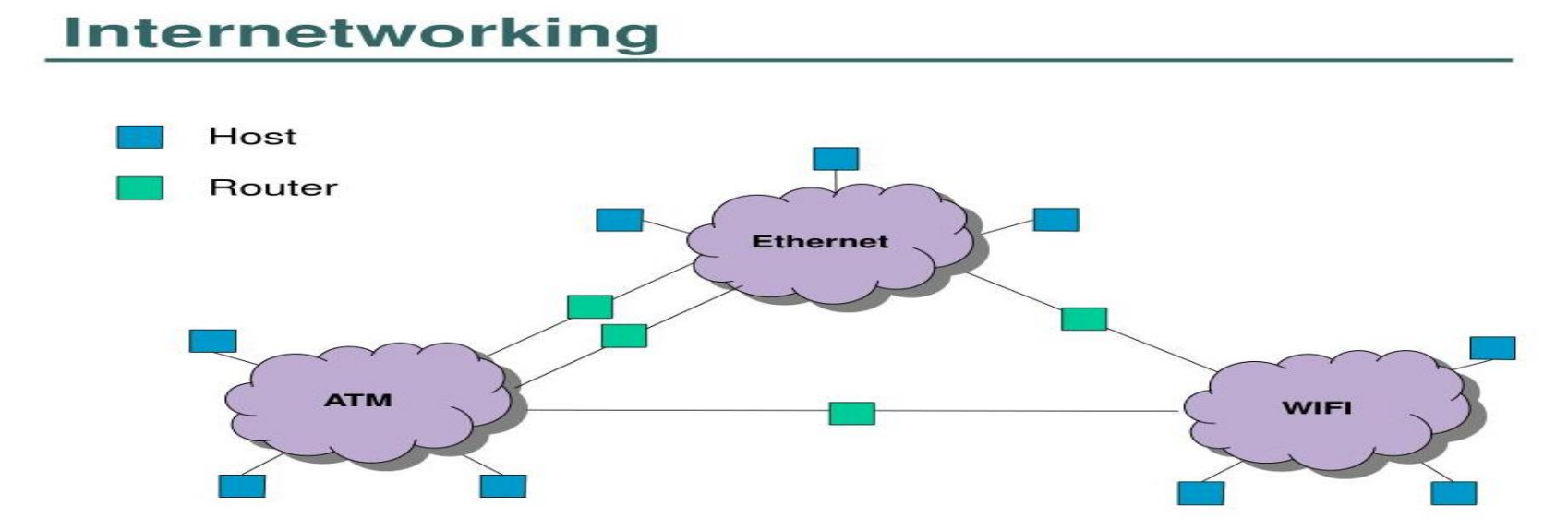
Devices:

- Repeaters, bridges, gateways , routers,
- The Network Layer;
- Design issues,
- Routing algorithms,
- Congestion control Algorithms,
- Quality of service,
- Internetworking,
- Network-Layer in the internet.

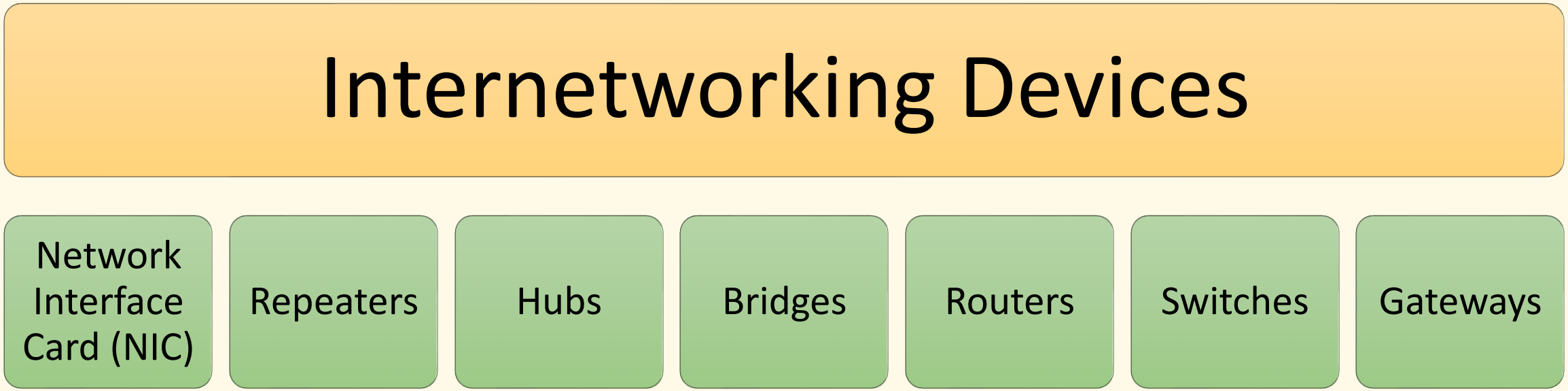
Internetworking means **connecting two or more separate computer networks** so they can communicate and share data with each other.
A single network = LAN (Local Area Network)
Connecting multiple LANs or WANs together = Internetworking
Example: Connecting your college LAN with the internet or another campus LAN.



Types of Internetworking

Feature	Connection-Oriented Internetworking	Connectionless Internetworking
Path setup	Yes, before sending data	No path setup
Path used	Fixed path for all packets	Each packet may take a different path
Reliability	High (packets arrive in order)	Less reliable (packets can be lost or out of order)
Example Protocols	X.25, Frame Relay	IP, UDP
Best for	Applications needing guaranteed delivery	Applications needing speed and flexibility

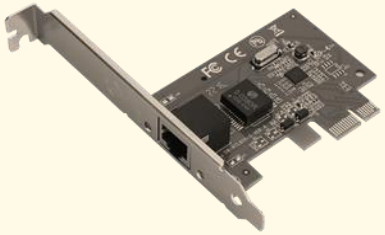
- **Internetworking Devices**
- These are **hardware devices** used to connect different computers, networks, or segments of a network so that data can be shared.



UNIT-IV

Devices:

Modem



NIC

Repeater

Hub

Switch

Router

Bridge

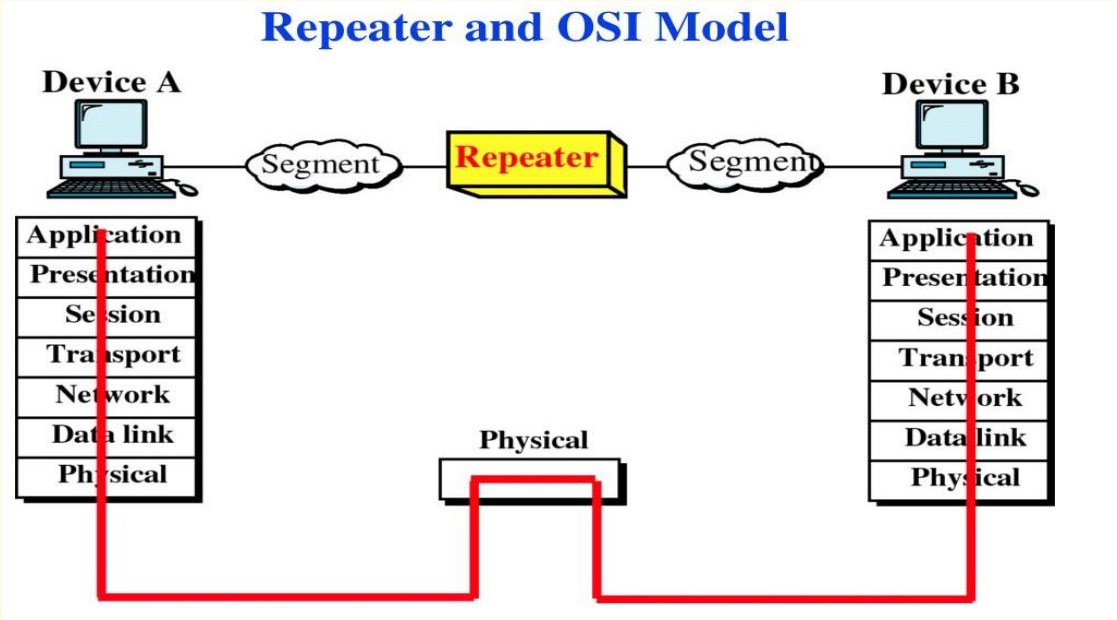
Gateway

Network Interface Card (NIC)

- A hardware component installed in a computer.
- Connects the computer to a network (wired or wireless).
- Converts data from the computer into signals suitable for transmission over the network.
- Each NIC has a unique **MAC address**.

2. Repeaters

- Regenerate and amplify(increase) signals in a network.
- Used when the distance between devices is too long and signals become weak.
- Work only at the **Physical Layer** (Layer 1 of OSI Model).
- Do not filter or direct data—just strengthen the signal.

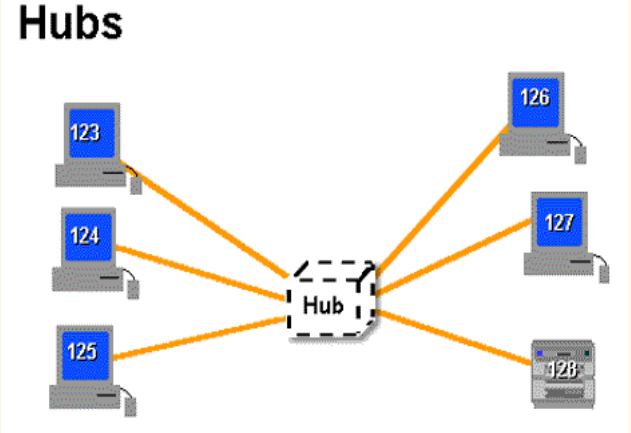


Types of Repeaters

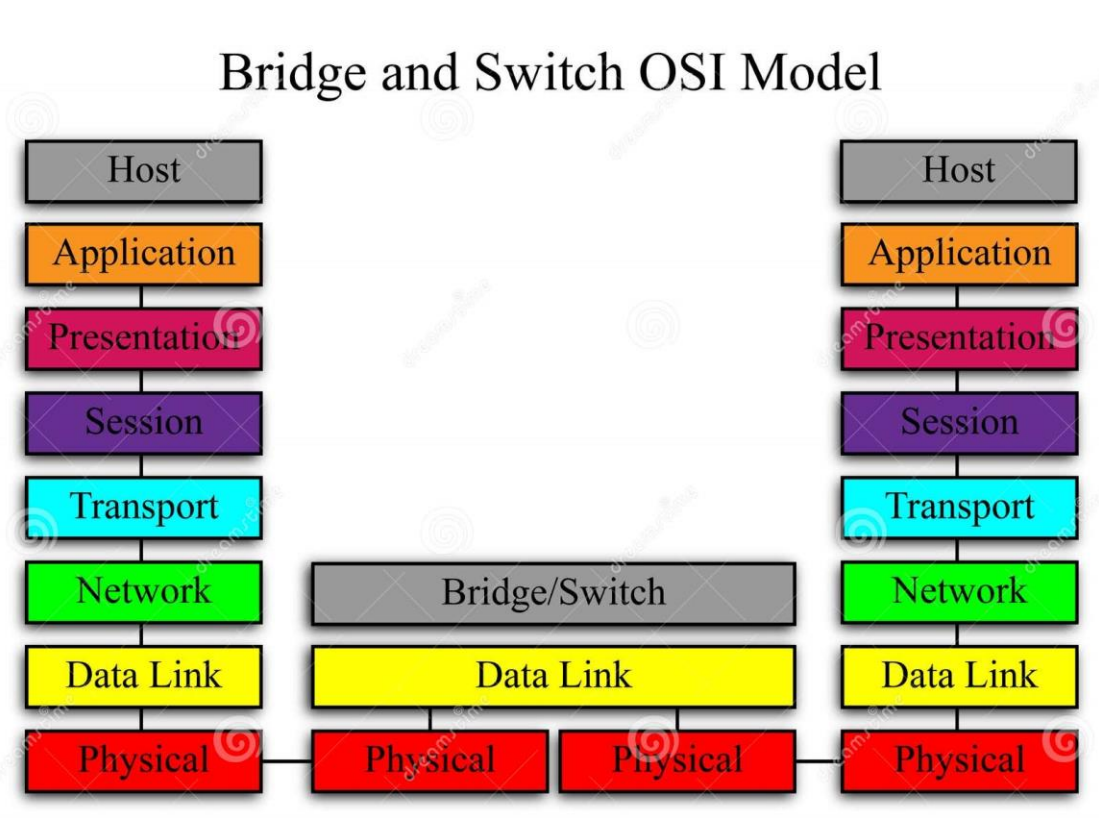
Type	Description	Use
Single-Port	Regenerates signal for one link.	Point-to-point links
Multipoint	Regenerates signal to multiple ports.	Large LANs, cable TV
Smart	Regenerates, monitors, detects errors, filters traffic.	Intelligent networks
Optical	Regenerates optical signals directly.	Long-distance fiber optics

3. Hubs

- A simple, central connection point for devices in a network.
- Broadcasts incoming data to all connected devices.
- Operate at the **Physical Layer**.
- Cannot filter or manage traffic → can cause network congestion.



Type	Description	Use/Example
Active Hub	Powered; amplifies & regenerates signals.	Large LANs to extend cable length.
Passive Hub	No power; just connects devices.	Small networks/short distance.
Smart Hub	Monitoring & management features.	Enterprise networks needing control.



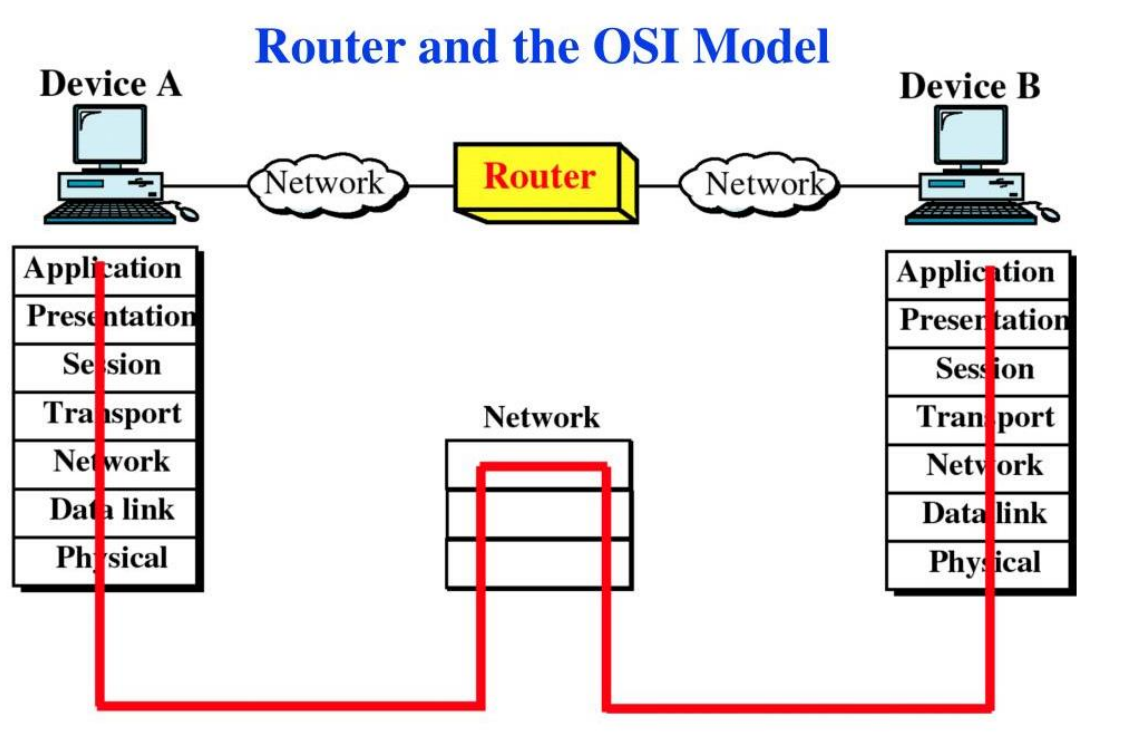
4. Bridges

- Connect two different network segments (like two LANs).
- Filter traffic using **MAC addresses**.
- Operate at the **Data Link Layer** (Layer 2).
- Reduce network traffic by dividing it into collision domains.

Type of Bridge	Description	Use/Example
Simple (Transparent) Bridge	Connects two LAN segments; forwards frames based on MAC addresses.	Small LAN expansion
Source-Route Bridge	Used in Token Ring networks; determines path from source to destination.	Token Ring LANs
Translational Bridge	Connects different types of networks (e.g., Ethernet ↔ Token Ring).	Heterogeneous (different) network interconnection

5. Routers

- Connect different networks (for example, LAN to WAN or to the Internet).
- Forward data packets based on **IP addresses**.
- Operate at the **Network Layer** (Layer 3).
- Choose the best path for data to travel.

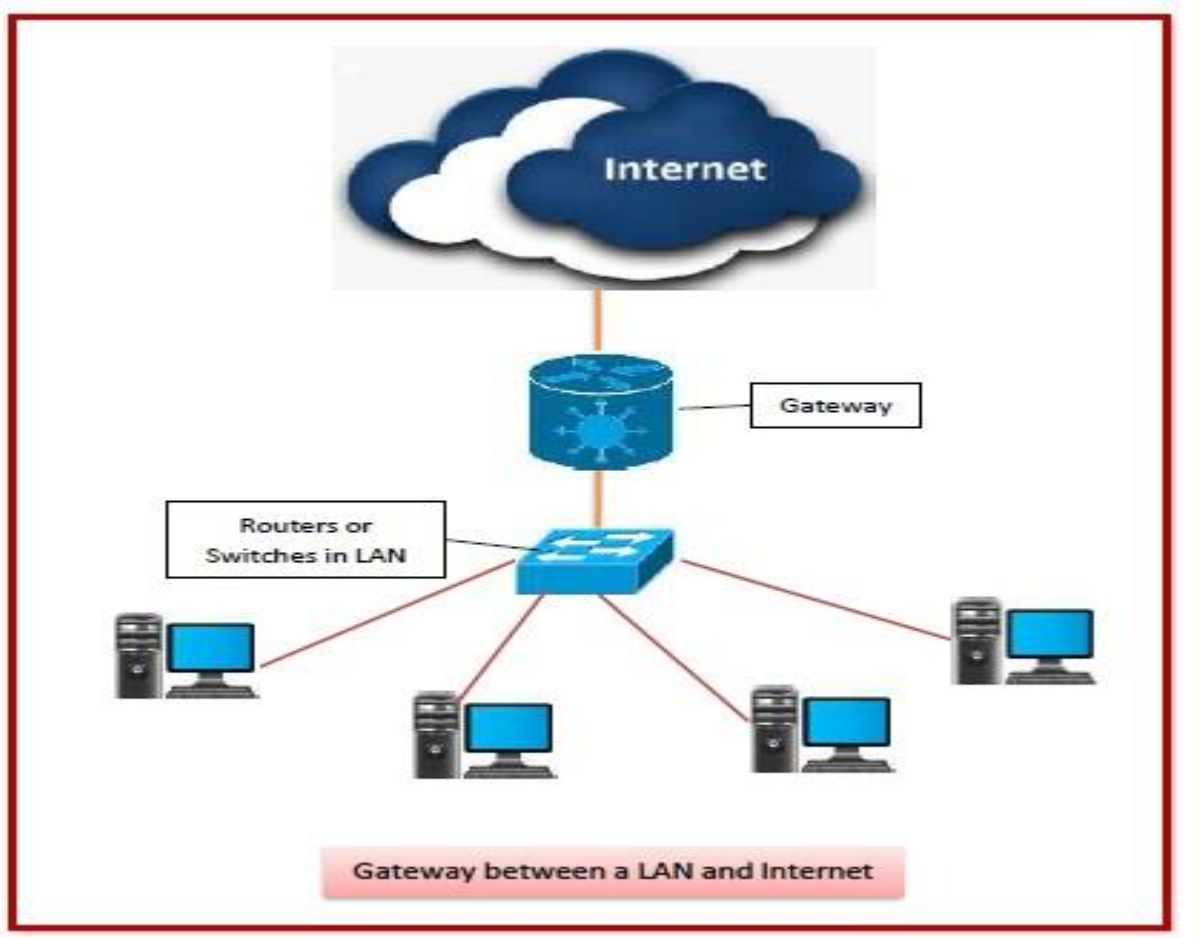


Type of Router	Description	Use/Example
Static Router	Routes are manually configured by the network administrator.	Small networks with fixed paths
Dynamic Router	Automatically updates routing tables using routing protocols .	Large or changing networks

Switch Type	Key Point
Cut-Through	Forwards as soon as it reads destination address → very fast, no error check.
Store-and-Forward	Receives full frame, checks for errors, then forwards → slower but reliable.

6. Switches

- Advanced version of hubs.
- Send data only to the intended recipient device using **MAC addresses**.
- Operate mainly at the **Data Link Layer**, but some work at Layer 3 (multilayer switches).
- Improve network efficiency by reducing collisions.



7. Gateways

- Connect networks using **different protocols** (e.g., LAN to a mainframe or Internet to private network).
- Perform protocol conversion.
- Work at **all layers of OSI Model** as needed.
- Used for communication between completely different networks.

Routing Algorithm (Definition)

A **routing algorithm** is a set of rules used by routers to determine the best path for data packets to travel from source to destination across a network.

Type	Description	Key Features / Example
Static Routing (Non-Adaptive)	Routes are manually entered by the network administrator and do not change automatically.	Simple, secure. Suitable for small or stable networks. Example: Manually configured routes in routers.
Dynamic Routing (Adaptive)	Routers automatically discover and update routes based on network changes using routing protocols.	Self-adjusting, better for large or changing networks. Example: RIP,. RIP → Routing Information Protocol

Feature	Static (Non-Adaptive)	Dynamic (Adaptive)
Configuration	Manual	Automatic
Adapts to Changes	No	Yes
Overhead	Low	Higher (needs CPU & bandwidth)
Best For	Small, stable networks	Large, frequently changing networks

Static Routing Algorithm

- Dijkstra Algorithm
- Bellman-ford Algorithm

Dijkstra's Algorithm (Definition)

Dijkstra's Algorithm is a **shortest path algorithm** used in networking and graph theory to find the **least-cost path** from one source node to all other nodes in a weighted graph.

Steps of Dijkstra's Algorithm (in short)

1.Initialize:

1. Set distance of source node = 0.
2. All other nodes = ∞ (infinity).

2.Visit the nearest node:

1. Select the unvisited node with the **smallest tentative distance**.

3.Update distances:

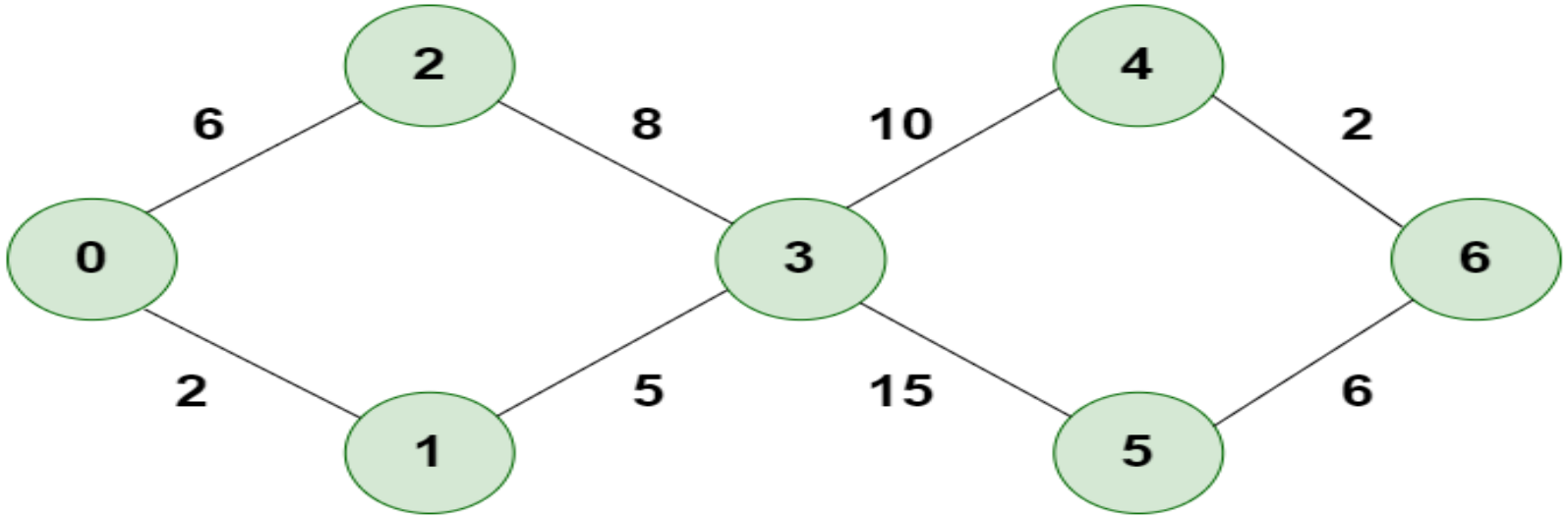
1. For each neighbor of this node, calculate new distances.
2. If new distance < old distance, update it.

4.Mark as visited:

1. Once a node is visited, it's finalized.

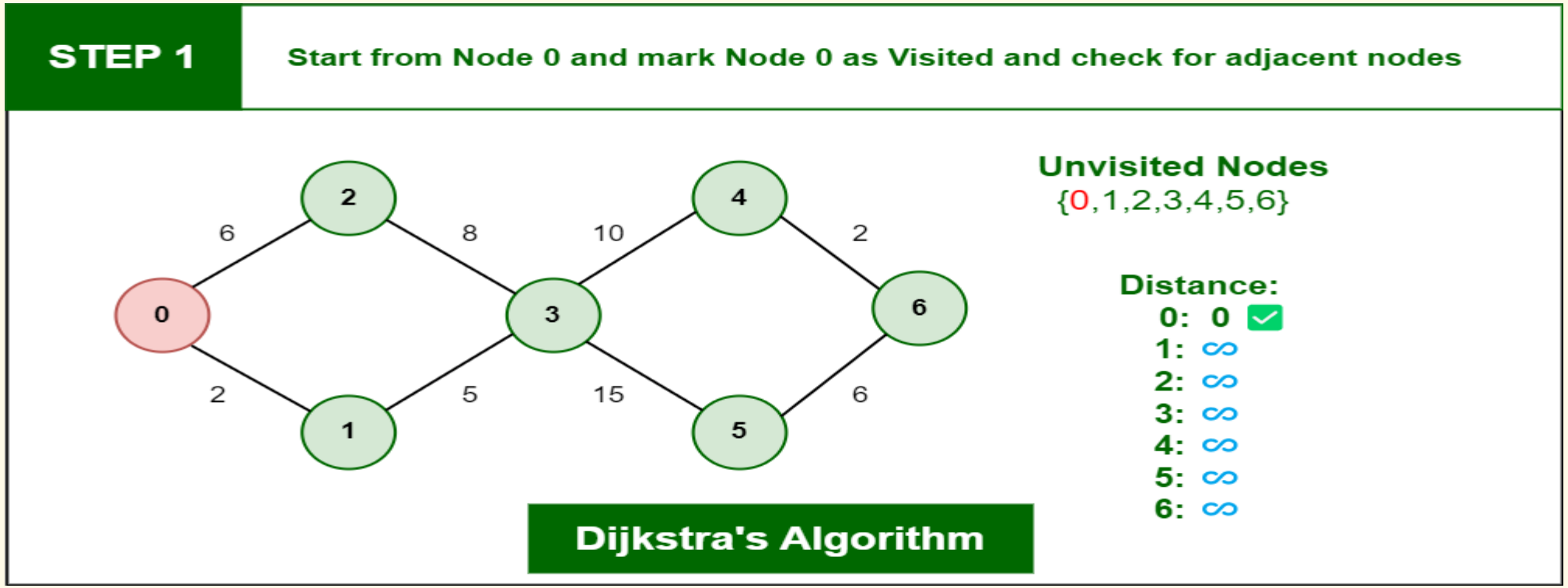
5.Repeat steps 2–4 until all nodes are visited.

Example Graph



Dijkstra's Algorithm

Step 1: Start from Node 0 and mark Node as visited as you can check in below image visited Node is marked red.



Step 2: Check for adjacent Nodes, Now we have to choices (Either choose Node1 with distance 2 or either choose Node 2 with distance 6) and choose Node with minimum distance. In this step **Node 1** is Minimum distance adjacent Node, so marked it as visited and add up the distance.

Distance: Node 0 -> Node 1 = 2

Step 3: Then Move Forward and check for adjacent Node which is Node 3, so marked it as visited and add up the distance, Now the distance will be:

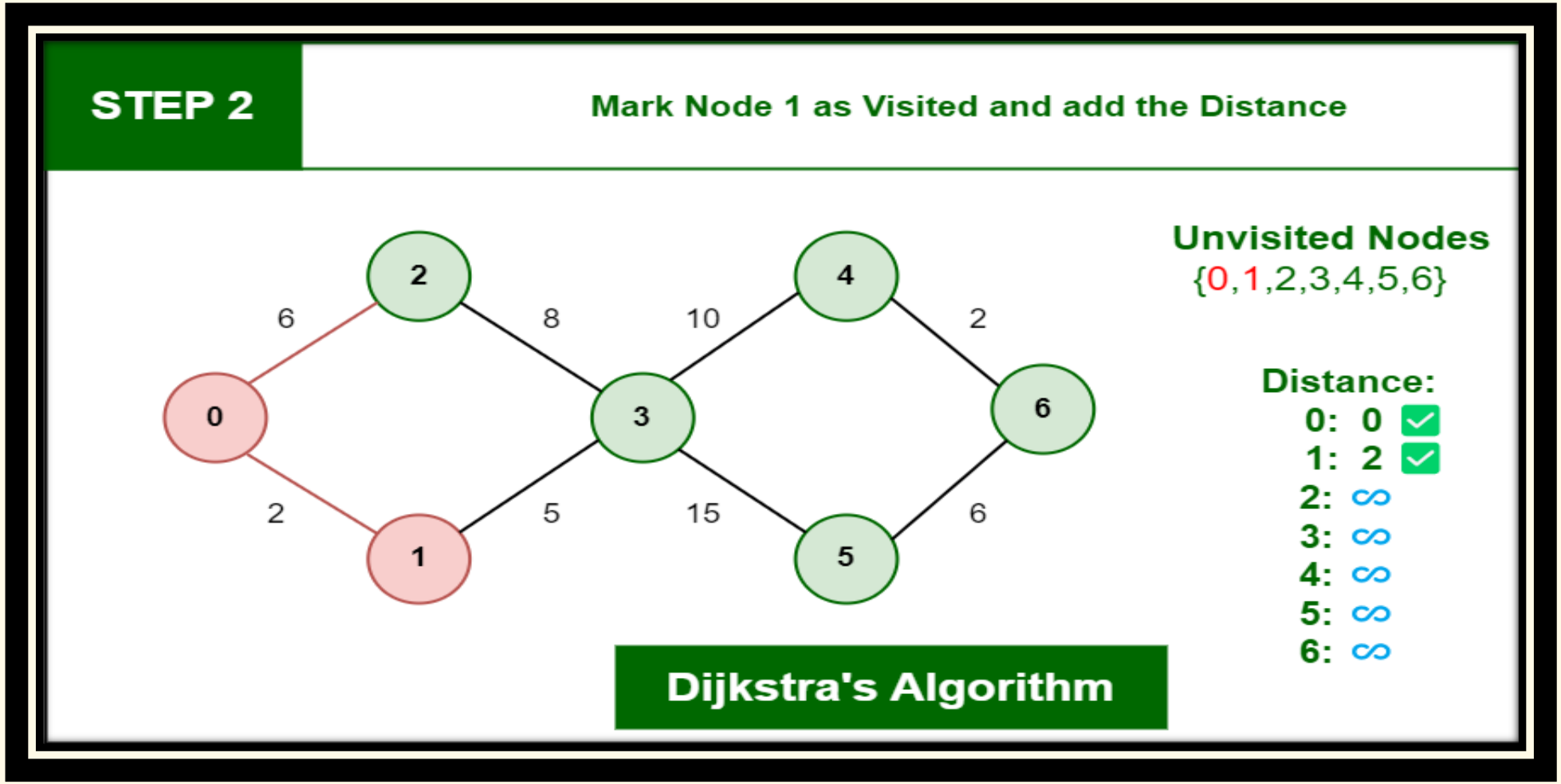
Distance: Node 0 -> Node 1 -> Node 3 = 2 + 5 = 7

Step 4: Again we have two choices for adjacent Nodes (Either we can choose Node 4 with distance 10 or either we can choose Node 5 with distance 15) so choose Node with minimum distance. In this step **Node 4** is Minimum distance adjacent Node, so marked it as visited and add up the distance.

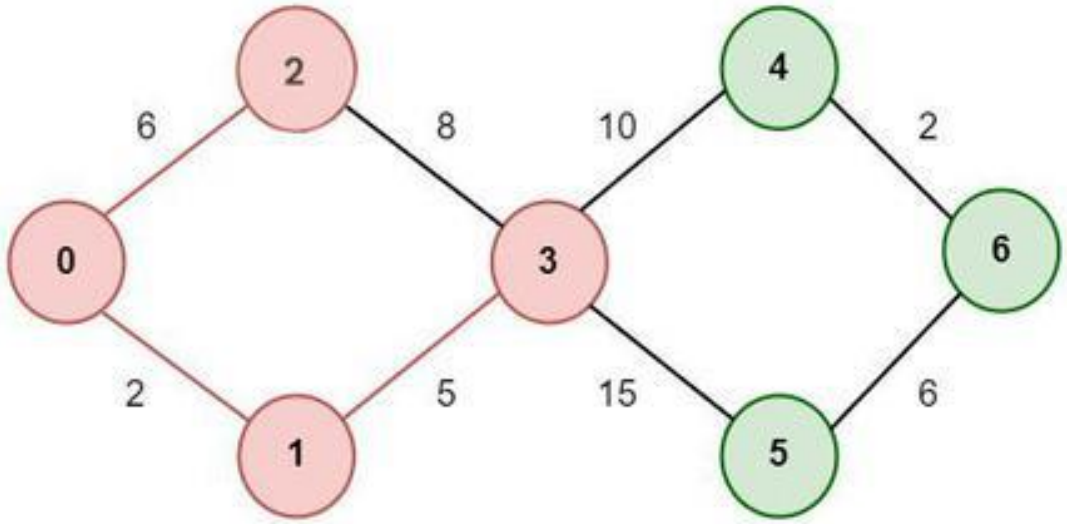
Distance: Node 0 -> Node 1 -> Node 3 -> Node 4 = 2 + 5 + 10 = 17

Step 5: Again, Move Forward and check for adjacent Node which is **Node 6**, so marked it as visited and add up the distance, Now the distance will be:

Distance: Node 0 -> Node 1 -> Node 3 -> Node 4 -> Node 6 = 2 + 5 + 10 + 2 = 19



STEP 3 Mark Node 3 as Visited after considering the Optimal path and add the Distance

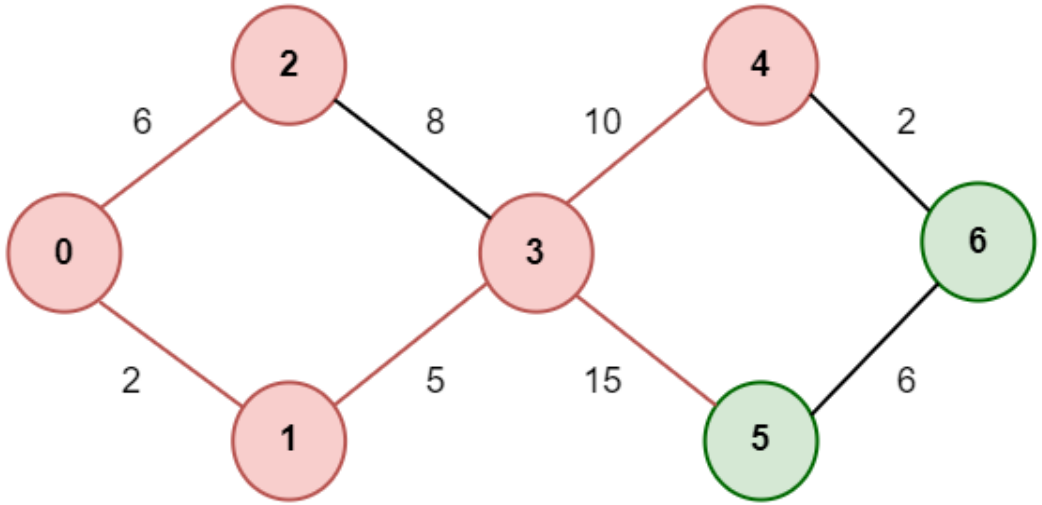


Unvisited Nodes
{0,1,2,3,4,5,6}

Distance:
0: 0 ✓
1: 2 ✓
2: 6 ✓
3: 7 ✓
4: ∞
5: ∞
6: ∞

Dijkstra's Algorithm

STEP 4 Mark Node 4 as Visited after considering the Optimal path and add the Distance

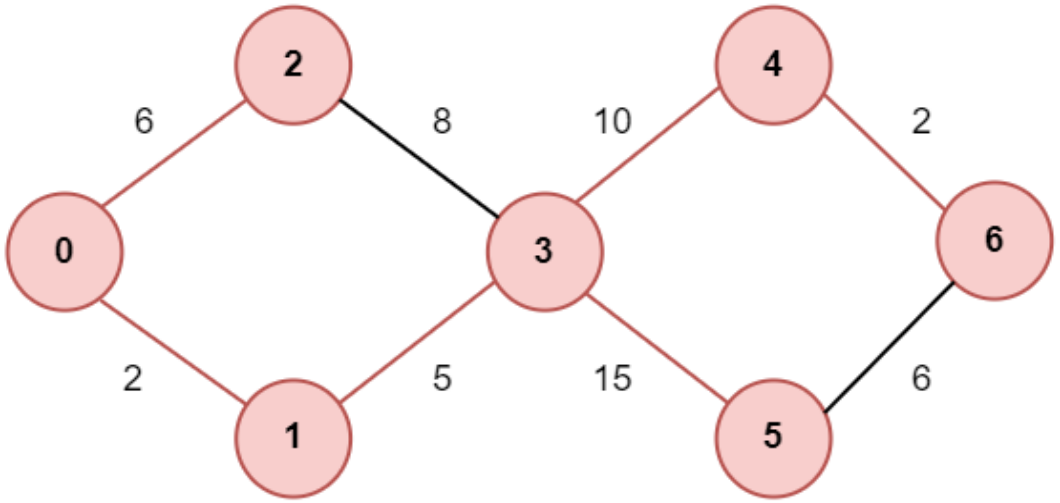


Unvisited Nodes
{0,1,2,3,4,5,6}

Distance:
0: 0 ✓
1: 2 ✓
2: 6 ✓
3: 7 ✓
4: 17 ✓
5: ∞
6: ∞

Dijkstra's Algorithm

STEP 5 Mark Node 6 as Visited and add the Distance



Unvisited Nodes
{0,1,2,3,4,5,6}

Distance:
0: 0 ✓
1: 2 ✓
2: 6 ✓
3: 7 ✓
4: 17 ✓
5: 22 ✓
6: 19 ✓

Dijkstra's Algorithm

So, the Shortest Distance from the Source Vertex is 19 which is optimal one

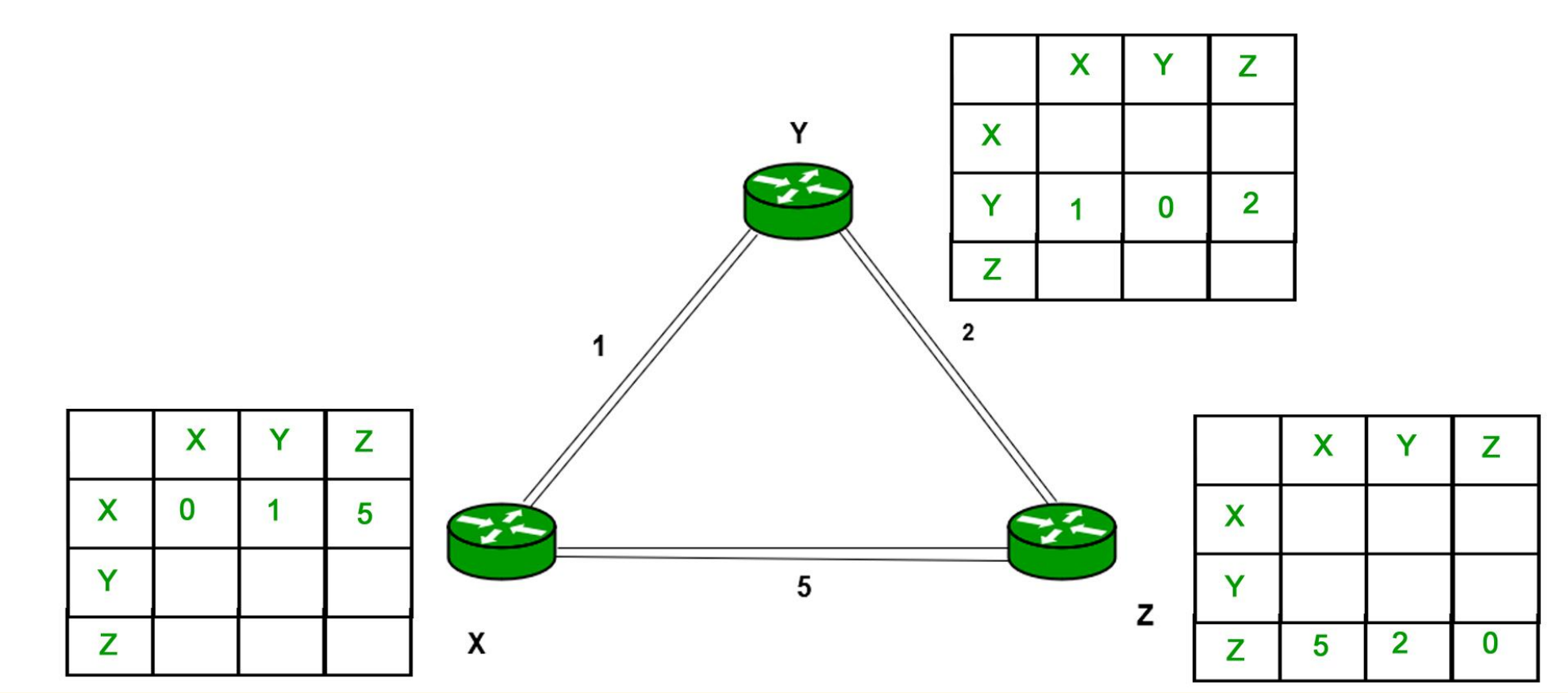
Bellman–Ford Algorithm (Definition)

Bellman–Ford is a **shortest path algorithm** that computes the least-cost path from a single source to all other vertices in a weighted graph. Unlike Dijkstra, **it works even with negative edge weights** (but no negative cycles).

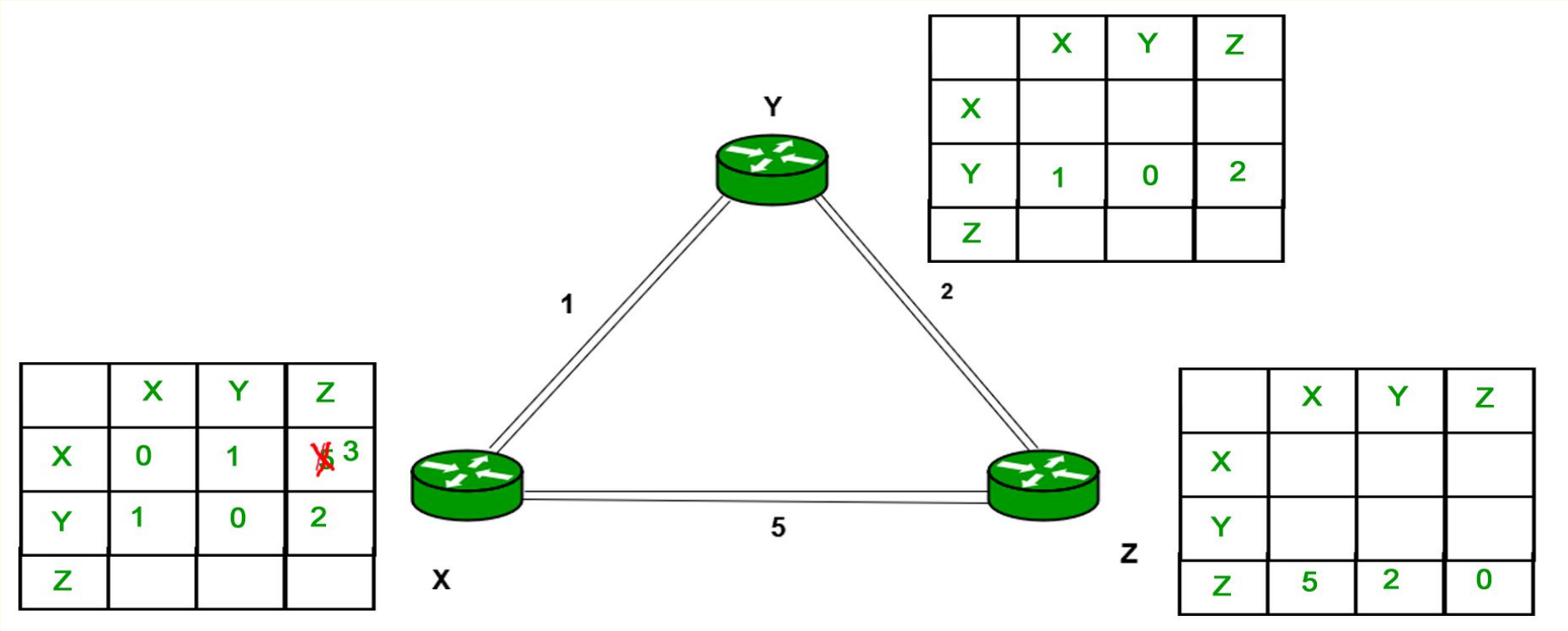
A **distance vector routing algorithm** is used by routers to determine the **best path to a destination** based on **distance metrics** (like cost, or delay).

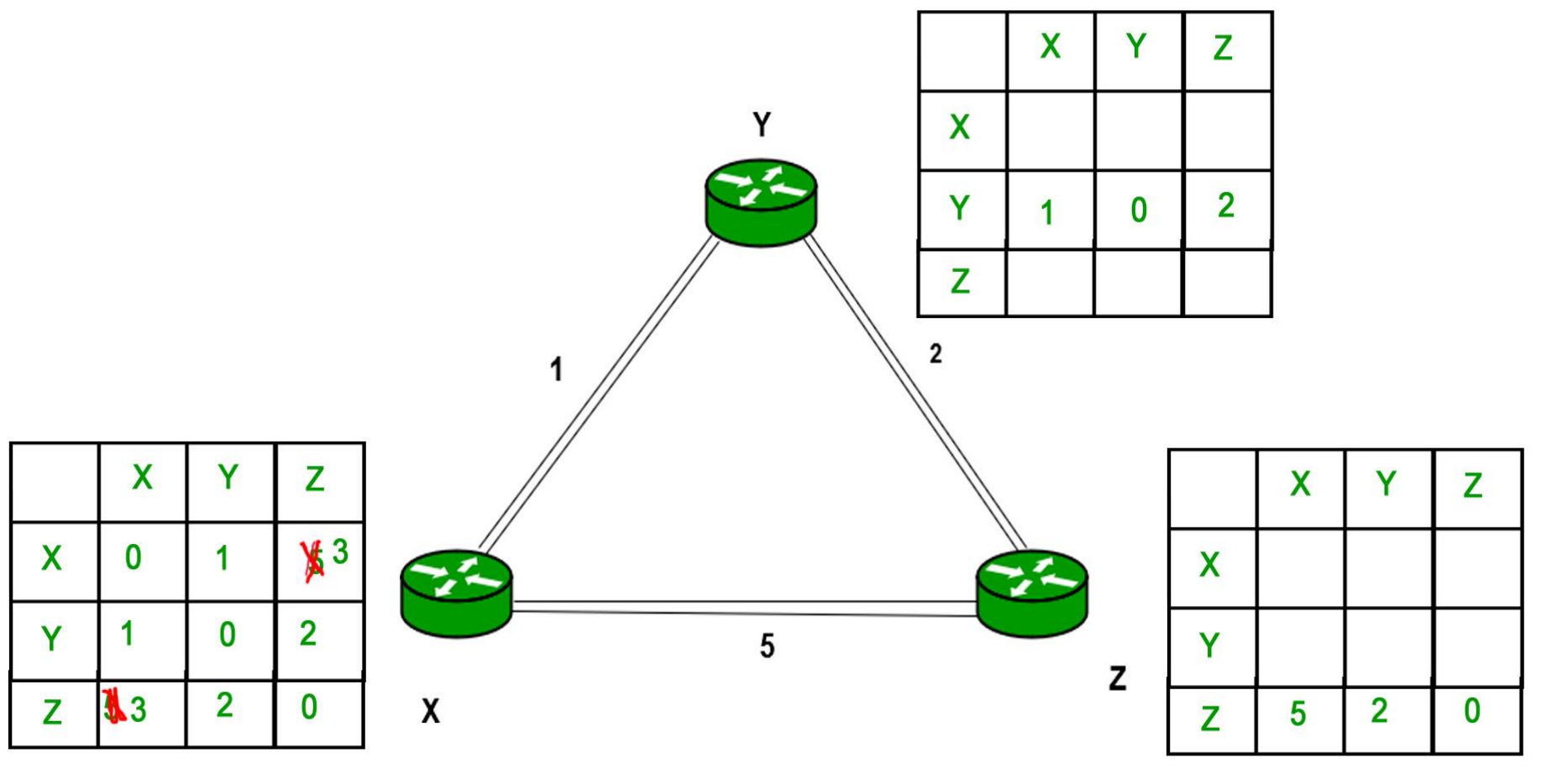
Example :

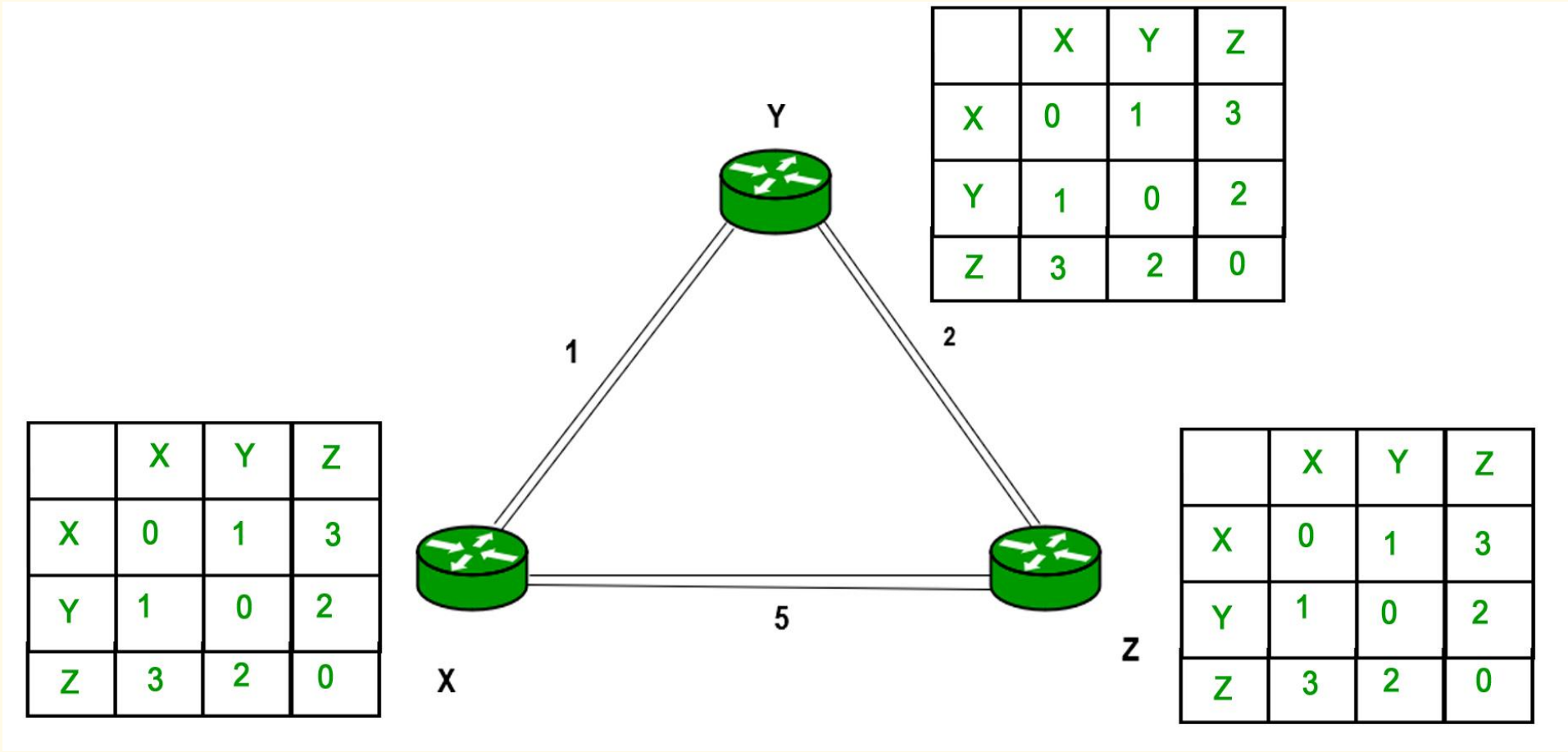
Consider 3-routers X, Y and Z as shown in figure. Each router have their routing table. Every routing table will contain distance to the destination nodes.



Consider router X , X will share it routing table to neighbors and neighbors will share it routing table to it to X and distance from node X to destination will be calculated using bellmen- ford equation.c







Dynamic (Adaptive) Routing

- Flood –Based Routing Algorithm
- Flow –Based Routing Algorithm
- Distance Vector Routing Algorithm
- Link-State Routing Algorithm
- Hierarchical Routing Algorithm

Routing Algorithm

Flood-Based Routing

Every incoming packet is sent (flooded) out of all outgoing links except the one it came from. Ensures delivery but creates heavy traffic.

Flow-Based Routing

Routing decisions are made based on current traffic (flow) in the network to avoid congestion and balance load.

Distance Vector Routing

Each router shares its routing table (distances to all nodes) with its neighbors periodically. Routes chosen based on the shortest distance (Bellman-Ford).

Link-State Routing

Each router knows the full topology and link costs. Routers compute shortest paths using Dijkstra's algorithm.

Hierarchical Routing

Network is divided into regions or levels. Routers handle only intra-region details and summarized info about other regions, reducing table size and complexity.

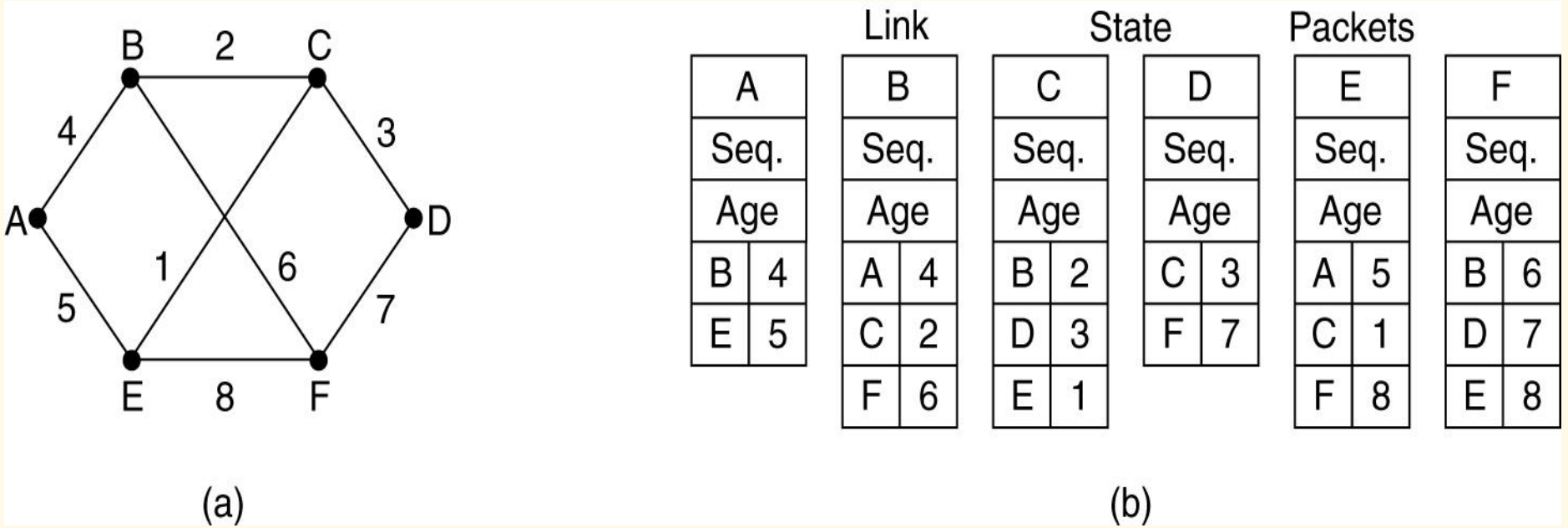
Unicast Routing - Link State Routing

Unicast means the transmission from a single sender to a single receiver. It is a point-to-point communication between the sender and receiver. There are various unicast protocols such as TCP, HTTP, etc.

- TCP (Transmission Control Protocol) is the most commonly used unicast protocol. It is a connection-oriented protocol that relies on acknowledgment from the receiver side.

- HTTP stands for HyperText Transfer Protocol. It is an object-oriented protocol for communication.

Building Link State Packets (Link- State Routing Algorithm)



(a) A network. (b) The link state packets for this network.

Network Layer (OSI Model – 3rd Layer)

•Responsible for **delivering packets** from the source host to the destination host across **multiple networks**.

•Handles **logical addressing** and **routing**.

Function	Explanation
Logical Addressing	Assigns IP addresses to devices (source & destination) so packets can be identified across networks.
Routing	Decides the best path for data packets to reach the destination using routing algorithms.
Packet Forwarding	Moves packets from the input interface to the output interface of a router.
Fragmentation & Reassembly	Breaks large packets into smaller pieces (fragments) to fit into data link layer frames and reassembles them at the destination.

Design Issue of Network Layer

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

1. Store-and-Forward Packet Switching

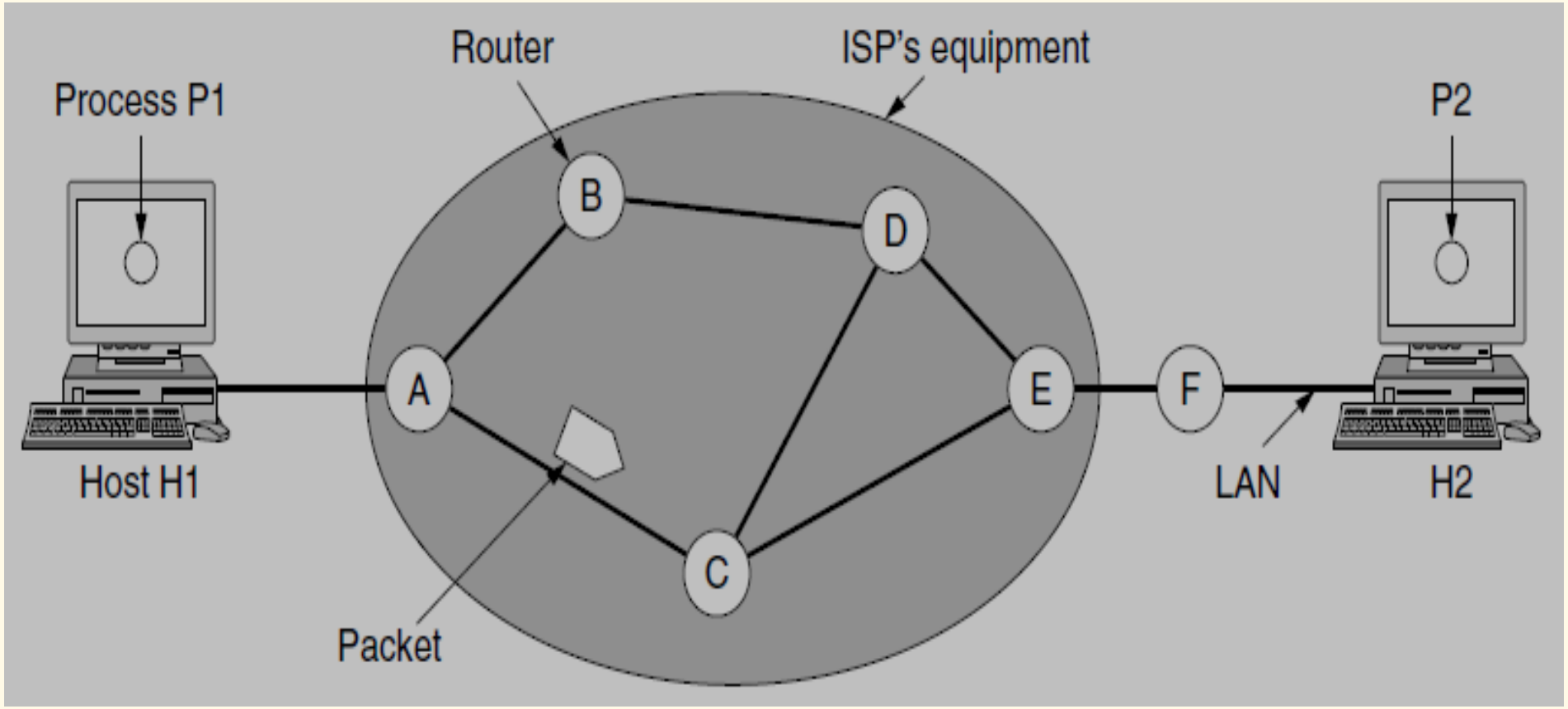
•Each packet is received completely at an intermediate node (router/switch), stored temporarily, then forwarded to the next node.

•Used in most modern networks including the Internet.

The environment of the network layer protocols

- Host sends packet to nearest router
- Packet stored there until it has fully arrived, then forwarded to next router

Store-and-Forward Packet Switching



2.Services Provided to the Transport Layer

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

Debate regarding design of the network layer:

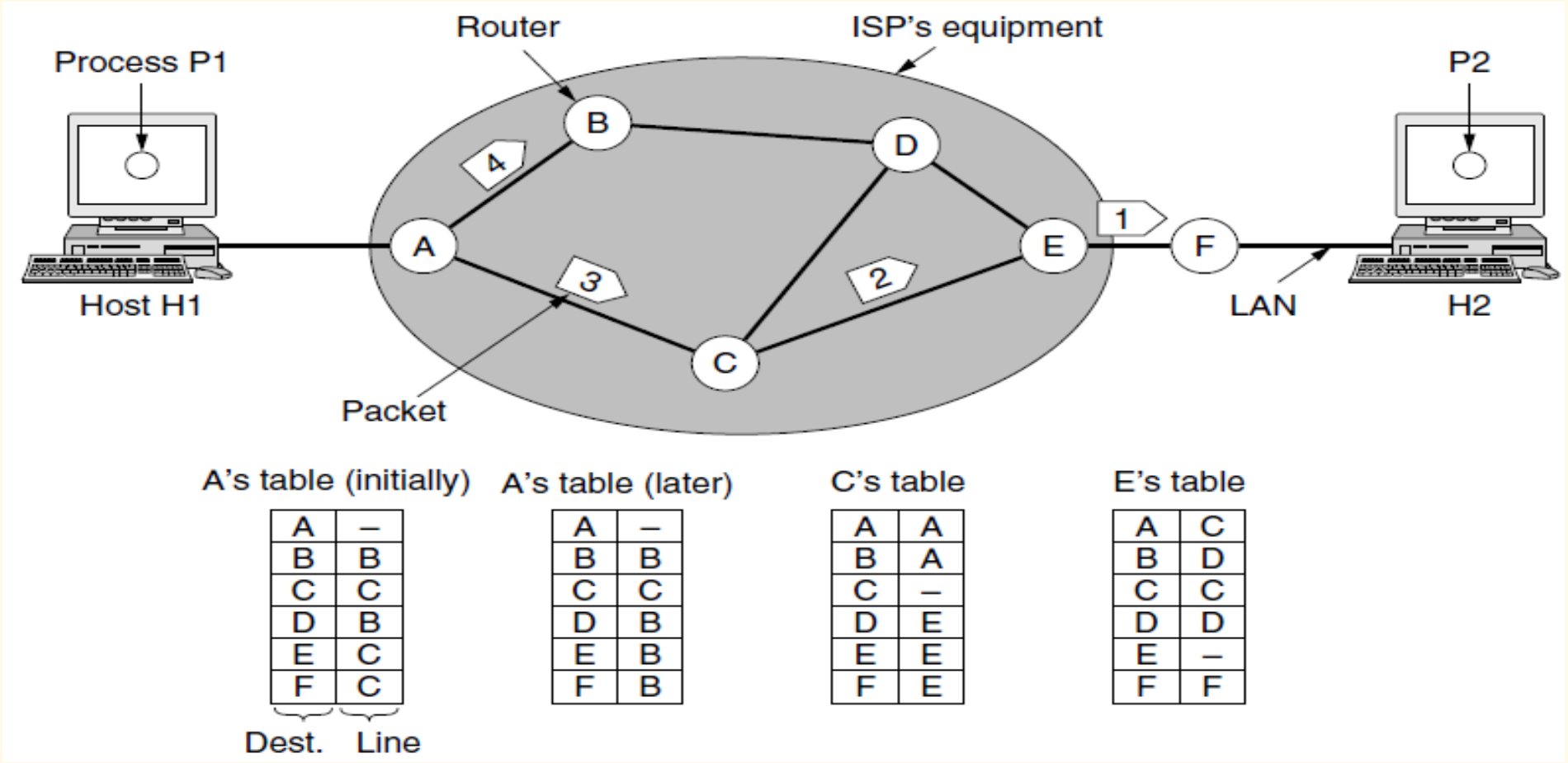
- Internet Camp – Connectionless, e.g., IP
- Telephone Companies – Connection-oriented, e.g., ATM

3.Implementation of Connectionless Services

- No setup before data transfer.
- Each packet carries full destination address.
- Routers treat packets independently (may take different paths).
- Example: **IP (Internet Protocol)**.

Routing within a datagram network

Routing Algorithm – Algorithm that manages the tables and makes the routing decisions



A's table (initially)

Dest.	Line
A	–
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	–
B	B
C	C
D	B
E	B
F	B

C's table

A	A
B	A
C	–
D	E
E	E
F	E

E's table

A	C
B	D
C	C
D	D
E	–
F	F

4. Implementation of Connection-Oriented Service

•A virtual circuit (VC) is established between source and destination before data transfer.

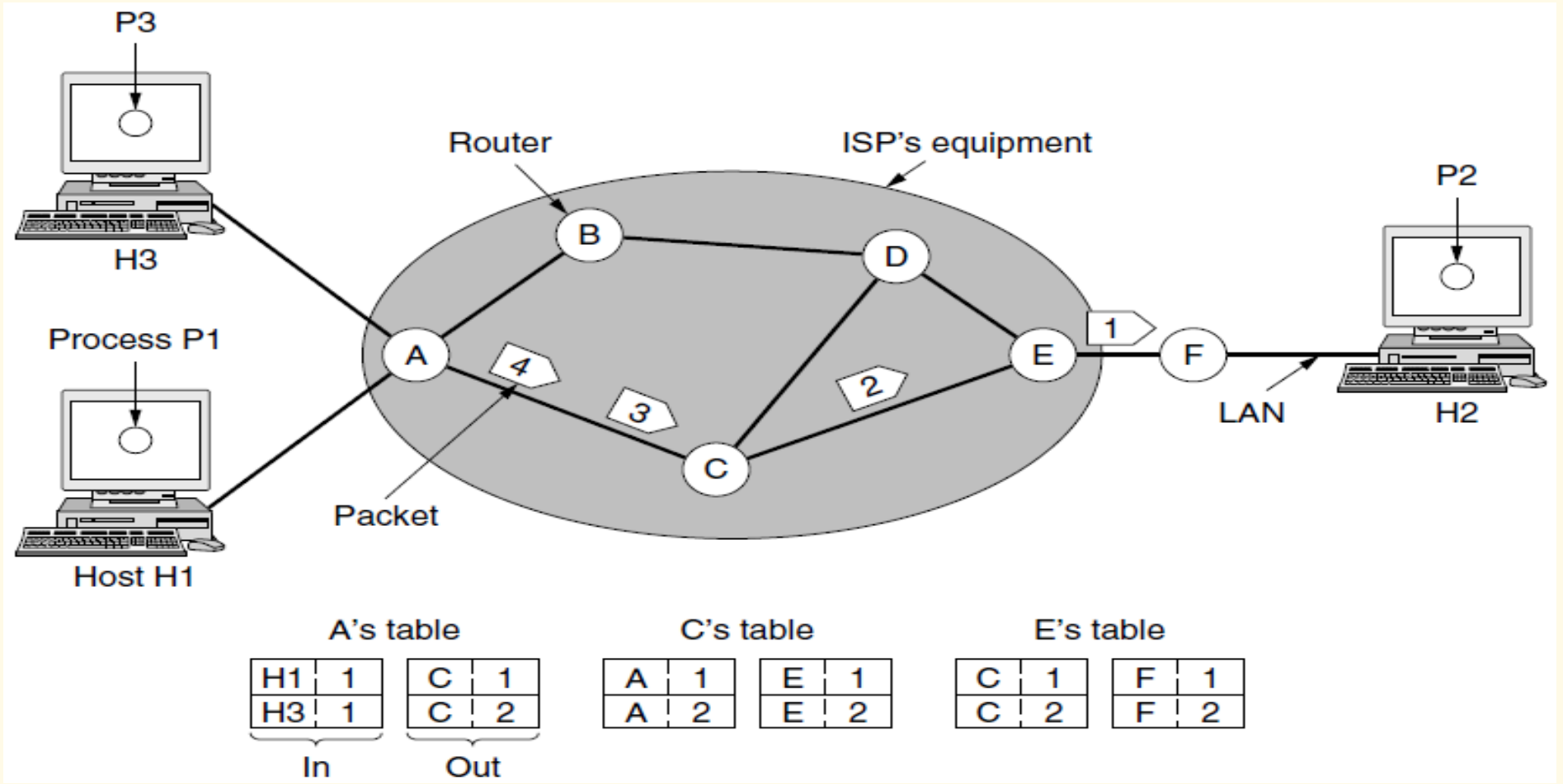
•All packets follow the same path.

•Routers maintain state information for each connection.

•Example: **ATM, Frame Relay**.

Fig: Routing within a virtual-circuit subnet

- **Label switching**, e.g., MPLS (Multi Protocol Label Switching - with IP packets wrapped in an
- MPLS header having a 20-bit connection identifier or label.)



Comparison of Virtual – Circuit and Datagram Subnets

Feature	Virtual Circuit (VC)	Datagram (Connectionless)
Setup	Requires setup phase before data transfer.	No setup phase needed.
Path	All packets follow the same path.	Each packet may follow a different path.
Address in Packet	Short VC number used.	Full destination address in each packet.
Reliability	More reliable, can reserve resources.	Less reliable, no resource reservation.
Example	ATM, Frame Relay.	Internet (IP).

What is Congestion?

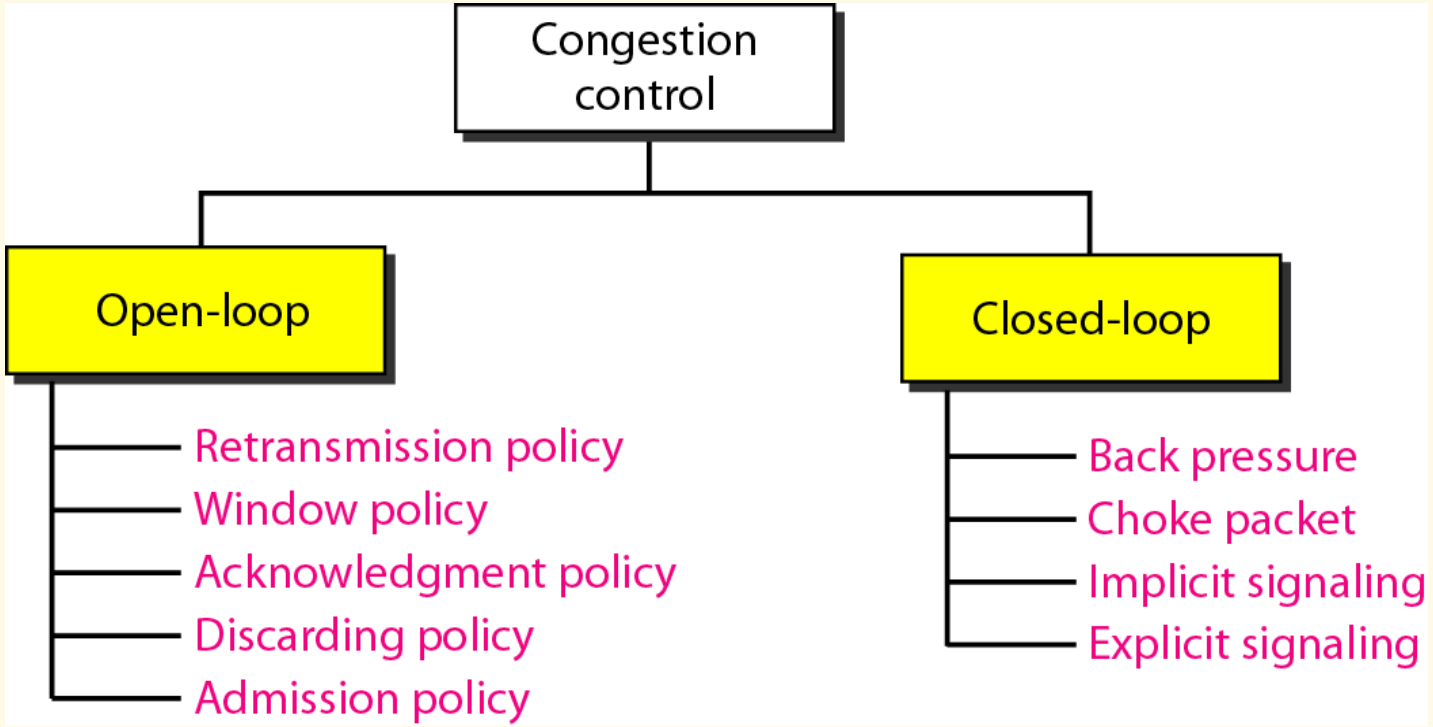
Congestion occurs when **too many packets** are present in the network (routers, switches, or links), causing:

- Packet delay
- Packet loss
- Reduced performance

What is Congestion Control?

It is a set of **techniques** to **prevent or remove congestion** in a network so that performance remains acceptable.

Congestion control categories



Causes of Congestion:

- Too much traffic (data sent faster than network can handle).
- Insufficient router buffer space.
- Slow links or router processing.
- Bursty data traffic.

Types of Congestion Control

Type	Meaning	Example
Open-Loop Control (Prevention)	Designed **before** congestion occurs. Preventive policies.	traffic shaping, proper routing, resource reservation.
Closed-Loop Control (Removal)	Reacts **after** congestion occurs. Uses feedback to adjust traffic.	Packet dropping, choke packets, load shedding.

Open-Loop Congestion Control

Retransmission Policy

Window Policy

Acknowledgement Policy

Discarding Policy

Admission Policy

Retransmission Policy

Decides when and how lost packets are retransmitted.

Poor retransmission policy (like frequent timeouts) can worsen congestion. Good policy reduces unnecessary retransmissions.

Window Policy

Uses a window (sliding window) to control the number of outstanding unacknowledged packets.

Prevents sender from flooding the network.

Acknowledgement Policy

Decides how/when acknowledgements (ACKs) are sent.

Delayed or combined ACKs reduce extra traffic; immediate ACKs increase load.

Discarding Policy

Routers discard less important packets during congestion.

Reduces network load and gives priority to critical packets.

Admission Policy

Decides whether to accept or reject new traffic into the network based on current load.

Prevents overload by refusing extra traffic.

Closed-loop congestion control detects congestion during operation and uses feedback (like choke packets) to dynamically reduce network load.

Closed - Loop Congestion Control

- Backpressure
- Choke Packet
- Implicit Signaling
- Explicit Signaling
  - Backward Signaling
  - Forward Signaling

1. Backpressure

•**Meaning:** When a router becomes congested, it **slows down or blocks incoming traffic** from the previous router.

2.Choke Packet

•**Meaning:** A special **control packet** sent by a congested router directly to the **source** to reduce its sending rate.

3. Implicit Signaling

•**Meaning:** Congestion is **inferred indirectly** by the sender from network behavior (no explicit message).

•**Examples:** Increased **packet delay**, **packet loss**, or dropped ACKs suggest congestion.

4. Explicit Signaling

•**Meaning:** Network **explicitly informs** the sender or receiver about congestion.

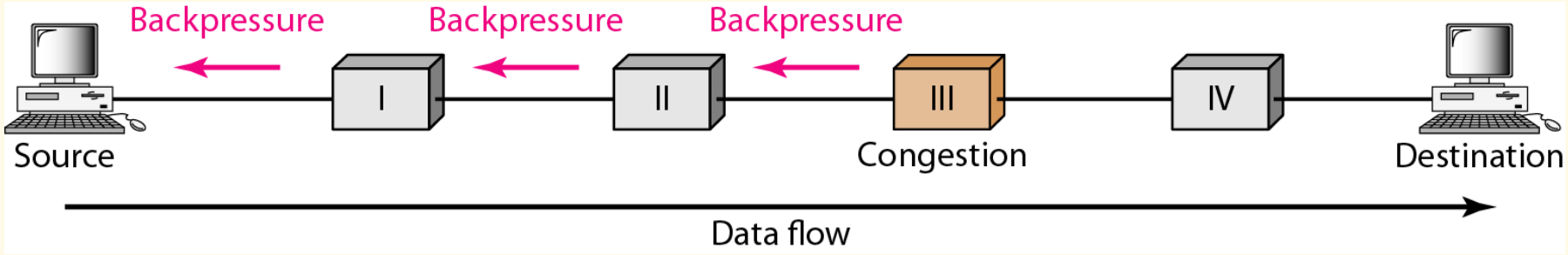
5. Backward Signaling

•**Meaning:** Congestion signal is sent **toward the source (upstream)**.

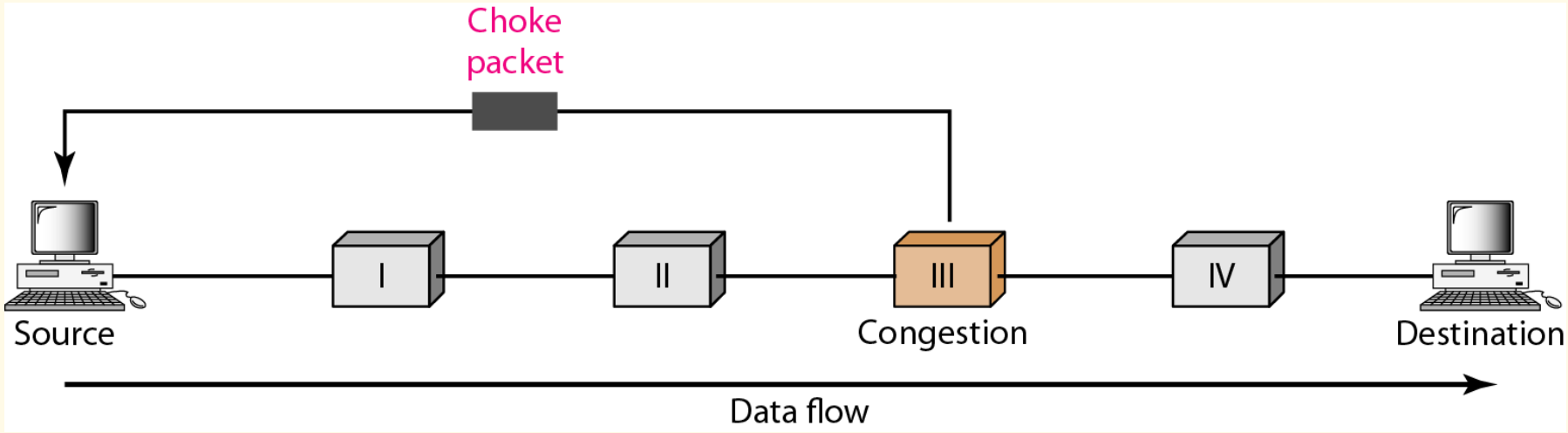
6. Forward Signaling

•**Meaning:** Congestion signal is sent **toward the destination (downstream)**.

Backpressure method for alleviating congestion



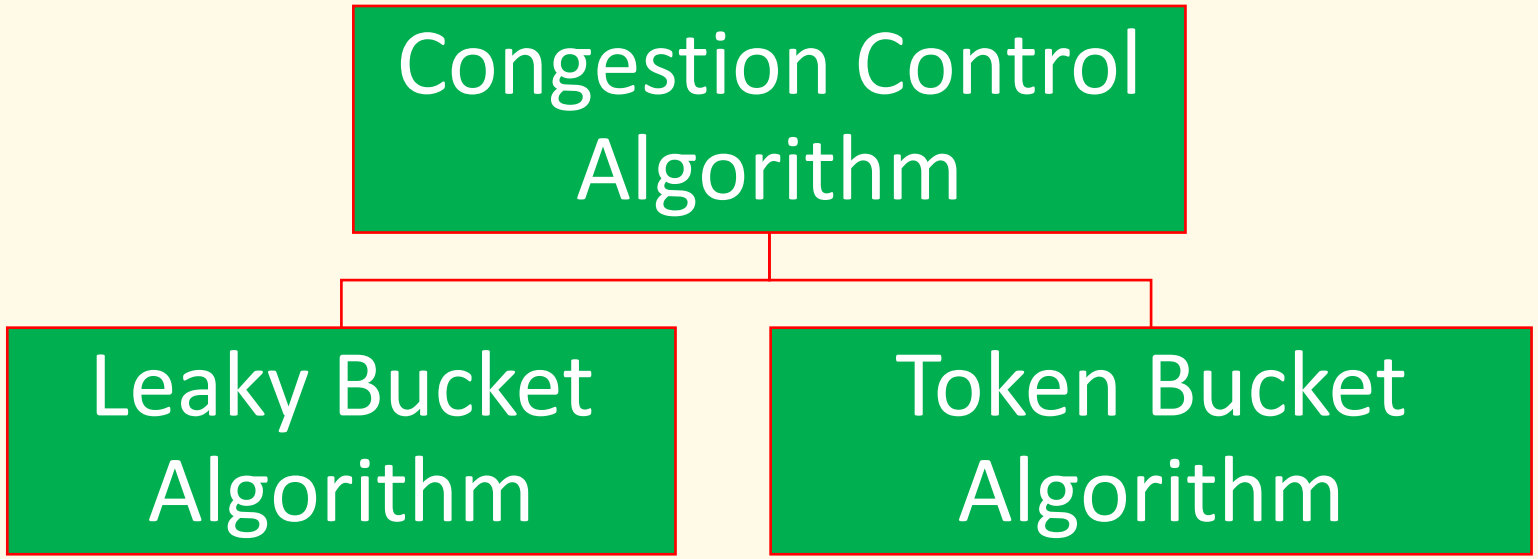
Choke packet



Term	Direction / Style	Key Idea
Backpressure	Backward (hop by hop)	Router slows incoming traffic.
Choke Packet	Backward (source)	Special control packet to slow sender.
Implicit Signaling	No explicit packet	Sender infers congestion by delay/loss.
Explicit Signaling	With explicit packet or bit	Network tells sender/receiver about congestion.
Backward Signaling	Toward the source	Asks to slow down.
Forward Signaling	Toward the destination	Informs receiver of congestion.

Congestion Control Algorithms

Congestion control algorithms can be divided into **open-loop (prevention)** and **closed-loop (removal)** techniques.

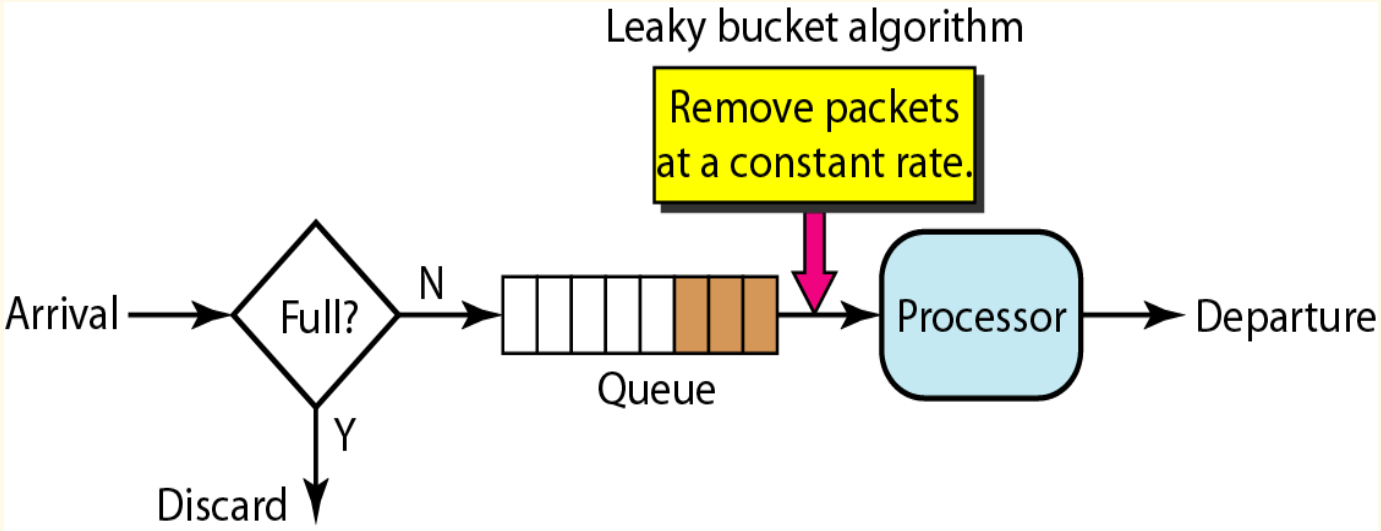


1. Leaky Bucket Algorithm (Traffic Shaping)

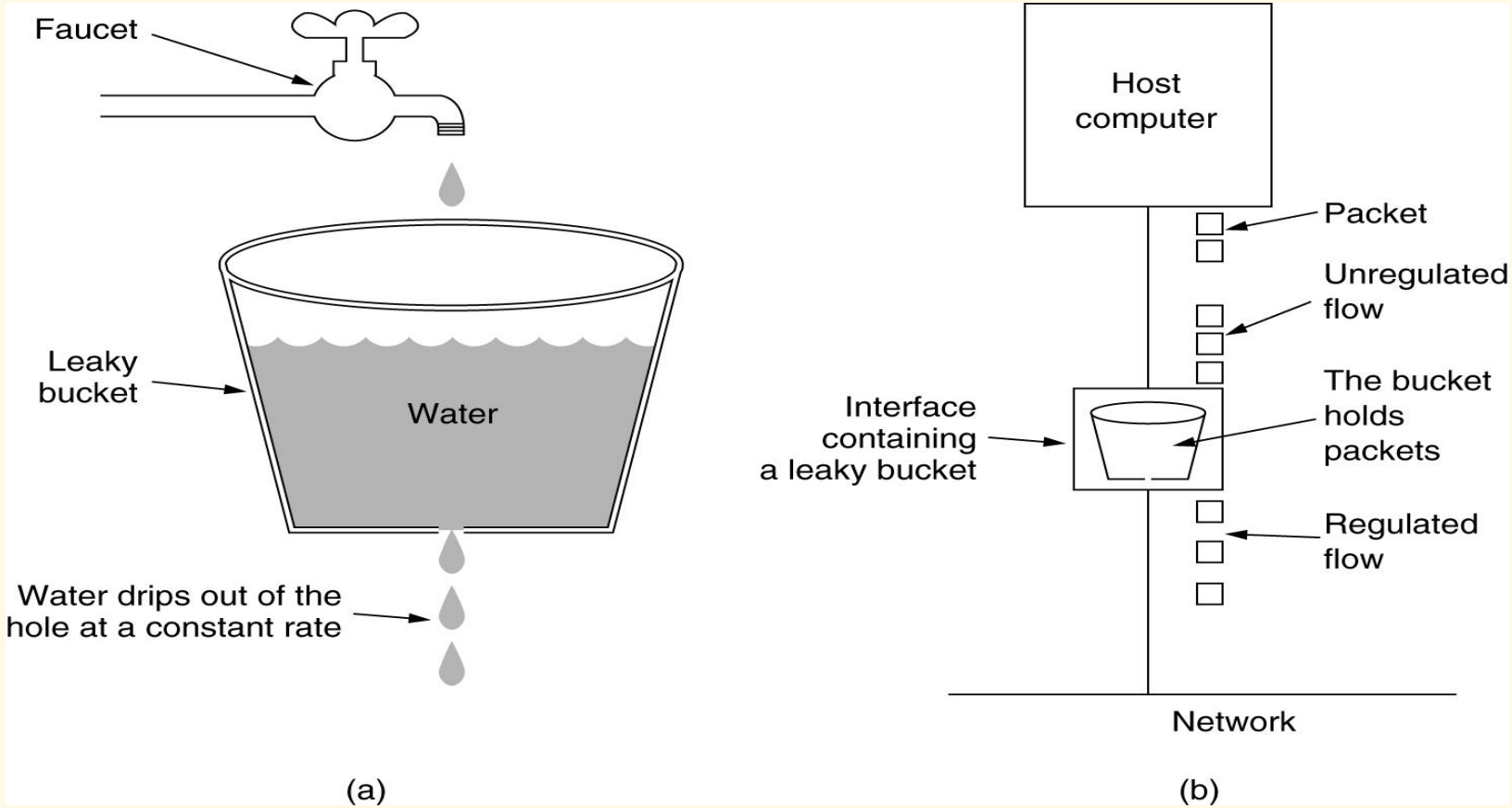
- **Idea:** Packets enter a bucket (fixed size).
- If the bucket is full, extra packets are discarded.
- Packets are sent out at a **constant rate**.
- **Prevents sudden bursts** of traffic.

A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.

Leaky bucket implementation



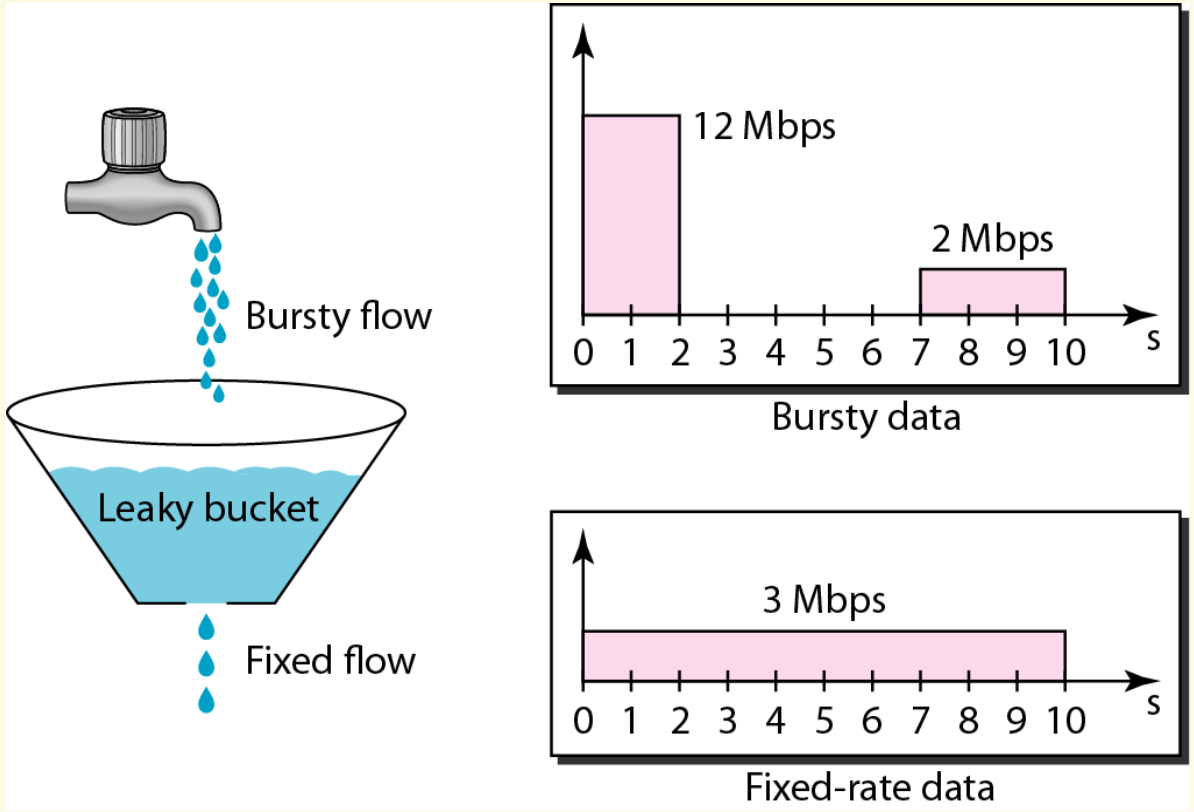
The Leaky Bucket Algorithm



(a) A leaky bucket with water.

(b) a leaky bucket with packets.

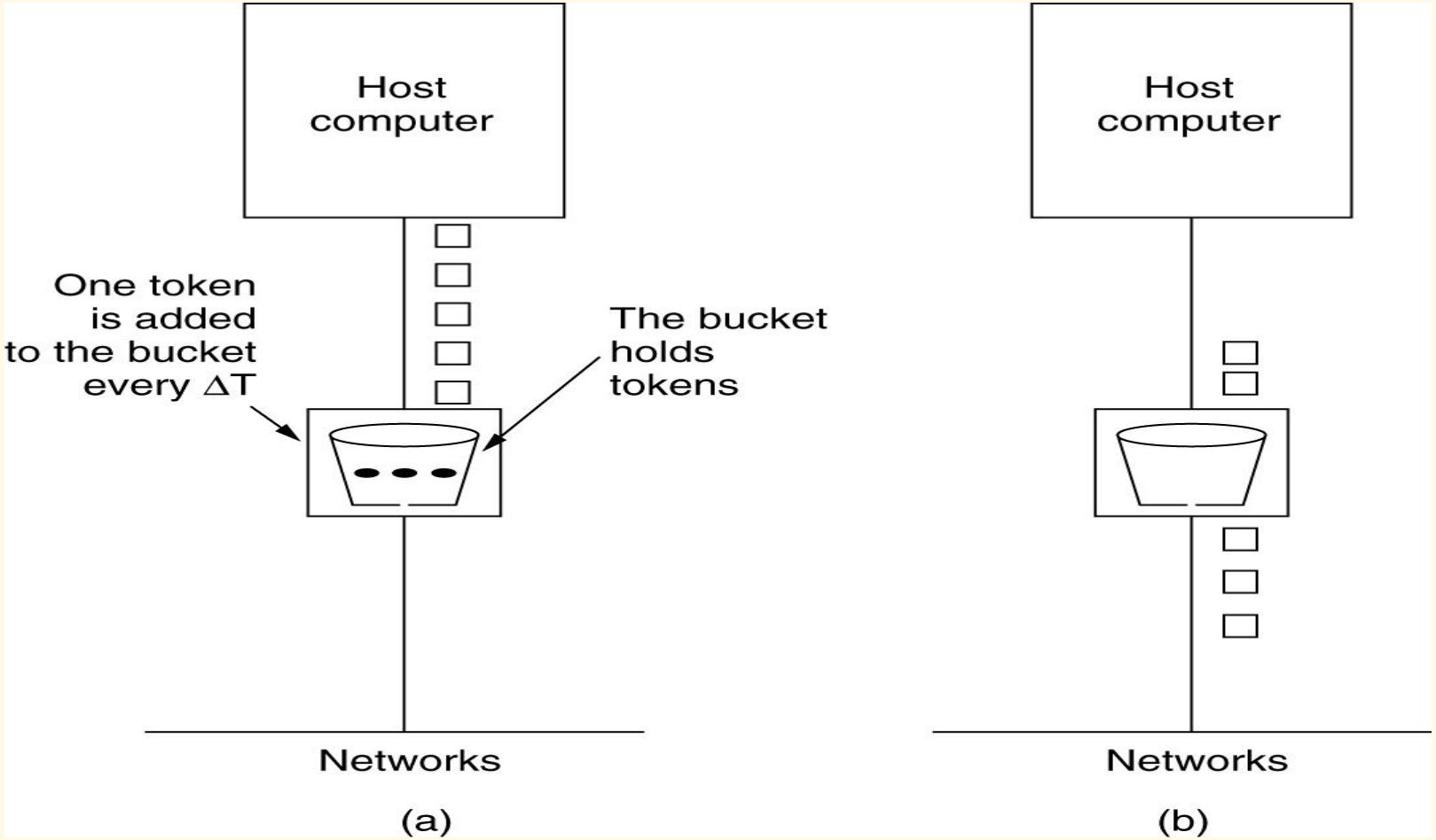
Leaky bucket



2. Token Bucket Algorithm (Traffic Shaping)

- **Idea:** Tokens are generated at a fixed rate.
- To send a packet, a token is needed.
- Allows **bursts of traffic** (if tokens are available) while controlling the average rate.

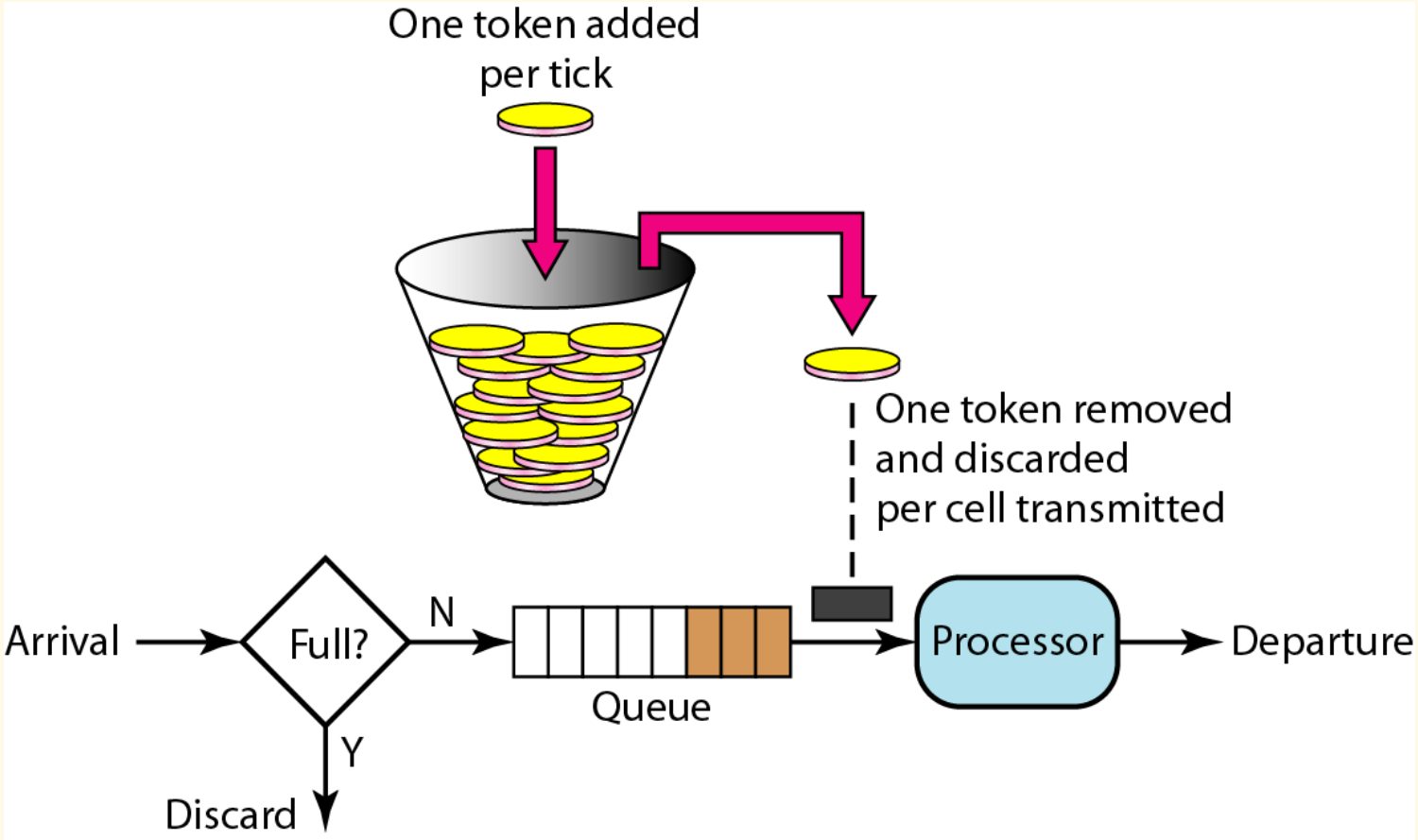
The token bucket allows bursty traffic at a regulated maximum rate.



The Token Bucket Algorithm

(a) Before. (b) After.

Token bucket



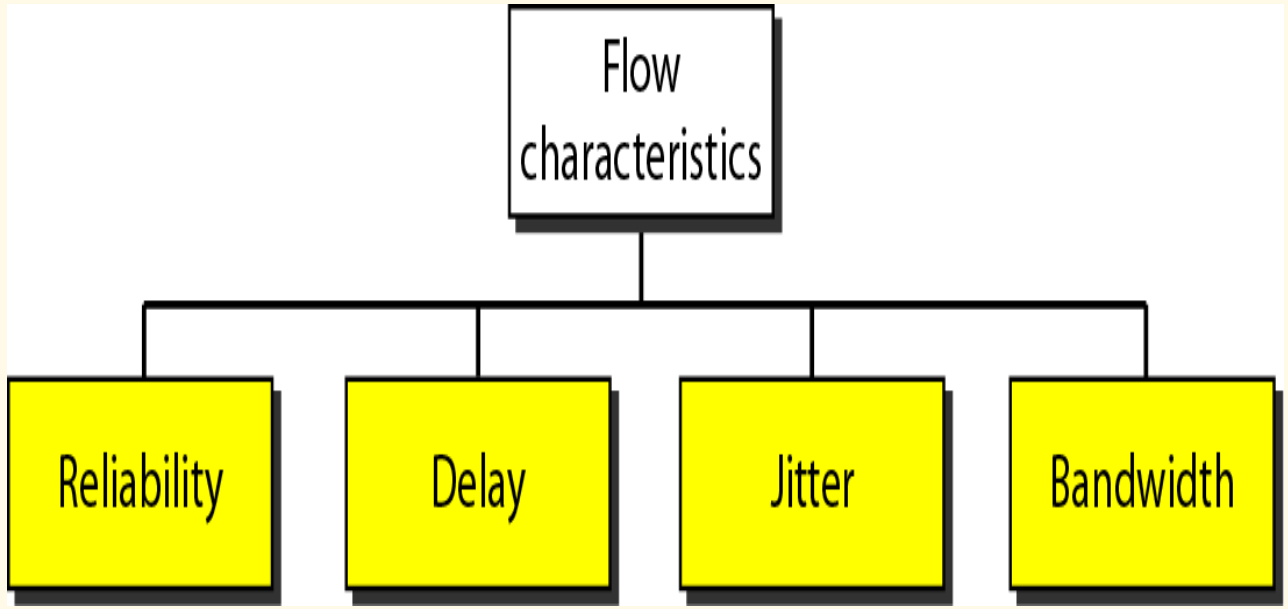
Algorithm	Type	Key Idea
Leaky Bucket	Open-loop	Sends packets at constant rate, discards overflow.
Token Bucket	Open-loop	Allows bursts but limits average rate.

QUALITY OF SERVICE

Quality of Service (QoS) refers to the ability of a network to **provide different priority levels** to different applications, users, or data flows, and to guarantee **performance measures** like bandwidth, delay, jitter, and reliability.

QoS ensures that important traffic (like video calls, VoIP, real-time apps) gets better service than normal traffic.

Flow characteristics



Parameter	Meaning	Why Important?
Bandwidth	Amount of data transmitted per second.	Ensures enough capacity for heavy apps (video, large files).
Delay (Latency)	Time taken for a packet to travel from source to destination.	Must be low for real-time apps (telephony, gaming).
Jitter	Variation in packet delay.	High jitter ruins voice/video quality.
Reliability	Probability of packet delivery without loss.	Important for file transfers, financial transactions.

How stringent the quality-of-service requirements are.

Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

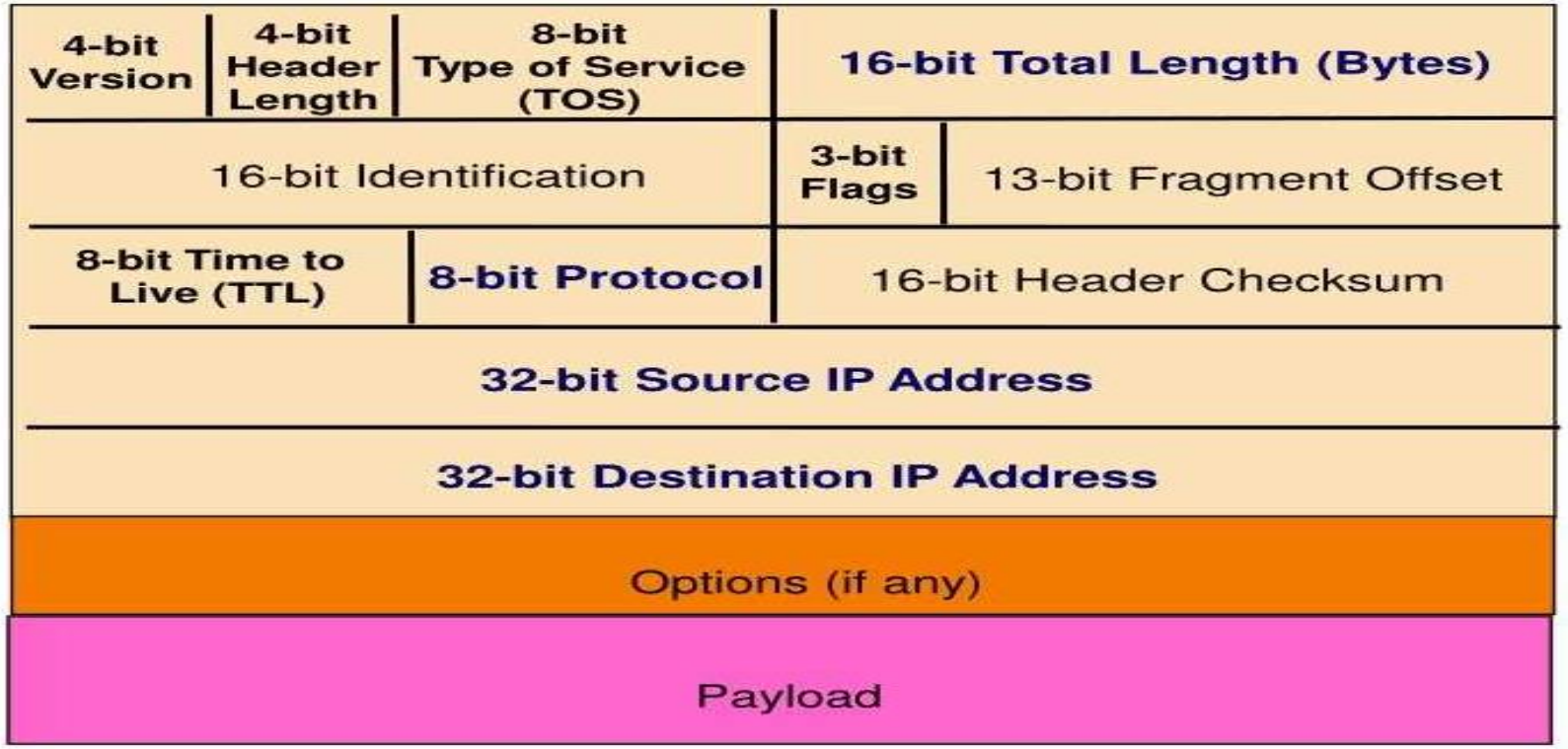
Categories of QoS and Examples

1. Constant bit rate
   - Telephony
2. Real-time variable bit rate
   - Compressed videoconferencing
3. Non-real-time variable bit rate
   - Watching a movie on demand
4. Available bit rate
   - File transfer

Network Layer in the Internet

- The **network layer** in the Internet is mainly implemented by the **Internet Protocol (IP)**.
- It is responsible for **host-to-host packet delivery** across multiple networks.
- Uses a **connectionless, best-effort service** (packets may be lost, duplicated, or arrive out of order).

IP Packet Structure



Fourteen Fields Comprise an IP Packet

1. **Version** – IP version (4 bits).
2. **Header Length (IHL)** – Size of header (4 bits).
3. **Type of Service (ToS/DSCP)** – QoS / priority info.
4. **Total Length** – Entire packet size (header + data).
5. **Identification** – Unique ID for packet fragments.
6. **Flags** – Control fragmentation.
7. **Fragment Offset** – Position of fragment in original packet.
8. **Time to Live (TTL)** – Limits packet lifetime (hops).
9. **Protocol** – Next layer protocol (TCP = 6, UDP = 17).
10. **Header Checksum** – Error check of header only.
11. **Source IP Address** – Sender's IP.
12. **Destination IP Address** – Receiver's IP.
13. **Options** – Extra control (optional, rarely used).
14. **Padding** – Fills to make header multiple of 32 bits.

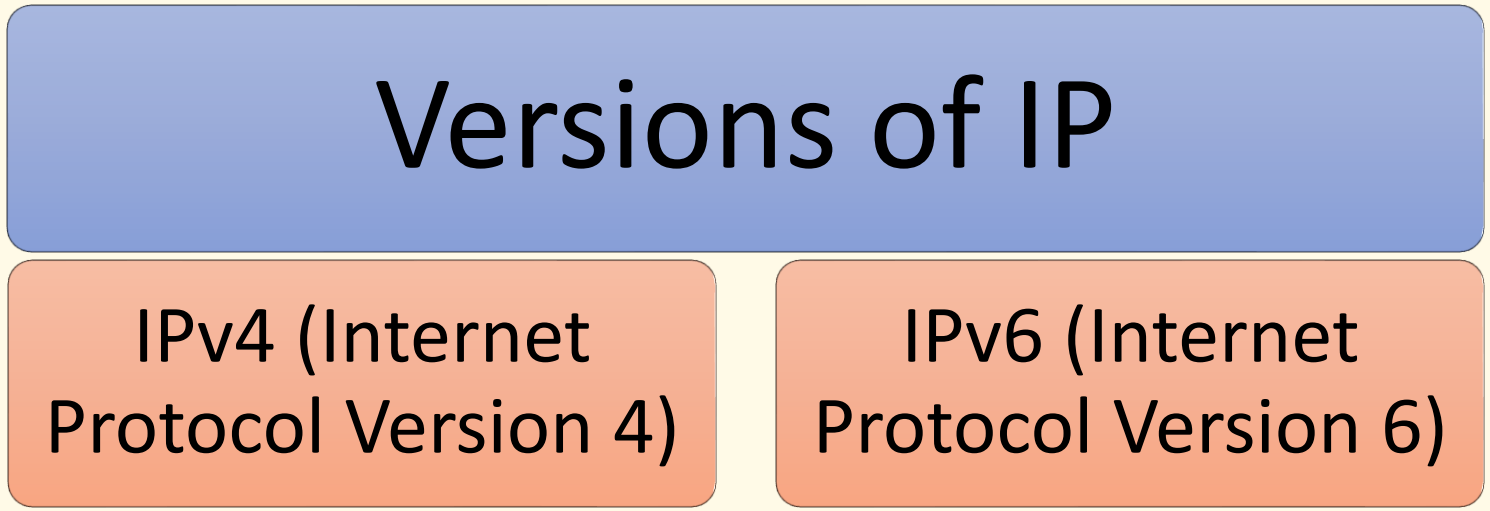
IP Address Format

An **IP Address** is a **unique identifier** for each device on a network.

It has two main parts:

1.Network ID → Identifies the network.

2.Host ID → Identifies the device (host) within that network.



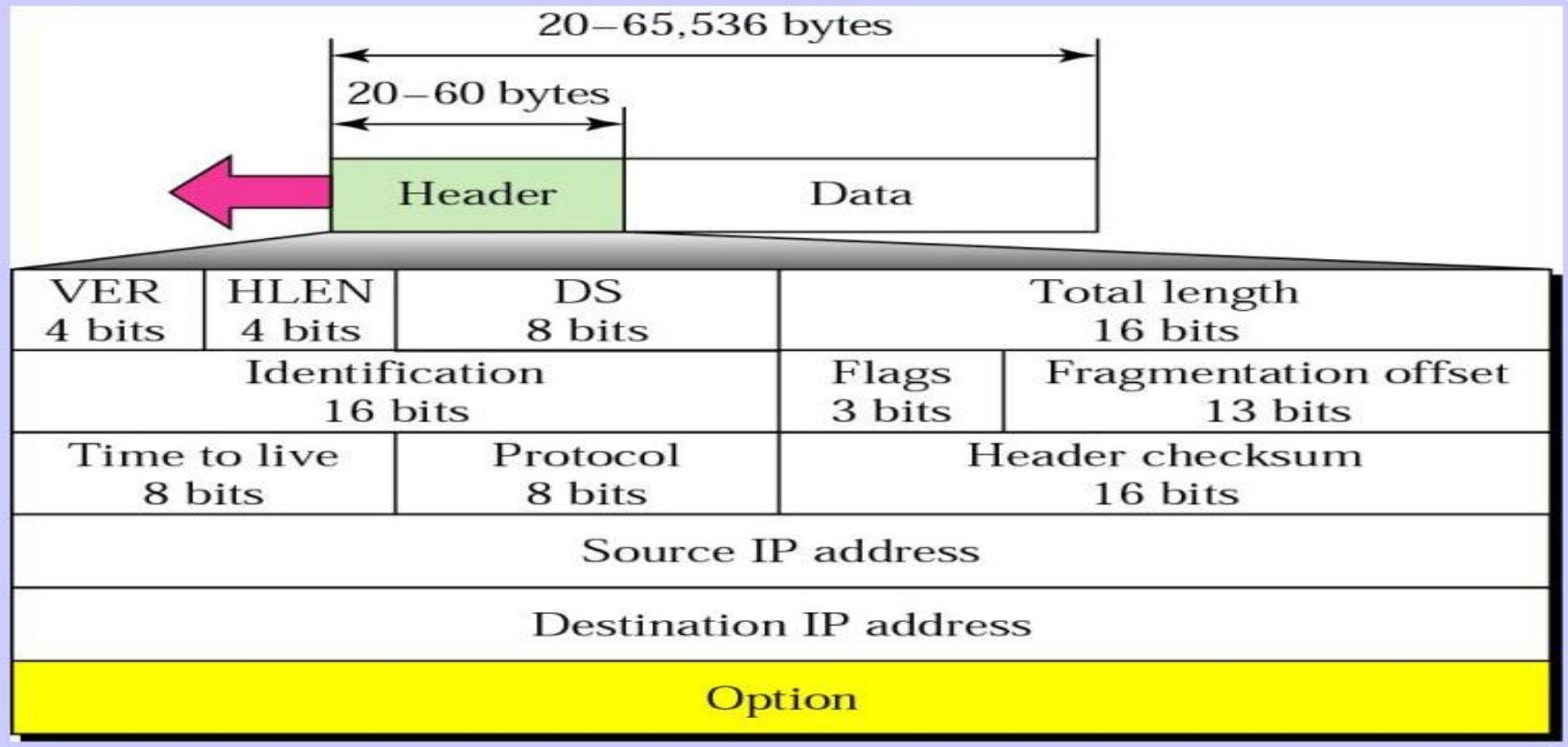
1. IPv4 Address Format

- **Size:** 32 bits (4 bytes).
- **Written as:** 4 decimal numbers (0–255) separated by dots.
  - Example: `192.168.1.10`
- **Structure:**
  - **Network part** + **Host part** (depends on class/subnet).
- **Classes:** A, B, C, D, E.

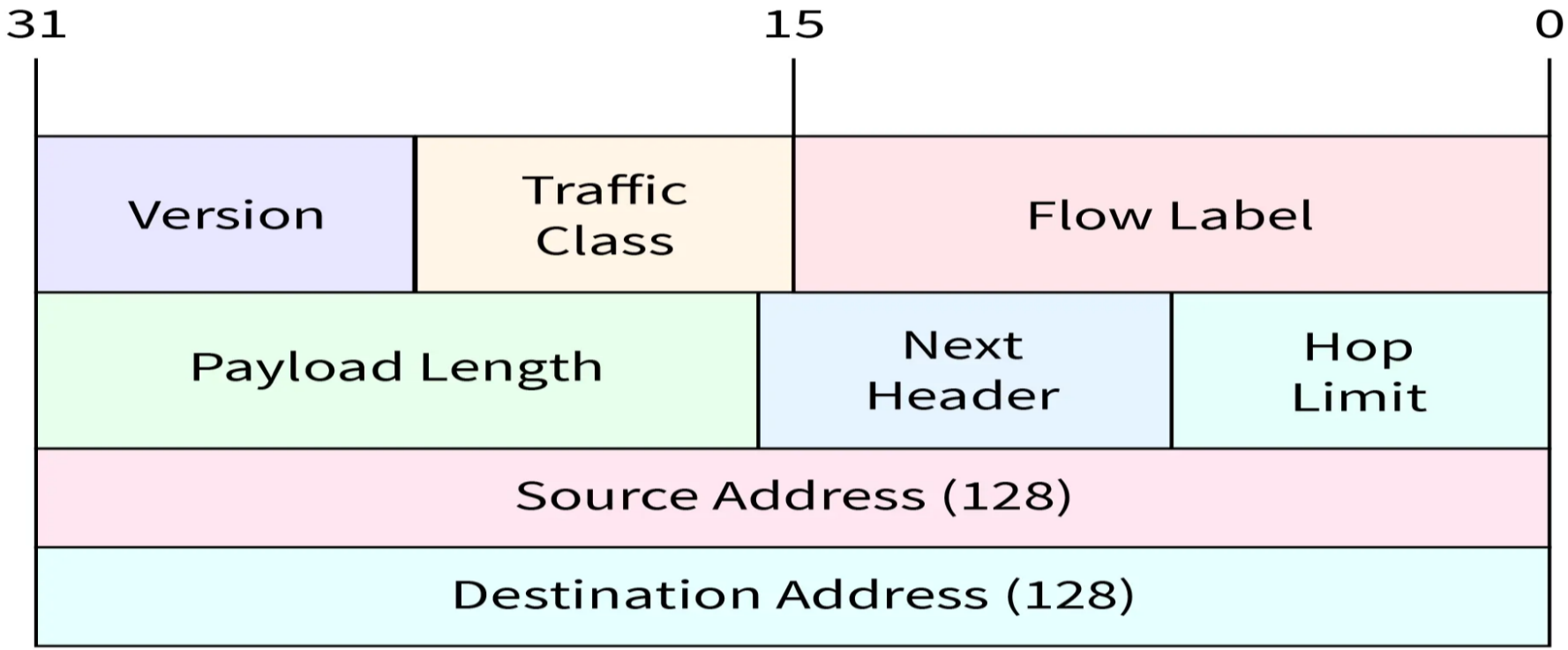
2. IPv6 Address Format

- **Size:** 128 bits (16 bytes).
- **Written as:** 8 groups of 4 hexadecimal digits separated by colons.
  - Example: `2001:0db8:85a3:0000:0000:8a2e:0370:7334`
  - Can be shortened: `2001:db8:85a3::8a2e:370:7334`
- **Structure:**
  - **Prefix (Network ID)** + **Interface ID (Host ID)**.
- **Types:** Unicast, Multicast, Anycast (no broadcast).

IPV4 datagram format



IPv6 Header Format



Feature	IPv4	IPv6
Address Size	32-bit	128-bit
Address Format	Dotted decimal (192.168.1.1)	Hexadecimal colon (2001:db8::1)
Header Size	20–60 bytes (variable)	Fixed 40 bytes
Total Addresses	~4.3 billion	~3.4×10^{38} (virtually unlimited)
Security	Optional (IPSec)	Mandatory (IPSec)
Broadcast	Supported	Not supported (uses multicast/anycast instead)
Configuration	Manual or DHCP	Auto-configuration supported
Fragmentation	Done by sender & routers	Done only by sender
Example Address	192.168.0.1	2001:db8::1

Internet Control Protocols

These are protocols that work **alongside IP** in the **network layer of the Internet** to provide additional functions like error reporting, address mapping, and group communication.

Internet Control Protocols

1.ARP Address Resolution Protocol
2.RARP Reverse Address Resolution Protocol
3.ICMP Internet Control Message Protocol
4.IGMP Internet Group Management Protocol

1. ARP (Address Resolution Protocol)

- **Purpose:** Maps a **logical IP address → physical MAC address**.
- **Works:**
  - Host sends an ARP request: *"Who has IP 192.168.1.2?"*
  - The device with that IP replies with its MAC address.
- **Use:** Used in LANs for packet delivery.

2. RARP (Reverse Address Resolution Protocol)

- **Purpose:** Maps a **physical MAC address → logical IP address**.
- **Works:**
  - Used by **diskless computers** when booting, to get their IP from a RARP server.
- **Note:** Rarely used today (replaced by **BOOTP, DHCP**).

3. ICMP (Internet Control Message Protocol)

•**Purpose:** Used for **error reporting and diagnostics**.

•**Functions:**

- Reports unreachable host/network.
- Helps in congestion control.
- Supports tools like **ping** (echo request/reply) and **traceroute**.

•**Example:** If a router can't deliver a packet, it sends an ICMP "Destination Unreachable" message.

4. IGMP (Internet Group Management Protocol)

•**Purpose:** Manages **multicast group membership**.

•**Works:**

- Allows hosts to join/leave multicast groups.
- Routers use IGMP to learn which groups have members in a subnet.

•**Use:** Streaming video, online conferences, IPTV, etc.