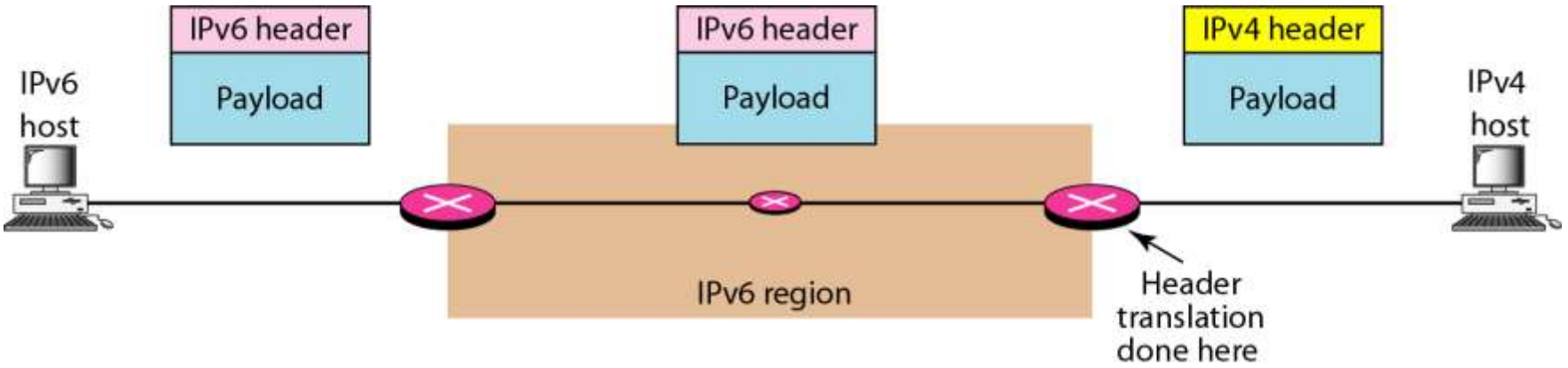
Concept: Devices run both IPv4 and IPv6 simultaneously.
How it works: The system decides which protocol to use based on the destination address.
Advantage: Full compatibility with both IPv4 and IPv6 networks

Figure 20.20 *Tunneling strategy*



Concept: IPv6 packets are encapsulated within IPv4 packets.
How it works: Used when two IPv6 networks communicate over an IPv4-only infrastructure.
Advantage: Allows IPv6 traffic to traverse IPv4 networks without modification.

Figure 20.21 *Header translation strategy*



Concept: Converts IPv6 headers to IPv4 headers (and vice versa).
How it works: Translation occurs at routers or gateways to allow communication between IPv4 and IPv6 hosts.
Advantage: Enables interoperability between IPv4-only and IPv6-only systems.

Table 20.11 *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.

ADDRESS MAPPING

The delivery of a packet to a host or a router requires two levels of addressing: logical and physical. We need to be able to map a logical address to its corresponding physical address and vice versa. This can be done by using either static or dynamic mapping.

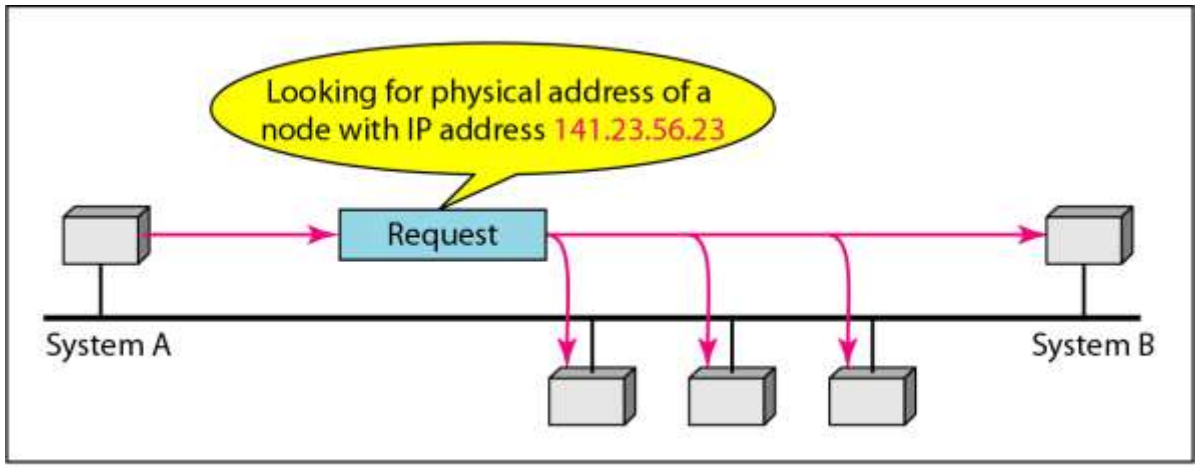
Topics discussed in this section:

Mapping Logical to Physical Address
Mapping Physical to Logical Address

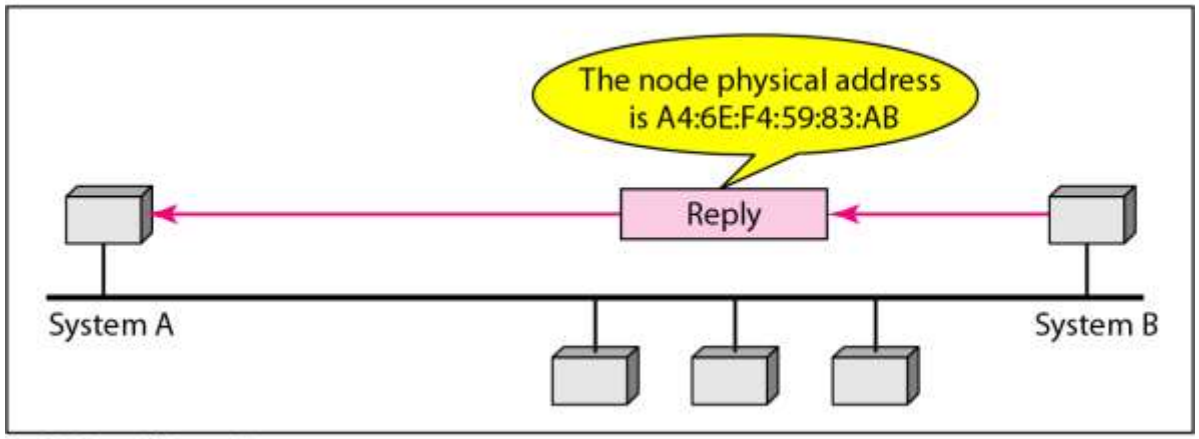
Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

Mapping Logical to Physical Address

Figure 21.1 *ARP operation*

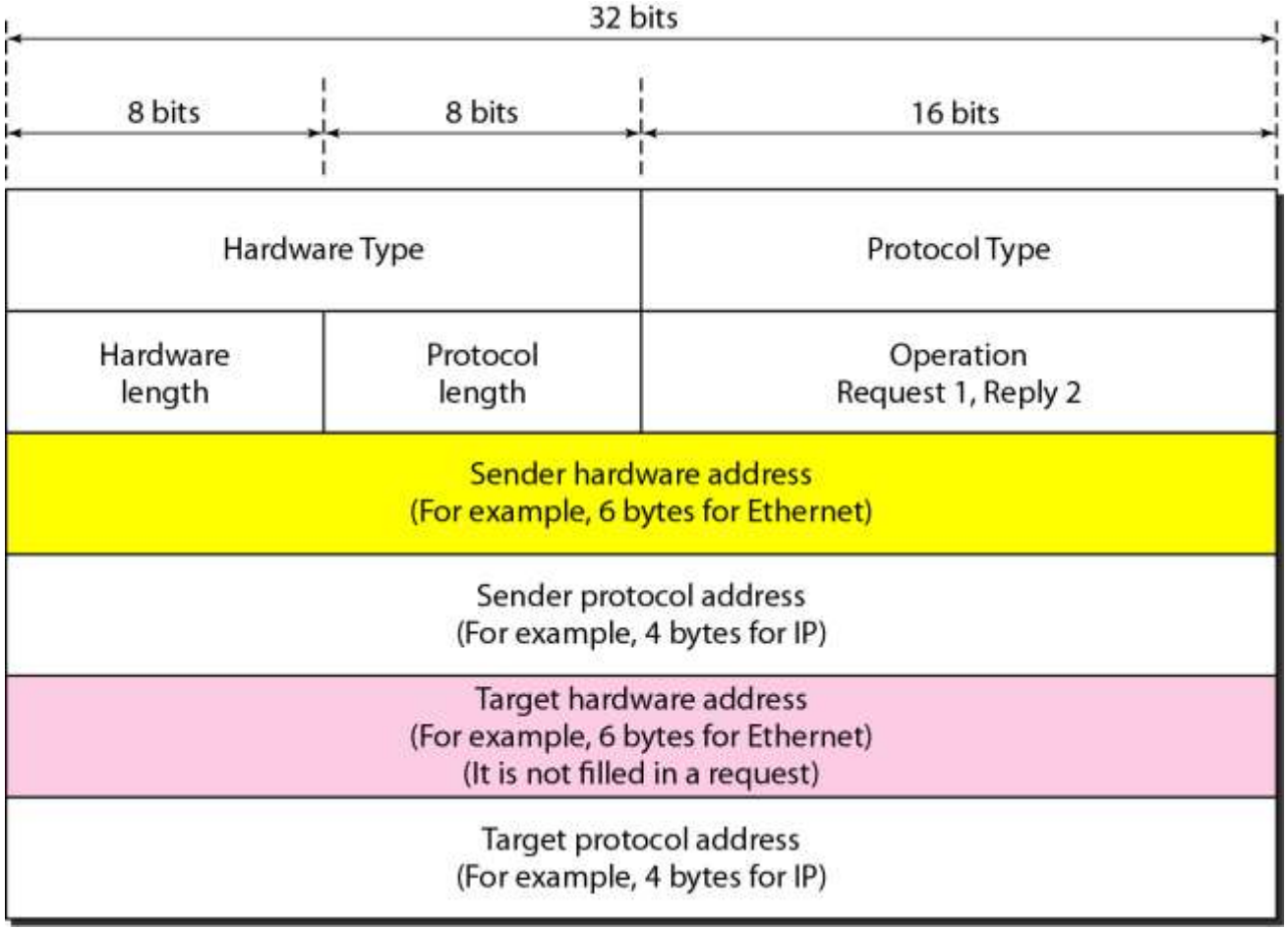


a. ARP request is broadcast



b. ARP reply is unicast

Figure 21.2 *ARP packet*



The fields arc as follows:

o Hardware type. This is a 16-bit field defining the type of the network on which ARP is running.

o Protocol type. This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 080016

o Hardware length. This is an 8-bit field defining the length of the physical address in bytes

o Protocol length. This is an 8-bit field defining the length of the logical address in bytes.

o Operation. This is a 16-bit field defining the type of packet. Two packet types are defined: ARP request (1) and ARP reply (2).

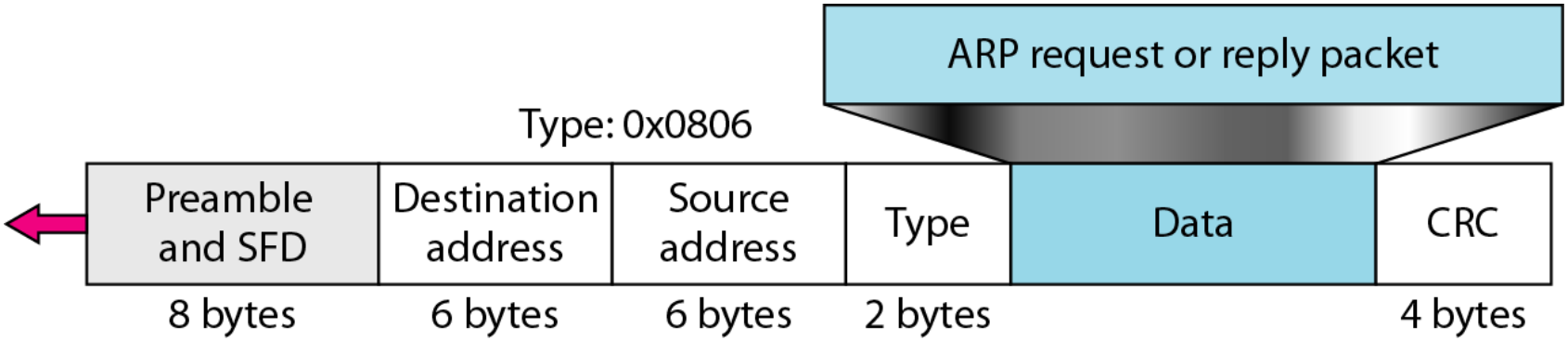
o Sender hardware address. This is a variable-length field defining the physical address of the sender. For example.

o Sender protocol address. This is a variable-length field defining the logical (for example, IP) address of the sender.

o Target hardware address. This is a variable-length field defining the physical address of the target.

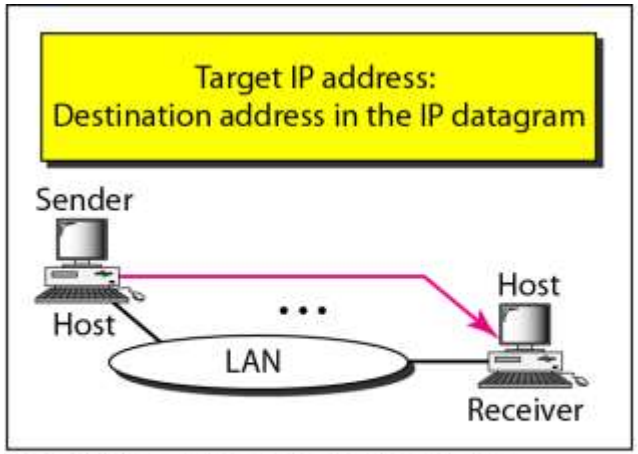
o Target protocol address. This is a variable-length field defining the logical (for example, IP) address of the target.

Figure 21.3 *Encapsulation of ARP (0X0806) packet*

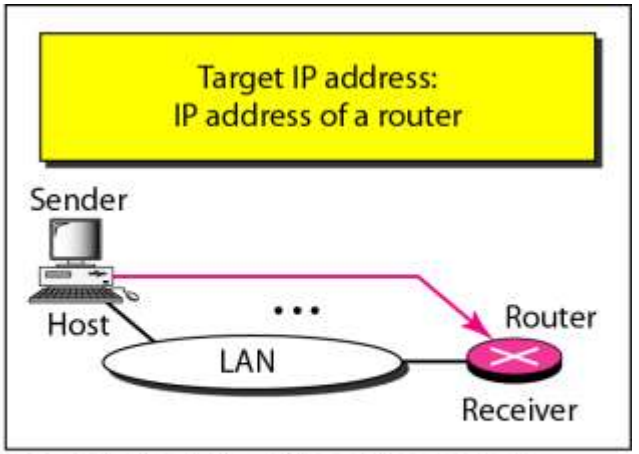


1. The sender knows the IP address of the target.
2. IP asks ARP to create an ARP request message, filling in the sender physical address, the sender IP address, and the target IP address. **The target physical address field is filled with Os.**
3. The message is passed to the data link layer where it is encapsulated in a frame by using the physical address of the sender as the source address and the physical broadcast address as the destination address.
4. Every host or router receives the frame. Because the frame contains a broadcast destination address, all stations remove the message and pass it to ARP. All machines except the one targeted drop the packet. The target machine recognizes its IP address.
5. The target machine replies with an ARP reply message that contains its physical address. The message is unicast.
6. The sender receives the reply message. It now knows the physical address of the target machine.

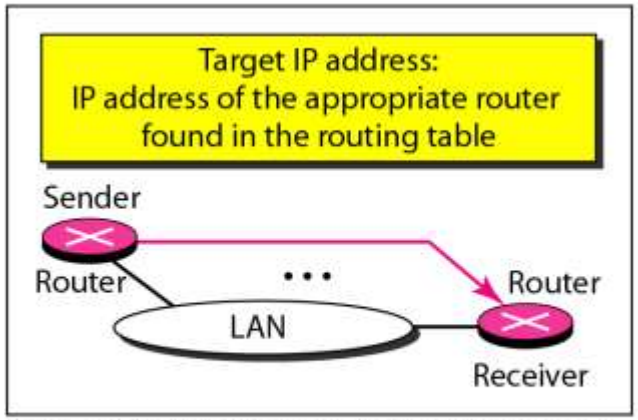
Figure 21.4 *Four cases using ARP*



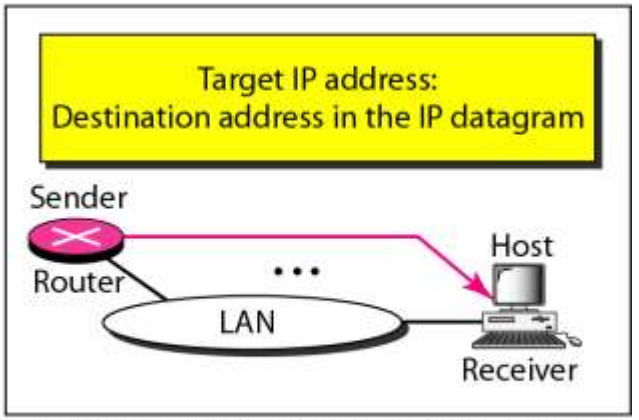
Case 1. A host has a packet to send to another host on the same network.



Case 2. A host wants to send a packet to another host on another network. It must first be delivered to a router.



Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.



Case 4. A router receives a packet to be sent to a host on the same network.

1. **Host to Host (Same Network):**
   - **The sender and receiver are on the same LAN.**
   - **The sender needs the MAC address of the destination host.**
   - **So, it maps the destination IP address → destination MAC address using ARP.**
2. **Host to Host (Different Network):**
   - **The sender wants to send data to a host on another network.**
   - **It checks the routing table or default gateway.**
   - **The router's IP address (next hop) is used for mapping.**
   - **So, it maps the router's IP address → router's MAC address.**
3. **Router to Router (Different Networks):**
   - **A router receives a packet meant for another network.**
   - **It looks up the next router's IP address in its routing table.**
   - **Then it maps that next router's IP address → next router's MAC address.**
4. **Router to Host (Same Network):**
   - **A router receives a packet meant for a host on its own network.**
   - **It maps the destination host's IP address → destination host's MAC address.**

Example 21.1

A host with IP address 130.23.43.20 and physical address B2:34:55:10:22:10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB. The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

Solution

Figure 21.5 shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses.

Figure 21.5 *Example 21.1, an ARP request and reply*

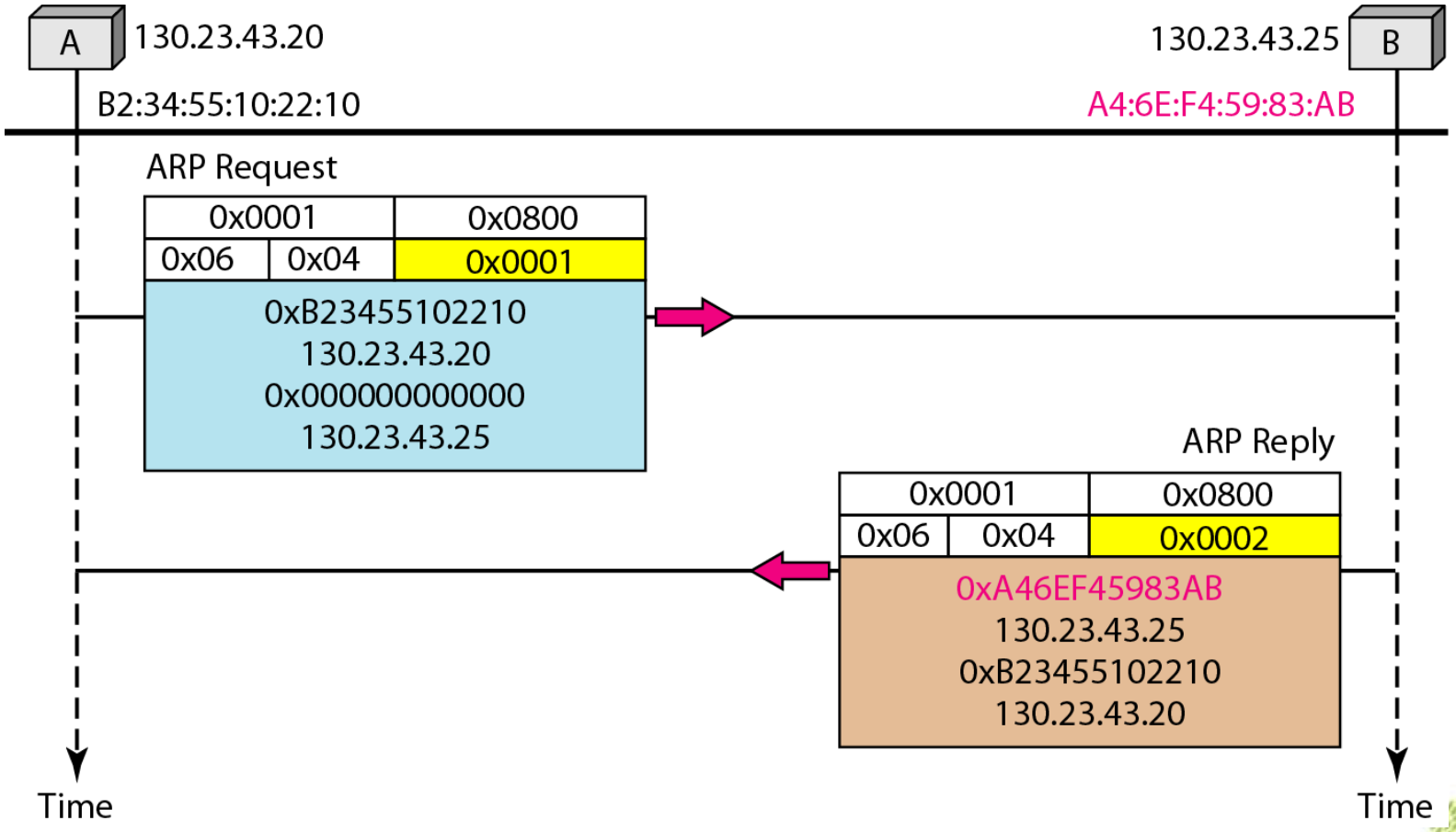
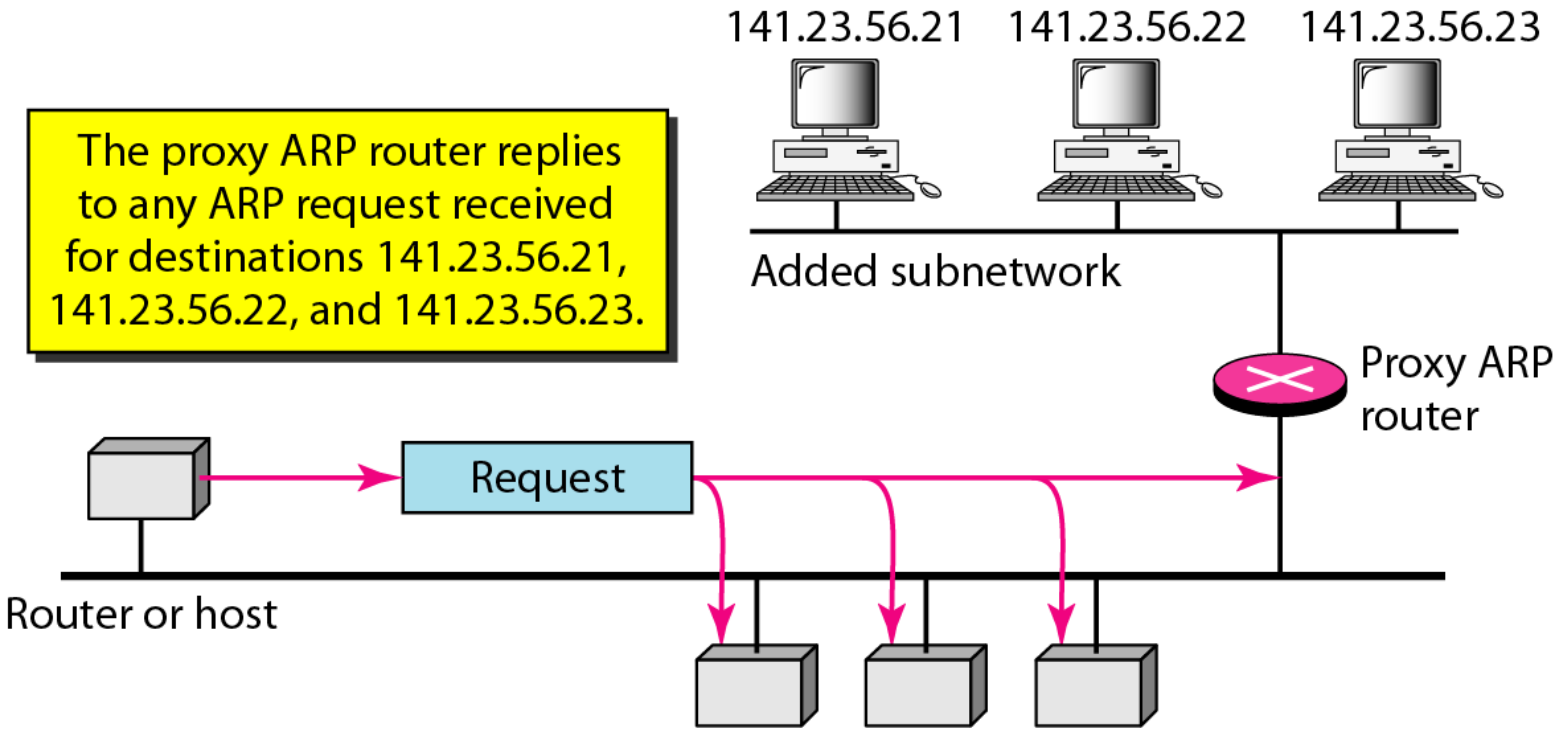


Figure 21.6 *Proxy ARP*



Mapping Physical to Logical Address: RARP(Reverse Address Resolution Protocol)

- **RARP is used to find the IP address (logical address) of a machine when it only knows its MAC address (physical address).**
- **It basically performs the reverse function of ARP.**
- **Normally, a computer reads its IP address from a configuration file stored on its disk.**
- **But a diskless machine (like some old workstations or network devices) cannot store configuration files.**
- **Such a machine only knows its physical address (from the network card).**
- **So, it sends a RARP request to ask:**

  "This is my MAC address — what is my IP address?"

How It Works

1. **The diskless machine sends a RARP request (broadcast) on the local network containing its MAC address.**
2. **A RARP server on the same network looks up its table of known mappings.**
3. **The server replies with the corresponding IP address.**
4. **The client (diskless machine) now knows its own IP and can start communication.**

Limitation of RARP

- **RARP works only within a single network because its requests use data link layer broadcasts, which cannot cross routers or subnets.**
- **It has been replaced by BOOTP and DHCP, which can:**
  - **Work across networks,**
  - **Provide dynamic IP assignment, and**

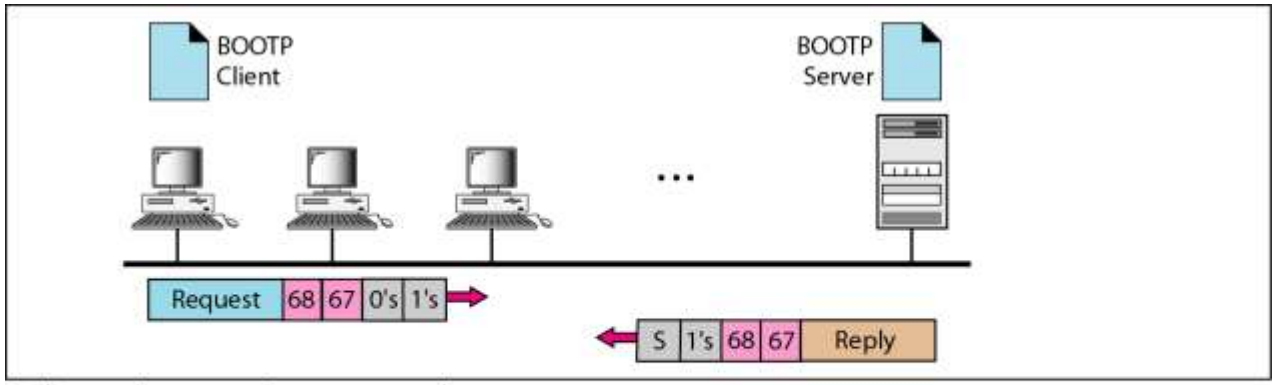
Mapping Physical to Logical Address: Bootstrap Protocol

- **Client Initialization:**
  - **A new device (e.g., diskless workstation) starts up and broadcasts a BOOTP request (BOOTREQUEST) message on the network.**
  - **The message includes the device's MAC address (its physical hardware address).**
- **Server Response:**
  - **A BOOTP server listens for requests and, upon receiving one, checks its configuration table.**
  - **This table maps each MAC address to a corresponding pre-assigned IP address.**

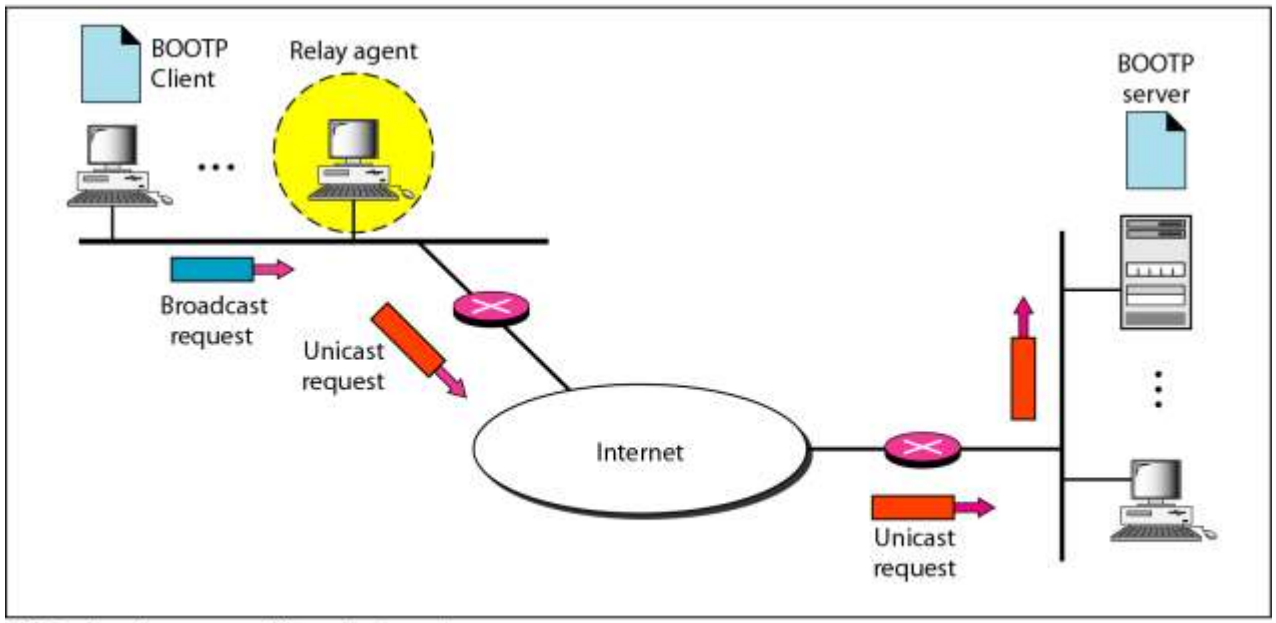
- **Assignment:**

  The server replies with a BOOTREPLY message containing:
  - **The assigned IP address (logical address),**
  - **The gateway address,**
  - **The subnet mask, and**
  - **The address of the boot file (for loading the OS or software).**
- **Client Configuration: The client uses this information to configure its network interface.**

Figure 21.7 *BOOTP client and server on the same and different networks*



a. Client and server on the same network



b. Client and server on different networks

Limitation of BOOTP:

- **Assigns a fixed IP address to a device based on its MAC address.**
- **The BOOTP server stores a table that permanently links each device's physical address (MAC) with its logical address (IP).**
- **The address mapping is static (fixed).**
- **If a device moves to another network, it cannot get a new IP automatically.**
- **The administrator must manually update the mapping**

Mapping Physical to Logical Address: DHCP (Dynamic Host Configuration Protocol)

- **Automatically assigns IP addresses — both static and dynamic.**
- **How it works:**
  - **DHCP maintains two databases:**
    1. **Static database: Permanent IP-to-MAC bindings (like BOOTP).**
    2. **Dynamic database: A pool of available IPs for temporary allocation.**
  - **The DHCP server "leases" an IP address for a limited time.**
  - **When the lease expires, the client must:**
    - **Renew the lease, or**
    - **Stop using the IP address (if renewal denied).**
  - **This allows efficient reuse of IP addresses**

Type	Description	Use Case
Static Allocation	Fixed IP address linked to a device's MAC address (manual entry).	For servers, printers, routers (permanent devices).
Dynamic Allocation	IP address temporarily assigned from the pool.	For mobile/laptop users, or devices that move between networks.

Error Reporting Using ICMP (Internet Control Message Protocol)

- **The Internet Control Message Protocol (ICMP) is used by network devices—such as routers and hosts—to send error messages and diagnostic information about network problems in the IP layer.**
- **ICMP helps in reporting errors that occur during packet delivery.**
- **It does not correct the error but simply informs the sender so that corrective action can be taken.**

Error Type	ICMP Message Type	Meaning
Destination Unreachable	Type 3	Packet couldn't reach its destination (e.g., host, network, or port unreachable).
Source Quench *(obsolete)*	Type 4	Informed sender to slow down transmission due to congestion.
Time Exceeded	Type 11	TTL (Time To Live) expired before reaching destination.
Parameter Problem	Type 12	Error in the header field of the IP packet.
Redirect Message	Type 5	Informs sender about a better route to the destination.