# Unit IV: Network Layer

Dr. Manishika Rawat

# NETWORK LAYER

Network Layer Design Issues, Network Address Translation, Internet Protocol (IP): IPv4 and IPv6 addressing; IP Addressing Techniques: Classful Addressing, Classless Addressing, Network and Host Identification, Loopback Address, Broadcast Address, Address Masking; Networks and Subnetworks: Subnetting, Subnet Mask, Supernetting; Network-Layer Protocols: ARP, RARP, IP datagram; Internetworking: Routing and Routing protocols (distance-vector and link-state); Interior and Exterior Gateway Protocol concepts; Routing Algorithms including Dijkstra's algorithm and distributed Bellman-Ford algorithm; Example protocols: OSPF, RIP, BGP, Encapsulation and Tunneling, Congestion Control, Quality of Service, Introduction of Wireshark Tool.

# Network Layer

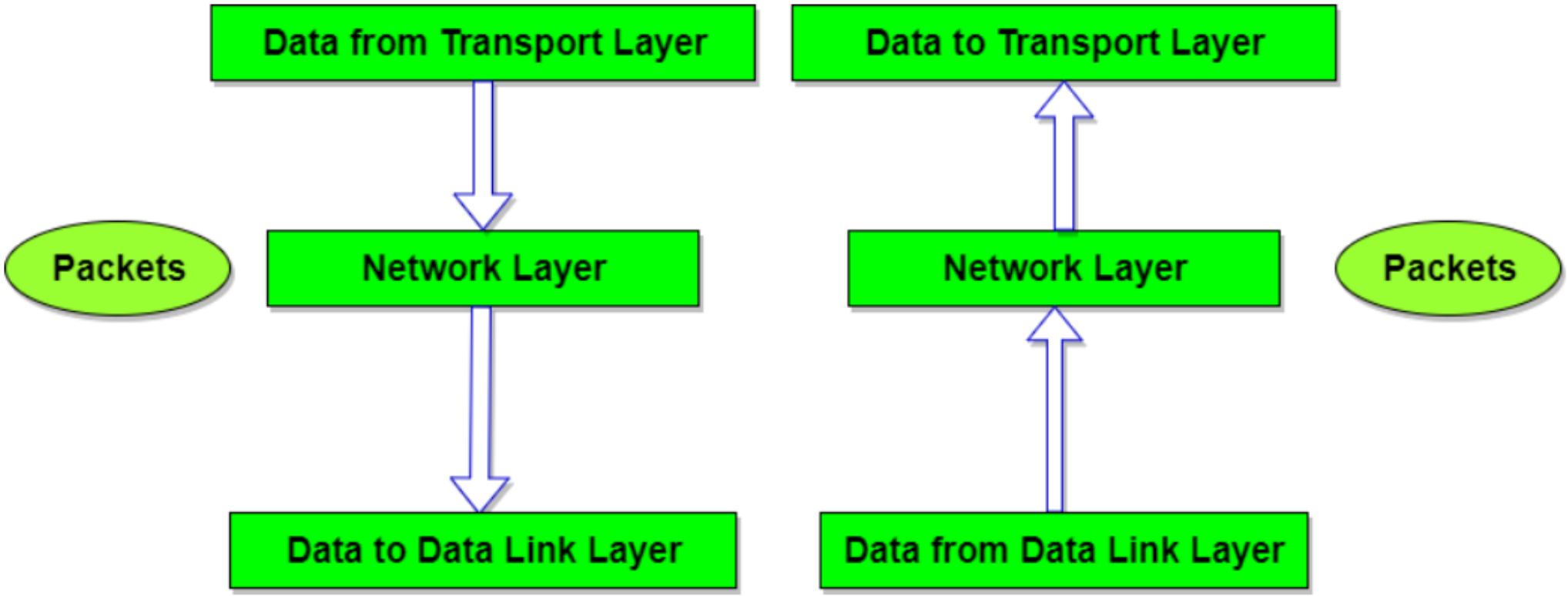
- The Network Layer is the **third layer** of the OSI model.
- It handles the **service requests from the transport layer** and further **forwards the service request to the data link layer**.
- The network layer translates the **logical addresses into physical addresses**
- It determines the **route from the source to the destination** and also manages the traffic problems such as switching, routing and controls the congestion of data packets.
- The main **role of the network layer is to move the packets from sending host to the receiving host.**

# Functions of Network Layer

- **Routing:** When a packet reaches **the router's input link, the router will move the packets to the router's output link**. For example, a packet from S1 to R1 must be forwarded to the next router on the path to S2.
- **Logical Addressing:** The data link layer implements the physical addressing and network layer implements the logical addressing. Logical addressing is also used to **distinguish between source and destination system.** The network layer **adds a header to the packet which includes the logical addresses of both the sender and the receiver.**

# Functions of Network Layer

- **Internetworking:** This is the main role of the **network layer that it provides the logical connection between different types of networks**.
- **Fragmentation:** The fragmentation is a process of **breaking the packets into the smallest individual data units** that travel through different networks.
  - **If the packets are too large for delivery, they are fragmented** i.e., broken down into smaller packets.


Data from Transport Layer
Data to Transport Layer
Packets
Network Layer
Network Layer
Packets
Data to Data Link Layer
Data from Data Link Layer

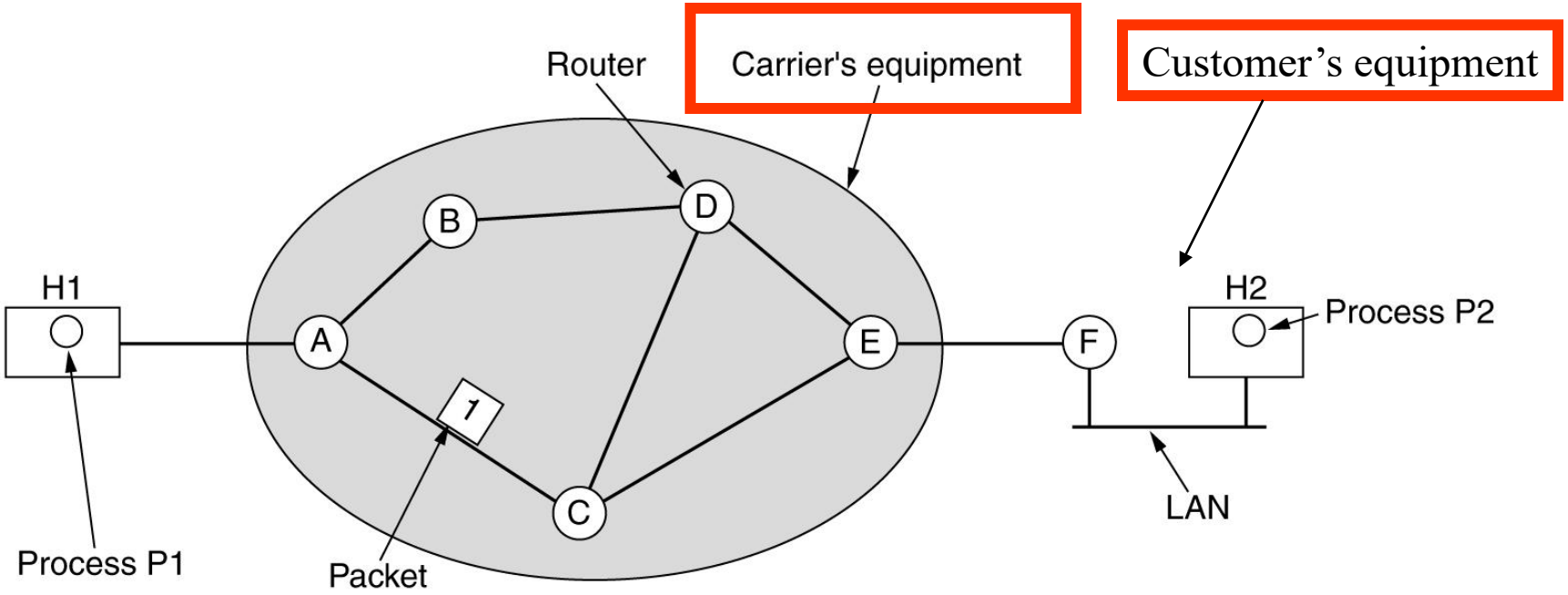
# Features

- Main responsibility of Network layer is **to carry the data packets from the source to the destination without changing or using it.**
- **If the packets are too large for delivery, they are fragmented** i.e., broken down into smaller packets.
- It decides **the route to be taken by the packets to travel from the source to the destination among the multiple routes available in a network** (also called as routing).
- The **source and destination addresses are added to the data packets inside the network layer.**

# Network Layer Design Issues

- **Store-and-Forward Packet Switching**
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

# Store-and-Forward Packet Switching




The environment of the network layer protocols.

- This equipment is used as follows:

- A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the carrier.
- The packet is stored there until it has fully arrived so the checksum can be verified.
- Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered.
- This mechanism is store-and-forward packet switching.

# Network Layer Design Issues

- Store-and-Forward Packet Switching
- **Services Provided to the Transport Layer**
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

# Services Provided to the Transport Layer

The network layer services have been designed with the following goals:

1. The services should be independent of the router technology.
2. The transport layer should be shielded from the number, type, and topology of the routers present.
3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

# Network Layer Design Issues

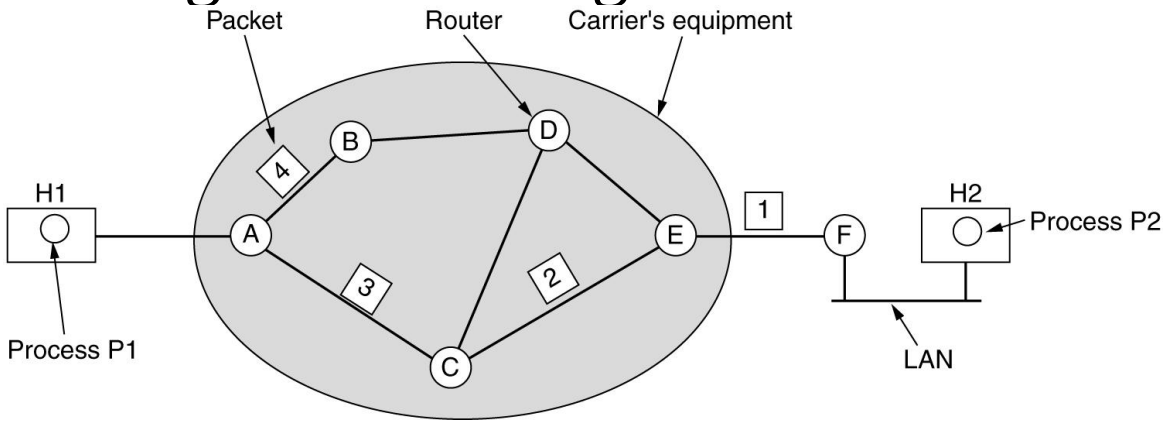
- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- **Implementation of Connectionless Service**
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

# Implementation of Connectionless Service

- If connectionless service is offered, packets are injected into the subnet individually and routed independently of each other.
- No advance setup is needed.
- In this context, the packets are frequently called datagrams and the subnet is called a datagram subnet.

# Implementation of Connectionless Service

## Routing within a diagram subnet.




A's table

| Dest. | initially Line | later Line |
|---|---|---|
| A | – | – |
| B | B | B |
| C | C | C |
| D | B | B |
| E | C | B |
| F | C | B |

C's table

| Dest. | Line |
|---|---|
| A | A |
| B | A |
| C | – |
| D | D |
| E | E |
| F | E |

E's table

| Dest. | Line |
|---|---|
| A | C |
| B | D |
| C | C |
| D | D |
| E | – |
| F | F |

**The question is:** a packet with a destination D arrives at router A. then which router will router A send this packet to?

# Network Layer Design Issues

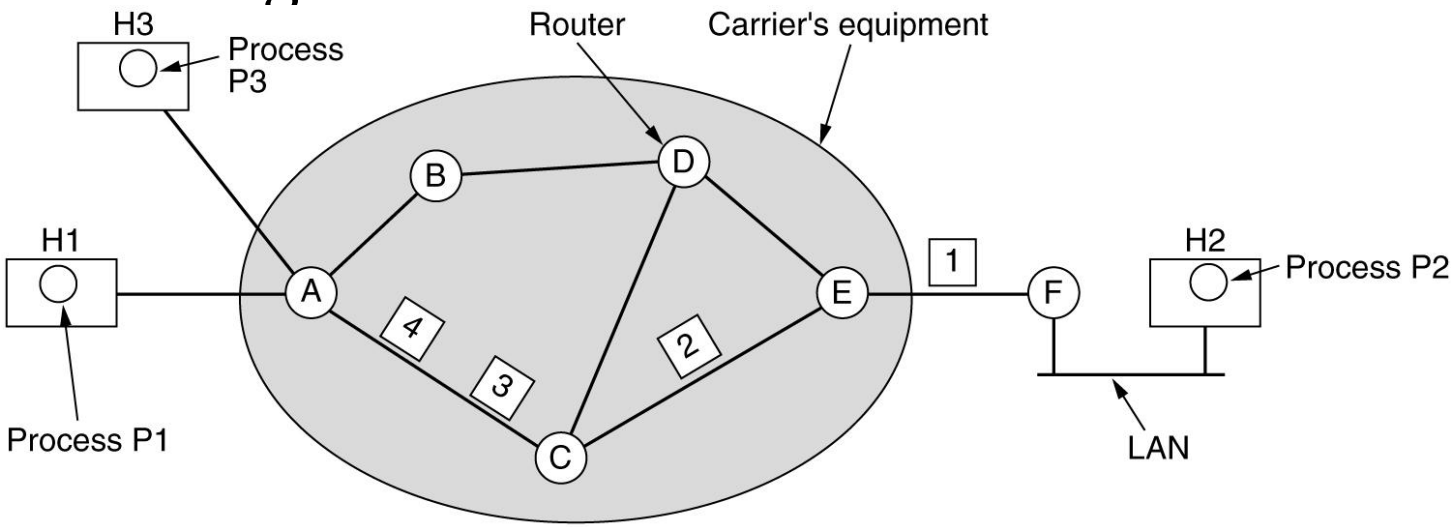
- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- **Implementation of Connection-Oriented Service**
- Comparison of Virtual-Circuit and Datagram Subnets

# Implementation of Connection-Oriented Service

- If connection-oriented service is used, a path from the source router to the destination router must be established before any data packets can be sent.
- This connection is called a VC (virtual circuit) and the subnet is called a virtual-circuit subnet.
- The idea behind virtual circuits is to avoid having to choose a new route for every packet sent. Instead, when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated.
- With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.

# Implementation of Connection-Oriented Service

Routing within a virtual-circuit subnet.




A's table

| In | | Out | |
|---|---|---|---|
| H1 | 1 | C | 1 |
| H3 | 1 | C | 2 |

C's table

| | | | |
|---|---|---|---|
| A | 1 | E | 1 |
| A | 2 | E | 2 |

E's table

| | | | |
|---|---|---|---|
| C | 1 | F | 1 |
| C | 2 | F | 2 |

# Network Layer Design Issues

- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- **Comparison of Virtual-Circuit and Datagram Subnets**

# Comparison of Virtual-Circuit and Datagram Subnets

| Issue | Datagram subnet | Virtual-circuit subnet |
|---|---|---|
| Circuit setup | Not needed | Required |
| Addressing | Each packet contains the full source and destination address | Each packet contains a short VC number |
| State information | Routers do not hold state information about connections | Each VC requires router table space per connection |
| Routing | Each packet is routed independently | Route chosen when VC is set up; all packets follow it |
| Effect of router failures | None, except for packets lost during the crash | All VCs that passed through the failed router are terminated |
| Quality of service | Difficult | Easy if enough resources can be allocated in advance for each VC |
| Congestion control | Difficult | Easy if enough resources can be allocated in advance for each VC |

- Inside the subnet, several trade-offs exist between virtual circuits and datagrams.
- One trade-off is between router memory space and bandwidth.
    - Virtual circuits allow packets to contain circuit numbers instead of full destination addresses. If the packets tend to be fairly short, a full destination address in every packet may represent a significant amount of overhead and hence, wasted bandwidth. The price paid for using virtual circuits internally is the table space within the routers. Depending upon the relative cost of communication circuits versus router memory, one or the other may be cheaper.
- Another trade-off is setup time versus address parsing time.
    - Using virtual circuits requires a setup phase, which takes time and consumes resources. However, figuring out what to do with a data packet in a virtual-circuit subnet is easy: the router just uses the circuit number to index into a table to find out where the packet goes. In a datagram subnet, a more complicated lookup procedure is required to locate the entry for the destination.

# IP Address

- Whatever connects to the internet must have a public (globally unique) IP address.
- An IP address is an identifier for a particular machine on a particular network. It is part of a scheme to identify computers on the internet.
- The main tasks of IP are:
  - The **addressing** of the computers, and the **fragmentation** of packets.

**There are two types of IP addresses:**

- **Internet Protocol version 4 (IPv4):** currently used version of Internet Protocol.

- **Internet Protocol version 6 (IPv6):** the upcoming replacement for IPv4. It contains some major improvements and new features.

# IP Address

- Let us understand it with another example, like if someone wants to send you a mail then he/she must have your home address.

- Similarly, your computer too needs an address so that other computers on the internet can communicate with each other without the confusion of delivering information to someone else’s computer. And that is why each computer in this world has a unique IP Address.

- Or in other words, an IP address is a unique address that is used to identify computers or nodes on the internet.

# IP Address

- This address is just a string of numbers written in a certain format. It is generally expressed in a set of numbers for example 192.155.12.1.

- Here each number in the set is from 0 to 255 range. Or we can say that a full IP address ranges from 0.0.0.0 to 255.255.255.255.

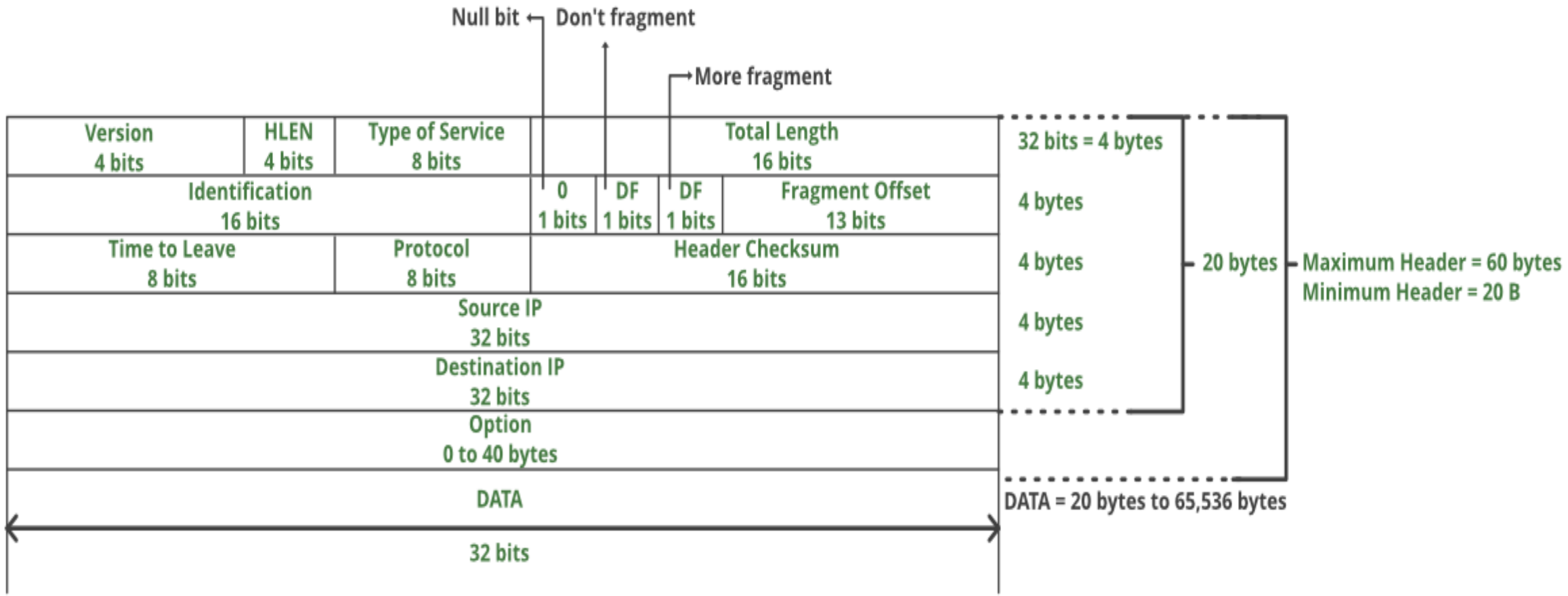
- And these IP addresses are assigned by IANA(known as Internet Corporation For Internet Assigned Numbers Authority).

# IP Datagram format

**Format of an IP Datagram**

- The format of data that can be recognized by IP is called an IP datagram. It consists of two components, namely, the header and data, which need to be transmitted.
- The fields in the datagram, except the data, have specific roles to perform in the transmission of data.
- Every field in the IP datagram has a fixed size except for the IP Options field, which can be 20–60 bytes in length. The sending computer sends a message to the protocol in the same layer on the destination computer by using the header.

# IP Datagram format



# IP Datagram format

- **VERSION:** Version of the IP protocol (4 bits), which is 4 for IPv4
- **HLEN:** IP header length (4 bits), which is the number of 32 bit words in the header. The minimum value for this field is 5 and the maximum is 15.
- **Type of service:** Low Delay, High Throughput, Reliability (8 bits)
- **Total Length:** Length of header + Data (16 bits), which has a minimum value 20 bytes and the maximum is 65,535 bytes.
- **Identification:** Unique Packet Id for identifying the group of fragments of a single IP datagram (16 bits)
- **Flags:** 3 flags of 1 bit each : reserved bit (must be zero), do not fragment flag, more fragments flag (same order)

# IP Datagram format

- **Fragment Offset:** Represents the number of Data Bytes ahead of the particular fragment in the particular Datagram. Specified in terms of number of 8 bytes, which has the maximum value of 65,528 bytes.
- **Time to live:** Datagram's lifetime (8 bits), It prevents the datagram to loop through the network by restricting the number of Hops taken by a Packet before delivering to the Destination.
- **Protocol:** Name of the protocol to which the data is to be passed (8 bits)
- **Header Checksum:** 16 bits header checksum for checking errors in the datagram header
- **Source IP address:** 32 bits IP address of the sender
- **Destination IP address:** 32 bits IP address of the receiver
- **Option:** Optional information such as source route, record route. Used by the Network administrator to check whether a path is working or not.

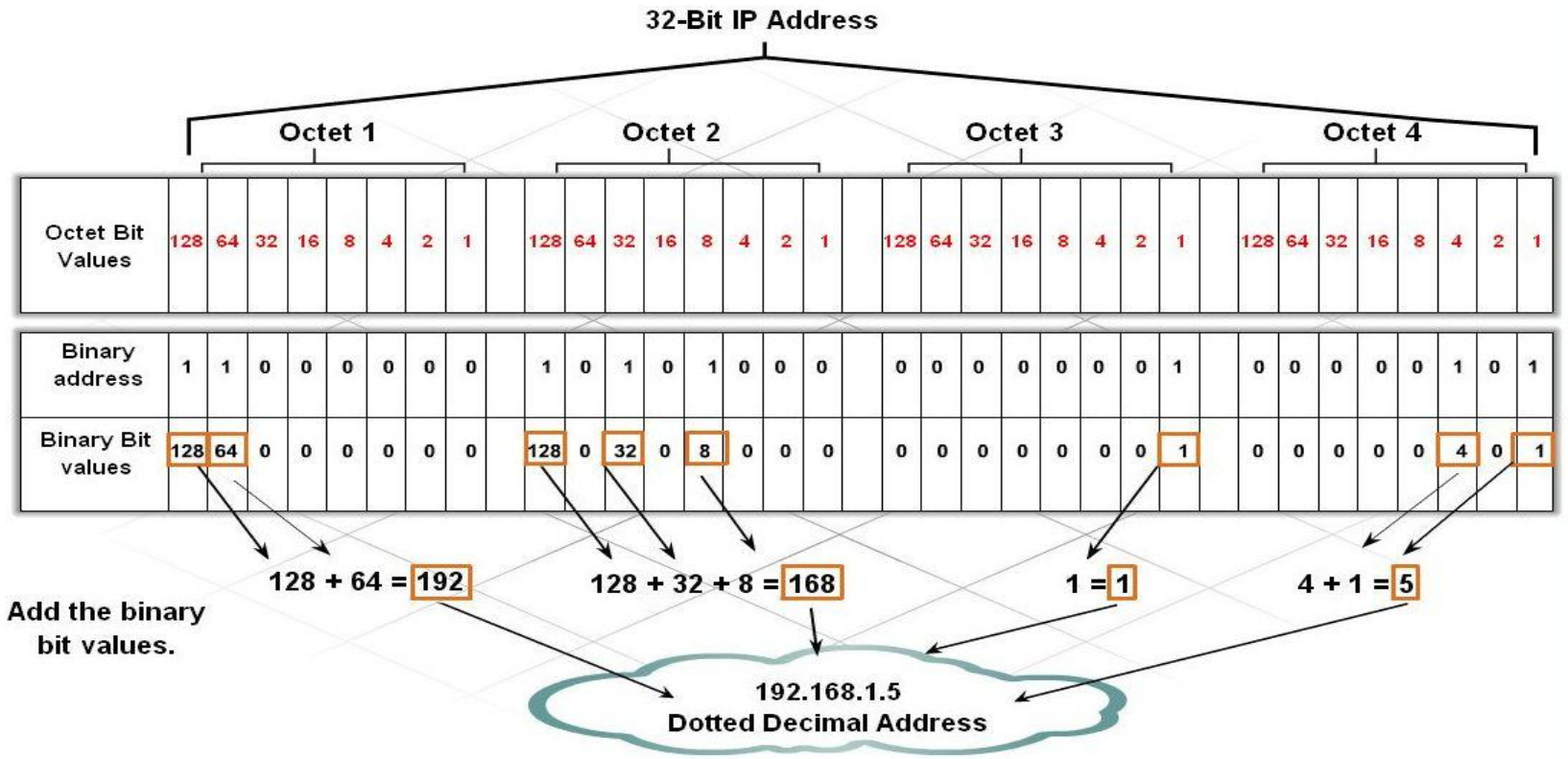
# IPv4

- Internet Protocol version 4. It consists of 4 numbers separated by the dots. Each number can be from 0-255 in decimal numbers.
- But computers do not understand decimal numbers, they instead change them to binary numbers which are only 0 and 1.
- Therefore, in binary, this (0-255) range can be written as (00000000 – 11111111).
- Since each number N can be represented by a group of 8-digit binary digits.
- So, a whole IPv4 binary address can be represented by 32-bits of binary digits. In IPv4, a unique sequence of bits is assigned to a computer,
- so a total of (2^32) devices approximately = 4,294,967,296 can be assigned with IPv4.
- IPv4 can be written as: **189.123.123.90**

# IP address

There are two prevalent notations to show an IPv4 address:

1. binary notation
2. dotted-decimal notation.



# Binary notation vs Dot-decimal notation

In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte.

The following is an example of an IPv4 address in binary notation: 01110101 10010101 00011101 00000010

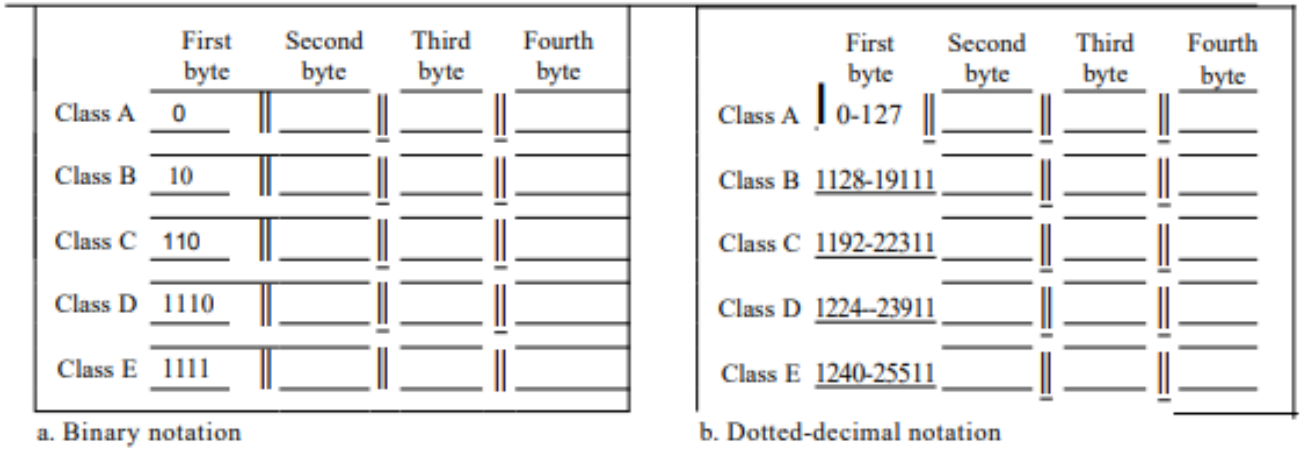
Dot-decimal notation is a presentation format for numerical data.

The following is an example of an IPv4 address in DD notation: **189.123.123.90**

# Classful addressing:

- IPv4 addressing, at its inception, used the concept of classes. This architecture is called classful addressing.
- In classful addressing, the address space is divided into five classes: A, B, C, D and E

- We can find the class of an address when given the address in binary notation or dotted-decimal notation.
  - If the address is given in binary notation, the first few bits can immediately tell us the class of the address.
  - If the address is given in decimal-dotted notation, the first byte defines the class.



# Classful Addressing

In classful addressing, an IP address in class A, B, or C is divided into netid (network address) and hostid (host address). These parts are of varying lengths, depending on the class of the address.

- **Network address,** address of the network within the Internet (used by gateways for routing IP packets between networks).

- **Host address**, address of the computer within the network (used for delivering packets to a particular network interface within the network).

# Classes of IP addresses

**Class A:** allows 128 networks, 16 million hosts each.
The IP address start from 1.0.0.0 to 127.255.255.255, and the mask address is **255.0.0.0**

**Class B:** allows 16,382 networks, 65,534 hosts each.
The IP address start from 128.0.0.0 to 191.255.255.255, and the mask address is **255.255.0.0**

**Class C:** allows 2 million networks, 254 hosts each.

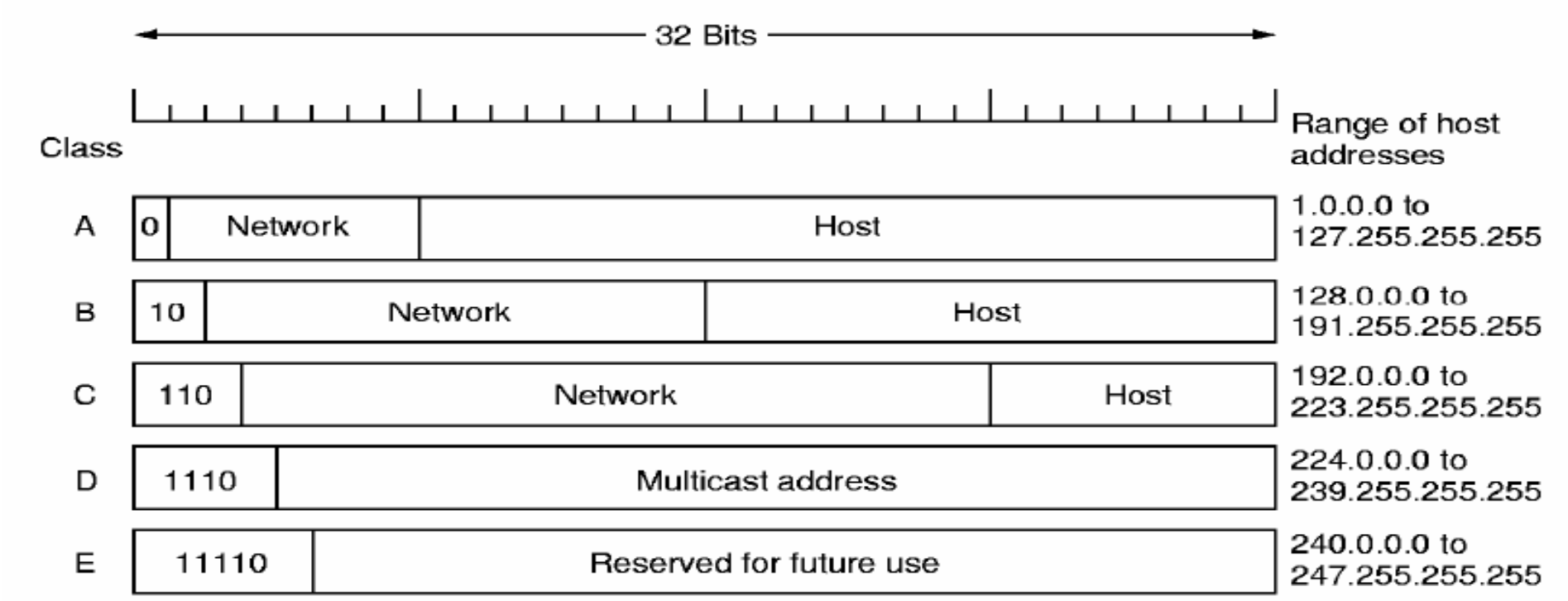
The IP address start from 192.0.0.0 to 223.255.255.255, and the mask address is **255.255.255.0**

**Class D:** multicast networks The IP address start from 224.0.0.0 to 239.255.255.255.

**Class E:** reserved for future use. From 240 to 255 and the 255.255.255.255 used for broadcast to all the subnet.

# Cont..

- One of the benefits of **classful addresses** is that they provide a hierarchy to the network through the use of the network ID. This translates into an efficient routing environment because it is easy for a router to determine what networks can be grouped together and treated as a single routing entry.



# Limitations of Classful Addressing

- One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size.

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 |

- Class A addresses were designed for large organizations with a large number of attached hosts or routers: too large for almost any organization leading to wasted addresses
- Class B addresses were designed for midsize organizations with tens of thousands of attached hosts or routers: wastage of addresses
- Class C addresses were designed for small organizations with a small number of attached hosts or routers: small number of hosts
- Class D, E addresses : Each address in this class is used to define one group of hosts on the Internet. The Internet authorities wrongly predicted a need for 268,435,456 groups.

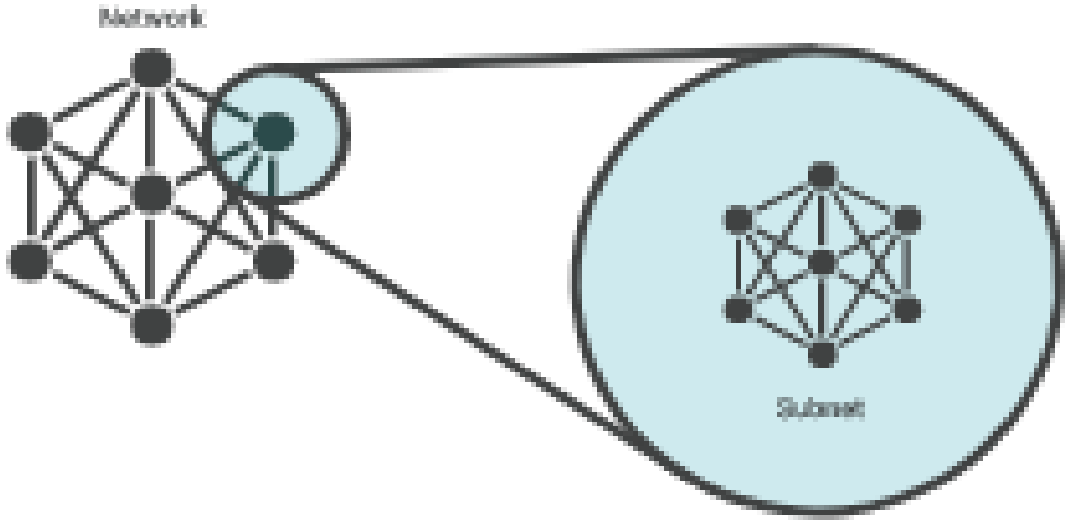
# Limitations of Classful Addressing

- If we have a class B with a Flat Network, the number of host will be more than $2^{16} = 65536$ hosts,
- Managing this network with this number of host is too tricky and the performance of this network will get down because of the heavy load. In other word, any single broadcast can slowdown the network.

- Therefore, the solution is the subnetting. Subnetting means divide or separate the single network into multiple networks that can reduce the loading from one network.
- The advantage of using subnetting is:-

  1. Reduce the traffic and the increase the performance.
  2. The smaller network can be easier to manage.

# IPv4 - Subnetting



# IPv4 - Subnetting

**What is a subnet?**

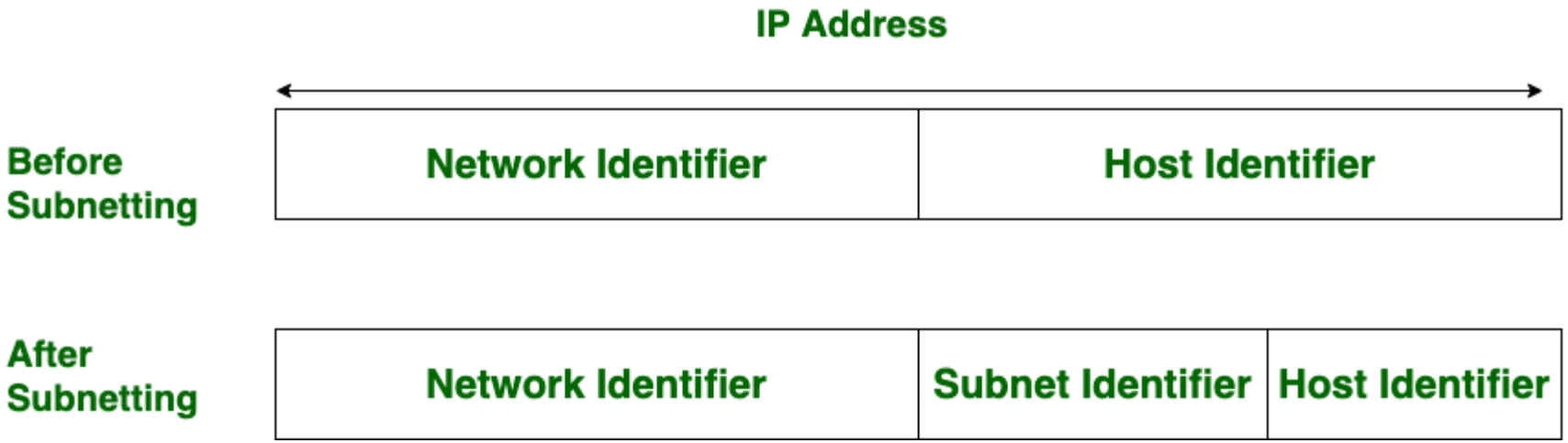
A subnet, or subnetwork, is a network inside a network. Subnets make networks more efficient. Through subnetting, network traffic can travel a shorter distance without passing through unnecessary routers to reach its destination.

# IPv4 - Subnetting

**Why is subnetting necessary?**

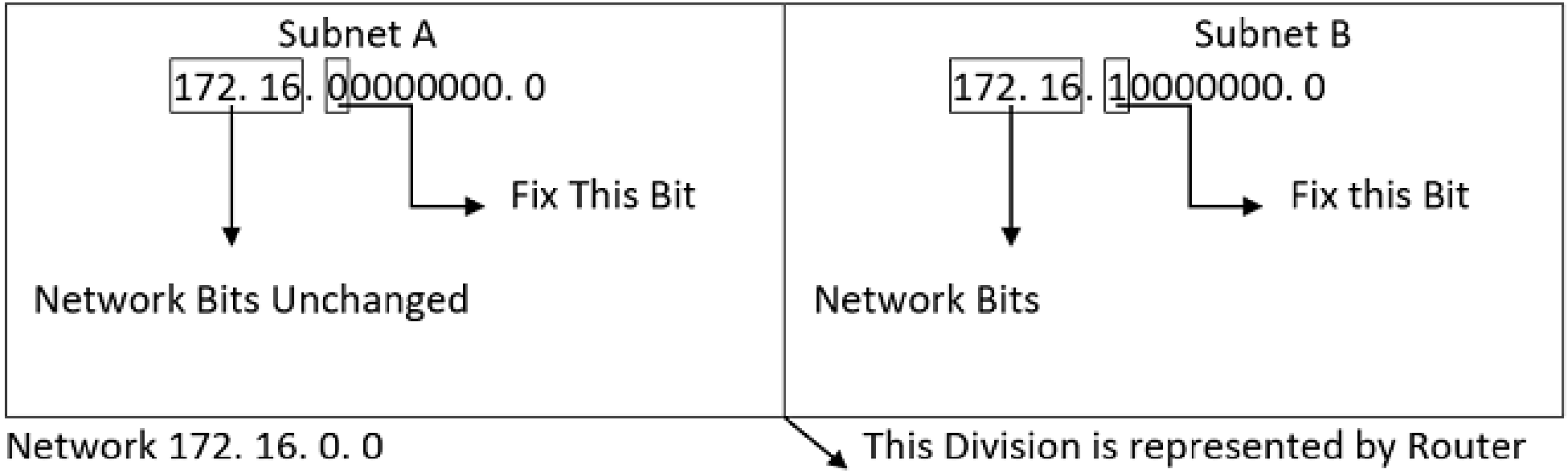
- As the previous example illustrates, the way IP addresses are constructed makes it relatively simple for Internet routers to find the right network to route data into.
- However, in a Class A network (for instance), there could be millions of connected devices, and it could take some time for the data to find the right device. This is why subnetting comes in handy: subnetting narrows down the IP address to usage within a range of devices.
- Because an IP address is limited to indicating the network and the device address, IP addresses cannot be used to indicate which subnet an IP packet should go to. Routers within a network use something called a subnet mask to sort data into subnetworks.

# IPv4 - Subnetting



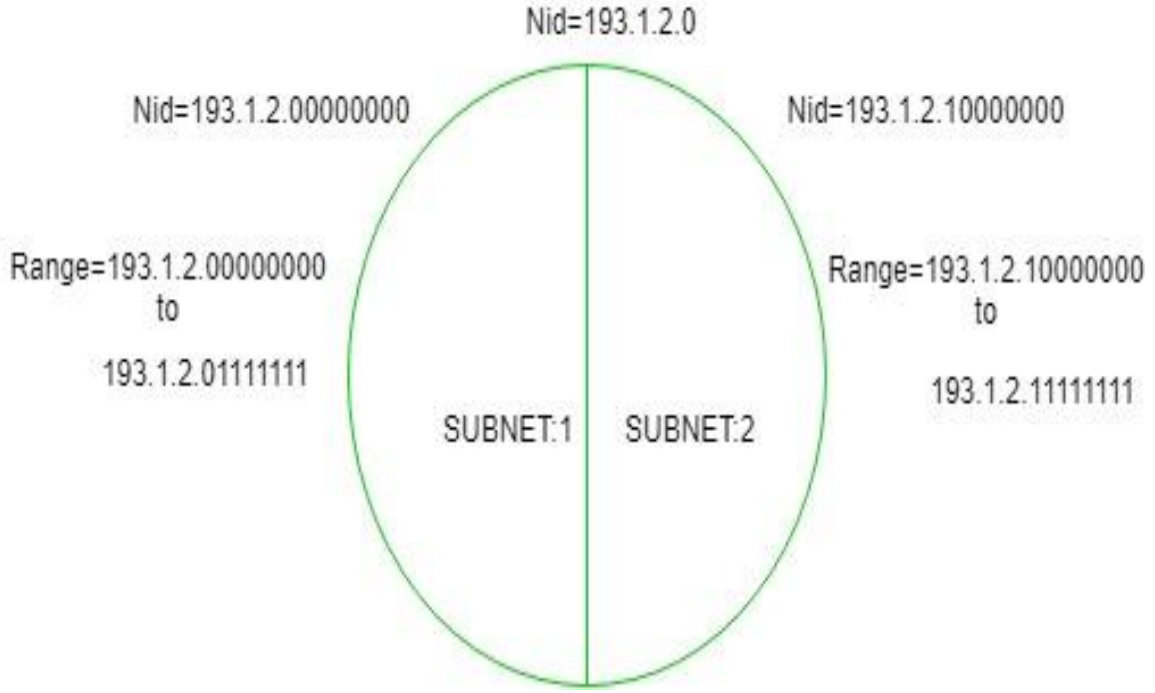
# IPv4 - Subnetting

For the construction of the subnets, we usually check the MSB (Most Significant Bit) bits of the host ID and if found wrong we make it right. In order to create two network subnets, we fix one of the host's MSB (Most Significant Bit) bits in the table below. We are unable to alter network bits since doing so would alter the entire network.



# IPv4 - Subnetting

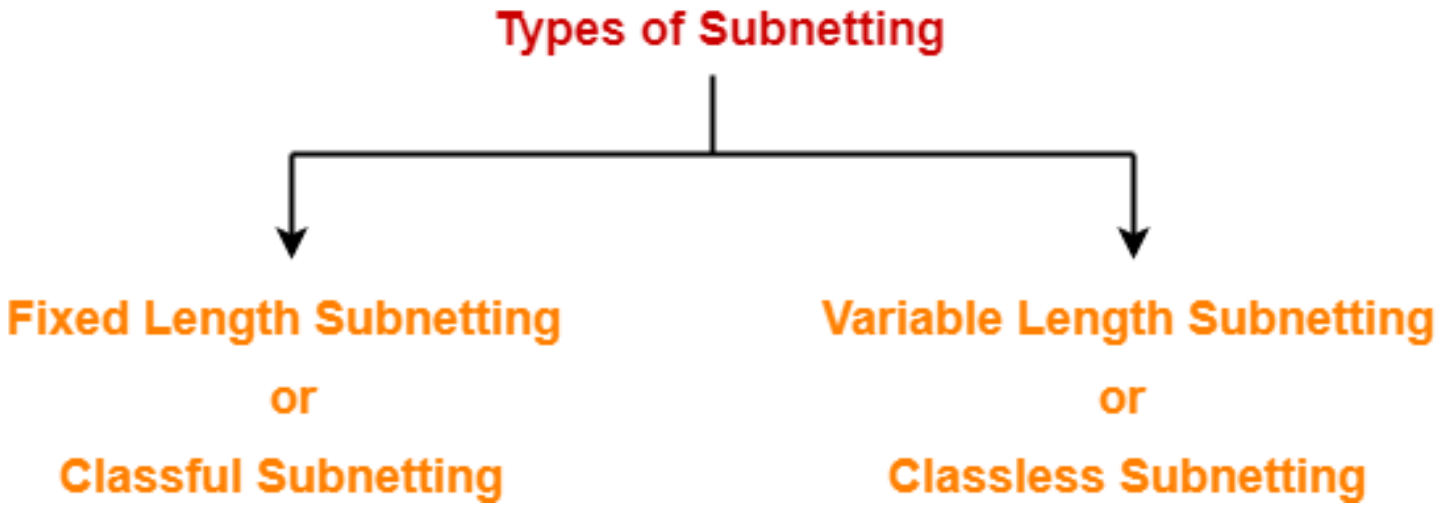
- We need a subnet mask to identify a subnet, which is created by substituting the number "1" for each Network ID bit and the amount of bits we reserve for Host ID to create the subnet.
- A data packet from the internet is intended to be forwarded to the specified subnet network using the subnet mask.
- The network can be divided into two parts: To divide a network into two parts, you need to choose one bit for each Subnet from the host ID part.



# IPv4 - Subnetting

**Types of Subnetting-**

Subnetting of a network may be carried out in the following two ways-



# IPv4 - Subnetting

**1. Fixed Length Subnetting-**

Fixed length subnetting also called as classful subnetting divides the network into subnets where-

- All the subnets are of same size.
- All the subnets have equal number of hosts.
- All the subnets have same subnet mask.

# IPv4 - Subnetting

**2. Variable Length Subnetting-**

Variable length subnetting also called as classless subnetting divides the network into subnets where-

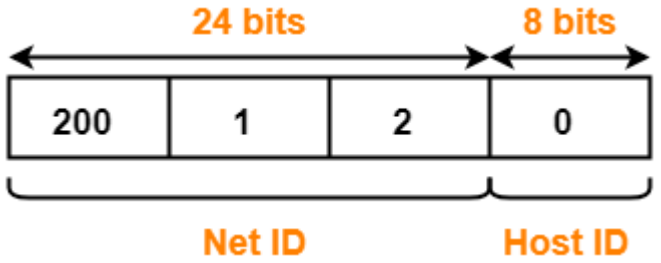
- All the subnets are not of same size.
- All the subnets do not have equal number of hosts.
- All the subnets do not have same subnet mask.

# IPv4 - Subnetting

**Example-01**

Consider we have a big single network having IP Address 200.1.2.0.

We want to do subnetting and divide this network into 2 subnets.

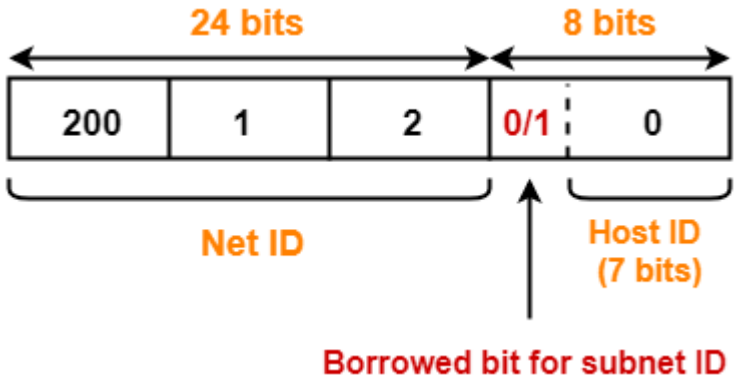



Clearly, the given network belongs to class C.

If borrowed bit = 0, then it represents the first subnet.
If borrowed bit = 1, then it represents the second subnet.

IP Address of the two subnets are-

200.1.2.00000000 = 200.1.2.0
200.1.2.10000000 = 200.1.2.128



# IPv4 - Subnetting

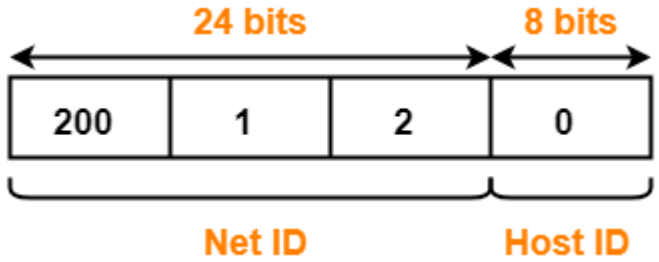
**Example-02**

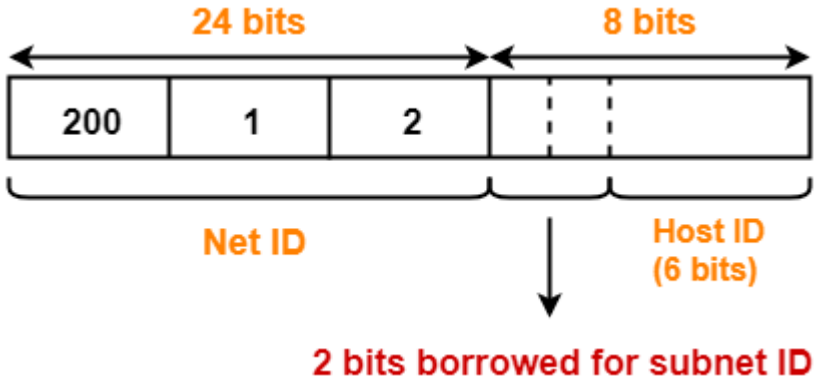
Consider-

We have a big single network having IP Address 200.1.2.0.

We want to do subnetting and divide this network into 4 subnets.

Clearly, the given network belongs to class C.



# IPv4 - Subnetting

If borrowed bits = 00, then it represents the 1st subnet.

If borrowed bits = 01, then it represents the 2nd subnet.

If borrowed bits = 10, then it represents the 3rd subnet.

If borrowed bits = 11, then it represents the 4th subnet.

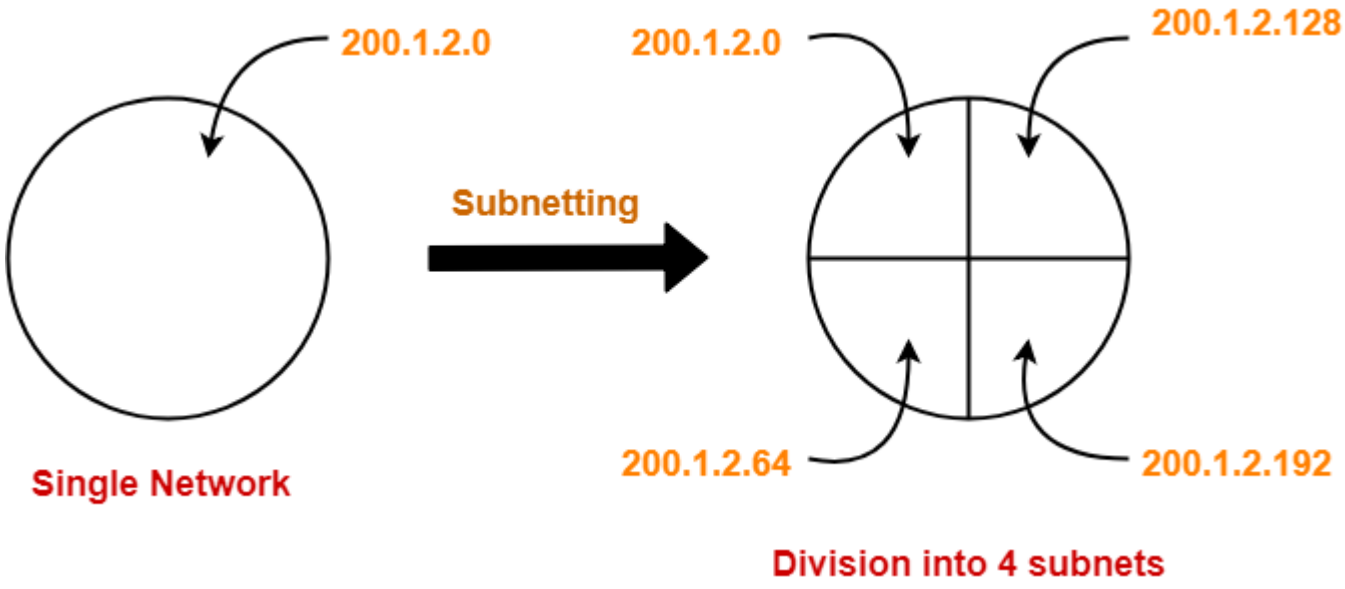
**IP Address of the four subnets are-**

200.1.2.00000000 = 200.1.2.0

200.1.2.01000000 = 200.1.2.64

200.1.2.10000000 = 200.1.2.128

200.1.2.11000000 = 200.1.2.192



# IPv4 - Subnetting

- Range of IP addresses in each subnet:
  - Subnet 1: 200.1.2.0- 200.1.2.63
  - Subnet 2: 200.1.2.64- 200.1.2.127
  - Subnet 3: 200.1.2.128- 200.1.2.191
  - Subnet 4: 200.1.2.192- 200.1.2.255

- Number of hosts in each subnet = 62

# Reserved and Restricted Addresses

- In any subnet/network, there are certain addresses that cannot be assigned to an individual device because they have a special purpose.
  - The subnet address is the first address in a range that identifies the subnet.
  - The broadcast address is the last address in the range, and all hosts on the subnet receive traffic if anything is sent to it.

- Assume that a subnet address is **172.31.9.0** with a mask of **255.255.255.0**.
  - The subnet address is **172.31.9.0**,
  - and the broadcast address is **172.31.9.255.**

# Cont...

There three important things that should be taken into our account when we thinking about subnetting:-

1. Network address – the first one
2. Broadcast address – the last one
3. Host addresses – everything in between

As well as, to find the number of hosts per subnet. We can use formal $2^x$ -2, where (x) is the number of unmasked bits (0’s) .

# Cont...

- For example, in 11000000, the number of zeros gives us $2^6 - 2 = 62$ hosts. In this example, there are 62 hosts per subnet and we make subtract because the first IP address reserve for the network address and the last one for the network broadcast.

- While when we want to find number of networks, we can use this formal $2^y$

  Where Y represent the number of masked bits, (1's). For example, in 11000000, the number of ones gives us $2^2 = 4$

# IPv4 - Subnetting

**How to Calculate Subnet Mask?**

For any given IP Address, the subnet mask is calculated-

- By setting all the bits reserved for network ID part and subnet ID part to 1.
- By setting all the bits reserved for host ID part to 0.

# IPv4 - Subnetting

**Subnet Mask Examples**

Now, let us discuss some examples on how to calculate subnet mask for any given network-

**Example-01:**

Consider we have a network having IP Address 200.1.2.0.

Clearly, this IP Address belongs to class C

In class C-

24 bits are reserved for the Network ID part.

8 bits are reserved for the Host ID part.

Subnet mask is obtained-

# IPv4 - Subnetting

By setting the first 24 bits to 1.

By setting the remaining 8 bits to 0.

So, Subnet mask

**= 11111111.11111111.11111111.00000000**

**= 255.255.255.0**

# Subnet masks:-

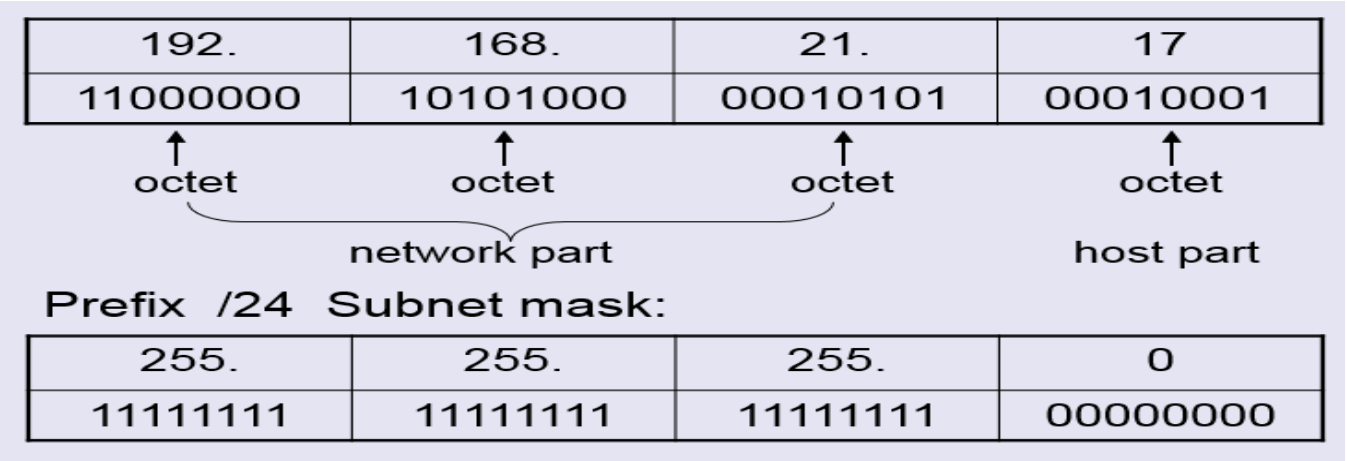
- A mask is a 32-bit binary number that is expressed in dotted decimal notation. By default, a mask contains two fields, the network field and the host field. These correspond to the network number and the locally administered part of the network address.
- When an administrator subnets, they are adjusting the way they view the IP address.
- Table 1: Default masks for classful addressing

| Address Class | Bits Used for Subnet Mask | | | | Dotted Decimal Notation |
|---|---|---|---|---|---|
| Class A | 11111111 | 00000000 | 00000000 | 00000000 | 255.0.0.0 |
| Class B | 11111111 | 11111111 | 00000000 | 00000000 | 255.255.0.0 |
| Class C | 11111111 | 11111111 | 11111111 | 00000000 | 255.255.255.0 |

# Cont…

- Routers and hosts still assume class subnet masks by default:

- Class A /8 255.0.0.0
- Class B /16 255.255.0.0
- Class C /24 255.255.255.0
-
- The figure below gives an example to class C mask address:




The first three octets represent the network part and the last octet represent the host part.

# IPv4 - Subnetting

Problem-01:

1. If the subnet mask 255.255.255.128 belongs to class C, find-
    1. Number of subnets
    2. Number of hosts in each subnet

Solution-

Given subnet mask

= 255.255.255.128

= 11111111.11111111.11111111.10000000

# IPv4 - Subnetting

Since 25 bits contain the value 1 and 7 bits contain the value 0, so-

Number of Net ID bits + Number of Subnet ID bits = 25

Number of Host ID bits = 7

Now,

It is given that subnet mask belongs to class C.

So, Number of Net ID bits = 24.

Substituting in the above equation, we get-

Number of Subnet ID bits= 25 – 24

= 1

# IPv4 - Subnetting

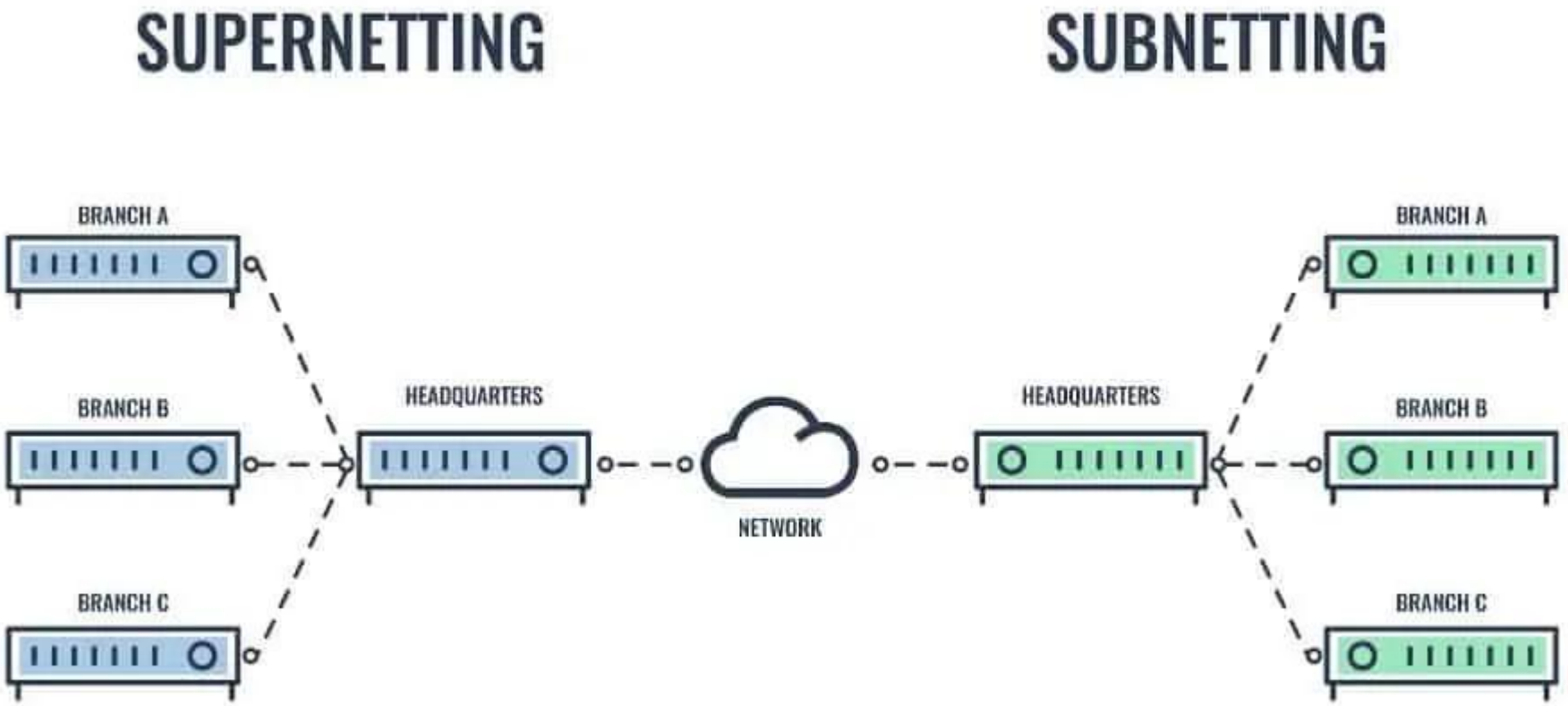
Thus,

Number of subnets = 2^1 = 2

Since number of Host ID bits = 7, so-

Number of hosts per subnet = 2^7 – 2 = 126

# IPv4 - Supernetting



# IPv4 - Supernetting

Supernetting is the opposite of Subnetting. In subnetting, a single big network is divided into multiple smaller subnetworks. In Supernetting, multiple networks are combined into a bigger network termed as a Supernetwork or Supernet.

Supernetting is mainly used in Route Summarization, where routes to multiple networks with similar network prefixes are combined into a single routing entry, with the routing entry pointing to a Super network, encompassing all the networks. This in turn significantly reduces the size of routing tables and also the size of routing updates exchanged by routing protocols.

# IPv4 - Supernetting

**How to Supernet a Network?**

All the networks are not considered suitable networks for aggregation. Some rules are defined for Supernetting the network. The network must follow below given three rules for aggregation.

- **Contiguous**: All the networks that are to be aggregated must be contiguous.
- **Same Size**: All the networks must have the same size in the power of 2.
- **Divisibility**: ID of the first network must be zero or must be divisible by the block size.

# IPv4 - Supernetting

Now let us understand the rules of aggregation by taking an example. Suppose four networks are having the network ID 201.1.0.0, 201.1.1.0, 201.1.2.0, and 201.1.3.0. Now let us check if these networks can be aggregated or not.

•**Rule 1: Contiguous-** As we can identify from IP addresses that these are class C networks. The first network range is from 201.1.0.0 To 201.1.0.255. The second network range starts from 201.1.2.0. The starting address of the second network is obtained by adding 1 to the last network starting IP address. In this way, we will check all network IP addresses, and that all are contiguous.

•**Rule2: Same Size-** As all the given IP addresses belong to class C and every network has 2^8=256 hosts.

# IPv4 - Supernetting

•**Rule3: Divisibility-** IP address of the first network must be divisible by the network's total size. The total size of the network in our example is 4∗2^8=2^10. When we divide the starting IP address with the network size, then we get the last 10 bits as a remainder. For making the IP address divisible by the size, the last ten digits must be zero. A **binary representation** of the first IP address of the above example is given below: 11001001.00000001.00000000. 00000000

Here last ten bits are zero. So it can be divisible by the size of the network. So all three conditions are satisfied for the IP addresses given in the example. These four networks can be combined to form a supernet. The supernet ID or the network ID for all four networks will be 201.1.0.0.

# IPv4 - Supernetting

**How to supernet a network?**

Combining these networks into one network: (A summarized route)
• 192.168.0.0/24
• 192.168.1.0/24
• 192.168.2.0/24
• 192.168.3.0/24

**Step 1:** Write all the IP Addresses in binary like so:
• 192.168.0.0/24
11000000.10101000.00000000.00000000

• 192.168.1.0/24
11000000.10101000.00000001.00000000

# IPv4 - Supernetting

**Step 1:** Write all the IP Addresses in binary like so:

• 192.168.2.0/24
11000000.10101000.00000010.00000000

• 192.168.3.0/24
11000000.10101000.00000011.00000000

# IPv4 - Supernetting

Step 2: Find matching bits from left to right

**11000000.10101000.000000** 00.00000000

**11000000.10101000.000000** 01.00000000

**11000000.10101000.000000** 10.00000000

**11000000.10101000.000000** 11.00000000

# IPv4 - Supernetting

**Step 3:** Re write the matching numbers and add the remaining zeros, because you are converting network bits into host bits.

This will be your NEW NETWORK ID, the route that you will be advertising. (A summarized route)

11000000.10101000.00000000.00000000 = 192.168.0.0

# IPv4 - Supernetting

**Step 4**: Find the new subnet mask. Put “1s” in the matching networking part, and all zeros in the host part.

11111111.11111111.11111100.00000000

This your new subnet mask 255.255.252.0

- Your new summarized route is 192.168.0.0/22

# Classless Addressing: Address Blocks

- 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

- 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.

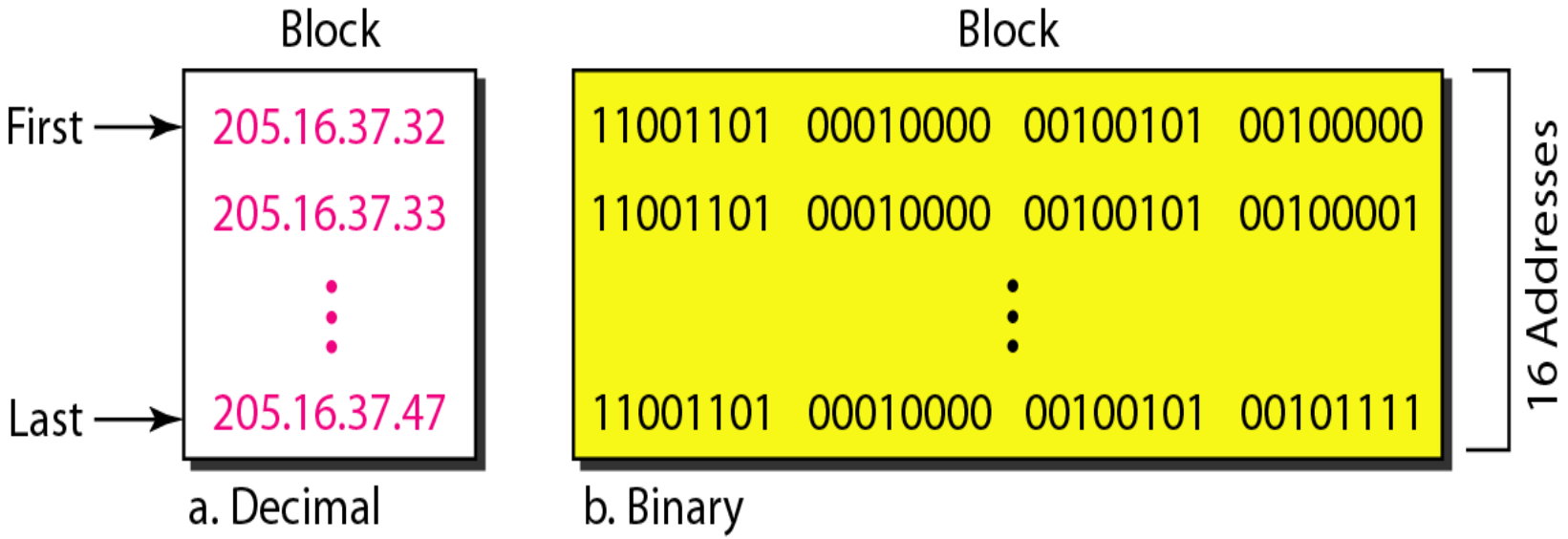
- For example,
  - a household may be given only two addresses;
  - a large organization may be given thousands of addresses.
  - An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.

# Cont…

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

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

# Cont…




We can see that the restrictions are applied to this block.

- The addresses are contiguous.
- The number of addresses is a power of $2^4 = 16$,
- 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.

# Mask and Address Blocks

In 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
- and the $/n$ defines the mask.

The address and the $/n$ notation completely define the whole block (the first address, the last address, and the number of addresses)

- The first address in the block can be found by setting the rightmost $32 - n$ bits to 0. The first address denotes the network address with which the external router sees the network.
- The last address in the block can be found by setting the rightmost $32 - n$ bits to 1.
- The number of addresses in the block can be found by using the formula $2^{32-n}$

# Mask and Address Blocks

Example: 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 and last address in the block?

- The binary representation is 1100110 00010000 00100101 00100111
- If we set 32 − 28 rightmost bits to 0, we get 11001101 00010000 00100101 00100000
  → 205.16.37.32 (First address)
- If we set 32 − 28 rightmost bits to 1, we get 11001101 00010000 00100101 00101111
  → 205.16.37.47 (Last address)
- The value of n is 28, which means that number of addresses is $2^{32-28}$ or 16

# Cont…

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.

Network Mask for 205.16.37.39/28 can be represented as 11111111 11111111 11111111 11110000 (twenty-eight 1's and four 0's). Find

a. The first address
b. The last address
c. The number of addresses

# Cont…

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 Is; 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 Os; 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 | | | |

# Network Addresses

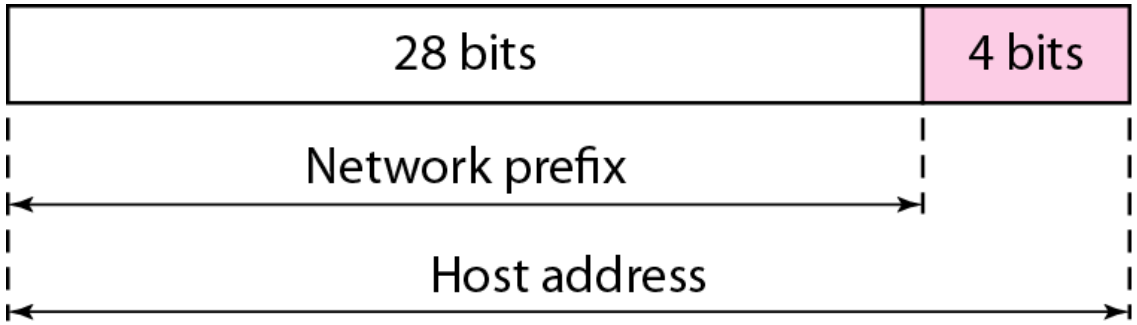
- A very important concept in IP addressing is the network address. When an organization is given a block of addresses, the organization is free to allocate the addresses to the devices that need to be connected to the Internet.
  - The first address in the class, however, is normally treated as a special address. The first address is called the network address and defines the organization network. It defines the organization itself to the rest of the world.

- The first address is the one that is used by routers to direct the message sent to the organization from the outside the organization network. The router has two addresses. One belongs to the granted block; the other belongs to the network that is at the other side of the router. We call the second address $x.y.z.t/n$ because we do not know anything about the network it is connected to at the other side. All messages destined for addresses in the organization block (205.16.37.32 to 205.16.37.47) are sent, directly or indirectly, to x.y.z.t/n. We say directly or indirectly because we do not know the structure of the network to which the other side of the router is connected

# Two-Level Hierarchy: No Subnetting

➢ An IP address can define only two levels of hierarchy when not subnetted. The $n$ left-most bits of the address $x.y.z.t/n$ define the network (organization network);

➢ $32 - n$ rightmost bits define the particular host (computer or router) to the network. The two common terms are prefix and suffix. The part of the address that defines the network is called the prefix.; the part that defines the host is called the suffix.



# Three-Levels of Hierarchy: Subnetting

An organization that is granted a large block of addresses may want to create clusters of networks (called subnets) and divide the addresses between the different subnets. The rest of the world still sees the organization as one entity.

However, internally there are several subnets. All messages are sent to the router address that connects the organization to the rest of the Internet; the router routes the message to the appropriate subnets.

The organization, however, needs to create small sub-blocks of addresses, each assigned to specific subnets. The organization has its own mask; each subnet must also has its own mask.

# Subnetting: Variable Length

Example: Consider an organization is given the block 17.12.40.0/26, which contains 64 addresses. The organization has three offices and needs to divide the addresses into three subblocks of 32, 16, and 16 addresses. Find out the subnet addresses and masks.

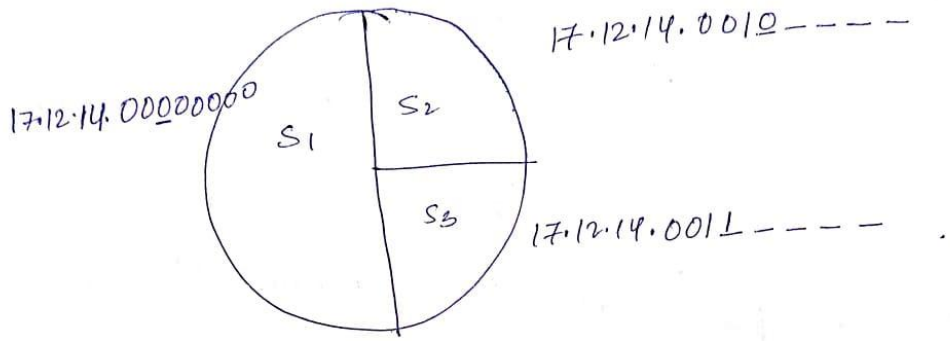
Sol:

We can find the new masks by using the following arguments:

- Suppose the mask for the first subnet is $n_1$, then $2^{32-n_1}$ must be 32 → $n_1 = 27$.
- Suppose the mask for the second subnet is $n_2$, then $2^{32-n_2}$ must be 16 → $n_2 = 28$.
- Suppose the mask for the third subnet is $n_3$, then $2^{32-n_3}$ must be 16 → $n_3 = 28$.

Q: 17.12.14.0/26 → total usees = 64

$G_1 \rightarrow 32$ : $G_2 = 16$ ; $G_3 = 16$ usees




$S_1$: 17.12.14.00000000/27

$S_2$: 17.12.14.00100000/28 ~~17.12.14.001~~

$S_3$: 17.12.14.00110000/28

Range: – $S_1$: 17.12.14.0/27 – 17.12.14.31/27

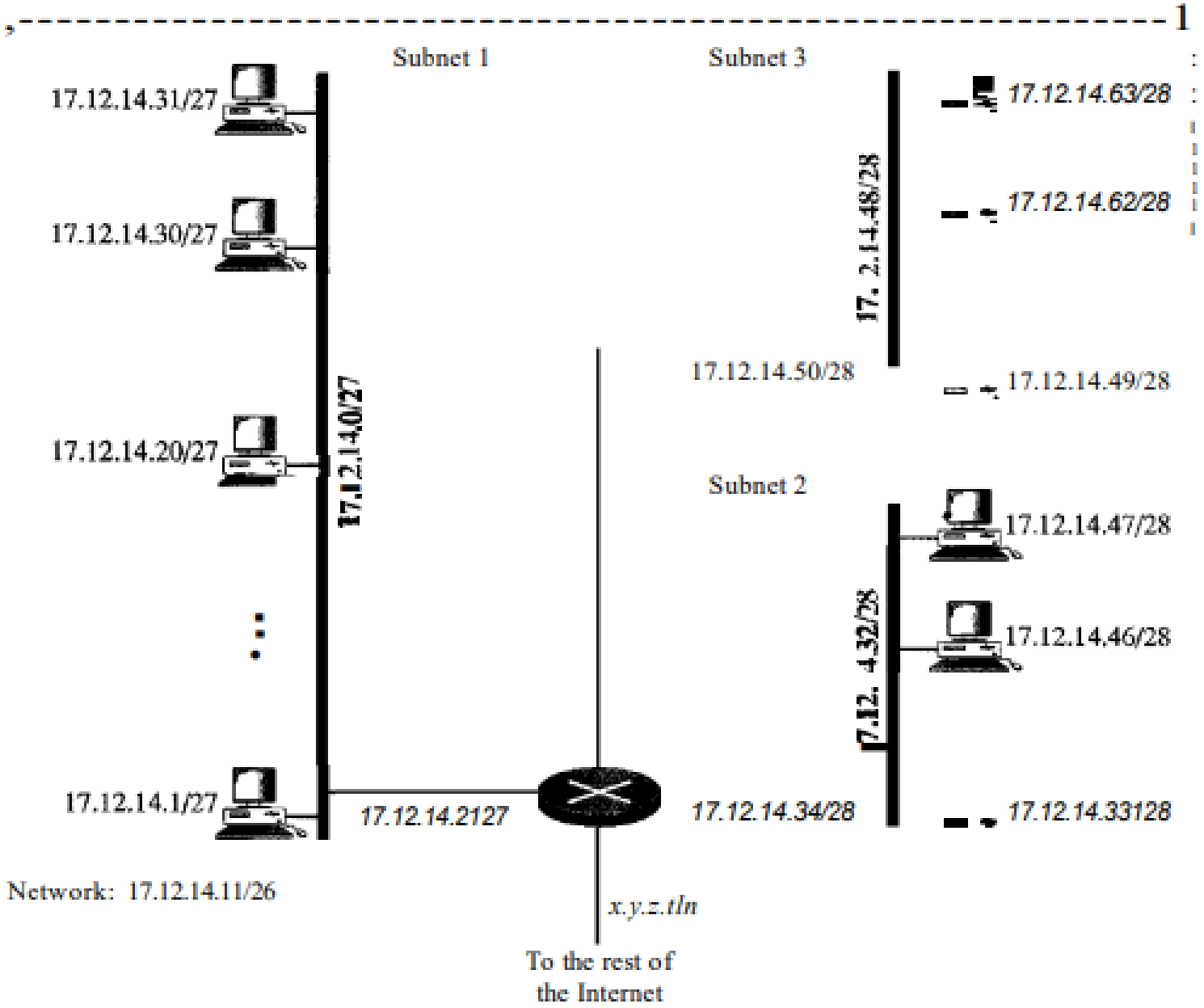
$S_2$: 17.12.14.32 – 17.12.14.00101111

– 17.12.14.47

$S_3$: 17.12.14.48 – 17.12.14.63

Masks: – $S_1$: 255.255.255.11100000 ⇌ 255.255.255.224

$S_2/S_3$: 255.255.255.11110000 ⇌ 255.255.255.240


Subnet 1
Subnet 3
17.12.14.31/27
17.12.14.30/27
17.12.14.20/27
17.12.14.1/27
17.12.14.0/27
17.12.14.2127
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
17.12.14.33128
Network: 17.12.14.11/26
x.y.z.tln
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 slash 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:** 90.100.127.128/25 → 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 = $128 \times 64 = 8{,}192$

***Number of granted address : 65,534***

***Number of allocated address : 40,960***

***Number of available address : 24,574***