Data Communications and Networking Fourth Edition

Forouzan

Chapter 19

Network Layer: Logical Addressing

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.*

Topics discussed in this section:

Address Space
Notations
Classful Addressing
Classless Addressing
Network Address Translation (NAT)
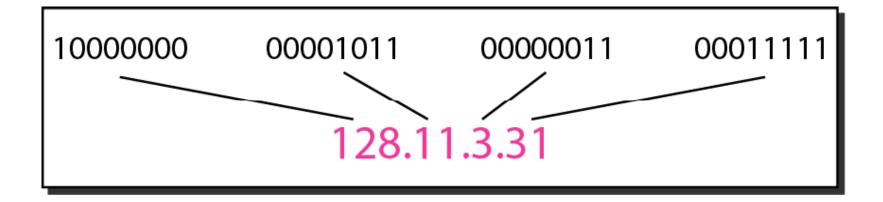
Note

An IPv4 address is 32 bits long.

Note

The IPv4 addresses are unique
and universal.

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

```
10000000       00001011       00000011       00011111
```

128.11.3.31

Note

Numbering systems are reviewed in Appendix B.

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.

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

Figure 19.2 *Finding the classes in binary and dotted-decimal notation*

	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10			
Class C	110			
Class D	1110			
Class E	1111			

a. Binary notation

	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation

Example 19.4

Find the class of each address.
a. **0**0000001 00001011 00001011 11101111
b. **110**00001 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

19.14

Note

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

Table 19.2 *Default masks for classful addressing*

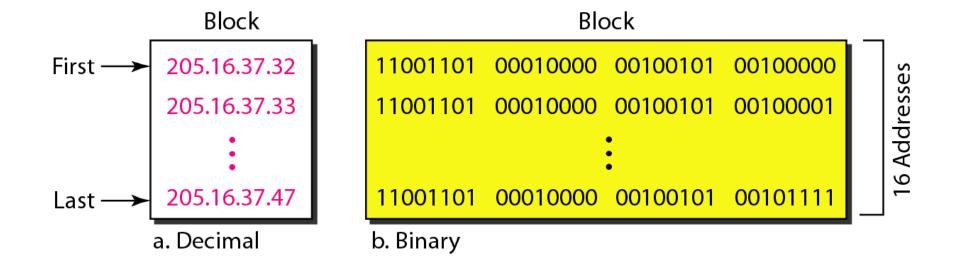| Class	Binary	Dotted-Decimal	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

Note

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

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*

Block

First → 205.16.37.32
205.16.37.33
⋮
Last → 205.16.37.47

a. Decimal

Block

11001101 00010000 00100101 00100000
11001101 00010000 00100101 00100001
⋮
11001101 00010000 00100101 00101111

16 Addresses

b. Binary

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.

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.

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

<div align="center">

11111111 11111111 11111111 11110000

</div>

(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.*

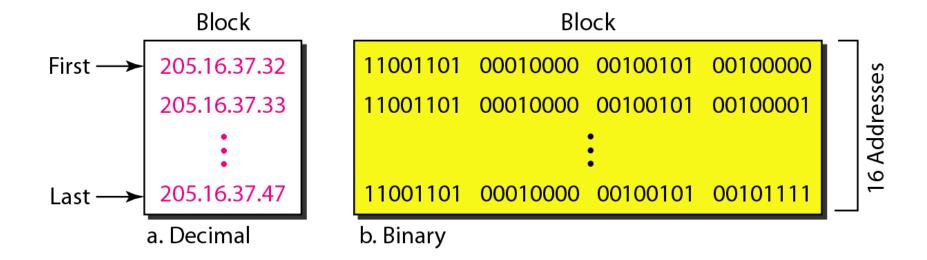| Address:	11001101	00010000	00100101	00100111
Mask complement:	**00000000**	**00000000**	**00000000**	**00001111**
Last address:	11001101	00010000	00100101	00101111

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:	**000000000 0000000 0000000 00001111**
Number of addresses:	15 + 1 = 16
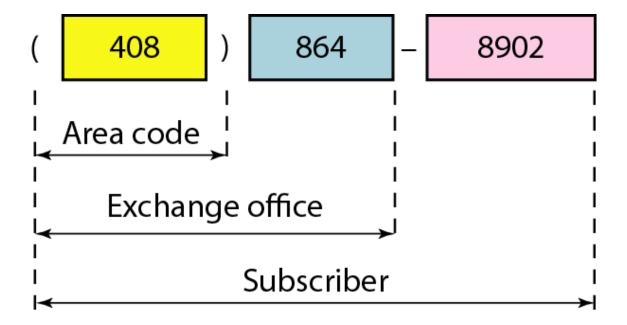
Figure 19.4 *A network configuration for the block 205.16.37.32/28*

Block

First →	205.16.37.32
	205.16.37.33
	⋮
Last →	205.16.37.47

a. Decimal

Block

11001101 00010000 00100101 00100000

11001101 00010000 00100101 00100001

⋮

11001101 00010000 00100101 00101111
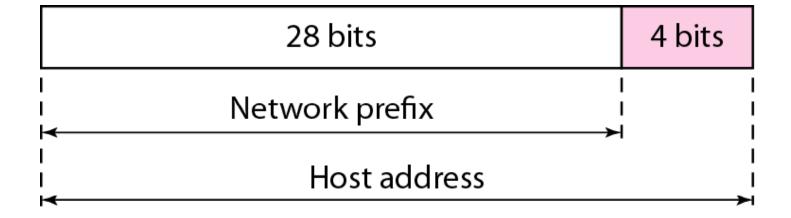
16 Addresses

b. Binary

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.
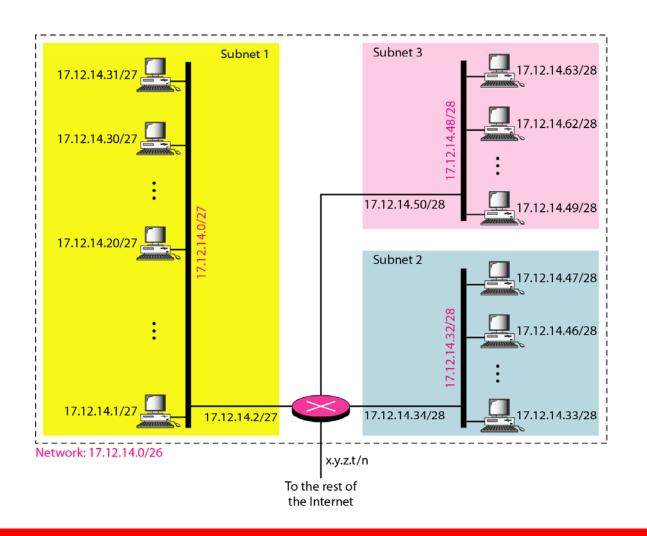
Figure 19.5 *Two levels of hierarchy in an IPv4 address*

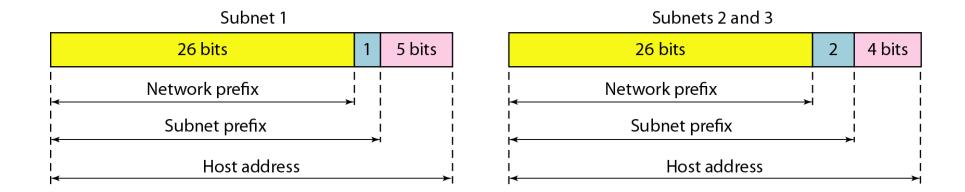# Figure 19.6 *A frame in a character-oriented protocol*

Each address in the block can be
considered as a two-level
hierarchical structure:
the leftmost *n* bits (prefix) define
the network;
the rightmost 32 − n bits define
the host.

Figure 19.7 *Configuration and addresses in a subnetted network*

Subnet 1

17.12.14.31/27

17.12.14.30/27

17.12.14.20/27

17.12.14.0/27

17.12.14.1/27   17.12.14.2/27

Network: 17.12.14.0/26

Subnet 3

17.12.14.63/28

17.12.14.62/28

17.12.14.48/28

17.12.14.50/28   17.12.14.49/28

Subnet 2

17.12.14.47/28

17.12.14.46/28

17.12.14.32/28

17.12.14.34/28   17.12.14.33/28

x.y.z.t/n

To the rest of
the Internet

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

Subnet 1

26 bits	1	5 bits

Network prefix

Subnet prefix

Host address

Subnets 2 and 3

26 bits	2	4 bits

Network prefix

Subnet prefix

Host address

19.37

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
```

19.40

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

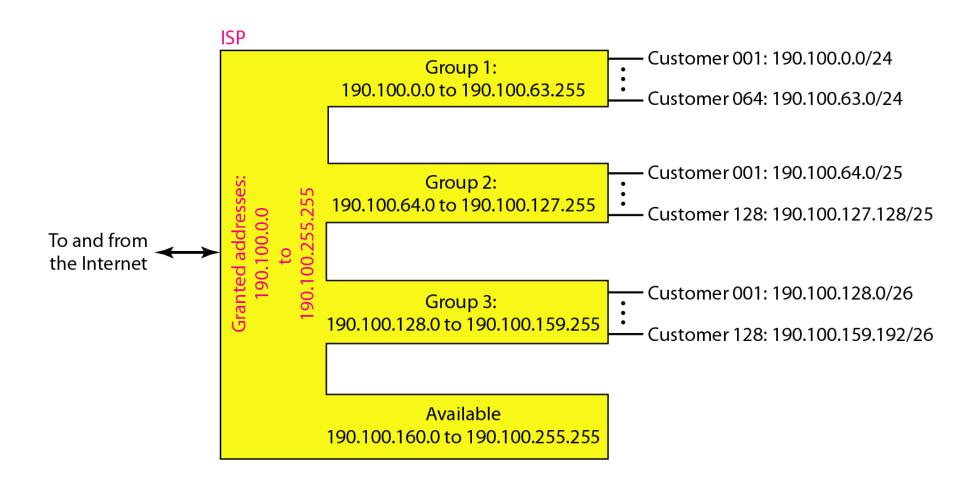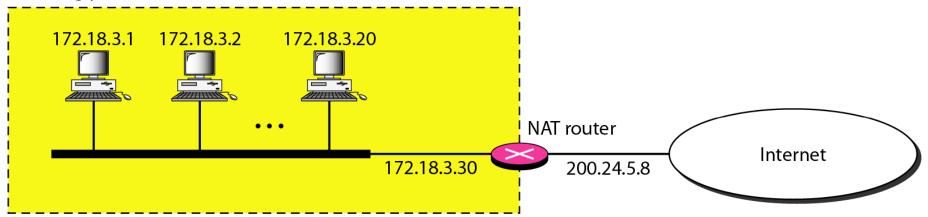**Figure 19.9** *An example of address allocation and distribution by an ISP*

ISP

Granted addresses: 190.100.0.0 to 190.100.255.255

To and from the Internet

Group 1:
190.100.0.0 to 190.100.63.255

Customer 001: 190.100.0.0/24
Customer 064: 190.100.63.0/24

Group 2:
190.100.64.0 to 190.100.127.255

Customer 001: 190.100.64.0/25
Customer 128: 190.100.127.128/25

Group 3:
190.100.128.0 to 190.100.159.255

Customer 001: 190.100.128.0/26
Customer 128: 190.100.159.192/26

Available
190.100.160.0 to 190.100.255.255

Table 19.3 *Addresses for private networks*
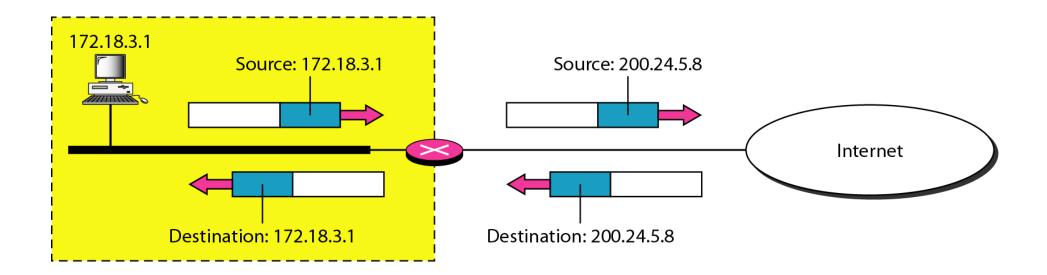
Range			Total
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

19.43

Figure 19.10 *A NAT implementation*

Site using private addresses

172.18.3.1 172.18.3.2 172.18.3.20

• • •

NAT router

172.18.3.30

200.24.5.8

Internet

19.44

Figure 19.11 *Addresses in a NAT*

172.18.3.1

Source: 172.18.3.1

Source: 200.24.5.8

Destination: 172.18.3.1

Destination: 200.24.5.8

Internet

Figure 19.12 *NAT address translation*
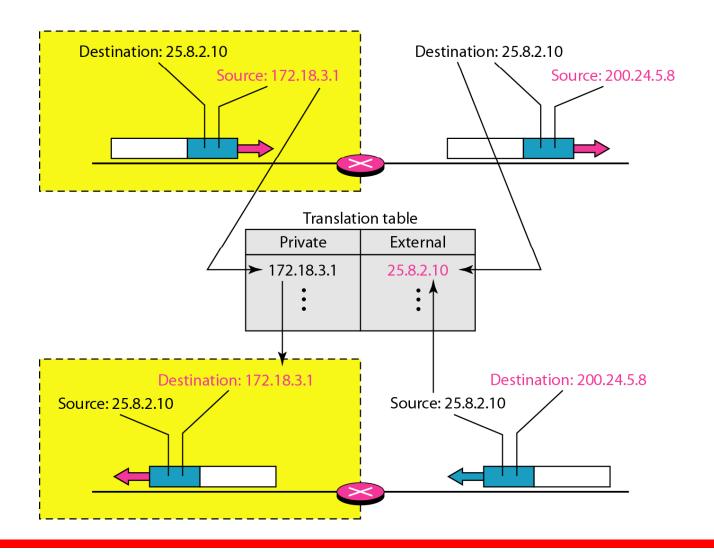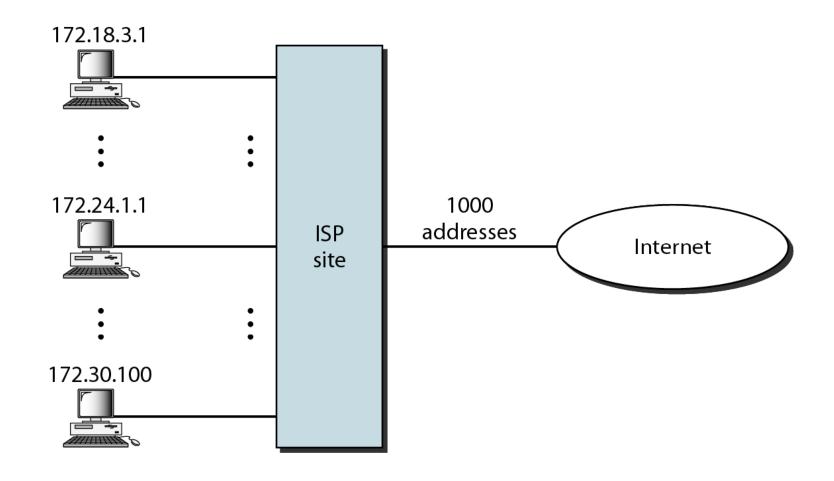
Table 19.4 *Five-column translation table*

Private Address	Private Port	External Address	External Port	Transport Protocol
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
. . .	. . .	. . .	. . .	. . .

19.47

Figure 19.13 *An ISP and NAT*

172.18.3.1

172.24.1.1

172.30.100

ISP
site

1000
addresses

Internet

19.48

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.
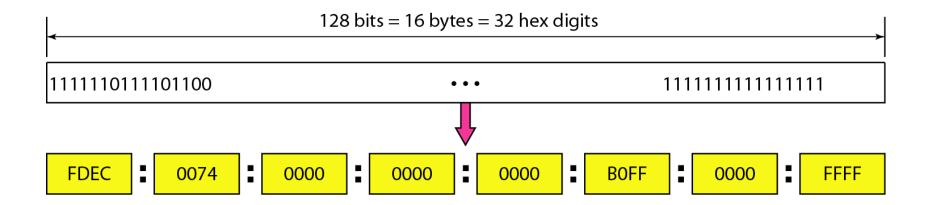
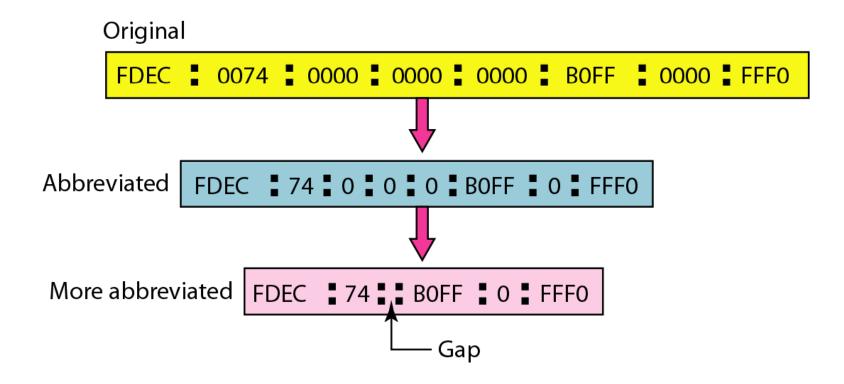**Topics discussed in this section:**

Structure
Address Space

Note

An IPv6 address is 128 bits long.

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

128 bits = 16 bytes = 32 hex digits

| 1111110111101100 | • • • | 1111111111111111 |

| FDEC | : | 0074 | : | 0000 | : | 0000 | : | 0000 | : | B0FF | : | 0000 | : | FFFF |

Figure 19.15 *Abbreviated IPv6 addresses*

Original

| FDEC | 0074 | 0000 | 0000 | 0000 | B0FF | 0000 | FFF0 |

Abbreviated

| FDEC | 74 | 0 | 0 | 0 | B0FF | 0 | FFF0 |

More abbreviated

| FDEC | 74 | | B0FF | 0 | FFF0 |

Gap

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
```
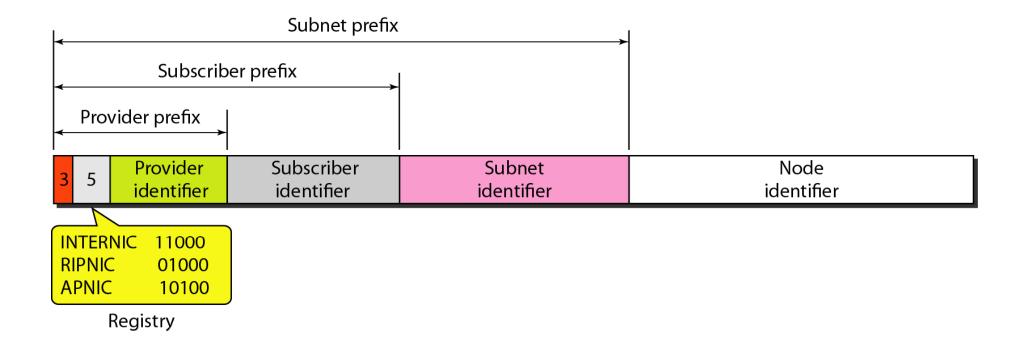
Table 19.5 *Type prefixes for IPv6 addresses*

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**

19.54
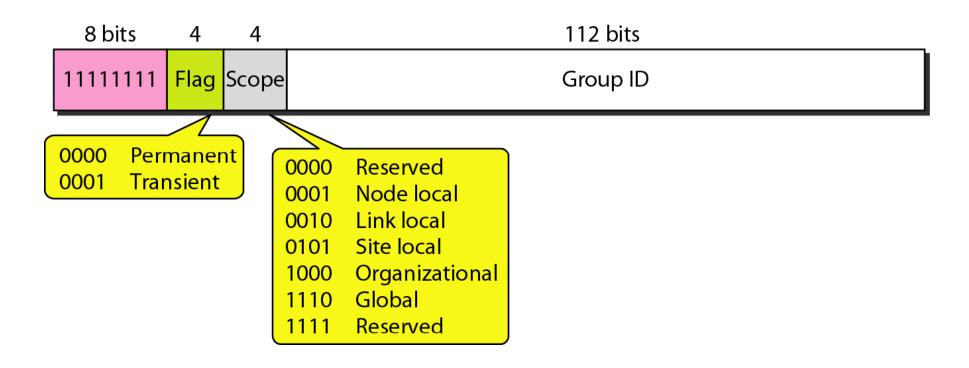
Table 19.5 *Type prefixes for IPv6 addresses (continued)*

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

19.55

Figure 19.16 *Prefixes for provider-based unicast address*

Figure 19.17 *Multicast address in IPv6*

Figure 19.18   *Reserved addresses in IPv6*

8 bits	120 bits	
00000000	All 0s	a. Unspecified

8 bits	120 bits	
00000000	0000000000000000...............00000000001	b. Loopback

8 bits	88 bits	32 bits	
00000000	All 0s	IPv4 address	c. Compatible

8 bits	72 bits	16 bits	32 bits	
00000000	All 0s	All 1s	IPv4 address	d. Mapped
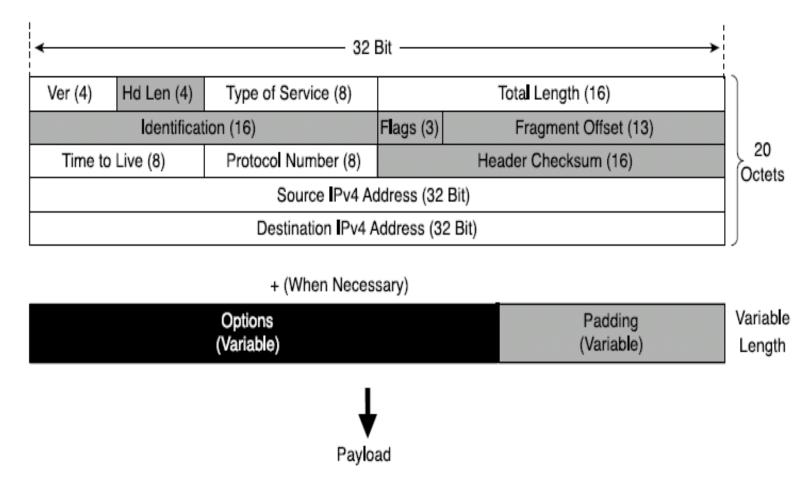
Figure 19.19 *Local addresses in IPv6*

10 bits	70 bits	48 bits	
1111111010	All 0s	Node address	a. Link local

10 bits	38 bits	32 bits	48 bits	
1111111011	All 0s	Subnet address	Node address	b. Site local

19.59

IPv4 Addressing Concepts and Their IPv6 Equivalents

IPv4 Address	IPv6 Address
Address Length – 32 bits	128 bits
Address Representation - decimal	hexadecimal
Internet address classes	Not applicable in IPv6
Multicast addresses (224.0.0.0/4)	IPv6 multicast addresses (FF00::/8)
Broadcast addresses	Not applicable in IPv6
Unspecified address is 0.0.0.0	Unspecified address is ::
Loopback address is 127.0.0.1	Loopback address is ::1
Public IP addresses	Global unicast addresses
Private IP addresses (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16)	Site-local addresses (FEC0::/10)
Autoconfigured addresses (169.254.0.0/16)	Link-local addresses (FE80::/64)

IPv6 Packet Format   **1**

IPv4 Header Structure

- basic IPv4 header contains 12 fields.
- each field of the IPv4 header has a specific use.
- Shaded field are removed in IPv6.

IPv4 Header - Review

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

IPv4 Header - Review

- **Version (4 bits)**
  - Indicates the version of IP and is set to 4.

- **Internet Header Length (4 bits**)
  - Indicates the number of 4-byte blocks in the IPv4 header.
  - Because an IPv4 header is a minimum of 20 bytes in size, the smallest value of the Internet Header Length (IHL) field is 5.

- **Type of Service (4 bits)**
  - Indicates the desired service expected by this packet for delivery through routers across the IPv4 internetwork.

IPv6 Packet Format   **4**

IPv4 Header - Review

- **Total Length (16 bits)**
  - Indicates the total length of the IPv4 packet (IPv4 header + IPv4 payload) and does not include link layer framing.
- **Identification (16 bits)**
  - Identifies this specific IPv4 packet.
  - The Identification field is selected by the originating source of the IPv4 packet. If the IPv4 packet is fragmented, all of the fragments retain the Identification field value so that the destination node can group the fragments for reassembly.
- **Flags (3 bits)**
  - Identifies flags for the fragmentation process.
  - There are two flags—one to indicate whether the IPv4 packet might be fragmented and another to indicate whether more fragments follow the current fragment.
- **Fragment Offset (13 bits)**
  - Indicates the position of the fragment relative to the original IPv4 payload.

IPv6 Packet Format   **5**

IPv4 Header - Review

- **Time to Live (8 bits)**
  - Indicate the maximum number of links on which an IPv4 packet can travel before being discarded.
  - Originally used as a time count with which an IPv4 router determined the length of time required (in seconds) to forward the IPv4 packet, decrementing the TTL accordingly. When the TTL equals 0,an ICMP Time Expired-TTL Expired in Transit message is sent to the source IPv4 address and the packet is discarded.
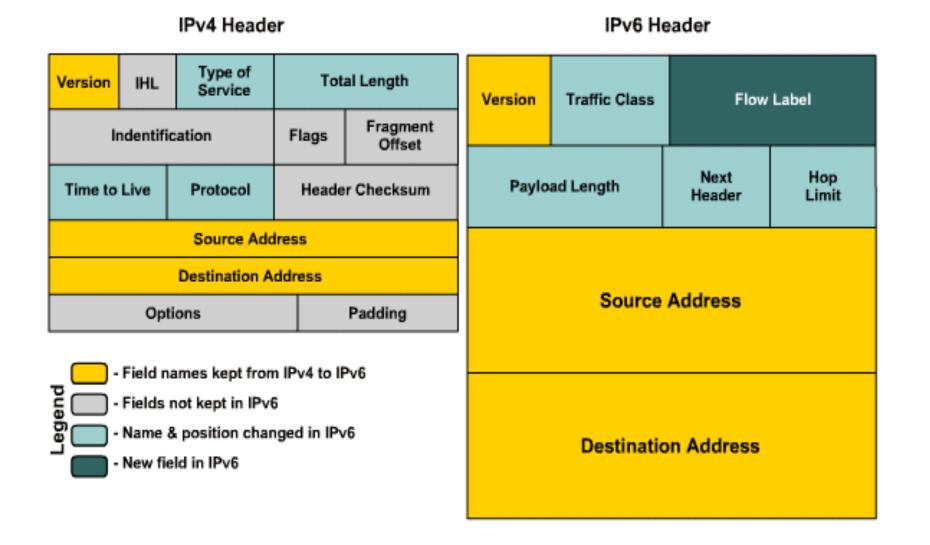- **Protocol (8 bits)**
  - Identifies the upper layer protocol.
  - For example, TCP uses a Protocol of 6, UDP uses a Protocol of 17, and ICMP uses a Protocol of 1.
  - The Protocol field is used to demultiplex an IPv4 packet to the upper layer protocol.

IPv6 Packet Format 6

IPv4 Header - Review
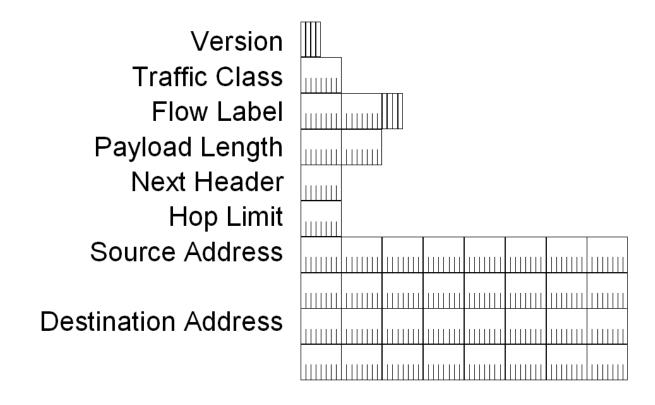
- **Header Checksum (16 Bits)**
  - Provides a checksum on the IPv4 header only.
  - The IPv4 payload is not included in the checksum calculation as the IPv4 payload and usually contains its own checksum..
- **Source Address (32 bits)**
  - Stores the IPv4 address of the originating host.
- **Destination Address (32 bits)**
  - Stores the IPv4 address of the destination host.
- **Options (multiple of 32 bits)**
  - Stores one or more IPv4 options.

IPv6 Packet Format   7

IPv4 vs IPv6 Header

IPv6 Packet Format

IPv6 Header

Version

Traffic Class

Flow Label

Payload Length

Next Header

Hop Limit

Source Address

Destination Address

IPv6 Packet Format   **9**

IPv6 Header Fields

- Based on these rules, RFC 2460 defines the following IPv6 header fields:

    1. **Version (4 bits)**
        - 4 bits are used to indicate the version of IP and is set to 6

    - **Traffic Class (8 bits)**
        - same function as the Type of Service field in the IPv4 header.

    1. **Flow Label (20 bits)**
        - identifies a flow and it is intended to enable the router to identify packets that should be treated in a similar way without the need for deep lookups within those packets.
        - set by the source and should not be changed by routers along the path to destination.

IPv6 Header Fields

4. **Payload Length (16 bits)**
   - With the header length fixed at 40 bytes, it is enough to indicate the length of the payload to determine the length of the entire packet.

5. **Next Header (8 bits)**
   - Indicates either the first extension header (if present) or the protocol in the upper layer PDU (such as TCP, UDP, or ICMPv6).

6. **Hop Limit (8 bits)**
   - In IPv6, the IPv4 TTL was appropriately renamed Hop Limit because it is a variable that is decremented at each hop, and it does not have a temporal dimension.

IPv6 Packet Format 11

IPv6 Header Fields

7. **Source IPv6 Address (128 bits)**

   - Stores the IPv6 address of the originating host.

8. **Destination IPv6 Address (128 bits)**

   - Stores the IPv6 address of the current destination host.

Values of the Next Header Field

Value (in decimal)	Header
0	Hop-by-Hop Options Header
6	TCP
17	UDP
41	Encapsulated IPv6 Header
43	Routing Header
44	Fragment Header
46	Resource ReSerVation Protocol
50	Encapsulating Security Payload
51	Authentication Header
58	ICMPv6
59	No next header
60	Destination Options Header

IPv6 Packet Format   **13**