Network Layer: Logical Addressing

Basic Services

- Basic services provided by a network layer are:
  - **Packetizing**
  - **Routing and Forwarding**
  - Other services
    - **Error control:**
      - Checks only header not whole datagram
      - ICMP uses error control
    - **Flow control:**
      - Not directly provide any flow control
      - Upper layer protocols uses this
    - **Congestion control**
    - **Quality of service (QoS)**
    - **Security**

Packetizing

- Encapsulation of the payload in a network-layer packet at the source and decapsulation of the payload from the network-layer packet at the destination.
  - Adds header that contains source and destination addresses
  - Few other information that requires by the network layer
  - If the packet is fragmented at the source or at the router along the path, this layer is responsible for waiting until all fragments arrive
  - Routers are not allowed to do decapsulation
    - Does only fragmentation if required
    - Look only inspect the addresses for forwarding purpose

Routing

- This is following some strategies to send the packets from source to destination.
- There may be more than one root possible from source to destination
  - But best path is chosen using the strategy called routing algorithms
  - These algorithms are called the routing protocol

Forwarding

- It is defined as the action applied by each router when a packet arrives at one of its interfaces.
- This decision is taken based on the *forwarding table or routing table* and the address present at the packet header.

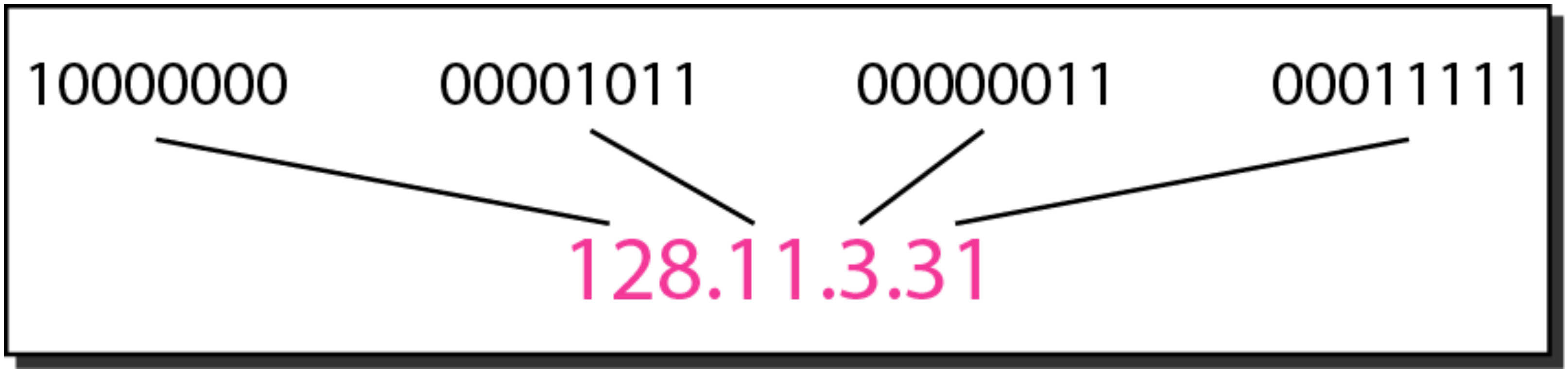
IPv4 Addresses

- ***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

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

Dotted-decimal notation and binary form



Example

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 and add dots for separation.

a. 129.11.11.239

b. 193.131.27.255

Example

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

Example

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.

Note

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

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

Find the class of each address.

a. <u>**0**</u>0000001 00001011 00001011 11101111

b. <u>**110**</u>00001 10000011 00011011 11111111

c. <u>**14**</u>.23.120.8

d. <u>**252**</u>.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.

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

Address Depletion

- Since the addresses are not distributed properly
  - Internet was faced with the problem of the addresses being rapidly used up
  - No more addresses are available for organizations and individuals

Example: Take Class A

- This class can be assigned to only 128 organizations in the world
- Each organization needs to have a single network with 1,67,77,216 nodes.
- There are may be few organizations that are large, hence most of the addresses are wasted.

Subnetting

- A classful address is divided into several blocks is called subnets
- Example:
  - Say a class C address:
    - Network address: 200.12.10.0. (*First address*)
    - Number of hosts: 256
    - Total usable address: 254
  - If two subnets we want then we fix the *first bit* of the last octate
  - So, for first network the IP addresses will vary for 200.12.10.0 to 200.12.10.127
  - For the second network the IP addresses will vary from 200.12.10.128 to 200.12.10.255

Supernetting

- Multiple small networks are connected to form a large networks
- Mostly used to reduce the complexity of the routing table and to overcome the increased network size
- Smaller block should have contiguous IP addresses to form a large block of contiguous IP address

Supernetting

- **Rules for aggregation**
  - Contiguous
  - Same size network (2^n)
  - 1st network ID /2^n
- **Example:**
  - 200.1.0.0/24
  - 200.1.1.0/24
  - 200.1.2.0/24
  - 200.1.3.0/24
  - Size of network = $4 \times 2^8 = 2^{10}$

- **So, supernet ID: 200.1.0.0**
- **Supernet mask: 255.255.252.0**

Advantages of Classful addressing

- Given address we can easily find the class of the address
- Prefix length also can be found very easily as well as suffix length.

Classless addressing

- Gives us the larger address space
  - Length of the IP address is increased and hence the length of the IP packet
- **Long term solution:**
  - Gives us the Ipv6 version of IP address
- **Short term solution:**
  - Use the same address space (32 bits)
  - Changes the distribution process
  - Called also classless addressing

Classless addressing

- In this addressing, whole address space is divided into variable length blocks and belongs to no classes as earlier.
- The prefix in an address defines the block (number of networks) and this prefix is variable (ranges from 0-32)
- The suffix defines the node(device) restrictions
- We can have a block of 2^0, 2^1, 2^2, .. 2^{32} addresses.
  - Only restriction is that each block should have the addresses in *terms of power of 2*
- We can say classful addressing is a special case of classless addressing.

Prefix length: Slash notation

- Since the prefix length is not inherent like classful addressing
  - We need to separately give the length of the prefix
  - In this case the prefix length, ***n***, is added to the address, separated by a slash
  - This notation is called slash notation and formally known as ***Classless Interdomain Routing (CIDR)*** strategy.

Note

- **In IPv4 classless 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**
- **The */n* defines the mask.**

Extracting information from address

- Using the prefix length n
  - The number of addresses found in the block is $N = 2^{32-n}$.
  - To find the *first address*, we keep the n-leftmost bits and set the *(32-n) rightmost bits all to 0s.*
  - To find the *last address*, we keep the n leftmost bits and set the *(32-n) rightmost bits all to 1s.*
- Example: Address: 167.199.170.82/27
  - First address: 167.199.170.64/27
    - $(82)_d = (010\mathbf{10010})_b$ $(64)_d=(010\mathbf{00000})_b$
  - Last address: 167.199.170.95/27. $(95)_d=(010\mathbf{11111})_b$

Address Mask

- Another way of determining the first and last address of a block of addresses
- The address mask is 32 bits in which
  - The *n leftmost bits are set to 1s* and
  - The *rest of the bits (32-n) are set to 0s.*
- Defining a mask in this way helps to extract the information using bitwise operation
  - **The number of addresses in this block N= NOT(mask)+1**
  - **The first address in this block =(any address in this block) AND (mask)**
  - **The last address in this block = (Any address in this block) OR(NOT mask)**

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

Example : Consider the CIDR as /28

Solution

a. ***The first address can be found by ANDing the given addresses with the mask. ANDing here is done bit by bit.***

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

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.***

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

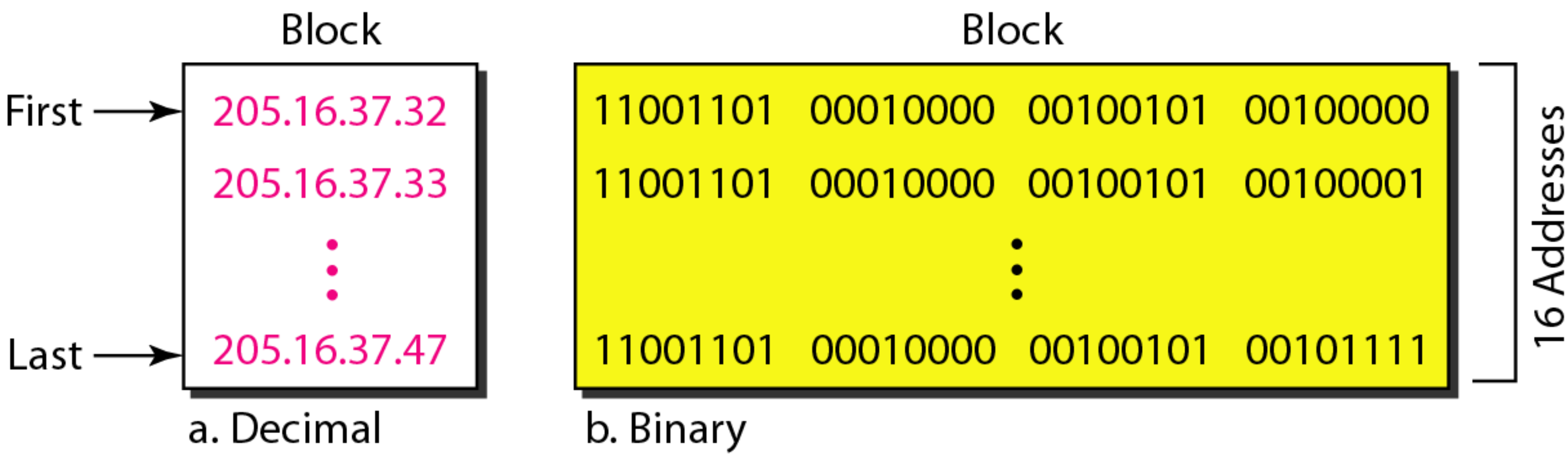
Example (continued)

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 00000000 00000000 00001111**
Number of addresses:	15 + 1 = 16

A network configuration for the block 205.16.37.32/28

- **What are the range of addresses??**



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.**

- **This first address is the one that is *assigned to the routers* to direct the message sent to the organization from outside.**

Two levels of hierarchy in an IPv4 address

Area code

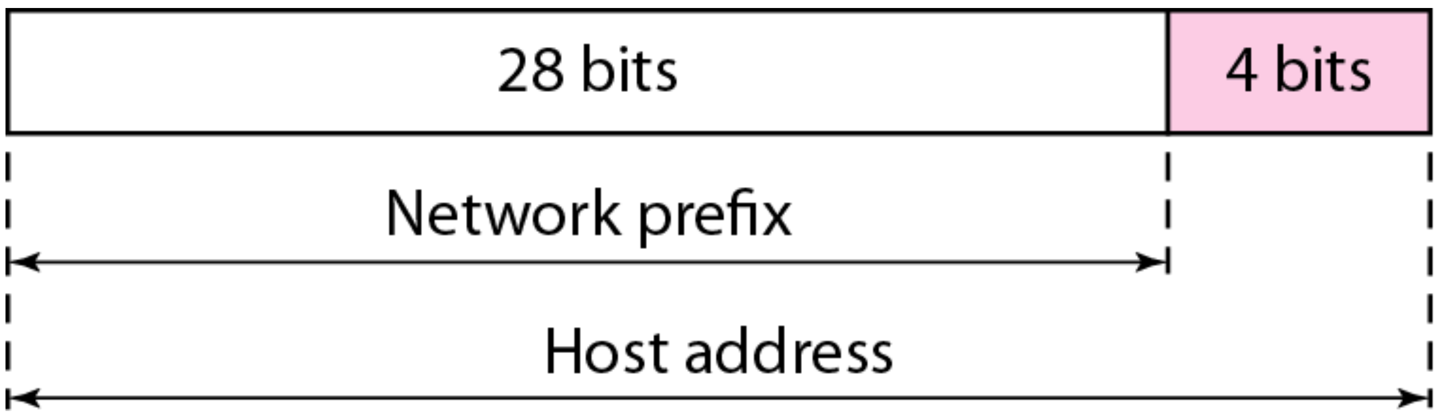
Telephone number: (+91) xxxx - nnnnnn

Operators code

Unique subscriber’s number

- IP address can define only two level of hierarchy when not subnetted.
- The n left-most bits defines the network and (32-n) bits defines a particular host of that network.

A frame in a character-oriented protocol

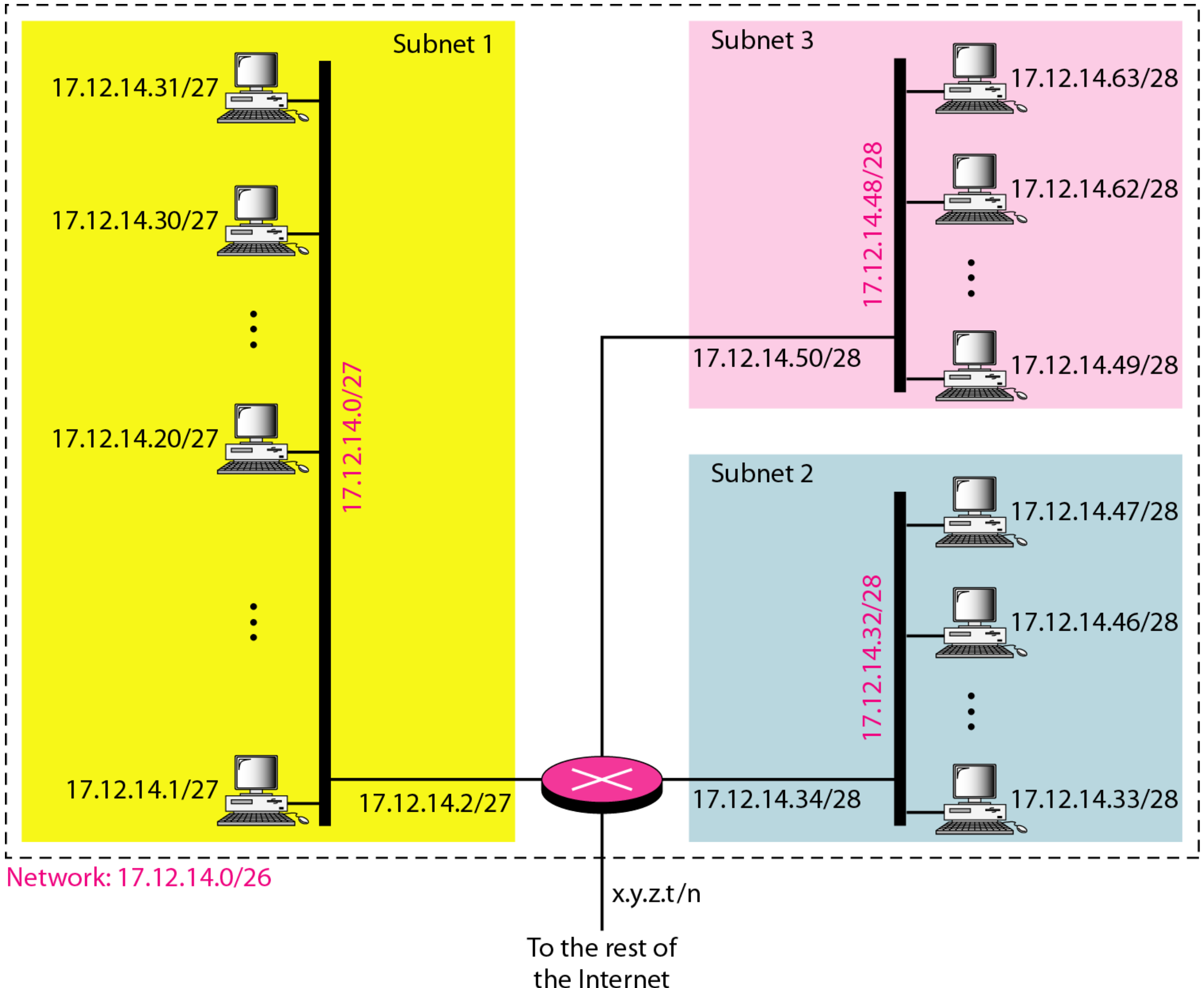


Three levels of hierarchy: Subnetting

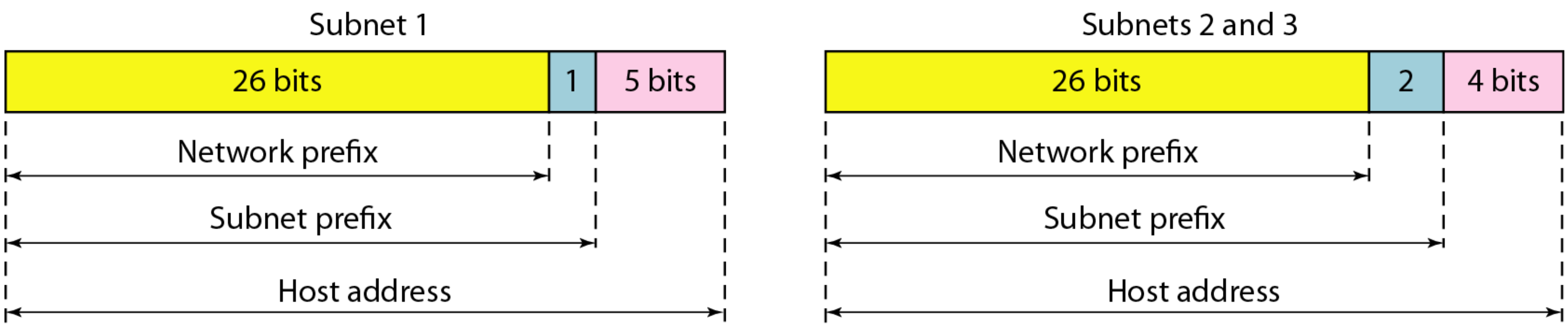
- An organization that is granted a large block of addresses may want to create clusters of networks (called subnets)
  - Divides the addresses between the different subnets
- All messages are sent to the router address that connects the organization to the rest of the Internet.
  - Router routes the message to the appropriate subnet.

Configuration and addresses in a subnetted network

- Assigned 64 addresses with the block **17.12.14.0/26**
- Three offices and needs to divide the addresses into *three subblocks of 32, 16 and 16 addresses*



Three-level hierarchy in an IPv4 address



Address Allocation

- Ultimate responsibility of address allocation is given to a global authority
  - Internet corporation of Assigned Names and Addresses (ICANN)
- But ICANN does not assign addresses to organizations
  - Assigns large block of addresses to an Internet service provider(ISP)
- Each ISP, divided its assigned block into smaller subblocks and grants the subblocks to its customers

Example

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 (continued)

Solution

Following figure shows the situation.

Group 1

For this group, each customer needs 256 addresses. This means that 8 ($\log_2 256$) 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 (continued)

Group 2

For this group, each customer needs 128 addresses. This means that 7 ($log_2 128$) 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 (continued)

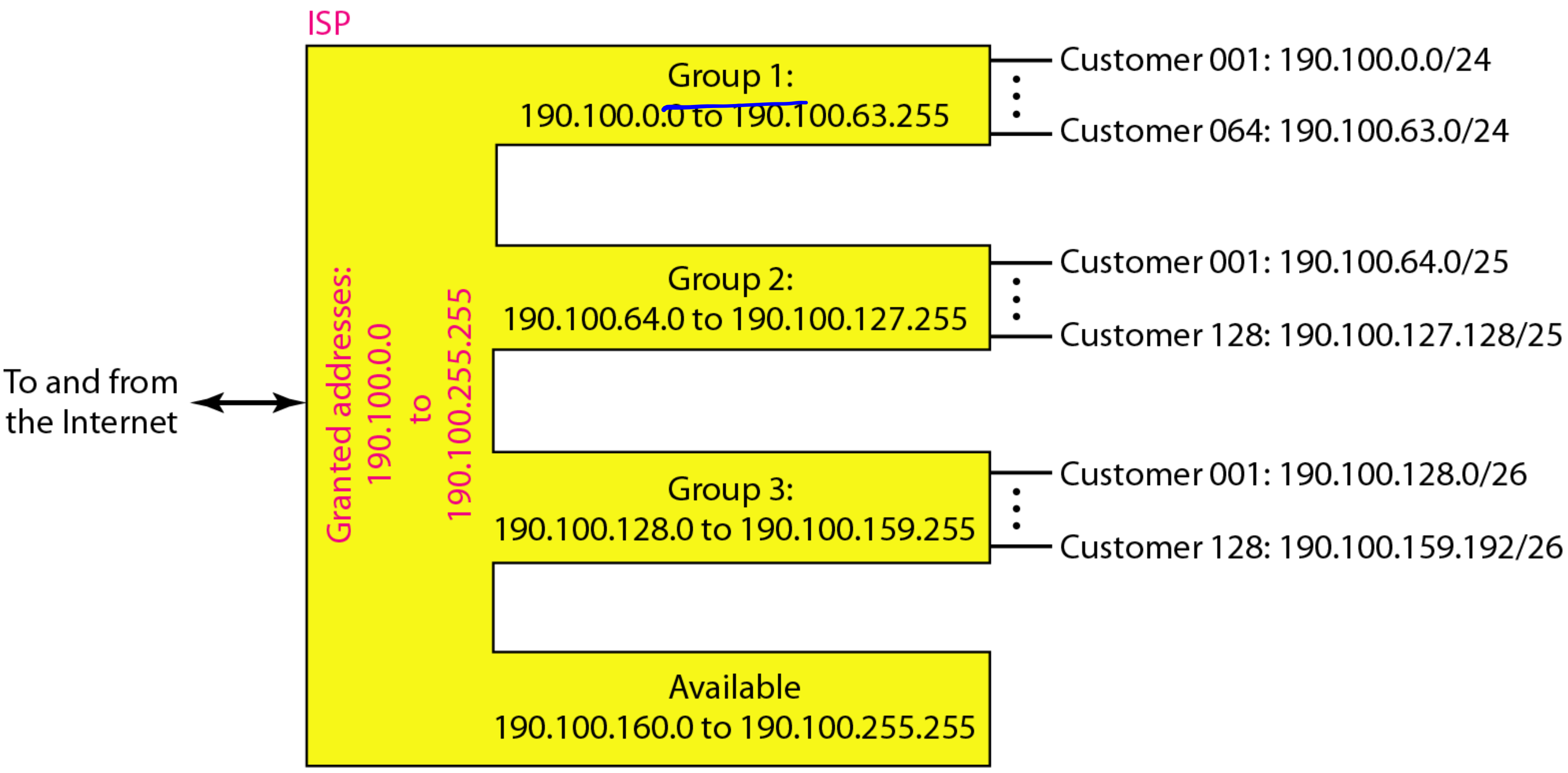
Group 3

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

An example of address allocation and distribution by an ISP

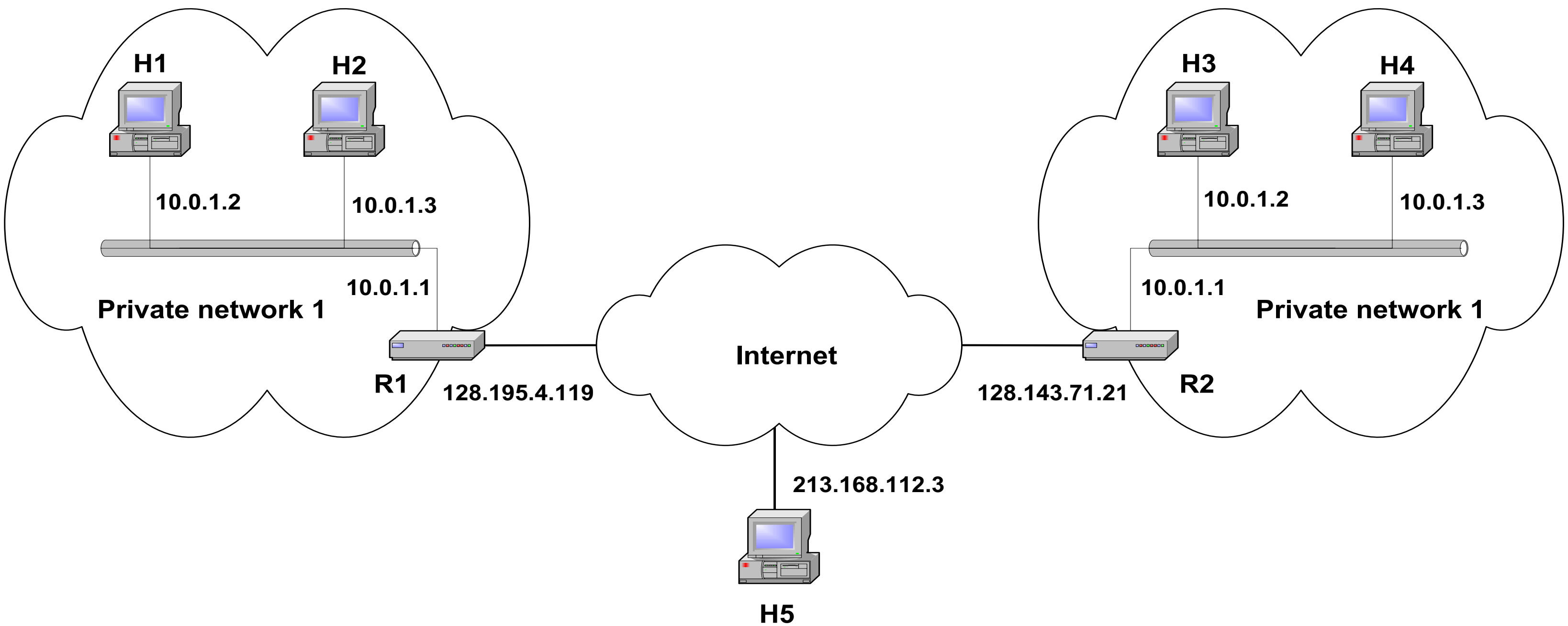


Private Network

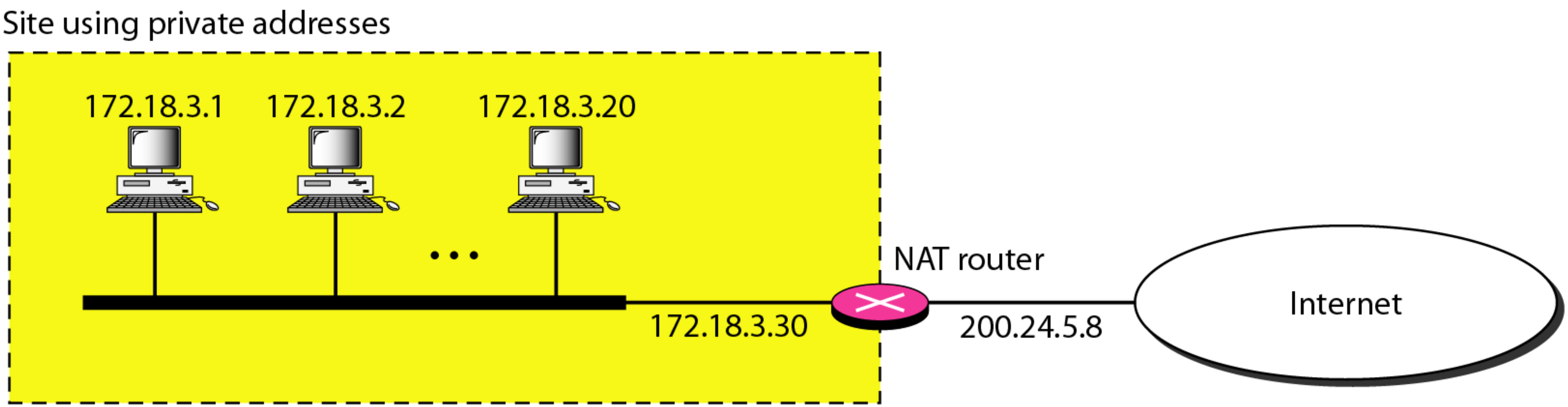
- *Private IP* network is an IP network that is not directly connected to the Internet
- IP addresses in a private network can be assigned arbitrarily.
  - Not registered and not guaranteed to be globally unique
- Generally, private networks use addresses from the following experimental address ranges (*non-routable addresses*):

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}

Private Network



A NAT implementation



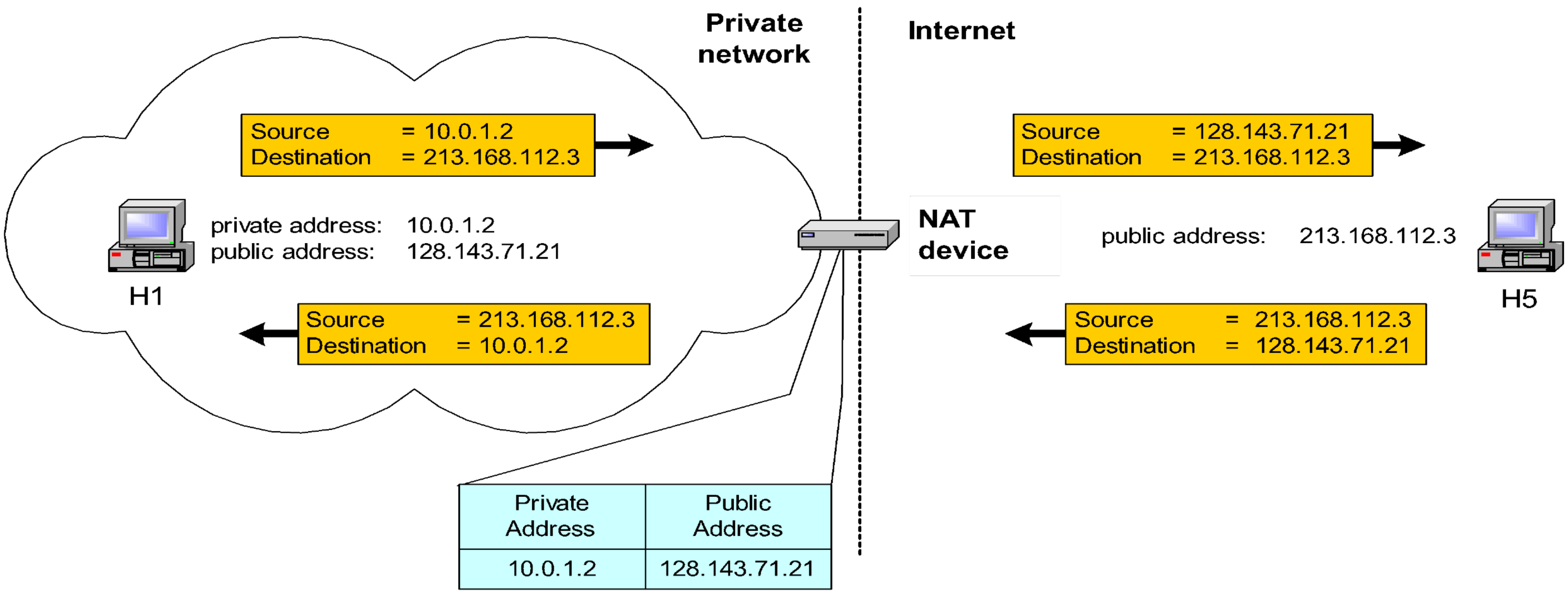
Network Address Translation (NAT)

- NAT is a router function where IP addresses (and possibly port numbers) of IP datagrams are replaced at the boundary of a private network

- NAT is a method that enables *hosts on private networks to communicate with hosts on the Internet*

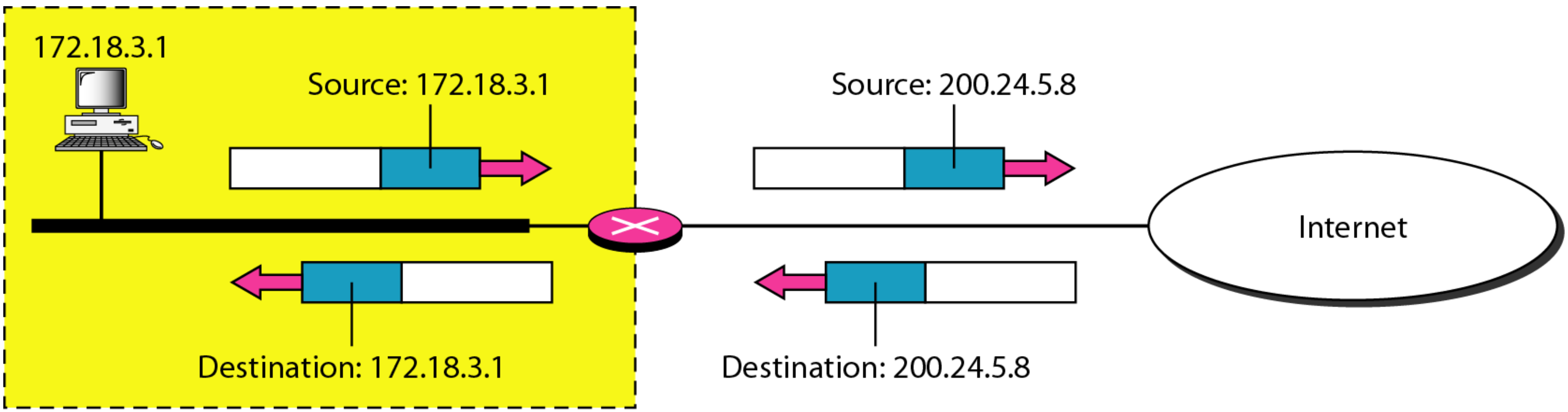
- NAT is run on routers that connect private networks to the public Internet, to replace the ***IP address-port pair*** of an IP packet with ***another IP address-port pair***.

Network Address Translation (NAT)



- NAT device has address translation table

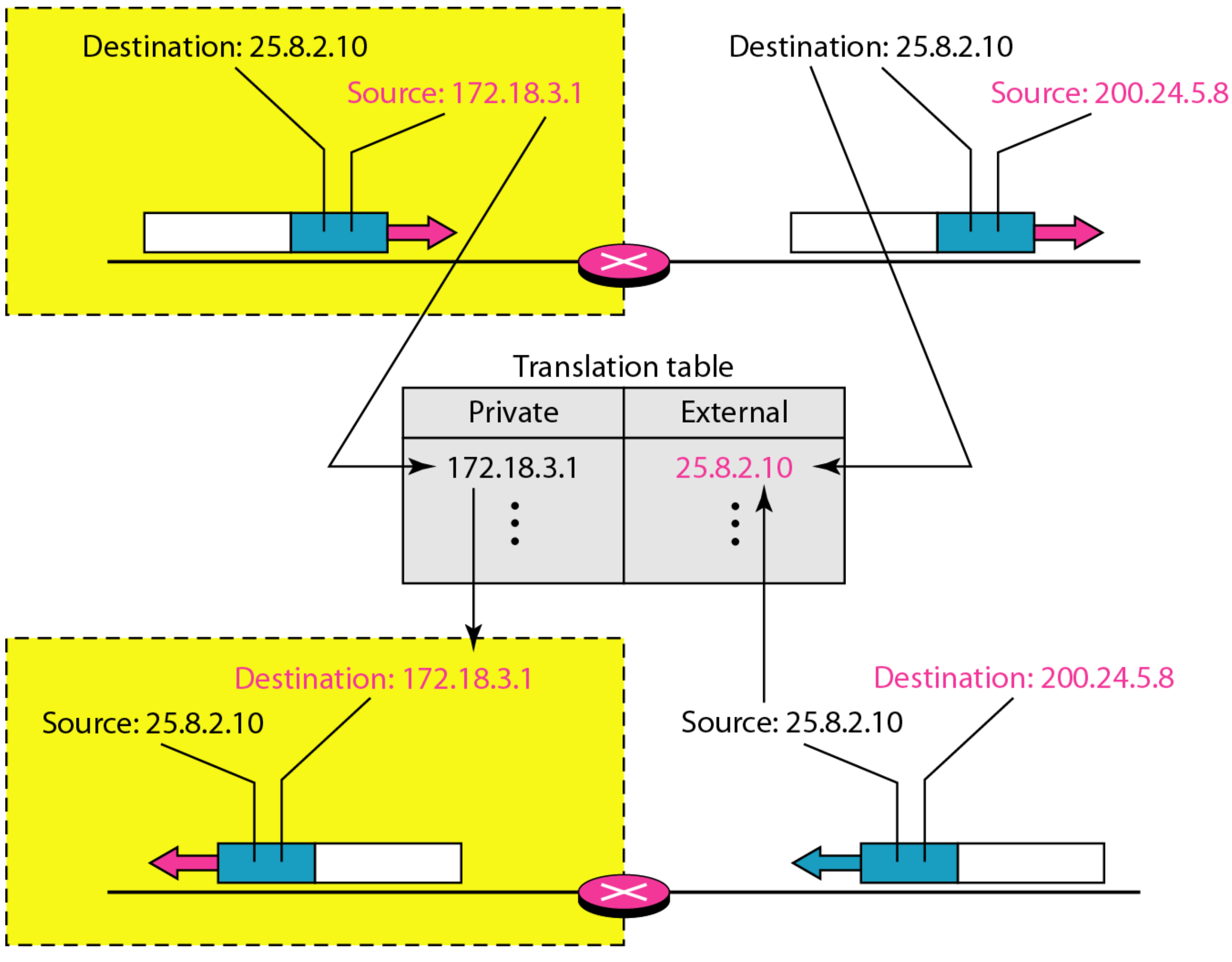
Addresses in a NAT



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
. . .	. . .	. . .	. . .	. . .

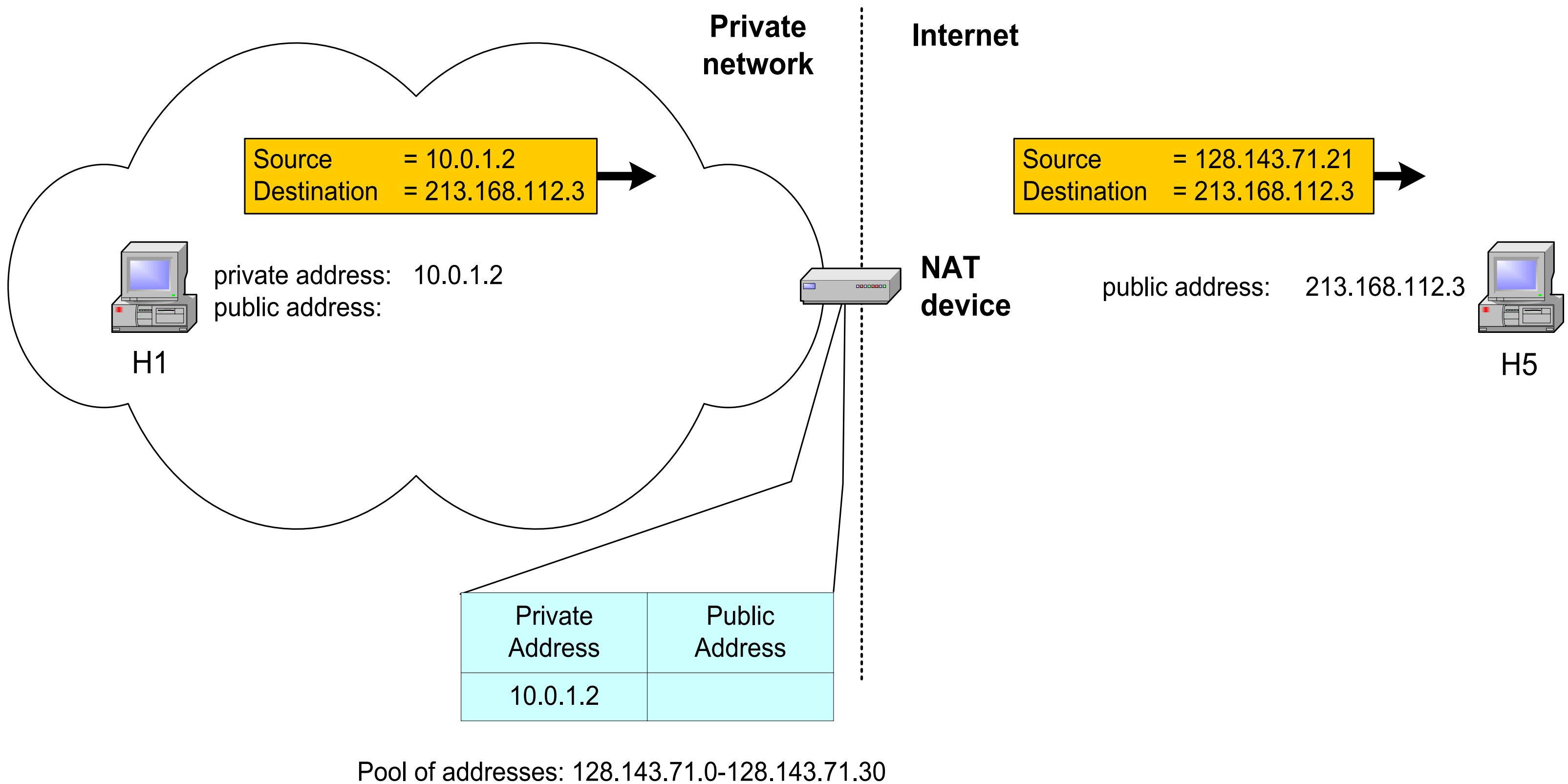
NAT address translation



Pooling of IP addresses

- **Scenario:** Corporate network has many hosts but only a small number of public IP addresses
- **NAT solution:**
  - Corporate network is managed with a private address space
  - NAT device, located at the boundary between the corporate network and the public Internet, manages a pool of public IP addresses
  - When a host from the corporate network sends an IP datagram to a host in the public Internet,
    - The NAT device picks a public IP address from the address pool, and binds this address to the private address of the host

Pooling of IP addresses

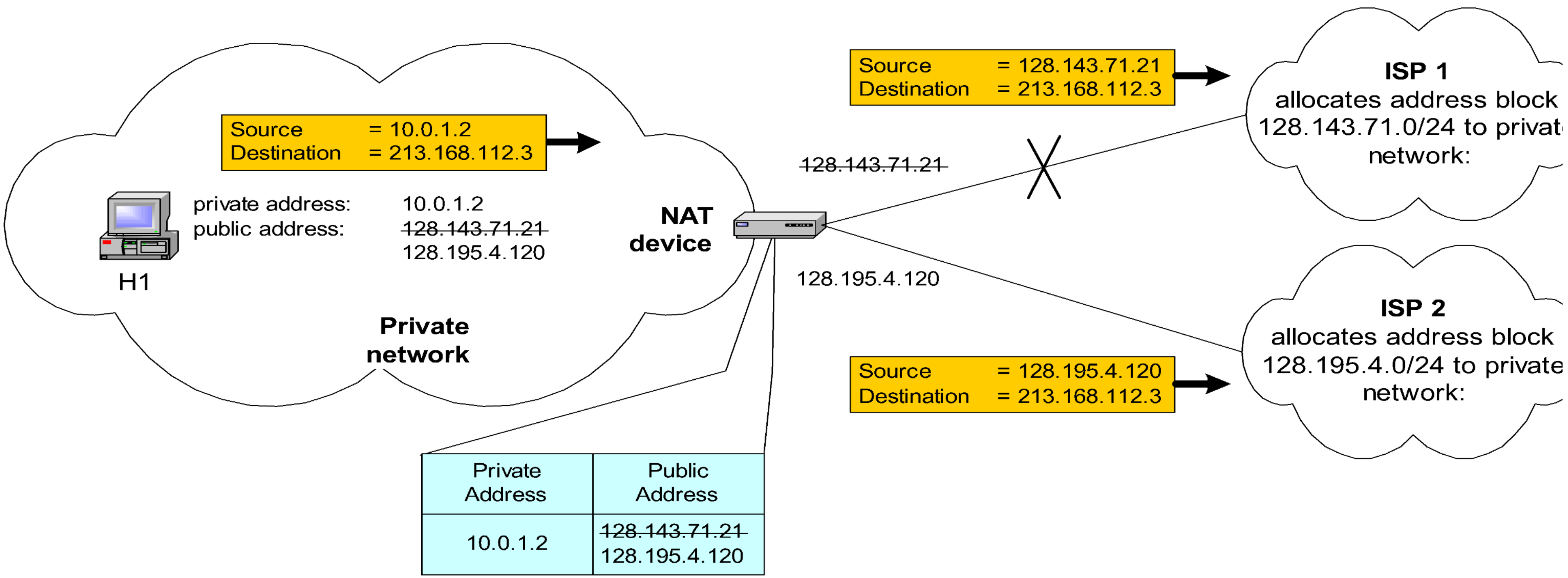


Supporting migration between network service providers

- **Scenario:** In CIDR, the IP addresses in a corporate network are obtained from the service provider. Changing the service provider requires changing all IP addresses in the network.
- **NAT solution:**
  - Assign private addresses to the hosts of the corporate network
  - NAT device has static address translation entries which bind the private address of a host to the public address.
  - Migration to a new network service provider merely requires an update of the NAT device. The migration is not noticeable to the hosts on the network.

  Note:
  - The difference to the use of NAT with IP address pooling is that the mapping of public and private IP addresses is static.

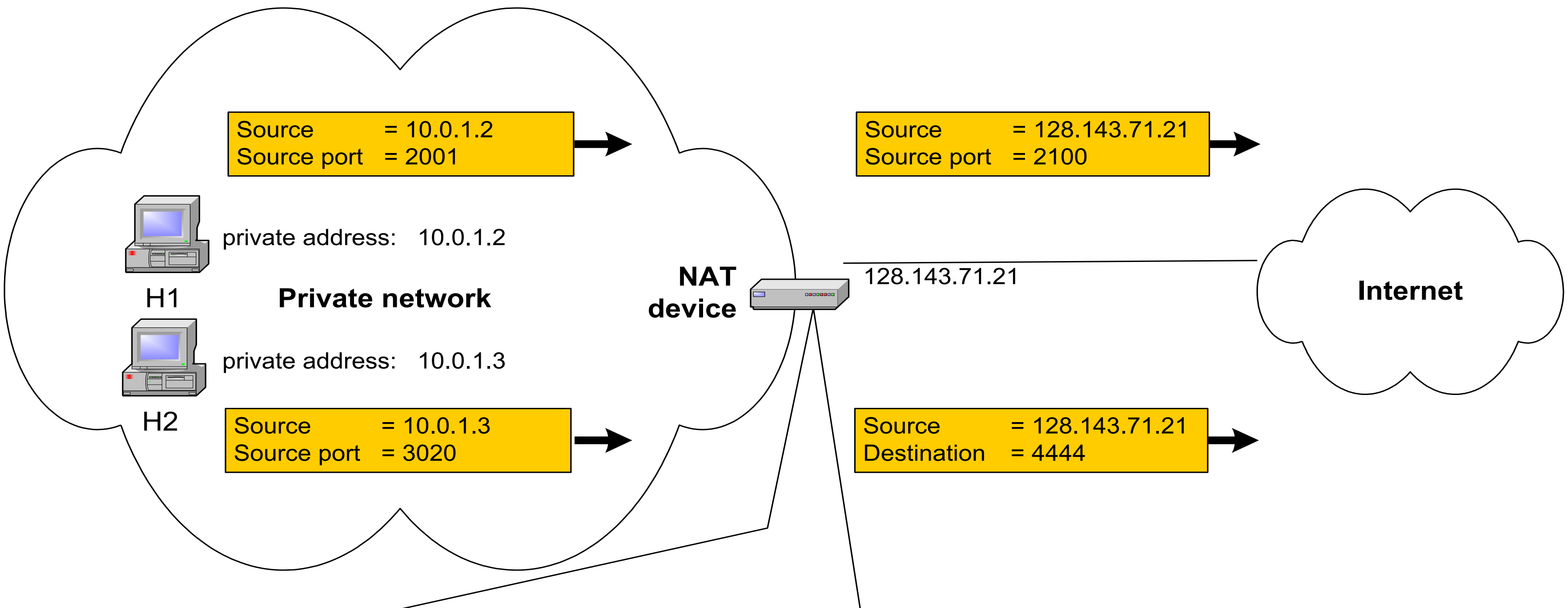
Supporting migration between network service providers



IP masquerading

- **Also called: Network address and port translation (NAPT), port address translation (PAT).**
- **Scenario:** Single public IP address is mapped to multiple hosts in a private network.
- **NAT solution:**
  - Assign private addresses to the hosts of the corporate network
  - NAT device modifies the port numbers for outgoing traffic

IP masquerading

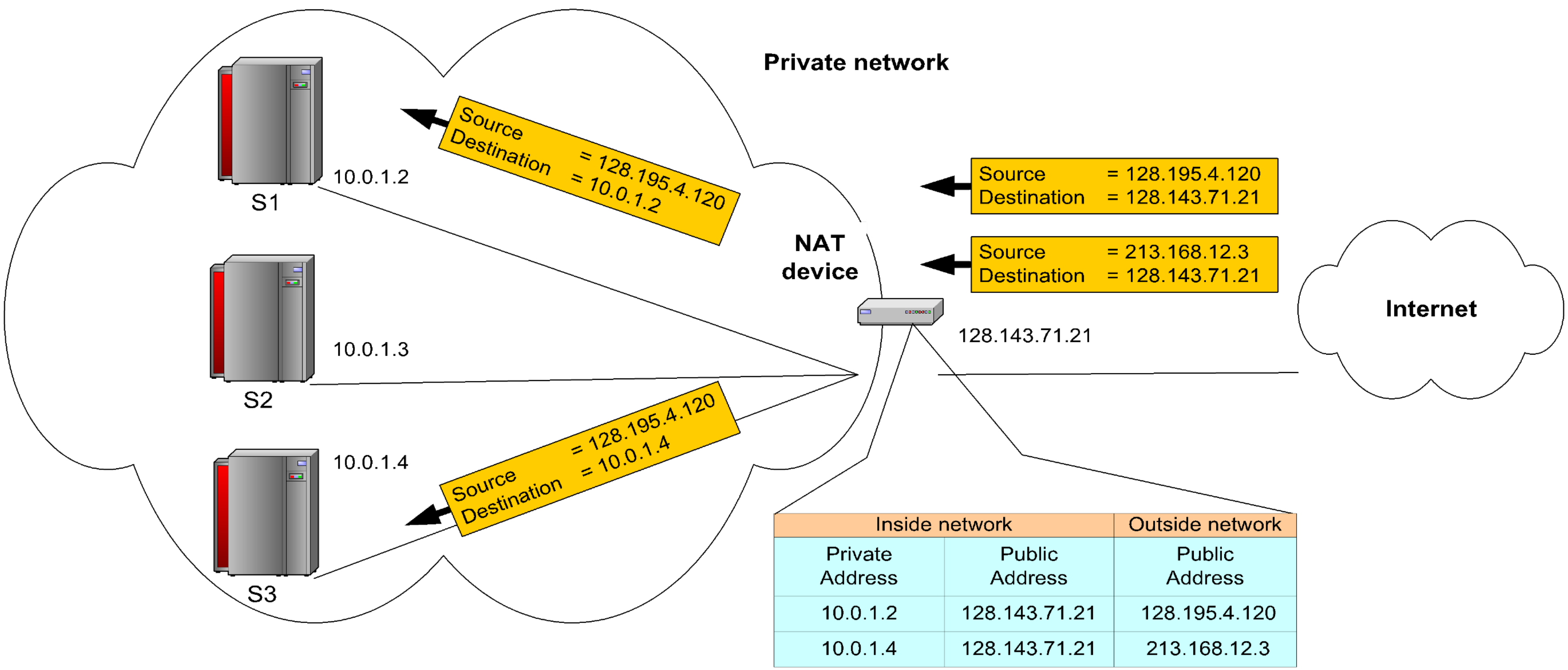


Private Address	Public Address
10.0.1.2/2001	128.143.71.21/2100
10.0.1.3/3020	128.143.71.21/4444

Load balancing of servers

- **Scenario:** Balance the load on a set of identical servers, which are accessible from a single IP address
- **NAT solution:**
  - Here, the servers are assigned private addresses
  - NAT device acts as a proxy for requests to the server from the public network
  - The NAT device changes the destination IP address of arriving packets to one of the private addresses for a server
  - A sensible strategy for balancing the load of the servers is to assign the addresses of the servers in a round-robin fashion.

Load balancing of servers



Inside network		Outside network
Private Address	Public Address	Public Address
10.0.1.2	128.143.71.21	128.195.4.120
10.0.1.4	128.143.71.21	213.168.12.3

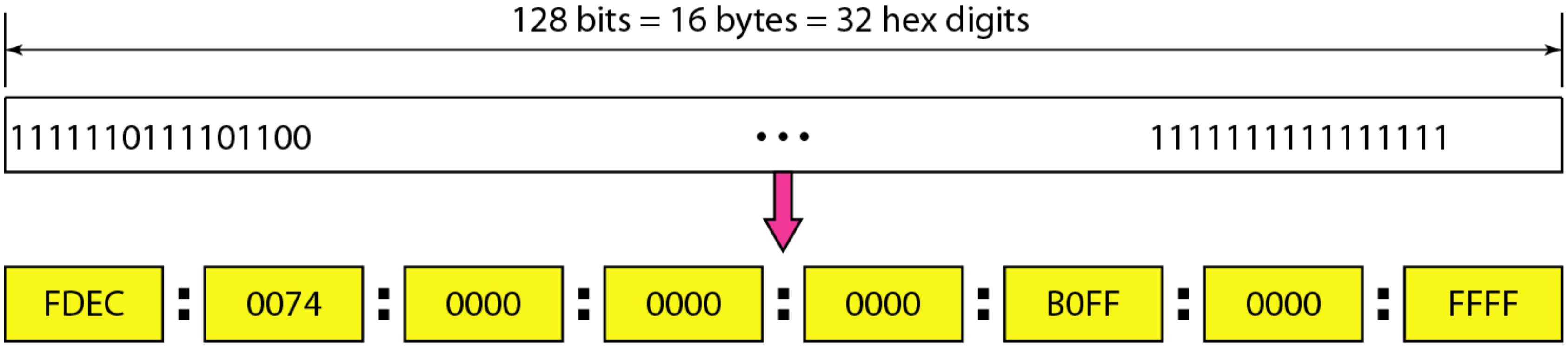
IPv6 ADDRESSES

- ***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.***

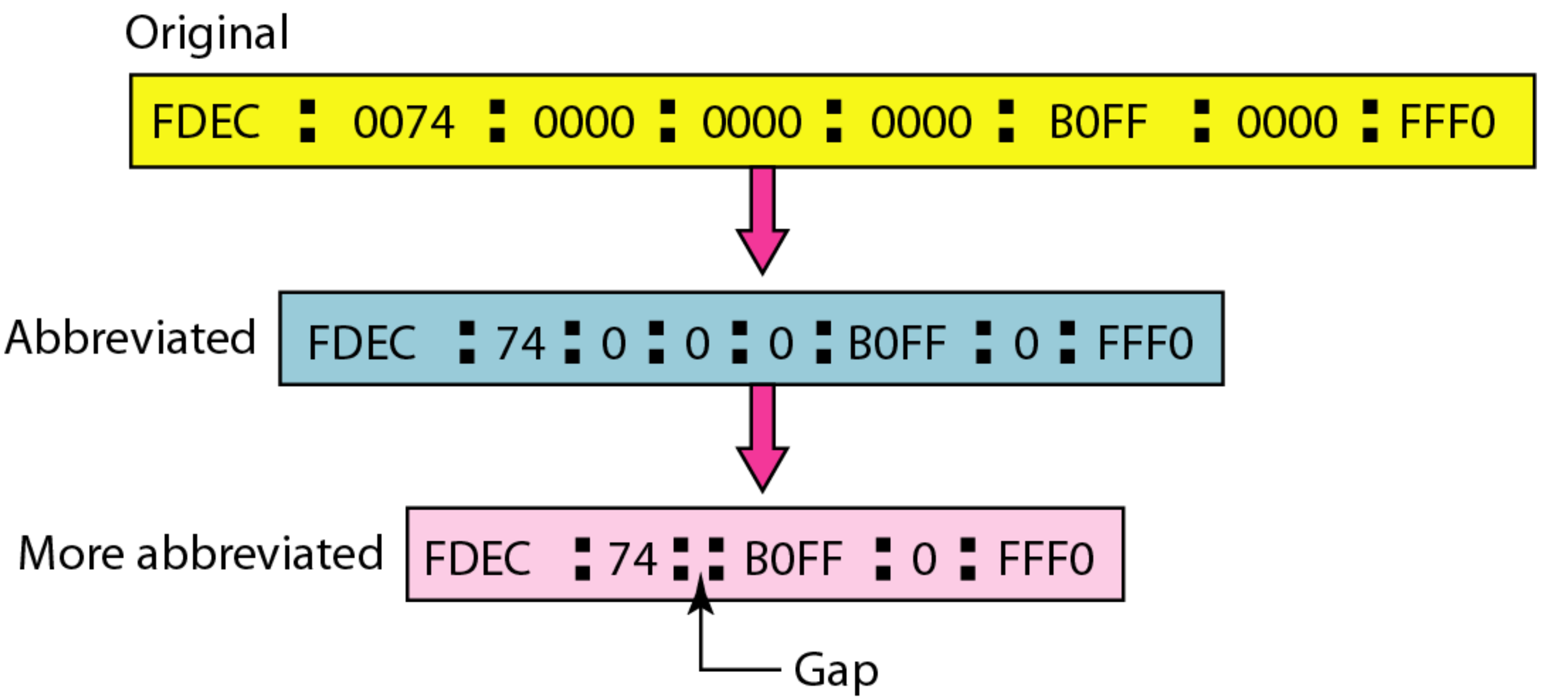
Note

An IPv6 address is 128 bits long.

IPv6 address in binary and hexadecimal colon notation



Abbreviated IPv6 addresses



Example

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
```

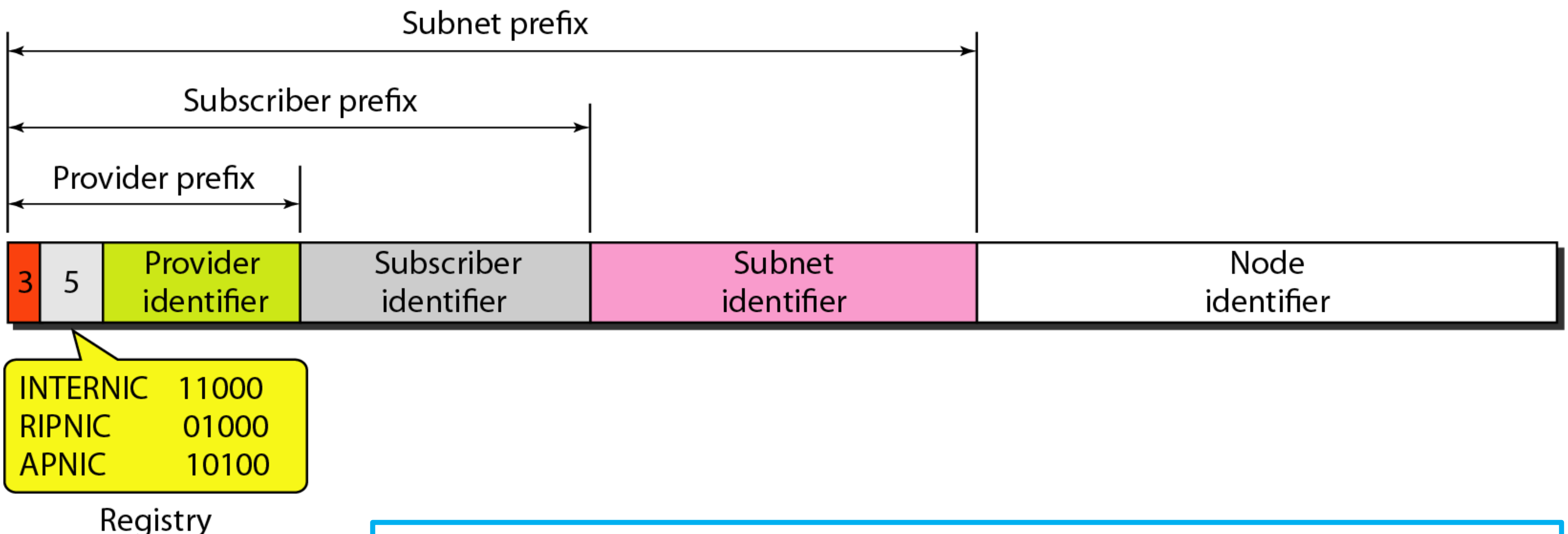
Type prefixes for IPv6 addresses

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**

Type prefixes for IPv6 addresses (continued)

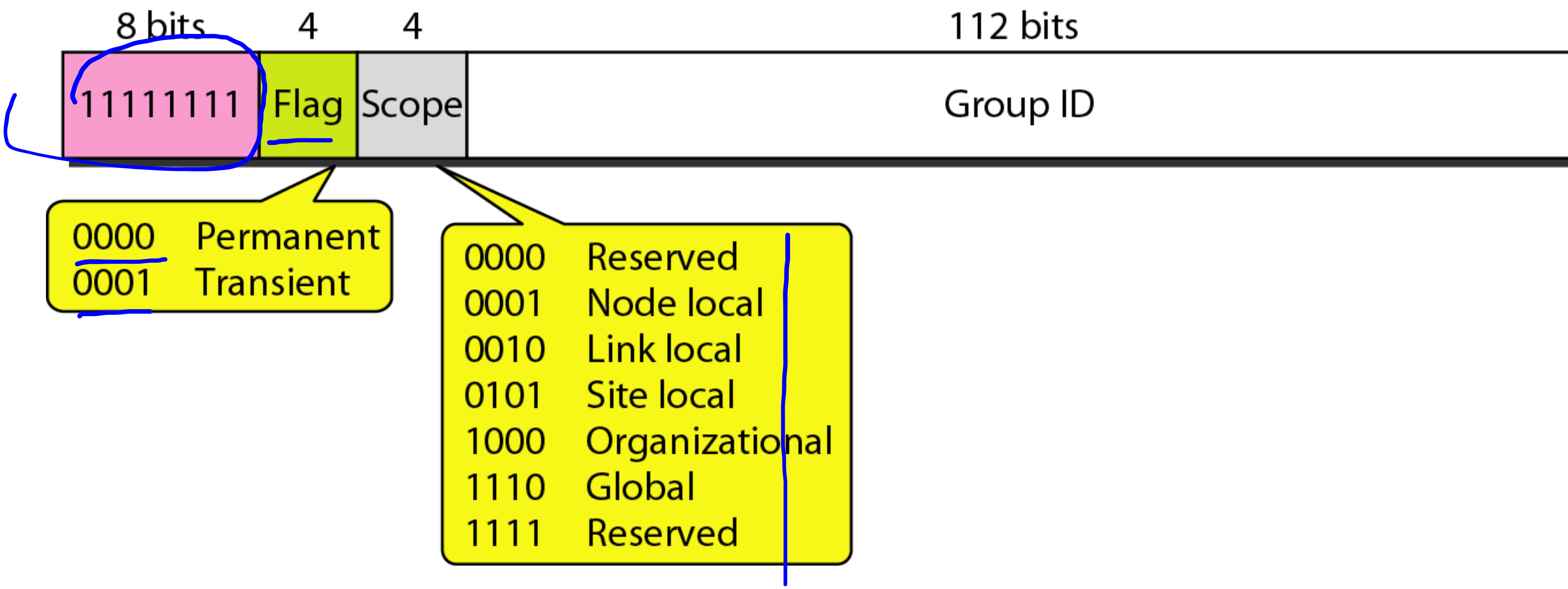
Type Prefix	*Type*	*Fraction*
011	Unassigned	1/8
100	Geographic-based unicast addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 1110 0	Unassigned	1/512
1111 1110 10	Link local addresses	1/1024
1111 1110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

Prefixes for provider-based unicast address



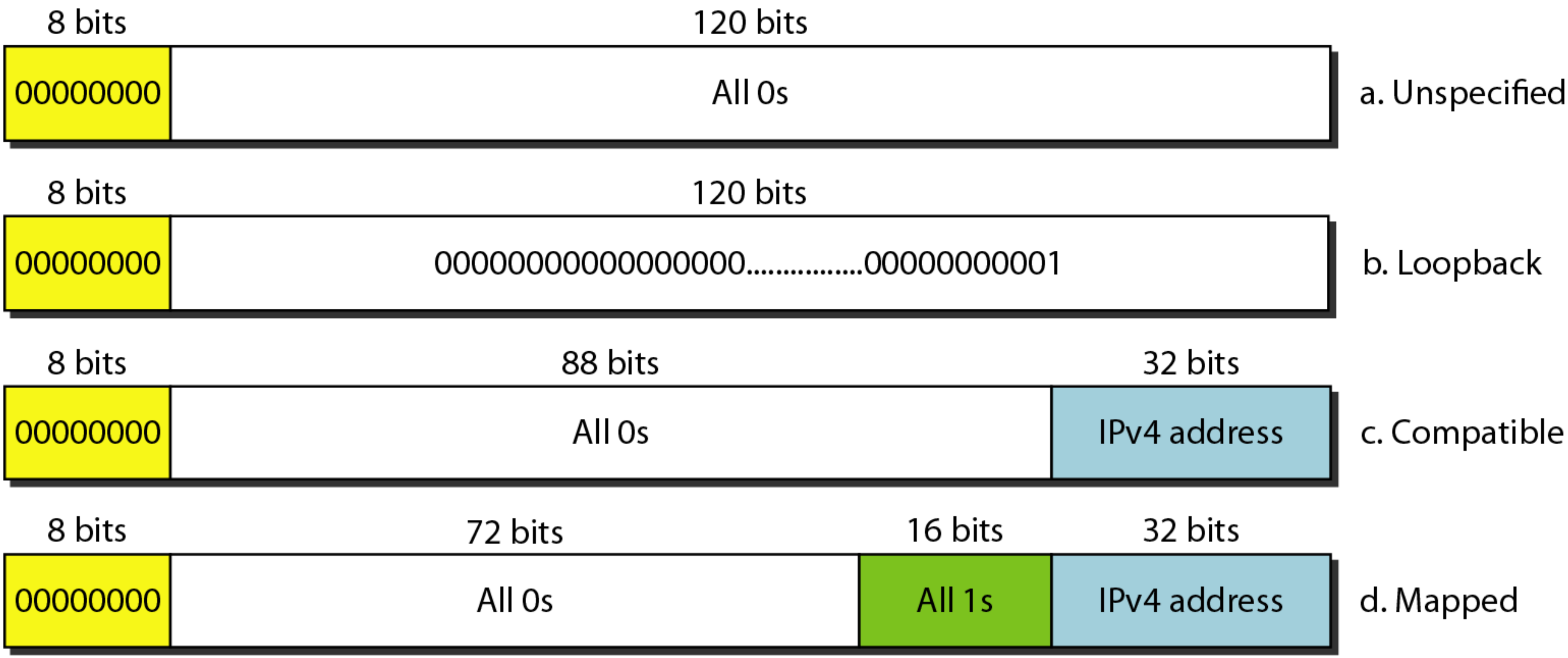
- **INTERNIC : This is the centre for North America**
- **RIPNIC : This is the centre for European registration and**
- **APNIC : This is for Asian and Pacific countries.**

Multicast address in IPv6



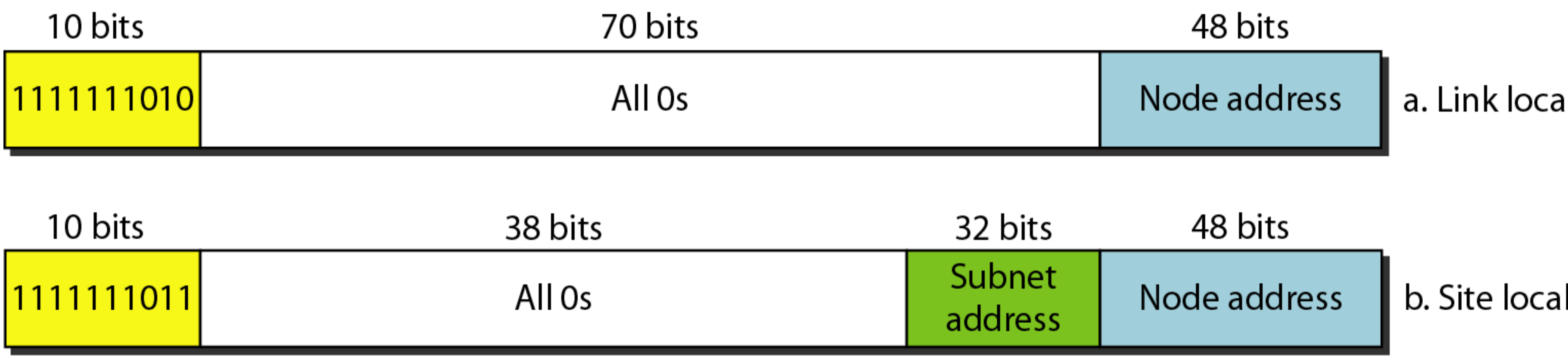
- A **permanent group address** is defined by the Internet authorities and can be accessed at all times.
- A **transient group address**, on the other hand, is used only temporarily.

Reserved addresses in IPv6



Local addresses in IPv6

- These addresses are used when an organization wants to use IPv6 protocol without being connected to the global Internet.
  - So, it is the same as the private network



- A link local address is used in an isolated subnet
- A site local address is used in an isolated site with several subnets.