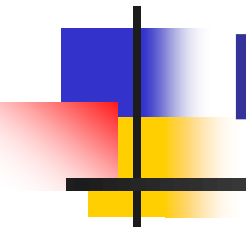


Network Layer

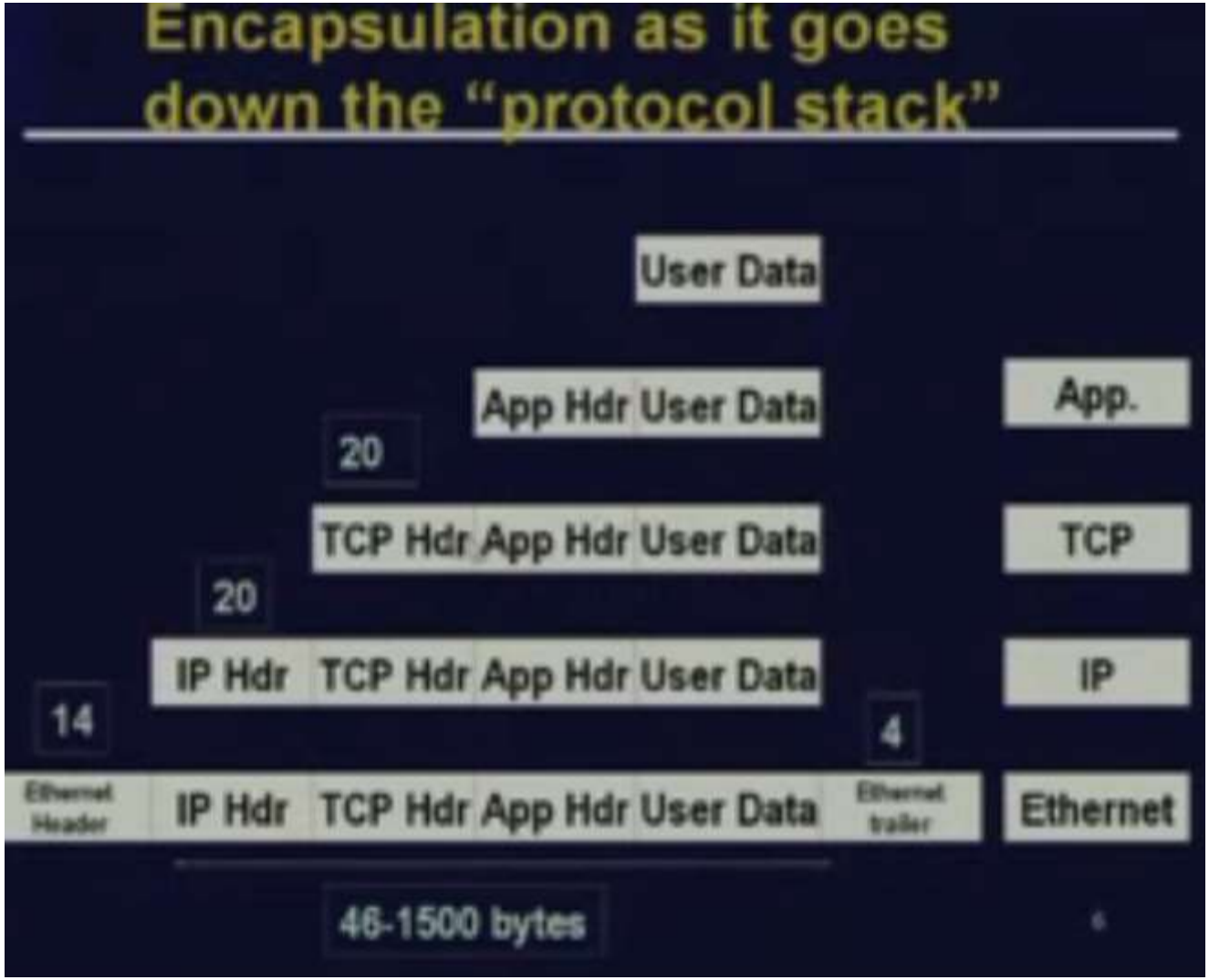
T Nishitha

- The network layer is concerned with getting packets from the source all the way to the destination.
- Getting to the destination may require making many hops at intermediate routers along the way.
- This function clearly contrasts with that of the data link layer, which has the more modest goal of just moving frames from one end of a wire to the other.
- Thus, the network layer is the lowest layer that deals with end-to-end transmission.

- Communication at the network layer is host-to-host(computer-to-computer); a computer somewhere in the world needs to communicate with another computer somewhere else in the world.
- Usually, computers communicate through the Internet.
- The packet transmitted by the sending computer may pass through several LANs or WANs before reaching the destination computer.
- For this level of communication, we need a global addressing scheme; we called this logical addressing. Today, we use the term IP address to mean a logical address in the network layer of the TCP/IP protocol suite.

- IP (Internet Protocol) is designed to connect networks that are
  - ✓ possibly managed by multiple organizations/people
  - ✓ May have different physical connections
  - ✓ May be connected via a sequence of arbitrary intermediaries

- Single protocol at network level insures packets will get from source to destination while allowing for flexibility


Encapsulation as it goes down the "protocol stack"
User Data
App Hdr User Data
App.
20
TCP Hdr App Hdr User Data
TCP
20
IP Hdr TCP Hdr App Hdr User Data
IP
14
4
Ethernet Header
IP Hdr TCP Hdr App Hdr User Data
Ethernet trailer
Ethernet
46-1500 bytes

- IPv4 is a “ **best effort connectionless** “ protocol(intermediate nodes will try to deliver packet to destination as best as possible. If its not possible, it will drop the packet)
- Eg router may drop the packets if the buffer is full
- It’s a datagram packet oriented protocol
- You can get an IP packet from anyone without any ‘setup’ or ‘connection establishment’
- Packets are normally routed using destination routing. You specify where packet is to go, not how it gets there.
- Each packet is routed independently
- ✓ Can be delivered out of order
- ✓ Might not be delivered at all

- When the network layer at the sending host receives a segment from the transport layer, it encapsulates the segment within an IP datagram, writes the destination address of the host (as well as other fields) on the datagram, and drops the datagram into the network.
- This process is similar to a person writing a letter, inserting the letter in an envelope, writing the destination address on the envelope, and dropping the envelope into a mailbox.
- Neither the Internet's network layer nor the postal service make any kind of preliminary contact with the destination before moving its "parcel" to the destination

IP Address

- The IP address is a 32-bit address
- ✓ Identifies the network and the host on a given network
- ✓ Divided into two parts- first part identifies the network, second part identifies the host on the network
- ✓ The format is not the same for each address

Three Address Types:

- Unicast Communication-destined for a single host
- Broadcast Communication- destined for all hosts on a network
- Multicast Communication- destined for a set of hosts that belong to a multicast group

19-1 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.

Note

An IPv4 address is 32 bits long.

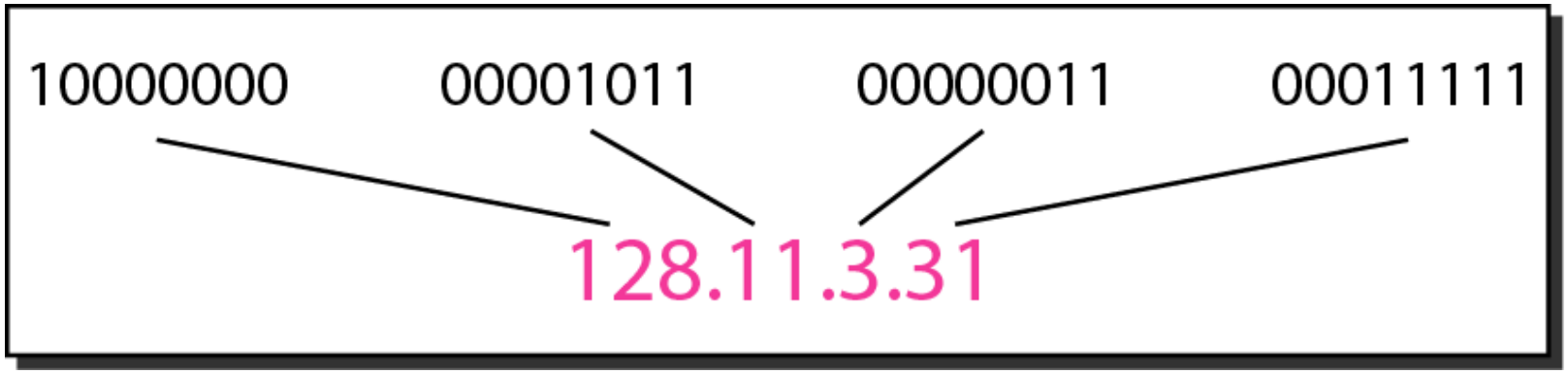
Note

The IPv4 addresses are unique and universal.

Note

The address space of IPv4 is 2^{32} or 4,294,967,296.

Figure 19.1 *Dotted-decimal notation and binary notation for an IPv4 address*



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.*

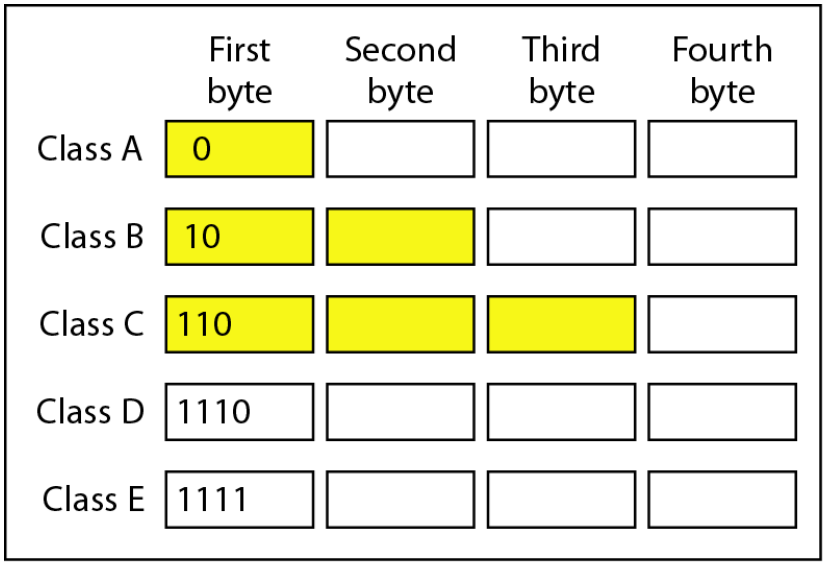
Classful Addressing:

- IPv4 addressing, at its inception, used the concept of classes. This architecture is called classful addressing.

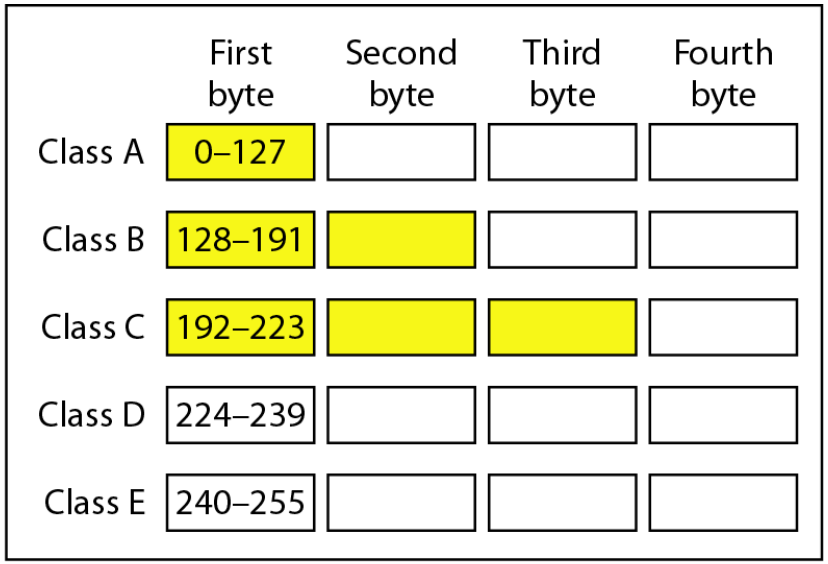
Note

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

Figure *Finding the classes in binary and dotted-decimal notation*



a. Binary notation



b. Dotted-decimal notation

Example 19.4

Find the class of each address.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

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.

c. The first byte is 14; the class is A.

d. The first byte is 252; the class is E.

Table 19.1 *Number of blocks and block size in classful IPv4 addressing*

Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Note

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

Table 19.2 *Default masks for classful addressing*

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

Class A

- Used for small number of networks and large number of hosts
- First byte (8 bits) represent the network address
- Last 3 bytes (24 bits) represent the host address
- Class A address have a first bit of 0
- Class A network addresses range from 0 to 126

Class B

- Provide an equal number of hosts and networks
- First two bytes are network address and last two bytes are host address
- First two bits of class B address are 10
- Network addresses range from 128 to 191

- Class C
  - ✓ Greater number of network addresses and fewer number of host addresses
  - ✓ First three bits are 110
  - ✓ Network addresses range from 192-223
- Class D
  - ✓ Used for special multicast addresses
  - ✓ First four bits are 1110
- Class E
  - ✓ Used for experimental purposes
  - ✓ First four bits are 1111

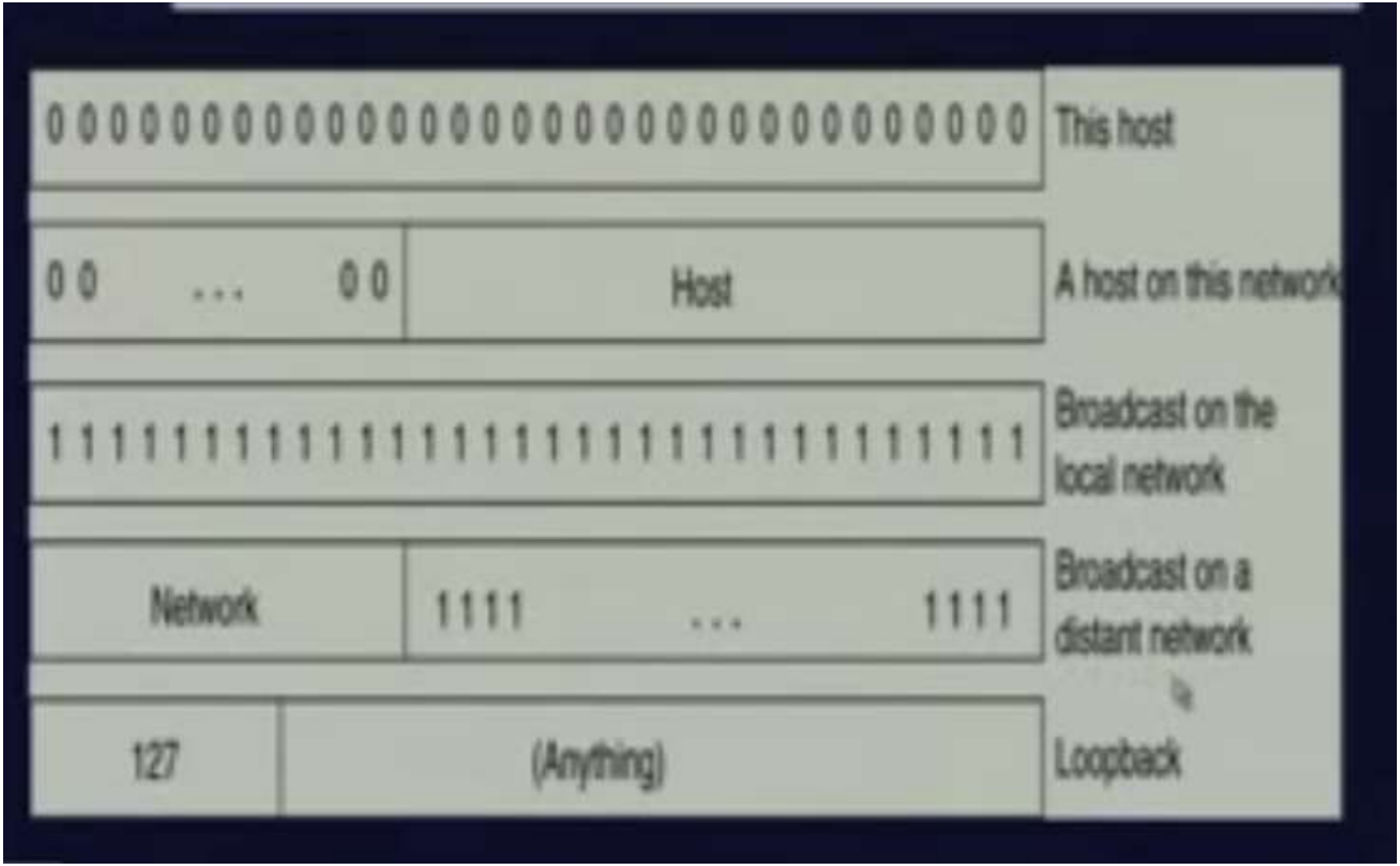
Special IP Addresses

- Special source addresses as part of an initializing procedure (e.g bootp)
- ✓ This host on this network

  NET=0, HostID=0
- ✓ Specified host on this network

  NET=0, HostID=this host
- Loopback addresses
- ✓ Loopback address- allows applications on same host to communicate using TCP/IP
- ✓ NetID= 127, HostId=anything

- Limited Broadcast- typically used for initialization.
  - ✓ Only appears on local cable/collision domain(you want to broadcast to the local network)
  - ✓ NetID= -1 and HostID=-1
- Net-directed Broadcast (to NetID)
  - ✓ Forwarded via router
  - ✓ NetID= NetId, HostID= -1
- Subnet-directed Broadcast (to NetID, subnetid)
  - ✓ NetID=NetId, subnetid= subnetid, HostId= -1


0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
This host
0 0 ... 0 0
Host
A host on this network
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Broadcast on the local network
Network
1 1 1 1 ... 1 1 1 1
Broadcast on a distant network
127
(Anything)
Loopback

Problems & Subnets

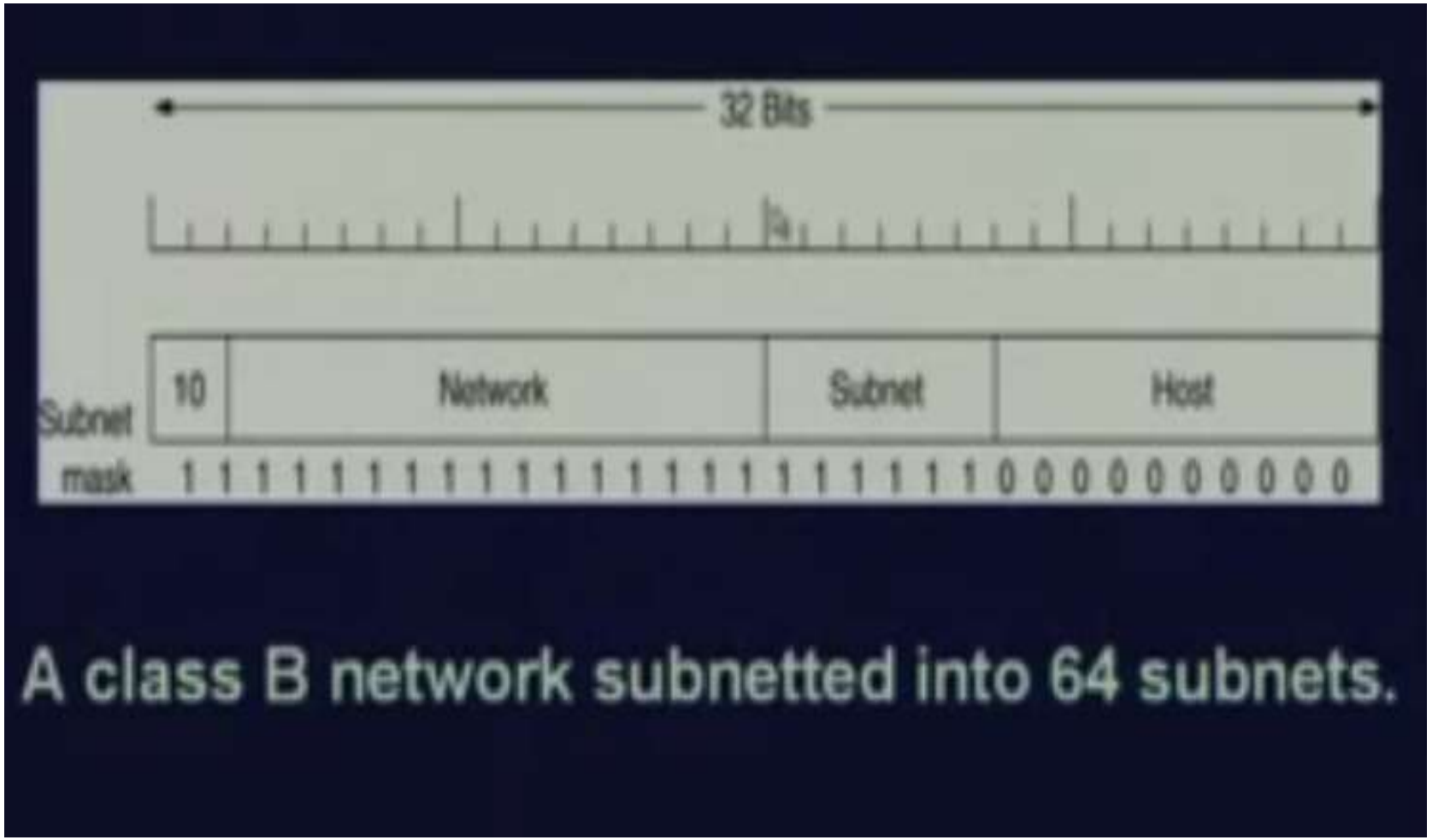
- A few companies got class A networks. Many institutions got class B networks
- Most people get class C address
- Class A is too big and class C is too small
- Allegedly, broadcast would go to an entire network. Obviously impractical for a Class A network, that's 16,777,216 hosts
- Eg: Suppose consider class B network containing 10,000 nodes. If all these 10,000 in one network broadcast IP address asking for MAC address (using ARP) , then traffic will be too much. Since network is too big, we have to break it up into smaller parts ie subnetworks. To do that we need some more bits to identify subnetwork

Subnets:

- Subnets are used to divide a large network into smaller networks
- Each address allows for one network address and many hosts (ie all hosts are on the same network)
- Subnet masks are used to create many subnets within the same network address

Subnet masks:

- An IP address has two components, the network address and the host address
- A subnet mask separates the IP address into the network address and host address.
- Subnetting further divides the host part of an IP address into a subnet and host address
- It is called a subnet mask because it is used to identify network address of an IP address by performing bitwise AND operation on the netmask
- Subnet mask is made by setting network bits to all 1's and setting host bits to all 0's



A class B network subnetted into 64 subnets.

Subnet Example #1

IP Address : 144.97.16.132

Subnet Mask : 255.255.255.192

11111111 11111111 11111111 11000000

10000010 01100001 00010000 10000100

Network

10000010 01100001 00010000 10000000

144.97.16.128

Host

00000000 00000000 00000000 00000100

4

- As it is class B, first two bytes defines network address
- Then 8+2 bits represents the subnetwork. Beyond that we represent bits with 0’s.

Subnet Example #2

IP Address : 144.97.17.132

Subnet Mask : 255.255.254.0

11111111 11111111 11111110 00000000

10000010 01100001 00010001 10000100

Network

10000010 01100001 00010000 00000000

144.97.16.

Host

00000000 00000000 00000001 10000100

1.132

Supernetting

- The time came when most of the class A and class B addresses were depleted; however,there was still a huge demand for midsize blocks.
- The size of a class C block with a maximum number of 256 addresses did not satisfy the needs of most organizations.
- Even a midsize organization needed more addresses. One solution was supernetting.

- In supernetting, an organization can combine several class C blocks to create a larger range of addresses. In other words, several networks are combined to create a supernetwork or a supemet.
- An organization can apply for a set of class C blocks instead of just one. For example, an organization that needs 1000 addresses can be granted four contiguous class C blocks. The organization can then use these addresses to create one supernetwork.
- Supernetting decreases the number of Is in the mask. For example, if an organization is given four class C addresses, the mask changes from /24 to /22

Note

Classful addressing, which is almost obsolete, is replaced with classless addressing.

Classless Addressing

- To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented.
- In this scheme, there are no classes, but the addresses are still granted in blocks.

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.
- For example, a household may be given only two addresses; a large organization may be given thousands of addresses.
- An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.

Restriction

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

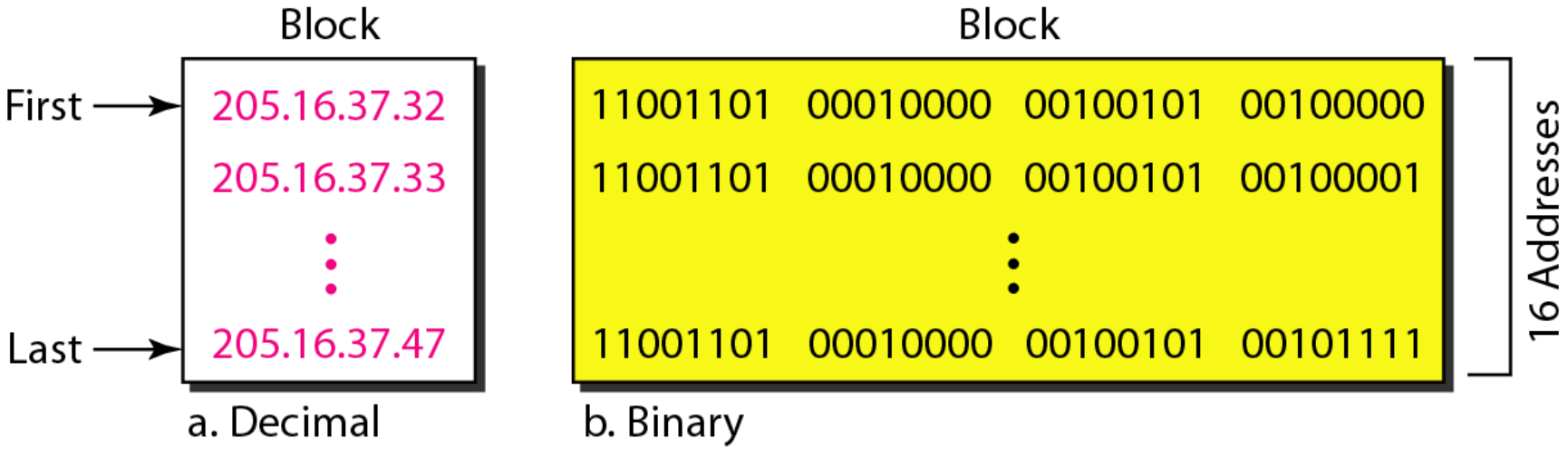
1. The addresses in a block must be contiguous, one after another.
2. The number of addresses in a block must be a power of 2 (I, 2, 4, 8, ...).
3. The first address must be evenly divisible by the number of addresses

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.

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



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.

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 0010000

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}.

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.*

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

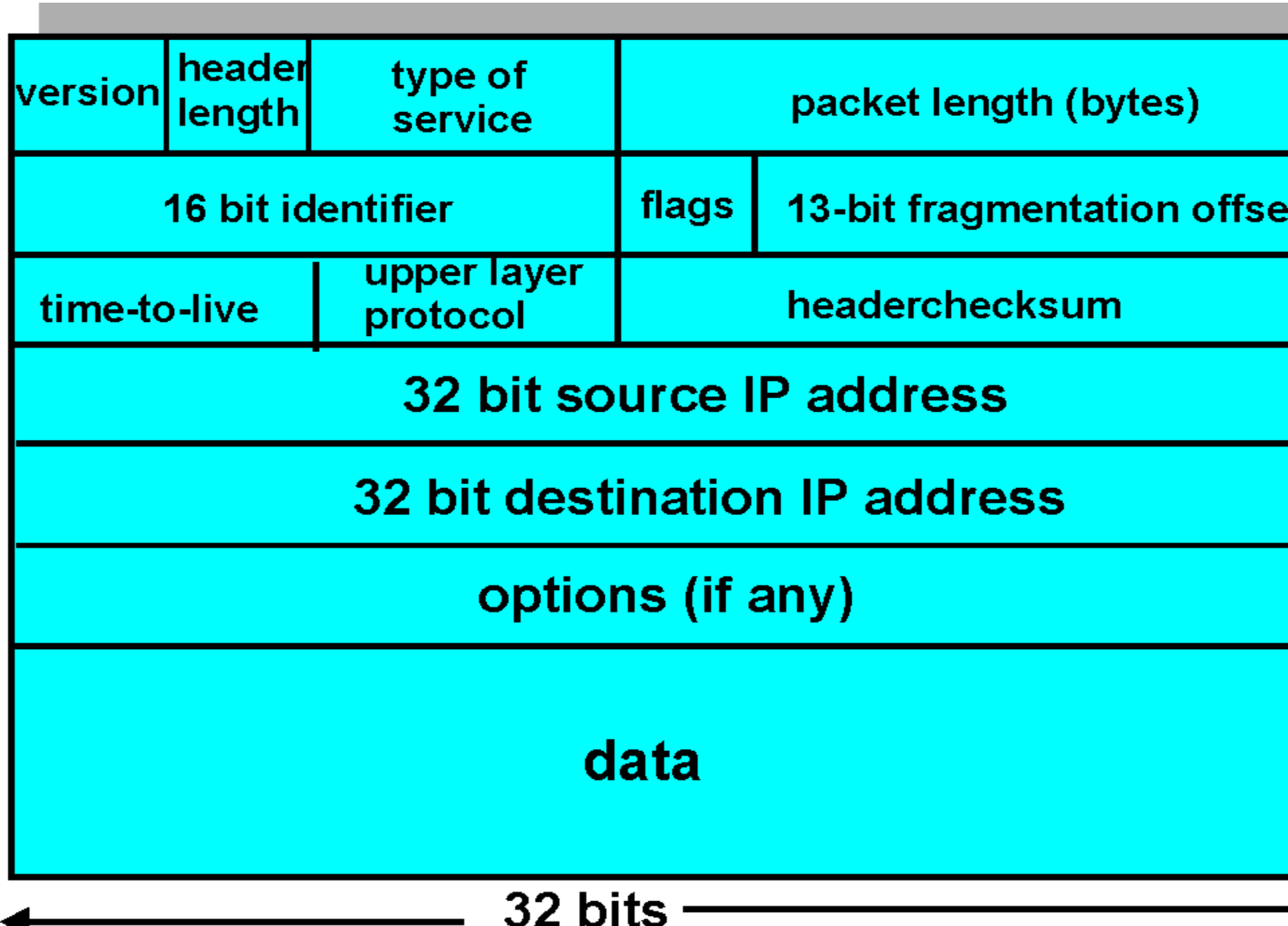
Example 19.9 (continued)

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

Datagram:

- Packets in the IPV4 layer are called datagrams
- A datagram is a variable length packet consisting of two parts:
- ✓ Header
- ✓ Data
- The header is 20 to 60 bytes in length and contains information essential to routing and delivery


version
header length
type of service
packet length (bytes)
16 bit identifier
flags
13-bit fragmentation offse
time-to-live
upper layer protocol
headerchecksum
32 bit source IP address
32 bit destination IP address
options (if any)
data
32 bits

- Version-4 bit field specifying the IP version. Currently 4.
- Header length- specified in 32- bit words. Range is 5..15 words, or 20 – 60 bytes. (as we have options)
- Type of service (8 bits)
- Message length ,in bytes
- Identification-This field is used in fragmentation
- Flags- This field is used in fragmentation.
- Fragmentation offset- This field is used in fragmentation.

- Time to live field- upper limit on the number of hops a message can go before being dropped(to identify packet loop)
- Protocol- identifies TCP,UDP, ICMP, etc
- Header Checksum- checksum of just the TCP/IP header
- Source address
- Destination address
- options

Fragmentation :

- A datagram can travel through different networks. Each router decapsulates the IPv4 datagram from the frame it receives, processes it, and then encapsulates it in another frame.
- The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled.
- The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.
- For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format

Maximum Transfer Unit (MTU)

- Each data link layer protocol has its own frame format in most protocols.
- when a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size, which is defined by the restrictions imposed by the hardware and software used in the network.

- We must divide the datagram to make it possible to pass through these networks. This is called fragmentation.
- The source usually does not fragment the IPv4 packet. The transport layer will instead segment the data into a size that can be accommodated by IPv4 and the data link layer in use.
- When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but with some changed.
- A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU. In other words, a datagram can be fragmented several times before it reaches the final destination.

Fields Related to Fragmentation

The fields that are related to fragmentation and reassembly of an IPv4 datagram are the identification, flags, and fragmentation offset fields.

- Identification-The identification number helps the destination in reassembling the datagram. It knows that all fragments having the same identification value must be assembled into one datagram.
- Flags – 3 bit field
  - ✓ Reserved
  - ✓ Do not fragment
  - ✓ More fragment
- Fragmentation offset- This 13-bit field shows the relative position of this fragment with respect to the whole datagram. It is the offset of the data in the original datagram measured in units of 8 bytes.

Checksum

- The implementation of the checksum in the IPv4 packet follows the same principles.
- First, the value of the checksum field is set to O. Then the entire header is divided into 16-bit sections and added together. The result (sum) is complemented and inserted into the checksum field.
- The checksum in the IPv4 packet covers only the header, not the data.

IPV6 Design Goals:

- IPV4 was very successful ,but the limited addresses pose problems
- Routing information were not inherent in addresses (if there is geographical info inbuilt in it, then routing becomes easier)
- Experience had shown that some aspects of IPV4 are problematic: option headers, fragments

Simplifications for IPV6

- Move to 128-bit addresses
- Assign a fixed format to all headers
- Remove the header checksum
- Use “extension headers” rather than options
- Remove the hop-by-hop segmentation procedure.

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

Note

An IPv6 address is 128 bits long.

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

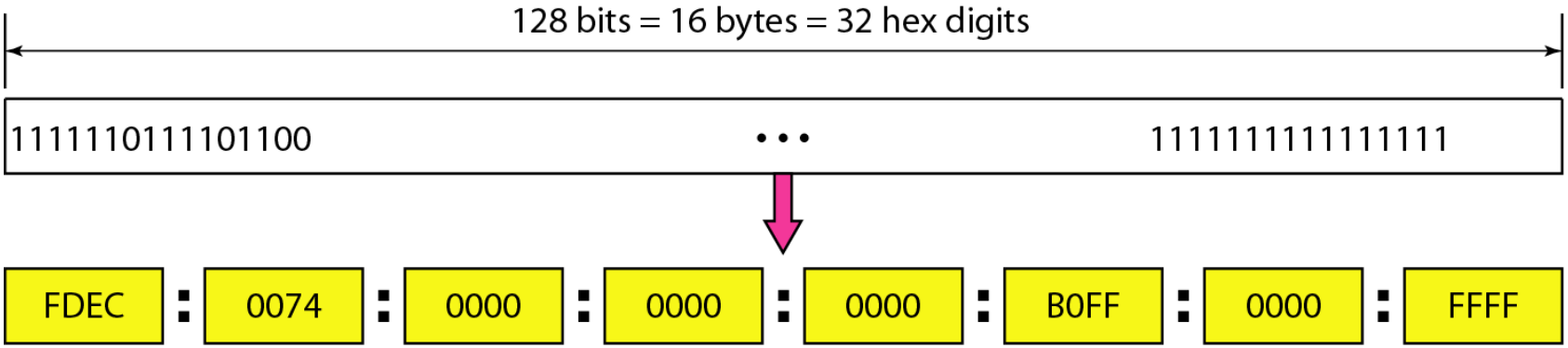
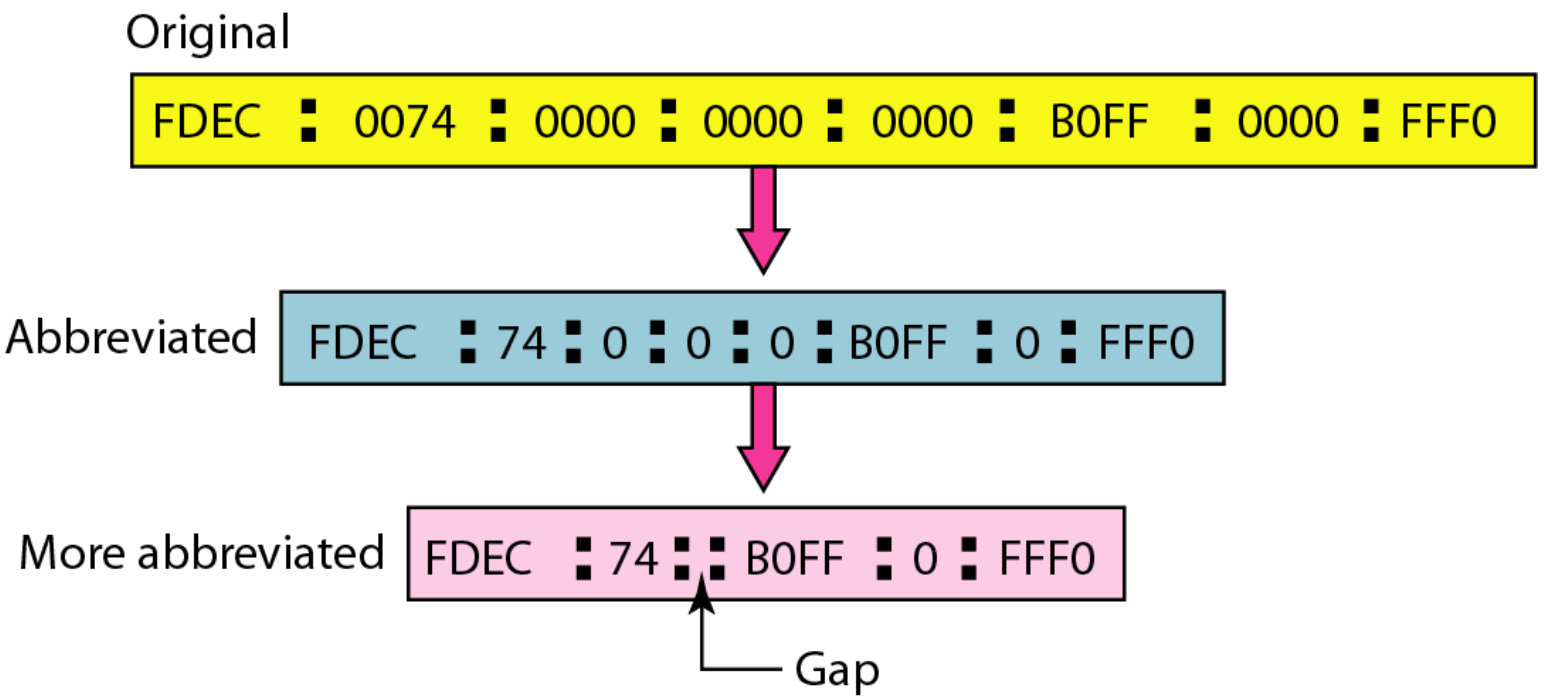


Figure 19.15 *Abbreviated IPv6 addresses*



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
```

Table 19.5 *Type prefixes for IPv6 addresses*

Type Prefix	*Type*	*Fraction*
0000 0000	Reserved	1/256
0000 0001	Unassigned	1/256
0000 001	ISO network addresses	1/128
0000 010	IPX (Novell) network addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Reserved	1/16
001	Reserved	1/8
010	**Provider-based unicast addresses**	**1/8**

Table 19.5 *Type prefixes for IPv6 addresses (continued)*

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

Figure 19.16 *Prefixes for provider-based unicast address*

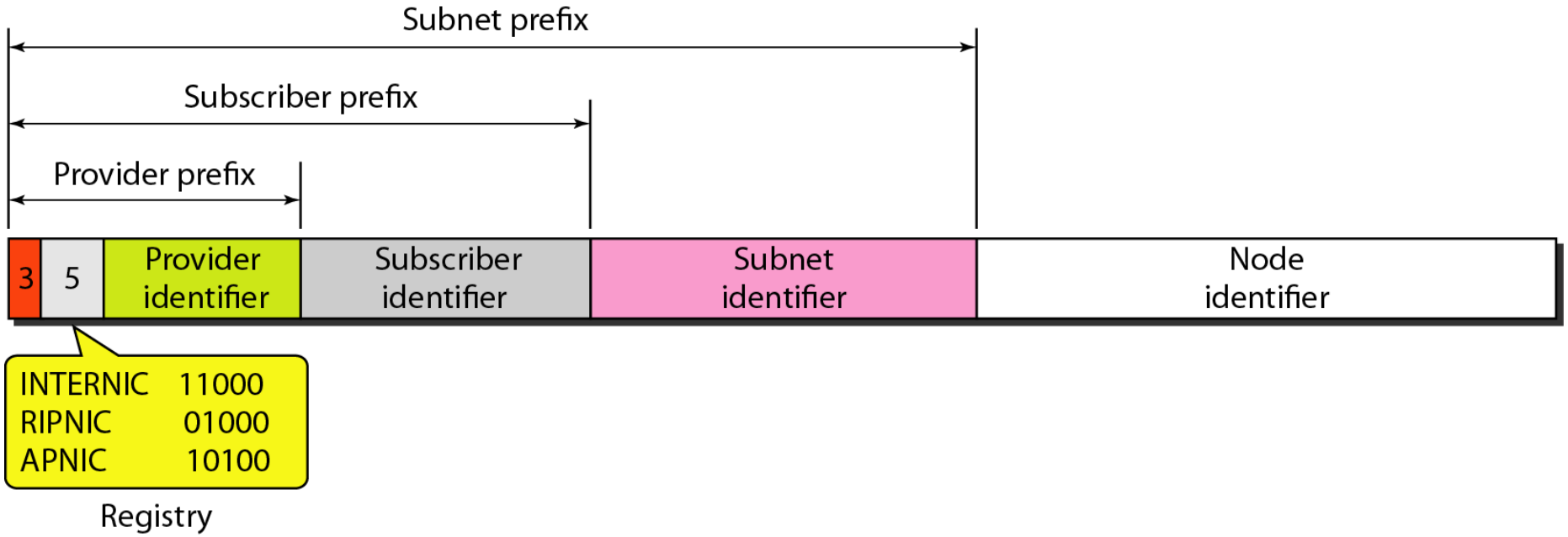


Figure 19.17 *Multicast address in IPv6*

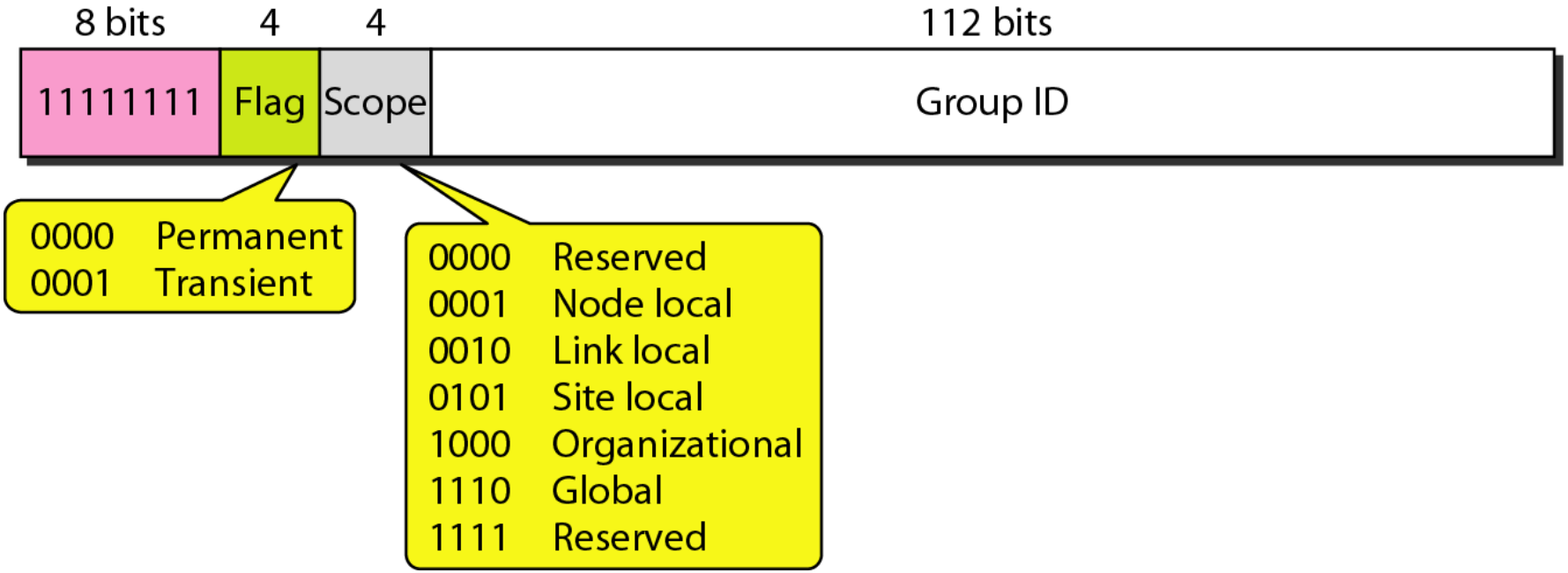


Figure 19.18 *Reserved addresses in IPv6*

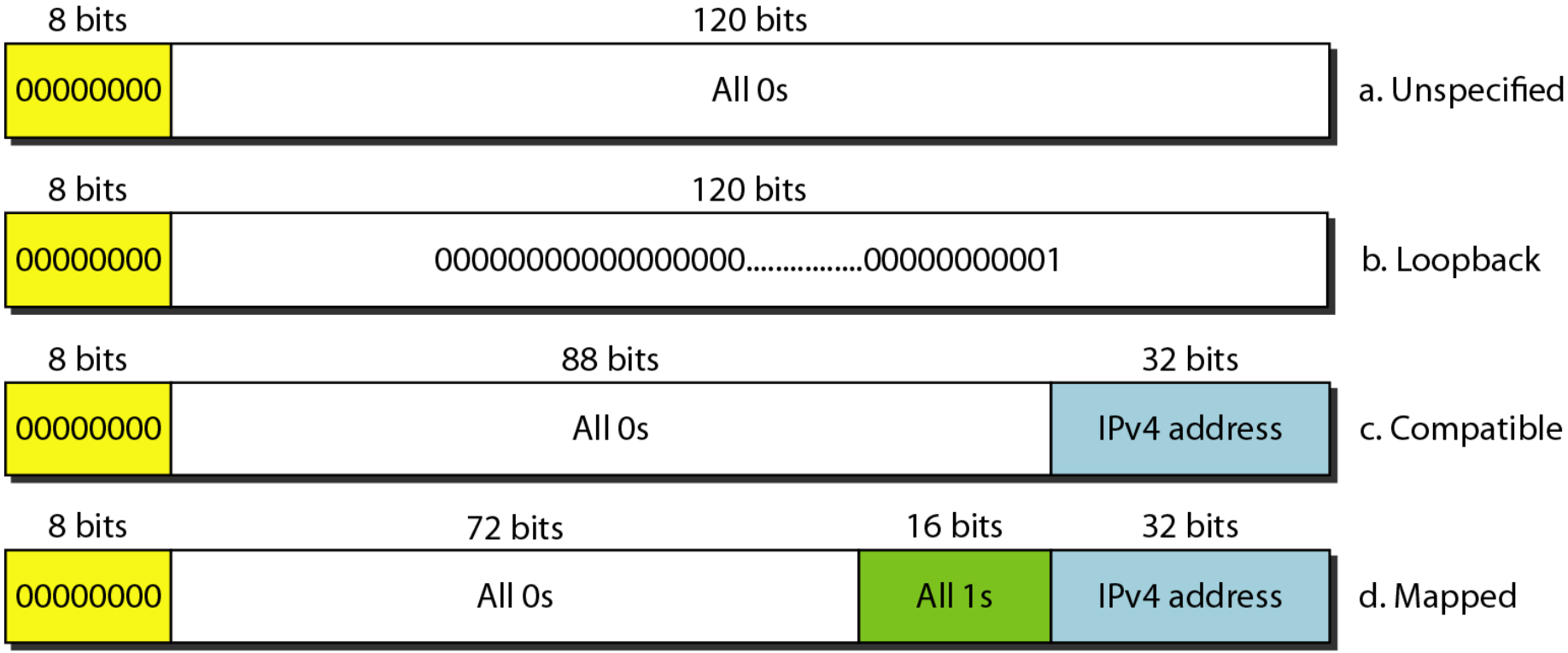


Figure 19.19 *Local addresses in IPv6*