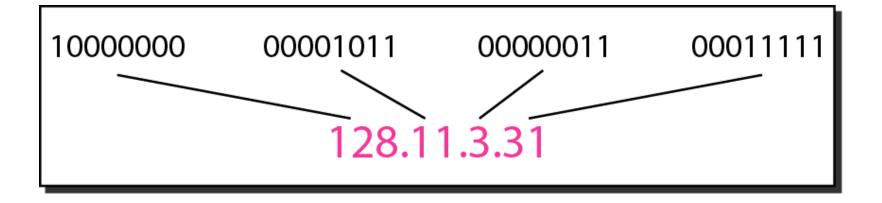
### Network Layer: Logical Addressing

### 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.

### Figure 1 Dotted-decimal notation and binary notation for an IPv4 address





- a. 10000001 00001011 00001011 11101111

### Solution

We replace each group of 8 bits with its equivalent decimal number and add dots for separation.

- a. 129.11.11.239
- b. 193.131.27.255



### 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.

- a. 01101111 00111000 00101101 01001110
- **b.** 11011101 00100010 00000111 01010010



### 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 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

### Find the class of each address.

- *a.* <u>0</u>00000001 00001011 00001011 11101111
- **b.** <u>110</u>000001 100000011 00011011 111111111
- **c.** <u>14</u>.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.

Note

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

 Table 2
 Default masks for classful addressing

Class	Binary	Dotted-Decimal	CIDR
A	1111111 00000000 00000000 00000000	<b>255</b> .0.0.0	/8
В	1111111 11111111 00000000 00000000	<b>255.255.</b> 0.0	/16
С	1111111 11111111 11111111 00000000	255.255.255.0	/24

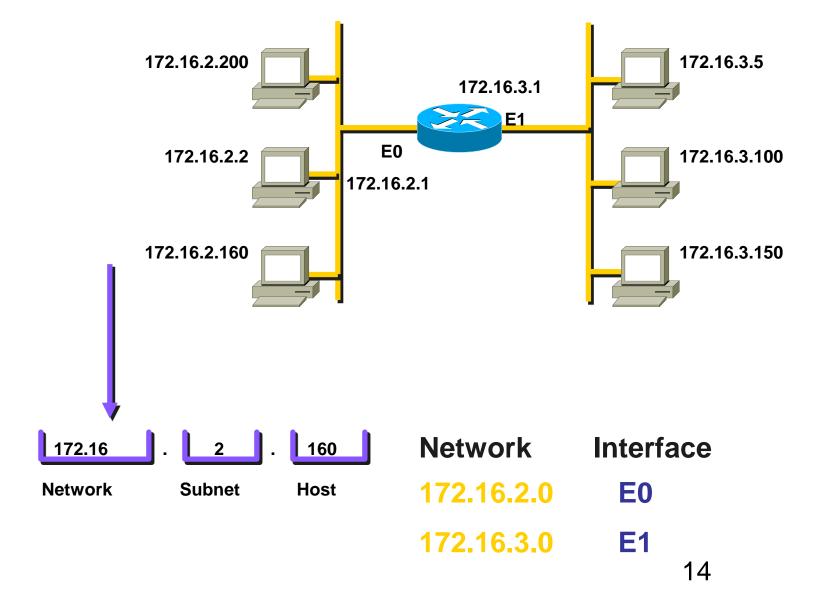


# Classful addressing, which is almost obsolete, is replaced with classless addressing.

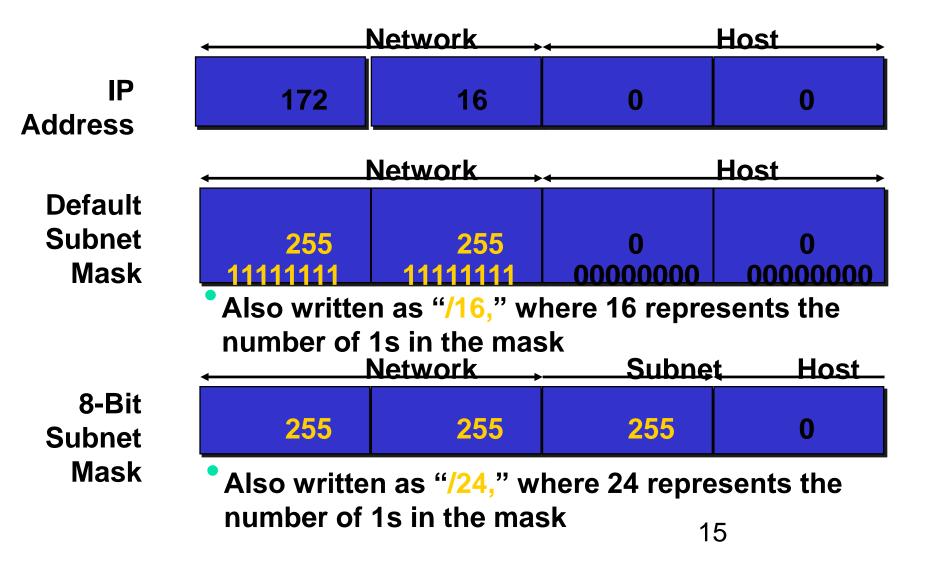
### Classless Inter-Domain Routing (CIDR)

- Basically the method that ISPs (Internet Service Providers) use to allocate an amount of addresses to a company, a home
- Ex: 192.168.10.32/28
- The slash notation (/) means how many bits are turned on (1s)

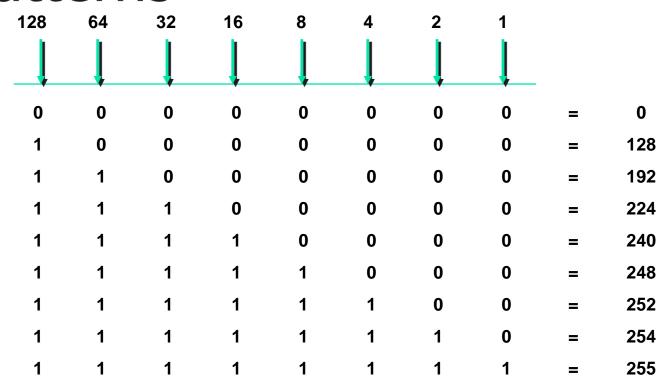
### **Subnet Addressing**



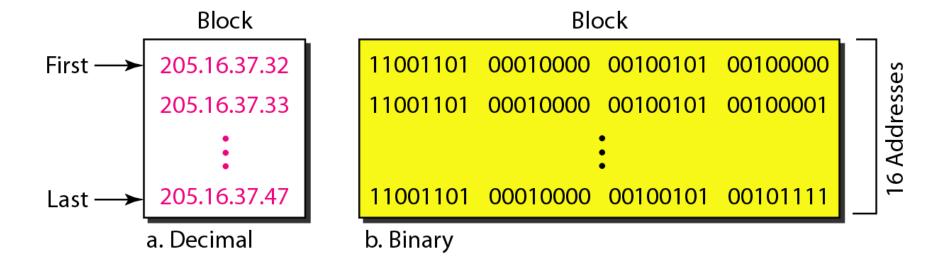
### Subnet Mask



# Decimal Equivalents of Bit Patterns



### Figure 3 A block of 16 addresses granted to a small organization





# 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.

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 3.



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

Find the last address for the block in Example 6.

### **Solution**

or

205.16.37.47

This is actually the block shown in Figure 3.



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

Find the number of addresses in Example 6.

### Solution

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

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 6 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 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 1111111 11110000

First address: 11001101 00010000 00100101 00100000

### Example 9 (continued)

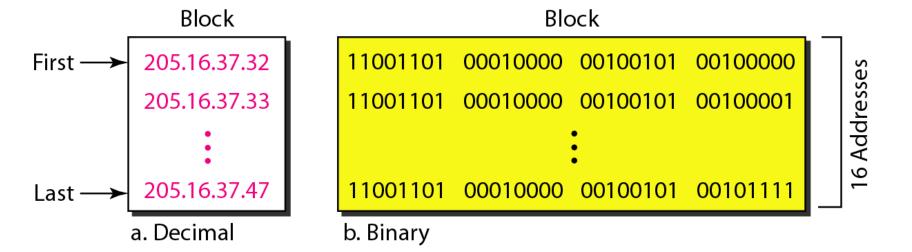
b. The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit. The result of ORing 2 bits is 0 if both bits are 0s; the result is 1 otherwise. The complement of a number is found by changing each 1 to 0 and each 0 to 1.

Address: 11001101 00010000 00100101 00100111

Mask complement: 00000000 00000000 00000000 00001111

Last address: 11001101 00010000 00100101 00101111

### Figure 4 A network configuration for the block 205.16.37.32/28





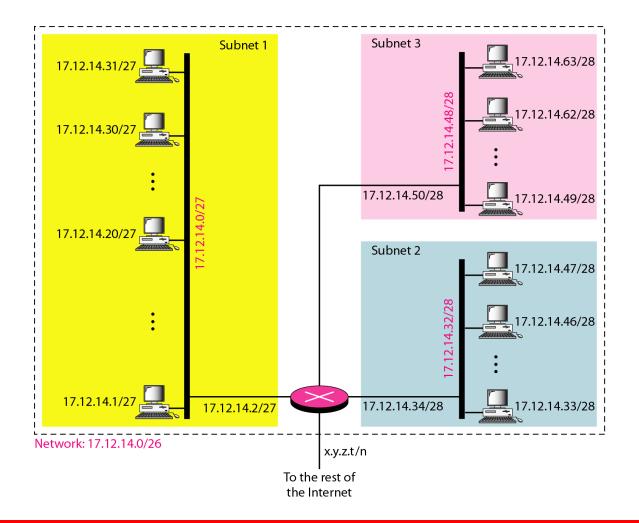
The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.



### Note

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 7 Configuration and addresses in a subnetted network



- 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 10 (continued)

### Solution

Figure 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

 $Total = 64 \times 256 = 16,384$ 



### Example 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

 $Total = 128 \times 128 = 16,384$ 

### Ex

### Example 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

 $Total = 128 \times 64 = 8192$ 

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 9 An example of address allocation and distribution by an ISP

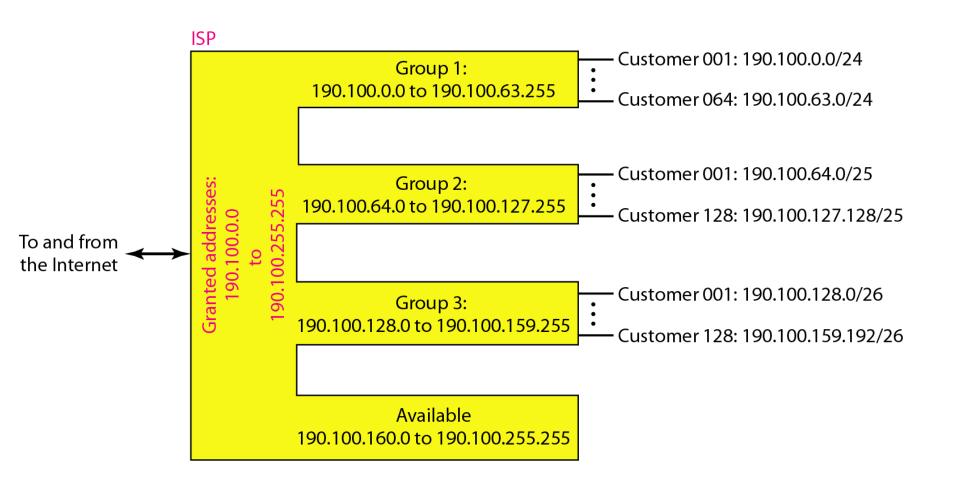
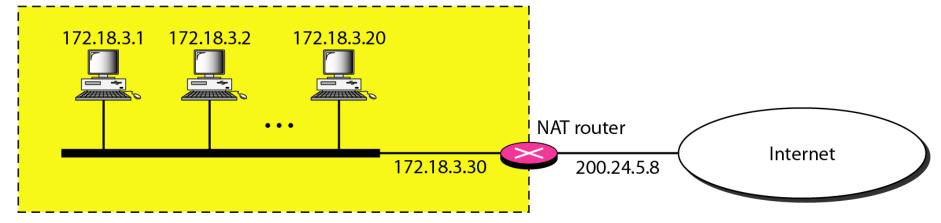


 Table 3
 Addresses for private networks

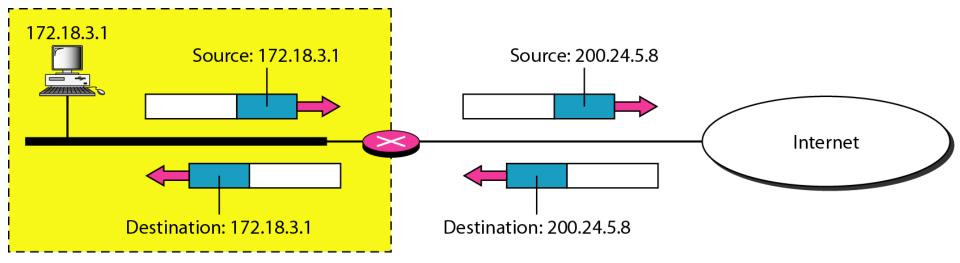
	Ran	ge	Total
10.0.0.0	to	10.255.255.255	$2^{24}$
172.16.0.0	to	172.31.255.255	$2^{20}$
192.168.0.0	to	192.168.255.255	$2^{16}$

### Figure 10 A NAT implementation

#### Site using private addresses



### Figure 11 Addresses in a NAT



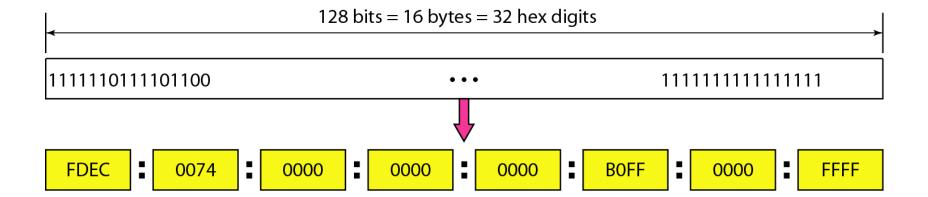
### 2 IPv6 ADDRESSES

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.

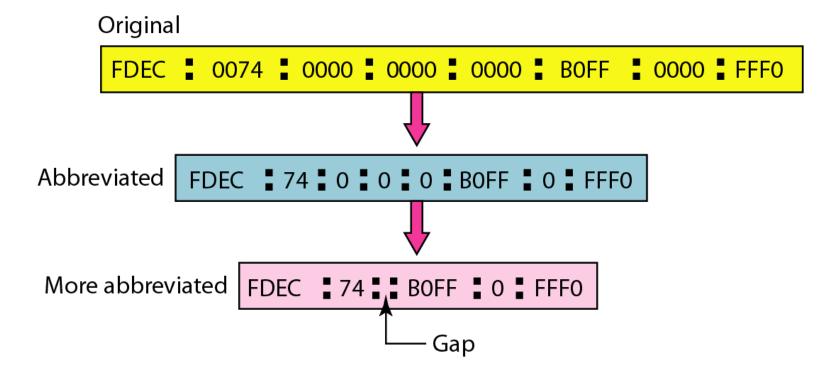


## An IPv6 address is 128 bits long.

### Figure 14 IPv6 address in binary and hexadecimal colon notation



### Figure 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

 0: 15:
 : 1: 12:1213

This means that the original address is.

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

		Dec.	Hex.	Binary	Dec.	Hex.	Binary
		0	0	0000	8	8	1000
IPv6 Address	Mototion	1	1	0001	9	9	1001
	Notation	2	2	0010	10	A	1010
		3	3	0011	11	В	1011
		4	4	0100	12	C	1100
		5	5	0101	13	D	1101
	One Hex digit = 4 bits	6	6	0110	14	E	1110
		7	7	0111	15	F	1111

2001:0DB8:AAAA:1111:0000:0000:0000:0100/64

1		2		3		4	5		6		7		8
2001	:	ODB8	:	AAAA	:	1111 :	0000	:	0000	:	0000	:	0100
16 bits		16 bits		16 bits		16 bits	16 bits		16 bits		16 bits		16 bits

- IPv6 addresses are 128-bit addresses represented in:
  - Eight 16-bit segments or "hextets" (not a formal term)
  - Hexadecimal (non-case sensitive) between 0000 and FFFF
  - Separated by colons

- How many addresses does 128 bits give us?
  - 340 undecillion (10<sup>36</sup>) addesses or ...
  - 340 trillion (10<sup>12</sup>) trillion trillion addresses or ...
  - "IPv6 could provide each and every square micrometer of the earth's surface with 5,000 unique addresses. Micrometer = 0.001 mm or 0.000039 inches" or....
  - I won't be the one presenting IPv7 at any Cisco Academy Conference.

### Rule 1: Leading 0's

- Two rules for reducing the size of written IPv6 addresses.
- The first rule is: Leading zeroes in any 16-bit segment do not have to be written.

```
3ffe : -0404 : 0001 : 1000 : 0000 : 0000 : 0ef0 : bc00

3ffe : 404 : 1 : 1000 : 0 : 0 : ef0 : bc00

3ffe : 0000 : 010d : 000a : 00dd : c000 : e000 : 0001

3ffe : 0 : 10d : a : dd : c000 : e000 : 1

ff02 : 0000 : 0000 : 0000 : 0000 : 0000 : 0500

ff02 : 0 : 0 : 0 : 0 : 0 : 500
```

### Rule 2: Double colon :: equals 0000...0000

- The second rule can reduce this address even further:
- Any single, contiguous string of one or more 16-bit segments consisting of all zeroes can be represented with a double colon.

```
ff02: 0000: 0000: 0000: 0000: 0000: 0000: 0000

ff02: 500

Second Rule

First Rule
```

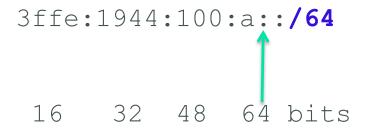
Ff02::500

Both of these are correct...

### **Network Prefixes**

 IPv4, the prefix—the network portion of the address—can be identified by a dotted decimal netmask or bitcount.

- IPv6 prefixes are always identified by bitcount (prefix length).
- Prefix length notation:



### **IPv6 Address Types**

