# Module 4
# Network Layer
## *Network Layer Addressing*
**BECE401L**

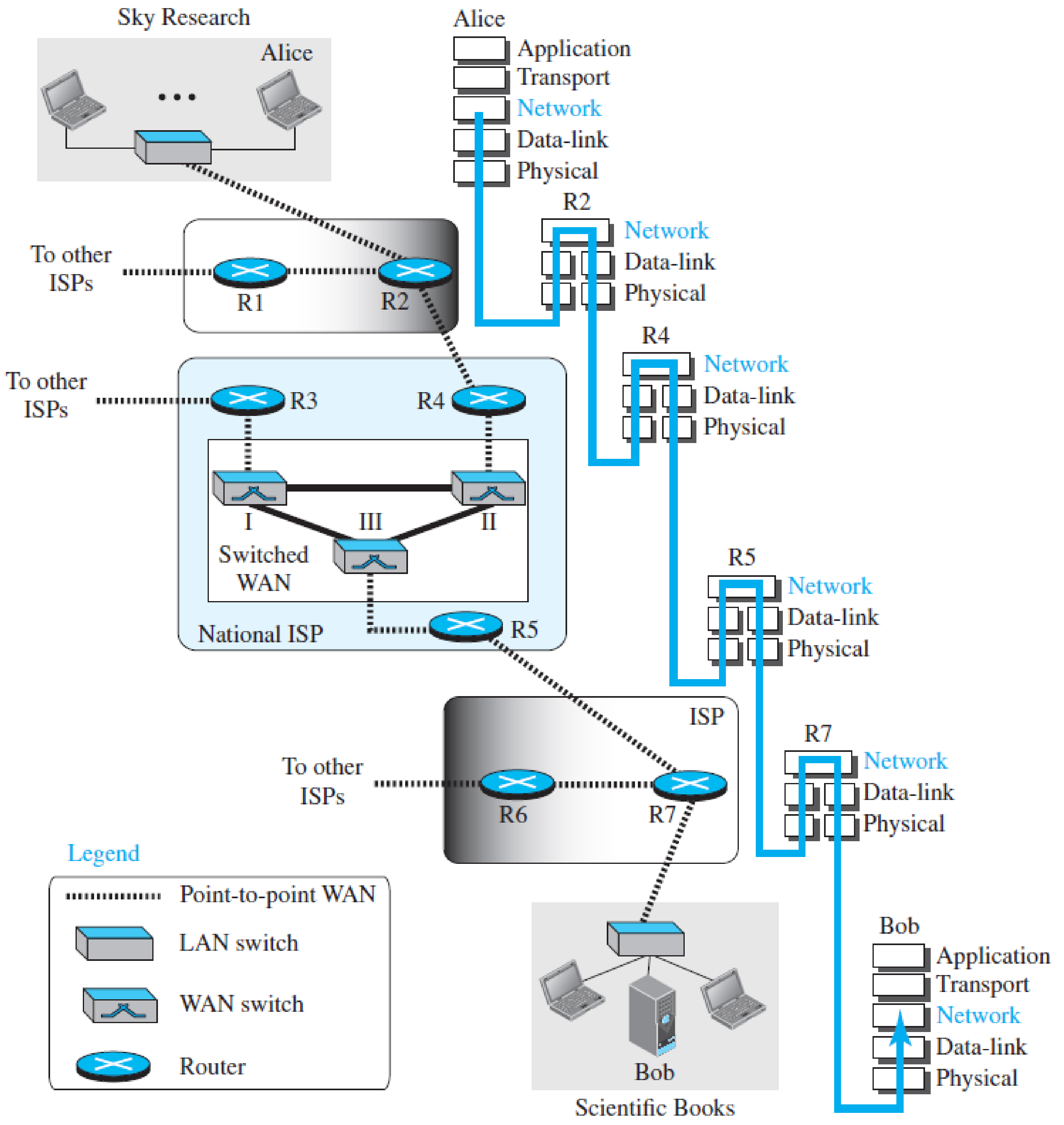
# Outline

Internetworking - IP Addressing - Subnetting - IPv4 and IPv6

# Network Layer

- Internet is an internetwork,
  - Combination of LANs and WANs

- Example:
  - The network layer is involved at the source host, destination host, and all routers in the path (R2, R4, R5, and R7).



# Network Layer Services

## Packetization

- **Encapsulating/Decapsulating** the payload in a network-layer packet at the **source/destination**.
- **Source is not allowed to change the content of the payload**
    - Unless it is too large for delivery
    - So needs to be fragmented.
- **For Fragmented Packet; NW layer is responsible:**
    - Waiting until all fragments arrive,
    - Reassembling them, and
    - Delivering them to the upper-layer protocol

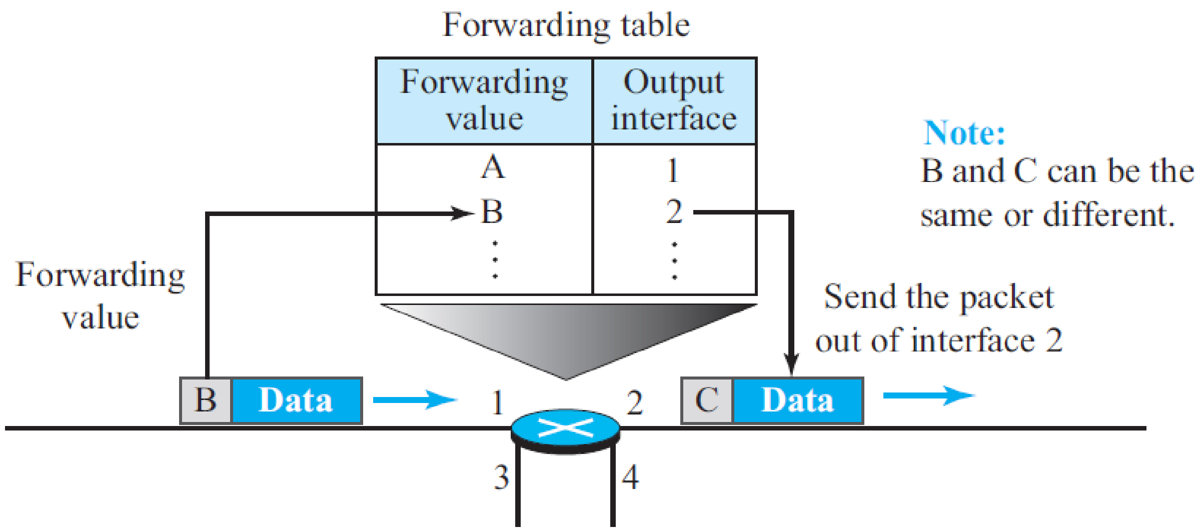
# Network Layer Services

## Routing

- NW Layer is responsible for routing the packet from its source to the destination.
- Various LANs/WANs and routers form the network
  - There is more than one route from the source to the destination.
  - **NW Layer finds the best one among these possible routes.**

## Forwarding

- **Routing** is used to create the decision-making tables for each router,
- ***Forwarding*** is the action applied by each router when a packet arrives.
- This decision-making table a is sometimes called the *forwarding table* and sometimes the *routing table*



# IPv4 Addressing

## Internet Address or IP address

- The identifier used in the IP layer of the TCP/IP protocol suite to identify the connection of each device to the Internet.

## IPv4 address: A 32-bit address

## IPv4 address are Uniquely and Universal

- Defines the connection of a host or a router to the Internet.
- IP address is the address of the connection, Not the host or the router:
  - As if the device is moved to another network, the IP address may be changed.
  - As each address defines one, and only one, connection to the Internet.
  - If a device has two connections to the Internet, via two networks, it has two IPv4 addresses.
- IPv4 addresses are universal
  - The addressing system must be accepted by any host that wants to be connected to the Internet.

# Address Space

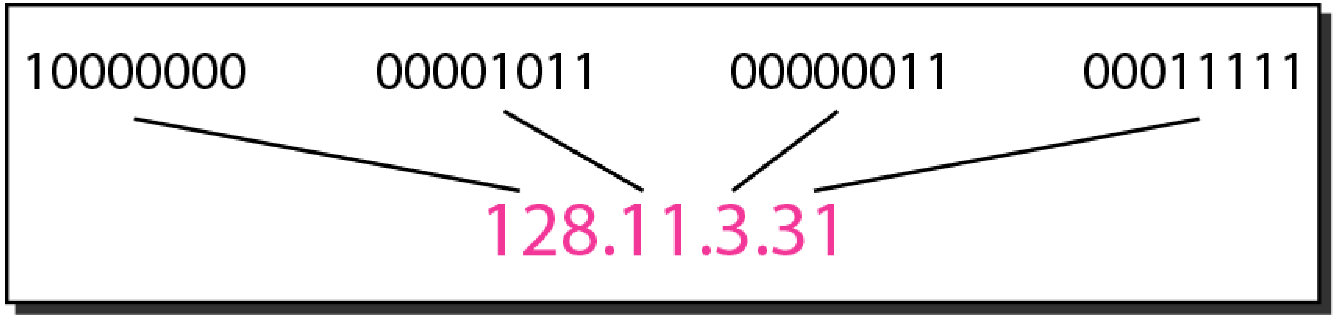
**Address space: Total number of addresses used by the protocol.**

**Address space of IPv4: $2^{32}$ or 4,29,49,67,296**

- If there were no restrictions, more than 4 billion devices could be connected to the Internet.

**Notation:**

- **Binary notation (base 2),**
- **Dotted-decimal notation and**
- **Hexadecimal notation (base 16).**




Dotted-decimal notation and binary notation for an IPv4 address

*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

*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.*

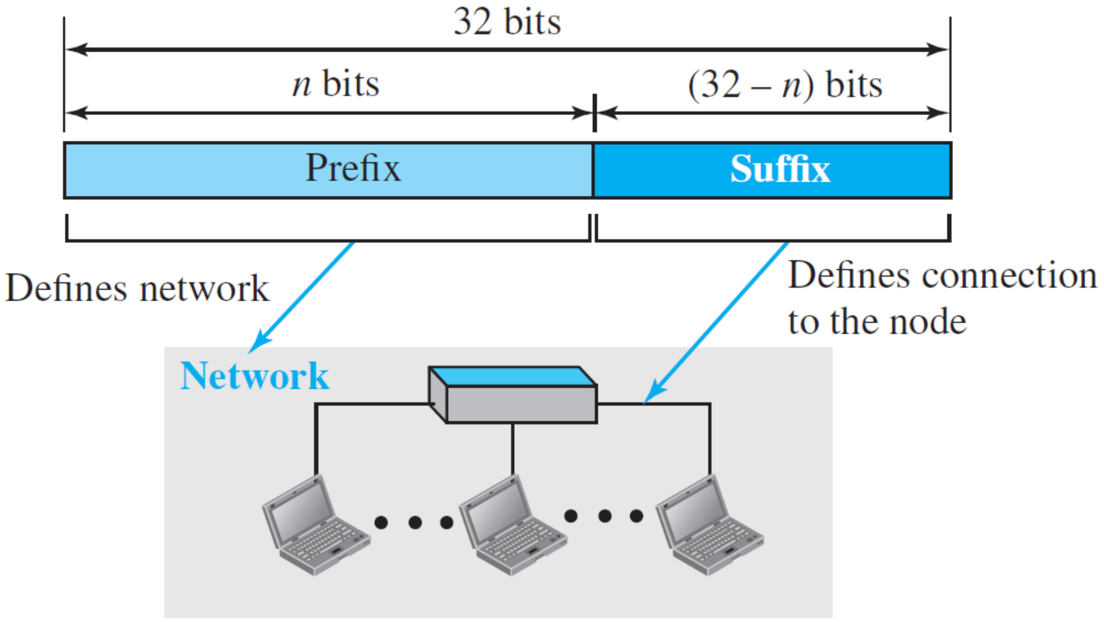
# Hierarchy in Addressing

## Communication network: The addressing system is hierarchical.

- ***Postal network:*** Includes the country, state, city, street, house number, and the name of the mail recipient.
- ***Telephone number:*** Includes the country code, area code, local exchange, and the connection.

## A 32-bit IPv4 address is also hierarchical:

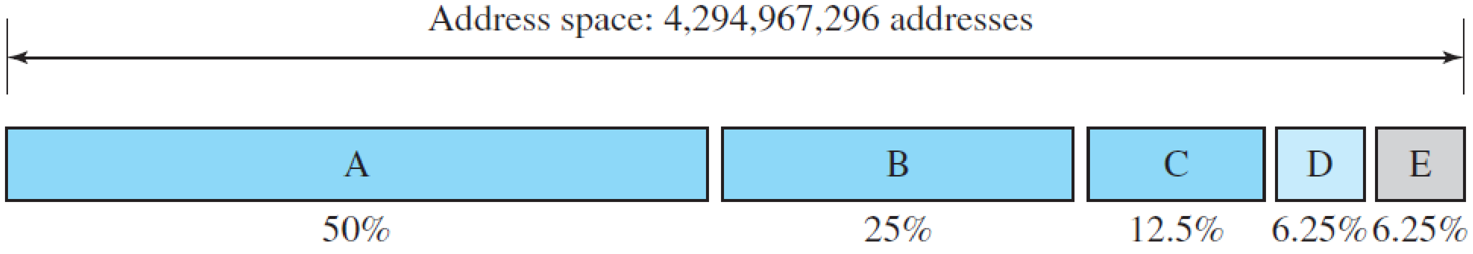
- **Prefix:**
  - Defines the network.
  - Length is *n* bits
  - Can be fixed length or variable length
    - Classful Addressing: Network identifier is fixed-length prefix (Obsolete).
    - Classless Addressing: Uses a variable-length network prefix.
- ***Suffix:***
  - Defines the node (connection of a device to the Internet).
  - Length is (32 − *n*) bits

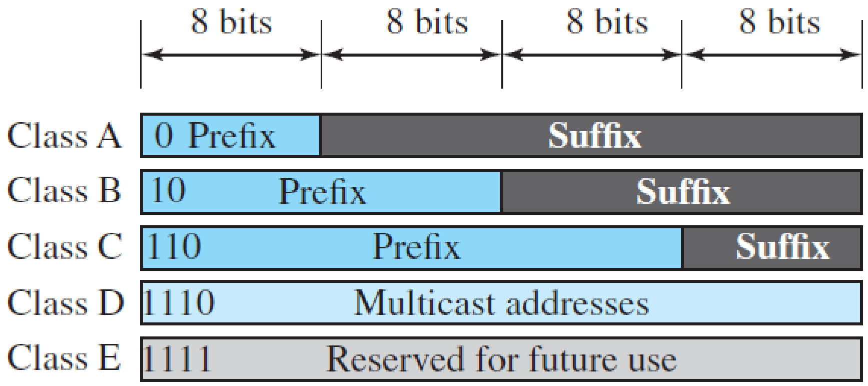


# Classful Addressing

**Classful Addressing:** the address space is divided into five classes: A, B, C, D, & E.

- **Class A:**
  - Network length is 8 bits,
  - The first bit, ‘0’, defines the class,
  - Only seven bits as the network identifier.
  - Only $2^7 = 128$ networks globally
- **Class B:**
  - Network length is 16 bits
  - 1st two bits, which are $(10)_2$, define the class
  - Only 14 bits as the network identifier.
  - Only $2^{14} = 16{,}384$ networks globally.
- **Class C:**
  - Network length is 24 bits
  - Addresses that start with $(110)_2$.
  - 3 bits define the class
  - Only 21 bits as the network identifier.
  - There are $2^{21} = 2{,}097{,}152$ networks globally.




| Class | Prefixes | First byte |
|---|---|---|
| A | $n$ = 8 bits | 0 to 127 |
| B | $n$ = 16 bits | 128 to 191 |
| C | $n$ = 24 bits | 192 to 223 |
| D | Not applicable | 224 to 239 |
| E | Not applicable | 240 to 255 |

- **Class D:** Not divided into prefix and suffix and is used for multicast addresses.
- **Class E:** All addresses that start with 1111 and is reserved for future use

# Classful Addressing

| Address Class | 1st Octet range in decimal | 1st Octet bits (Blue Dots do not change) | Network (N) and Host (H) Portion | Default mask (Decimal) | Number of possible networks and hosts per network |
|---|---|---|---|---|---|
| A | 0-127 | 00000000 - 01111111 | N.H.H.H | 255.0.0.0 | 128 Nets ($2^7$) 16,777,214 hosts ($2^{24}$-2) |
| B | 128-191 | 10000000 - 10111111 | N.N.H.H | 255.255.0.0 | 16,384 Nets ($2^{14}$) 65,534 hosts ($2^{16}$-2) |
| C | 192-223 | 11000000 - 11011111 | N.N.N.H | 255.255.255.0 | 2,09,150 Nets ($2^{21}$) 254 hosts ($2^8$-2) |
| D | 224-239 | 11100000 - 11101111 | NA (Multicast) | - | - |
| E | 240-255 | 11110000 - 11111111 | NA (Experimental) | - | - |

***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.*

# Classful Addressing

## Address Depletion

### Reason that classful addressing has become obsolete.

- As the addresses were not distributed properly
- The Internet was faced with the problem of the addresses being rapidly used up,
- Resulting in no more addresses.

### Example:

- Class A.
  - This class can be assigned to only 128 organizations in the world,
  - But each organization needs to have a single network (rest of the world) with 1,67,77,216 nodes (computers in this single network).
  - Since there may be only a few organizations that are this large
  - Most of the addresses in this class were wasted (unused).
- Class B addresses were designed for midsize organizations
  - But many of the addresses in this class also remained unused.
- Class C addresses have a completely different flaw in design.
  - The number of addresses that can be used in each network (256)
  - So small that most companies were not comfortable using a block in this address class.
- Class E addresses were almost never used, wasting the whole class.

# Classful Addressing

## Subnetting & Supernetting

To address the problem of address depletion in classful addressing, Two strategies were proposed:

### Subnetting

- class A or class B block is divided into several subnets.
- Each subnet has a larger prefix length than the original network.
- For example,
  - If a network in class A is divided into four subnets, each subnet has a prefix of $n_{sub} = 10$.
  - At the same time, if all of the addresses in a network are not used, subnetting allows the addresses to be divided among several organizations.
  - This idea did not work because most large organizations were not happy about dividing the block and giving some of the unused addresses to smaller organizations.

### Supernetting.

- Devised to combine several class C blocks into a larger block to be attractive to organizations that need more than the 256 addresses available in a class C block.
- This idea did not work either because it makes the routing of packets more difficult.

# Classless Addressing

Subnetting and Supernetting in Classful Addressing did not solve the Address Depletion Problem.

**With the growth of the Internet:**

- A **larger address space** was needed as a long-term solution.
- **IPv6 :** Requires length of IP addresses also be increased and Means the format of the IP packets needs to be changed.
- ***Classless Addressing:*** A short-term solution: *Use the same address space but to change the distribution of addresses*.

**Class privilege was removed from the distribution to compensate for the address depletion.**

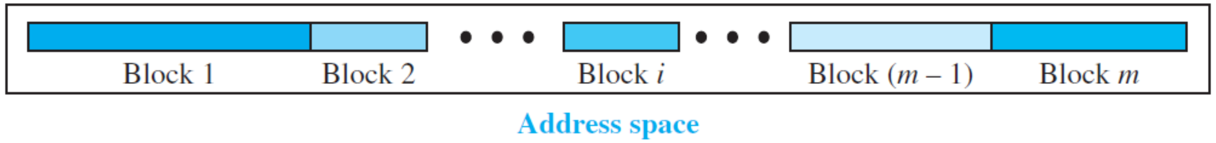
## Motivation for Classless Addressing.

- In 1990s, ISPs came into prominence provides Internet access for individuals, small businesses, and midsize organizations.
- An ISP is granted a large range of addresses
- Then subdivides the addresses (in groups of 1, 2, 4, 8, 16, and so on)
- Giving a range of addresses to a household or a small business via a dial-up modem, DSL, or cable modem to the ISP.
- Each customer needs some IPv4 addresses.

# Classless Addressing

In 1996, the new architecture called **Classless Addressing.**

- Variable-length blocks are used that belong to no classes.
- Whole address space is divided into variable length blocks.
- **Prefix in an address defines the block (network)**
- **Suffix defines the node (device).**
- Theoretically, A block of $2^0, 2^1, 2^2, \ldots, 2^{32}$ addresses.
- An organization can be granted one block of addresses.




Division of the whole address space into nonoverlapping blocks.

**Unlike classful addressing,**

- **Prefix length** in classless addressing is **variable**.
- Prefix length that ranges from 0 to 32.
- **Size of the network is inversely proportional to the length of the prefix.**
  - A small prefix means a larger network;
  - A large prefix means a smaller network.
- **Idea of classless addressing can be easily applied to classful addressing.**
  - An address in class A can be thought of as a classless address in which the prefix length is 8.
  - An address in class B can be thought of as a classless address in which the prefix is 16, and so on.
  - In other words, classful addressing is a special case of classless addressing.

# Classless Addressing

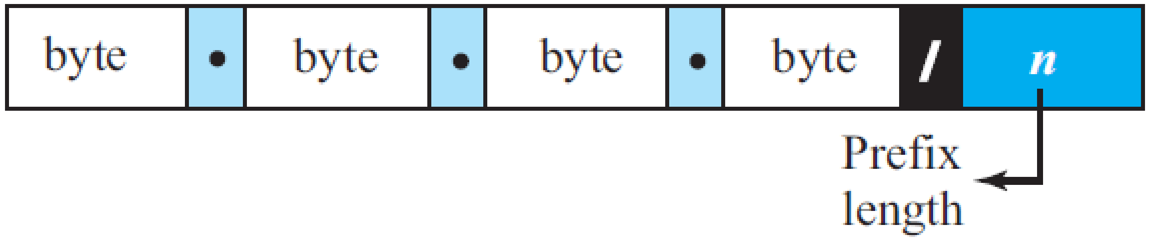
## Prefix Length

In classless addressing is how to find the prefix length if an address is given?

- A the prefix length is not inherent in the address.
- The length of the prefix is specified separately.

The **prefix length, *n*,** is added to the address, separated by a slash.

*Slash Notation* or *Classless Interdomain Routing* or *CIDR* strategy.




**Examples:**
12.24.76.8/8
23.14.67.92/12
220.8.24.255/25

# *IPv4 Addressing*
# Classless Addressing

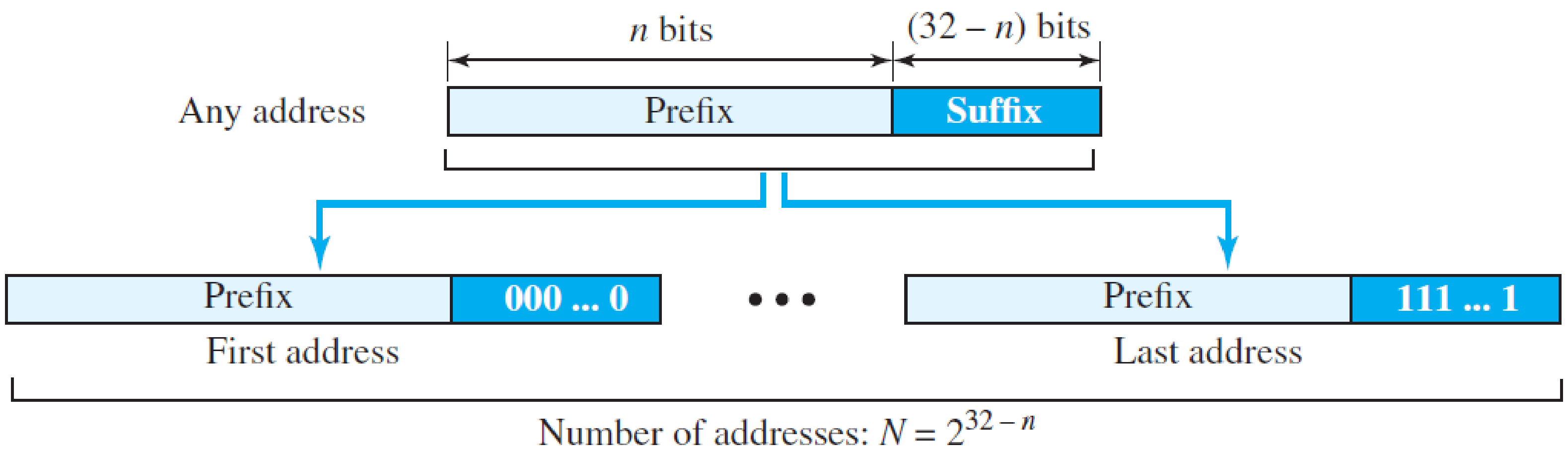
## *Extracting Information from an Address*

Three pieces of information about the block to which the address belongs:

- The number of addresses, The first address in the block, and The last address.

Since the value of prefix length, *n*, is given, we can easily find these three pieces of information, as shown in Figure.

- The number of addresses in the block is found as $N = 2^{32-n}$.
- 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.



**A classless address is given as 167.199.170.82/27.**

We can find the above three pieces of information as follows.

The **number of addresses** in the network is $2^{32-n} = 2^5 = 32$ addresses.

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.

| | | | | |
|---|---|---|---|---|
| Last address: 167.199.170.95/27 | 10100111 | 11000111 | 10101010 | 01011111 |

**A block of 16 addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the first and last 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* ***FIRST ADDRESS***

*11001101 00010000 00100101 00100000*

*or*

*205.16.37.32*

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

*11001101 00010000 00100101 00101111*

*or*

*205.16.37.47*

# Classless Addressing

## *Address Mask*

Another way to find the first and last addresses: *To use the address mask*.

### Address mask is a 32-bit number

- The *n* leftmost bits are set to 1s and
- The rest of the bits (32 − *n*) are set to 0s.

A computer can easily find the address mask because it is the complement of $(2^{32-n} - 1)$. using the three bit-wise operations NOT, AND, and OR.

- Number of addresses in the block N = **NOT** (mask) + 1.
- First address in the block = (Any address in the block) **AND** (mask).
- Last address in the block = (Any address in the block) **OR** [(**NOT** (mask)].

*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. 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.*

*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 |

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 |

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 |

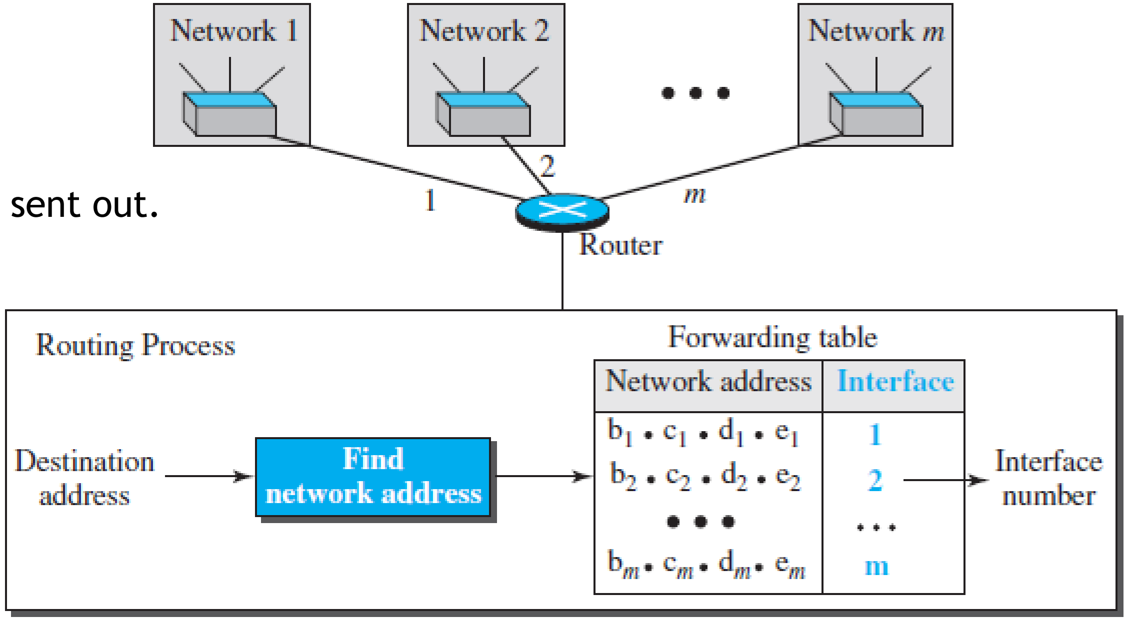
# Classless Addressing

## *Network Address*

Important because it is used in **routing a packet to its destination network**.

Example:

- Let us assume that an internet is made of *m* networks and a router with *m* interfaces.
- When a packet arrives at the router from any source host,
- The router needs to know to which network the packet should be sent:
  - From which interface the packet should be sent out?
- **After the network address has been found,**
- **The router consults its forwarding table**
  - To find the corresponding interface from which the packet should be sent out.




Network address is actually the identifier of the network

Each network is identified by its network address

# Classless Addressing

## *Block Allocation*

### How are the blocks allocated?

Responsibility of a global authority called the **Internet Corporation for Assigned Names and Numbers (ICANN)**.

- Assigns a large block of addresses to an ISP not to individual users.

**For the proper operation of the CIDR, Two restrictions need to be applied:**

- **The number of requested addresses, *N*,**
  - Needs to be a power of 2.
  - The reason is that $N = 2^{32-n}$ or $n = 32 - \log_2 N$.
  - If $N$ is not a power of 2, we cannot have an integer value for $n$.
- **The requested block needs to be allocated** where there is an adequate number of neighboring addresses available in the address space.
  - However, there is a restriction on choosing the first address in the block.
  - The first address needs to be divisible by the number of addresses in the block.
  - The reason is that the first address needs to be the prefix followed by $(32 - n)$ number of 0s.

$$\text{first address} = (\text{prefix in decimal}) \cdot 2^{32-n} = (\text{prefix in decimal}) \cdot N$$

# Classless Addressing

## *Subnetting*

More levels of hierarchy can be created using subnetting.

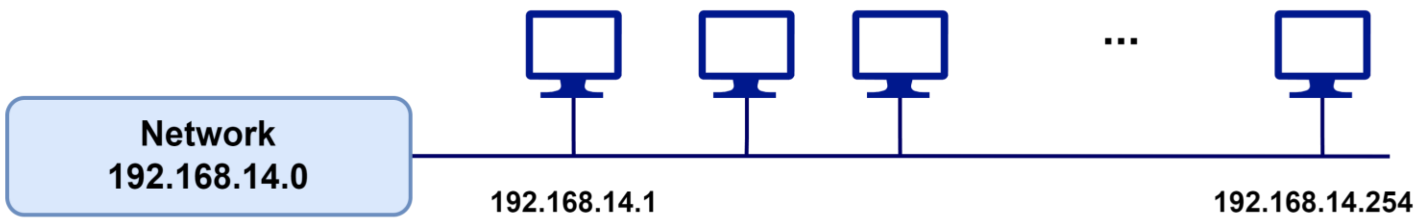
- An organization (may divide the range into several subranges and assign each subrange to a subnetwork (or subnet).
  - A subnetwork can be divided into several sub-subnetworks.
    - A sub-subnetwork can be divided into several sub-sub-subnetworks, and so on.

### *Designing Subnets*

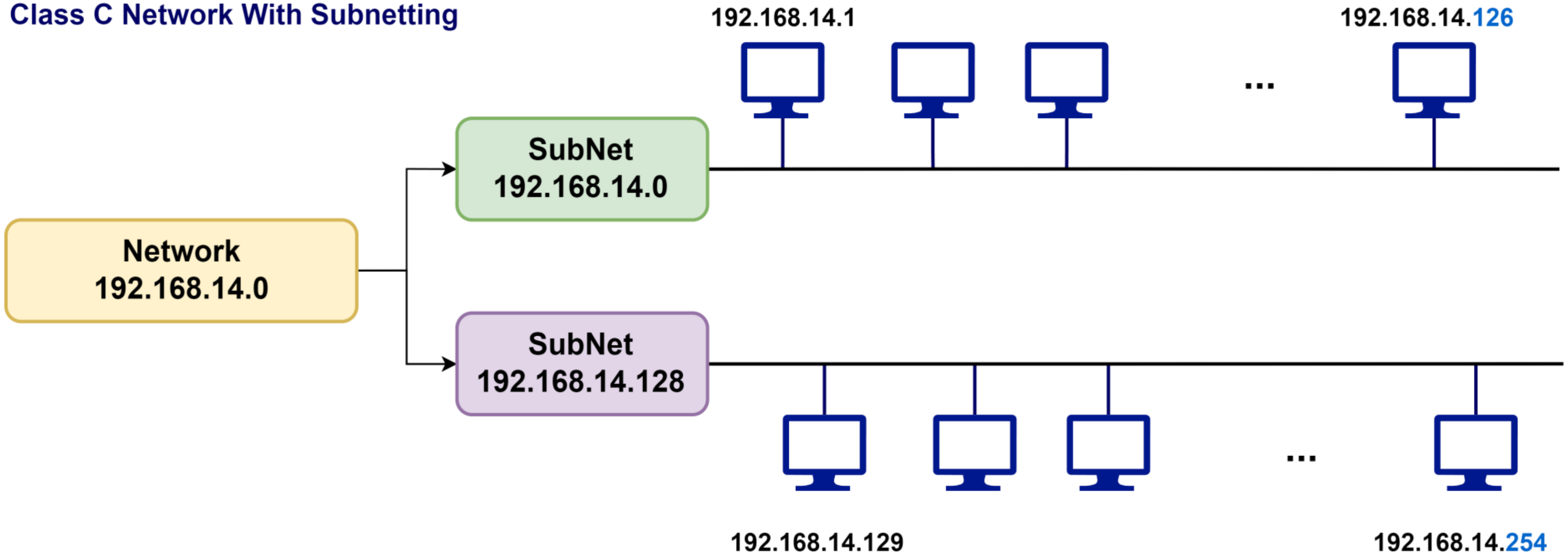
- The subnetworks in a network should be carefully designed to enable the routing of packets.
- Assume:
  - Number of addresses is N,
  - Prefix length is *n*,
  - Assigned number of addresses to each subnetwork is $N_{sub}$, and
  - Prefix length for each subnetwork is $n_{sub}$.

# Subnetting

Class C Network without Subnetting




Class C Network With Subnetting



# Classless Addressing

## *Subnetting*

Following steps for the proper operation of the subnetworks.

- The number of addresses in each subnetwork should be a power of 2.
- The prefix length for each subnetwork should be found using the following formula:

$$first\ address\ = (prefix\ in\ decimal) \times 2^{32-n} = (prefix\ in\ decimal).\ N.$$

- The starting address in each subnetwork should be divisible by the number of addresses in that subnetwork.

**Example:** An ISP has requested a block of 1000 addresses.

- Since 1000 is not a power of 2, 1024 addresses are granted.
- The prefix length is calculated as **$n = 32 - \log_2 1024 = 22$**.
- An available block, 18.14.12.0/22, is granted to the ISP.

### *Finding Information about Each Subnetwork*

- After designing the subnetworks, the information about each subnetwork, such as first and last address, can be found using the process we described to find the information about each network in the Internet.

# Subnetting

A subnetwork or subnet is a **logical subdivision of an IP network**.

The practice of dividing a network into two or more networks is called subnetting.

**Computers** that **belong** to **a subnet** are addressed with an **identical most-significant bit-group** in their **IP addresses**.

**Procedure:**

1. **Identify the class of the IP address and note the Default Subnet Mask.**
2. **Convert the Default Subnet Mask into Binary.**
3. **Note the number of hosts required per subnet and find the Subnet**
4. **Generator (SG) and octet position.**
5. **Generate the new subnet mask. 5. Use the SG and generate the network ranges (subnets) in the appropriate octet position.**

# Subnetting

**Example: Subnet the IP Address 216.21.5.0 into 30 hosts in Each Subnet**

**Procedure:**

1. Identify the class of the IP address and note the Default Subnet Mask. :
   - Class C
2. Convert the Default Subnet Mask into Binary.
   - 255.255.255.0
   - Binary : 1111111 . 1111111 . 1111111 . 00000000
3. Note the number of hosts required per subnet and find the Subnet
   - No. of Hosts per Subnet: 30
   - Binary : 11110 {5 Bits}
     - Reserve 5 bits in original subnet from right to left
4. Generator (SG) and octet position.
   - SG : 32 (1$^{st}$ 1 Bit from right to left)
   - Octet : 4$^{th}$
   - Binary : 1111111 . 1111111 . 1111111 . 11100000
   - 255.255.255.224 or 27 (No. of 1's )

# Subnetting

**Example: Subnet the IP Address 216.21.5.0 into 30 hosts in Each Subnet**

**Procedure:**

- Generate the new subnet mask.
    - 216.21.5.0 - 216.21.5.31 - Can communicate over Switch
    - 216.21.5.32 - 216.21.5.63
    - 216.21.5.64 - 216.21.5.95 - Inter Subnet Comm. Via. Router
    - 216.21.5.96 - 216.21.5.127
    - And so on …

# Example

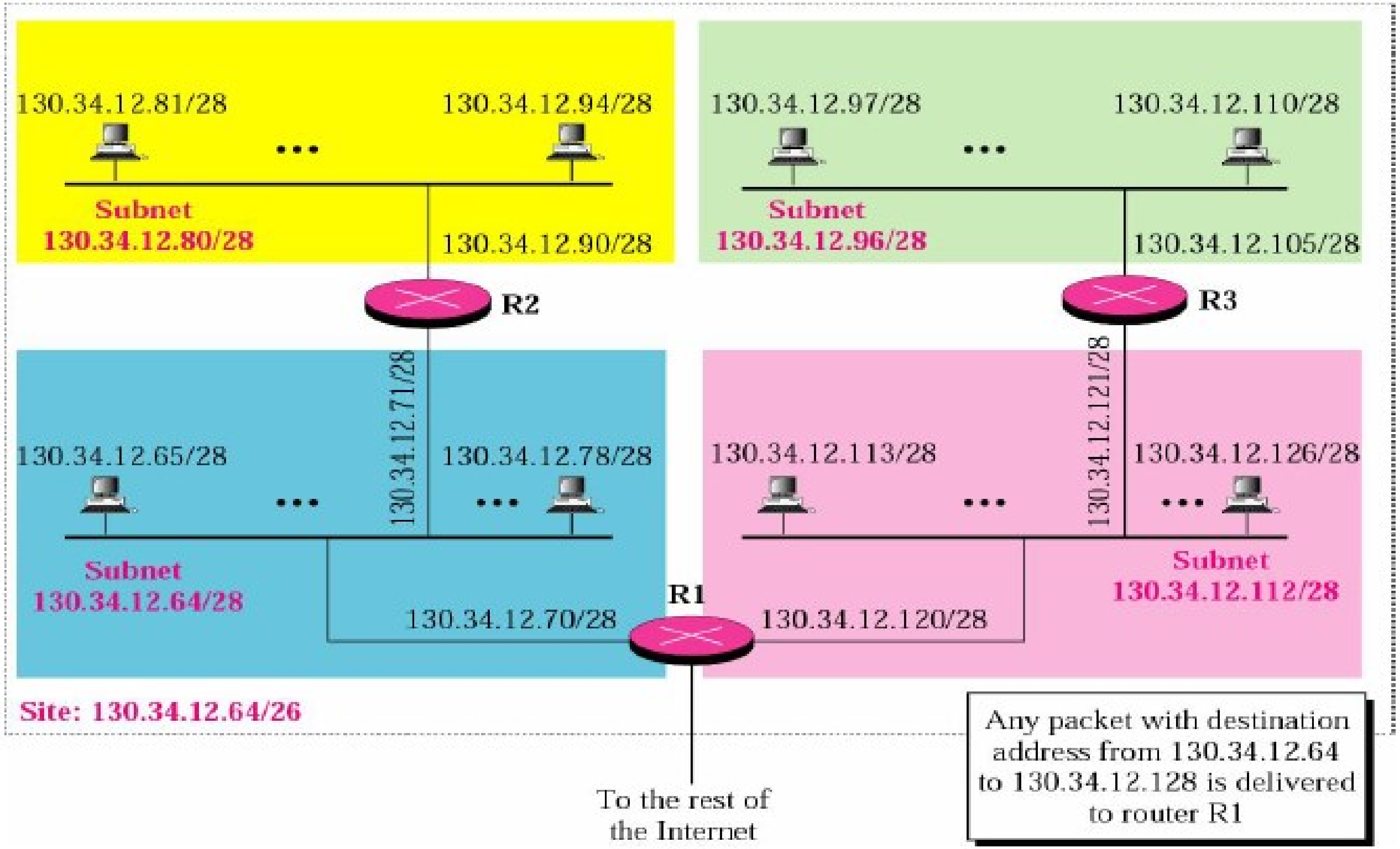
- An organization is granted the block 130.34.12.64/26.
- The organization needs 4 subnets.
- What is the subnet prefix length?
- Solution
  - We need 4 subnets
    - We need to add two more 1s (log2 4 = 2) to the site prefix.
  - The subnet prefix is then /28

# Example

- What are the subnet addresses and the range of addresses for each subnet in the previous example?

- Solution
  - *The site has* $2^{32-26} = 64$ *addresses.*
  - *Each subnet has* $2^{32-28} = 16$ *addresses.*

# Example


130.34.12.81/28
130.34.12.94/28
Subnet
130.34.12.80/28
130.34.12.90/28
130.34.12.97/28
130.34.12.110/28
Subnet
130.34.12.96/28
130.34.12.105/28
R2
R3
130.34.12.71/28
130.34.12.121/28
130.34.12.65/28
130.34.12.78/28
130.34.12.113/28
130.34.12.126/28
Subnet
130.34.12.64/28
Subnet
130.34.12.112/28
R1
130.34.12.70/28
130.34.12.120/28
Site: 130.34.12.64/26
To the rest of the Internet
Any packet with destination address from 130.34.12.64 to 130.34.12.128 is delivered to router R1

# Example

- The first address in the first subnet is 130.34.12.64/28
  - Note that the first address of the first subnet is the first address of the block.
  - The last address of the subnet can be found by adding 15 (16 −1) to the first address.
    - The last address is 130.34.12.79/28

- The first address in the second subnet is 130.34.12.80/28
  - Found by adding 1 to the last address of the previous subnet.
  - Again adding 15 to the first address, we obtain the last address, 130.34.12.95/28.

# Example

- Similarly, we find the first address of the third subnet to be 130.34.12.96/28 and the last to be 130.34.12.111/28

- Similarly, we find the first address of the fourth subnet to be 130.34.12.112/28 and the last to be 130.34.12.127/28

# Variable-Length Subnets

- In previous examples
  - All of subnets have the same mask
- Variable-length subnet
  - Design subnets of different sizes

# Example

- An organization is granted a block of addresses with the beginning address 14.24.74.0/24.
  - There are 2^(32−24)= 256 addresses in this block.
- The organization needs to have 11 subnets as shown below:

  a. two subnets, each with 64 addresses.

  b. two subnets, each with 32 addresses.

  c. three subnets, each with 16 addresses.

  d. four subnets, each with 4 addresses.

  Design the subnets

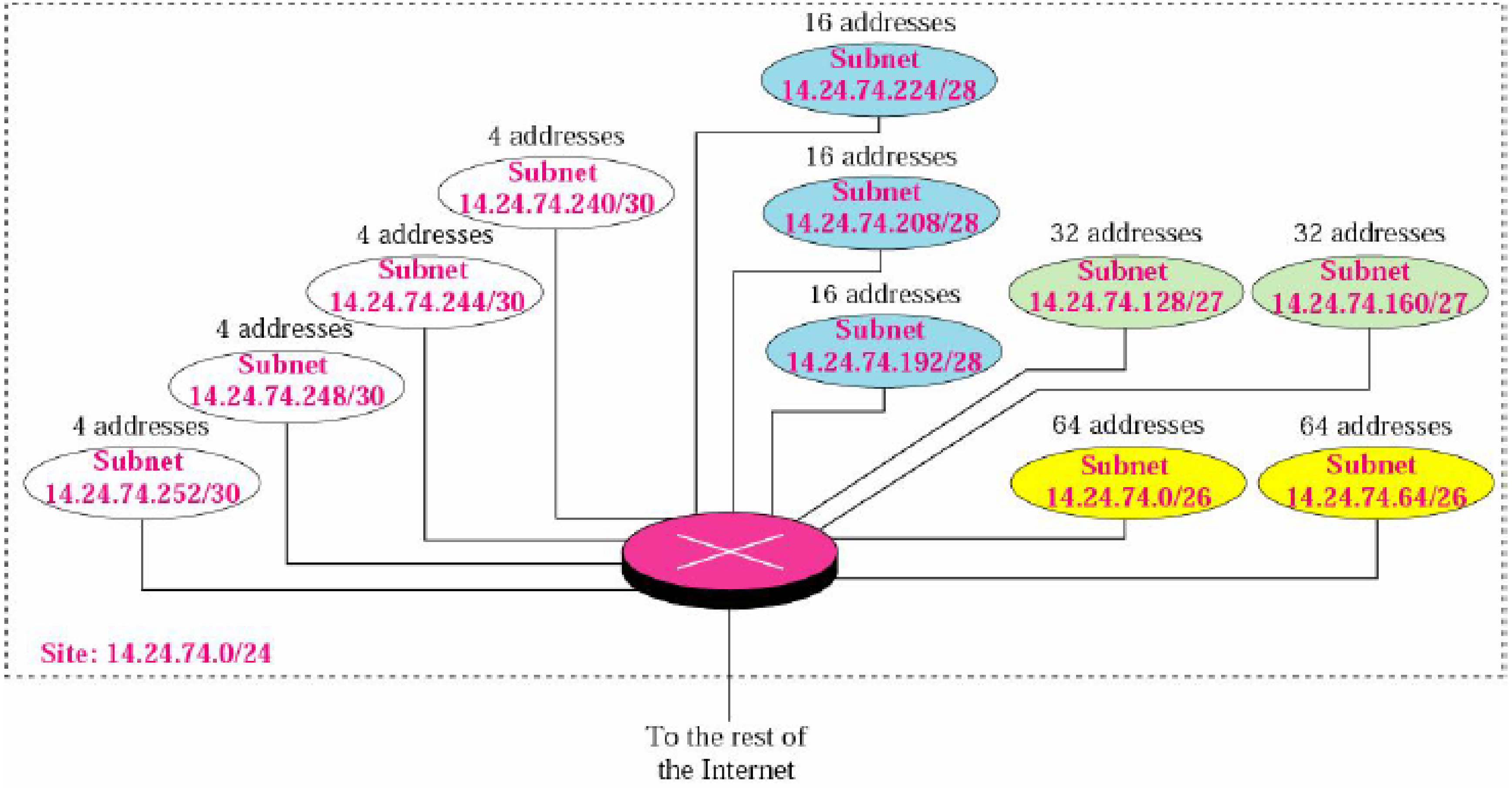
# Example : Solution

- The first 128 addresses are used for the first two subnets, each with 64 addresses.
  - The mask for each network is /26.
  - The subnet address for each subnet is given in the figure
- Use the next 64 addresses for the next two subnets, each with 32 addresses.
  - The mask for each network is /27.
  - The subnet address for each subnet is given in the figure.

# Example : Solution (Cont.)

- Use the next 48 addresses for the next three subnets, each with 16 addresses.
  - The mask for each network is /28.
  - The subnet address for each subnet is given in the figure
- Use the last 16 addresses for the last four subnets, each with 4 addresses.
  - The mask for each network is /30.
  - The subnet address for each subnet is given in the figure

# Example



16 addresses
Subnet
14.24.74.224/28
4 addresses
Subnet
14.24.74.240/30
16 addresses
Subnet
14.24.74.208/28
4 addresses
Subnet
14.24.74.244/30
32 addresses
Subnet
14.24.74.128/27
32 addresses
Subnet
14.24.74.160/27
16 addresses
Subnet
14.24.74.192/28
4 addresses
Subnet
14.24.74.248/30
4 addresses
Subnet
14.24.74.252/30
64 addresses
Subnet
14.24.74.0/26
64 addresses
Subnet
14.24.74.64/26
Site: 14.24.74.0/24
To the rest of
the Internet

# EXAMPLE

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

1. The first group has 64 customers; each needs 256 addresses.
2. The second group has 128 customers; each needs 128 addresses.
3. The third group has 128 customers; each needs 64 addresses.

Design the subblocks and give the CIDR notation for each subblock. Find out how many addresses are still available after these allocations.

## Group 1

For this group, each customer needs 256 addresses. This means the suffix length is 8 ($2^8 = 256$). The prefix length is then $32 - 8 = 24$.

01: 190.100.0.0/24 → 190.100.0.255/24

02: 190.100.1.0/24 → 190.100.1.255/24

……………………………………..

64: 190.100.63.0/24 → 190.100.63.255/24

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

## Group 2

For this group, each customer needs 128 addresses. This means the suffix length is 7 ($2^7 = 128$). The prefix length is then $32 - 7 = 25$. The addresses are:

001: 190.100.64.0/25 → 190.100.64.127/25

002: 190.100.64.128/25 → 190.100.64.255/25

128: 190.100.127.128/25 → 190.100.127.255/25

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

## Group 3

For this group, each customer needs 64 addresses. This means the suffix length is 6 ($2^6 = 64$). The prefix length is then $32 - 6 = 26$.

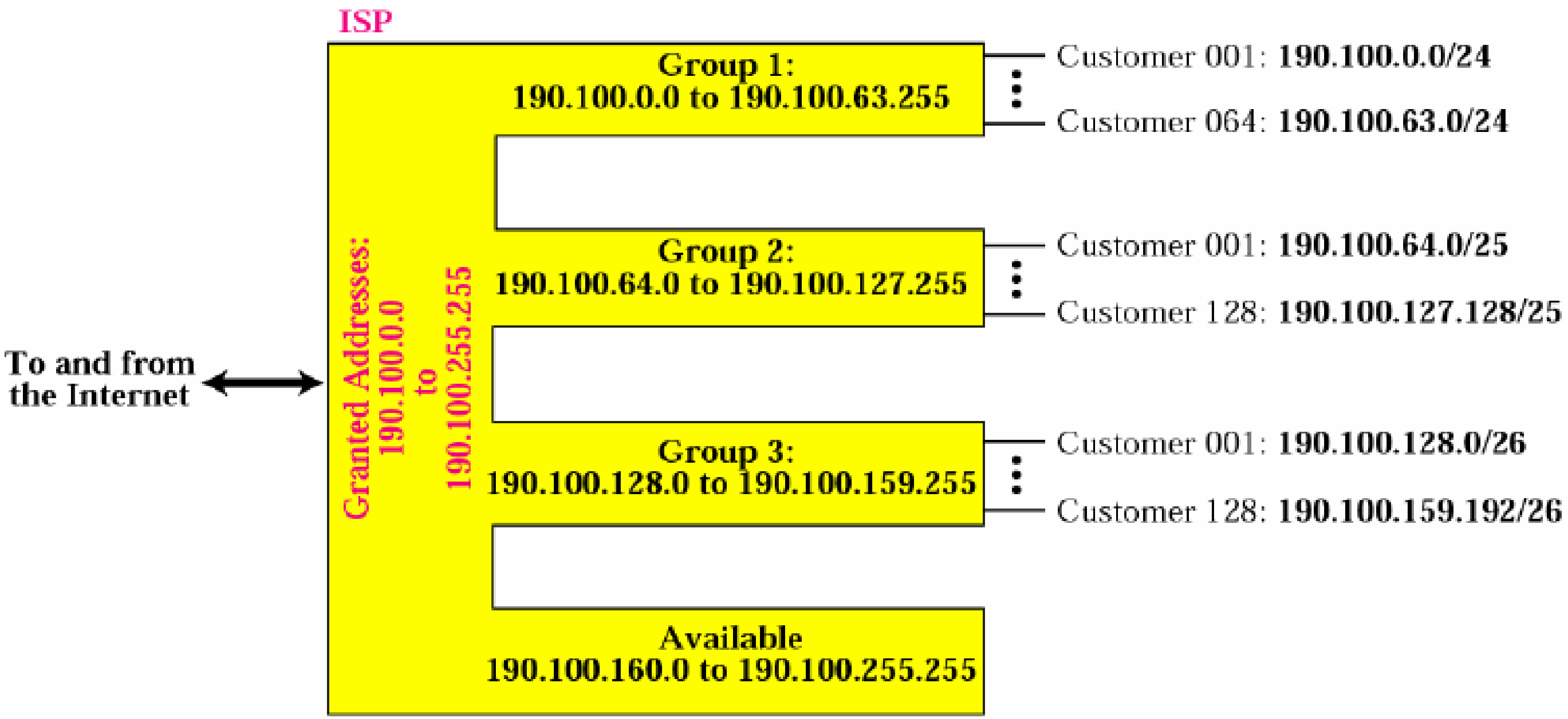
**001**:190.100.128.0/26 → 190.100.128.63/26

**002**:190.100.128.64/26 → 190.100.128.127/26

……………………………

**128**:190.100.159.192/26 → 190.100.159.255/26

**Total = 128 × 64 = 8,192**


ISP
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
Granted Addresses:
190.100.0.0
to
190.100.255.255
To and from
the Internet

# Reference

Forouzan, A. Behrouz. ***Data Communications & Networking***. 5th Edition. Tata McGraw-Hill Education.

**Chapter 18 Network Layer**