University of Technology
الجامعة التكنولوجية

Computer Science Department
قسم علوم الحاسوب

Network Switching and Routing 2
تبديل وتوجيه الشبكات 2

Prof. Dr. Rana Fareed Ghani
أ.د. رنا فريد غني

cs.uotechnology.edu.iq

Lecture 1
Routing Fundamentals

1. Introduction to Routing

1.1. What is Routing?

- **Routing** is the process of determining the best path for data packets to travel from the **source network** to the **destination network** across one or more intermediary networks. In other words, routing is responsible for forwarding data from one point in a network to another, ensuring that packets reach their correct destination.
- Every packet sent over the internet or a local network carries information about its **source** and **destination** addresses, and routers use this information to make forwarding decisions.

1.2. Some Definitions related to Routing

- **Network Communication**:
  - Data communication on a network involves sending small chunks of data called **packets**. Each packet includes both the source and destination IP addresses.
  - **Routing** enables packets to be directed from one network to another, ensuring data can travel across different routers, switches, and links.
- **The Role of Routers**:
  - A **router** is a network device responsible for forwarding packets from one network to another based on the destination address.
  - Routers use routing **tables** to decide where to send incoming packets. The routing table is a data structure that holds information about various network destinations and the next hop (next router or gateway) to reach those destinations.
- **Types of Routing**:
  - **Static Routing**: Involves manually configuring the paths that packets will take across the network. It is simple but not very flexible.
  - **Dynamic Routing**: Uses protocols like RIP, OSPF, and BGP to automatically learn and adjust routes based on network conditions (e.g., new routes, link failures).
- **Path Selection**:
  - **Routing protocols** determine the best path for data packets by considering factors like **distance**, **hop count**, **delay**, and **bandwidth**.
  - **Routing Metrics**: Routing protocols use metrics to evaluate the quality of a path. For instance, RIP uses hop count, while OSPF uses bandwidth and link reliability.
- **Routing Tables**:
  - Each router maintains a **routing table** that lists the best known routes to various destinations. It helps the router decide where to forward packets when they arrive.

- **Default Gateway**: If a packet's destination is unknown to the router, it is forwarded to the **default gateway**—the router that is responsible for reaching unknown destinations.

1.3. Importance of Routers in Network Communication

Routers are **critical devices** in any network infrastructure because they determine how data flows across different networks. They play a central role in directing network traffic efficiently and ensuring that packets are delivered to their correct destinations.

1.3.1. How Routers Make Forwarding Decisions

- **Forwarding Packets Based on Destination IP Address**:

  Routers are designed to **examine the destination IP address** of incoming packets. Once they receive a packet, they **compare** the destination IP address with their routing table to make forwarding decisions.
  If the destination address matches an entry in the routing table, the router will forward the packet to the next hop on the path toward the destination network.
- **Next Hop**:
  The next hop is usually either another router or the final destination. Routers use the **longest prefix match** method to select the best route, meaning they look for the most specific match between the destination address and the routing table entries.

- **Example**
  In IP routing, routers determine the optimal path for data packets by employing the **Longest Prefix Match (LPM)** algorithm. This method involves comparing the destination IP address of an incoming packet against entries in the router's forwarding table and selecting the entry with the most specific matching prefix. The "longest" prefix refers to the one with the greatest number of matching initial bits.
  Consider a router with the following forwarding table:

Destination Network	**Prefix Length**	**Next Hop**
192.168.0.0	/16	Router A
192.168.1.0	/24	Router B
192.168.1.128	/25	Router C

When a packet with the destination IP address **192.168.1.130** arrives, the router processes it as follows:

Match with 192.168.0.0/16: the first 16 bits match, covering addresses from 192.168.0.0 to 192.168.255.255.2. **Match with 192.168.1.0/24:** the first 24 bits match, covering addresses from 192.168.1.0 to 192.168.1.255.3. **Match with 192.168.1.128/25:** the first 25 bits match, covering addresses from 192.168.1.128 to 192.168.1.255. Among these, **192.168.1.128/25** has the longest prefix length (25 bits), making it the most specific match. Therefore, the router forwards the packet to **Router C**, as specified in the forwarding table. This process ensures that data packets are routed through the most precise path available, optimizing network efficiency and resource utilization.

1.3.2. **Network Layer Operations**:

Routers operate on the **Network Layer (Layer 3)** of the OSI model. They handle the **logical addressing** (IP addresses), unlike switches, which operate at the **Data Link Layer (Layer 2)** and deal with physical MAC addresses.
Each router in a network is responsible for forwarding packets between different networks (subnets), making decisions based on the **destination IP address** in the packet header. The following are the main responsibilities of routers in computer networks:

1. **Interconnecting Different Networks**:
   Routers connect and route traffic between different networks or subnets. In other words, they act as **gateways** between separate networks, enabling communication between devices that are on different networks.
   For example, a router may connect a local area network (LAN) to a wide area network (WAN) or connect multiple LANs within an organization.

2. **Traffic Optimization**:
   Routers also help in **optimizing network traffic** by determining the best path for data packets to travel. They use various routing algorithms and protocols (like RIP, OSPF, or BGP) to select the most efficient route, which can reduce latency, avoid network congestion, and ensure faster delivery.

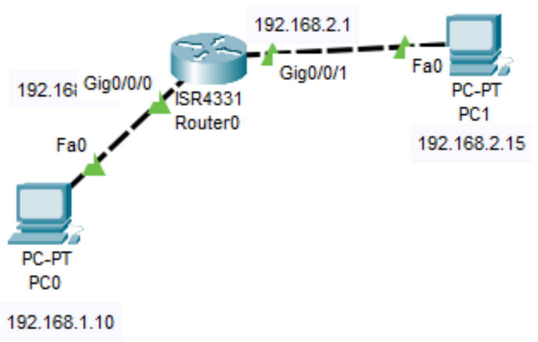
3. **Load Balancing**:
   Some routers also perform load balancing, distributing traffic across multiple links to avoid overloading a single path.

4. **Handling of Broadcasts and Multicasts**:
   Routers prevent **broadcast storms** (unnecessary traffic sent to all devices in a network) by not forwarding broadcast packets beyond their local network. This helps to maintain efficient network performance.
   Routers also handle **multicast** traffic, forwarding packets only to devices that have expressed interest in receiving them, which optimizes bandwidth usage.

5. **Security**:
   Routers play an important role in **network security**. They can act as a barrier between an internal network and external networks, implementing policies such as **firewall rules**, **access control lists (ACLs)**, and **NAT (Network Address Translation)** to filter traffic and protect against unauthorized access.

Example:

Imagine a network where **Router A** connects two subnets:
Subnet 1: 192.168.1.0/24
Subnet 2: 192.168.2.0/24
When a device in **Subnet 1** (e.g., IP address 192.168.1.10) wants to communicate with a device in **Subnet 2** (e.g., IP address 192.168.2.15), the packet from the source device will be sent to **Router A**, since the destination IP address doesn't match any local addresses in **Subnet 1**.
Router A will look at its routing table, find that the **192.168.2.0/24** network is reachable via its **interface connected to Subnet 2**, and forward the packet to the appropriate destination.
In this example, **Router A** is making a forwarding decision based on the destination IP address, ensuring the packet reaches the correct network and device.



The Packet Flow is as follows:
Device 1 (192.168.1.10) → Router A (Interface 1) → Router A (Interface 2) → Device 2 (192.168.2.15).

1.4. IP Addressing and Subnetting

- **IP Addressing**:

  IPv4 addressing is essential for network communication, and understanding the distinction between the **network portion** and the **host portion** is fundamental to configuring and managing IP addresses effectively.
- **IPv4 Address Structure**

  An **IPv4 address** is a 32-bit address, written in **dotted decimal format**, which consists of **four octets** (8 bits each) separated by periods.
  Example:
  192.168.1.10
  In binary, this would be:
  11000000.10101000.00000001.00001010
  Each octet represents a number between **0 and 255** (since 8 bits can represent values from 0 to 255).
- **Network Portion vs. Host Portion**

Network Portion:
The **network portion** of an IP address identifies the **network** to which the device (host) belongs. It is used by routers to determine where to send packets across networks.
The network portion is determined by the **subnet mask**.

- **Host Portion**:

The **host portion** identifies a specific **device (host)** within the network. Each host on a network must have a unique host portion within that subnet.
The host portion is what distinguishes one device from another within the same network.

- **Subnet Mask**

The **subnet mask** is a 32-bit number that helps to differentiate the **network portion** from the **host portion**. It works by using **1s** to represent the network portion and **0s** to represent the host portion.
Example:
IP Address: 192.168.1.10
Subnet Mask: 255.255.255.0
In binary, the subnet mask looks like:
11111111.11111111.11111111.00000000
This means that the **first 24 bits** (the 1s) represent the **network portion**, and the **last 8 bits** (the 0s) represent the **host portion**.

- **Breaking Down an IP Address**

Let's take the IP address 192.168.1.10 and a subnet mask of 255.255.255.0:
IP Address: 192.168.1.10
Network Portion: 192.168.1 (the first three octets)
Host Portion: .10 (the last octet)
Subnet Mask: 255.255.255.0
Network Bits: 255.255.255 → These are the **network bits**, covering the first three octets.
Host Bits: 0 → The last octet is for **host bits**.
Thus, **192.168.1.10** belongs to the **network 192.168.1.0** and represents a specific device (host) with the address 10.

- **Classful Addressing (Historical Context)**

In the past, IPv4 addresses were divided into classes, based on the size of the network portion. Here's a quick recap:

- **Class A**: 0.0.0.0 to 127.255.255.255

Default Subnet Mask: 255.0.0.0
Network Portion: 8 bits, Host Portion: 24 bits

- **Class B**: 128.0.0.0 to 191.255.255.255

Default Subnet Mask: 255.255.0.0
Network Portion: 16 bits, Host Portion: 16 bits

- **Class C**: 192.0.0.0 to 223.255.255.255

Default Subnet Mask: 255.255.255.0
Network Portion: 24 bits, Host Portion: 8 bits

- However, today, most networks use **CIDR (Classless Inter-Domain Routing)** notation, which allows more flexible subnetting.

  - **CIDR (Classless Inter-Domain Routing) Notation**

  CIDR notation represents the network and host portions more flexibly. Instead of using a default subnet mask, CIDR allows specifying the number of bits used for the network portion directly.
  Example: 192.168.1.10/24
  The /24 means that the first 24 bits of the address are for the **network portion**, leaving the last 8 bits for the **host portion**.

1.5. Subnets and Subnet masks

Subnets and subnet masks are fundamental concepts in networking that allow networks to be divided into smaller, more manageable segments. This improves efficiency, security, and performance within a network. Let's explore these concepts step by step.

1.5.1. What Is a Subnet?

A **subnet** (short for sub-network) is a smaller portion of a larger network. Subnetting is the process of dividing a large network into multiple smaller, logical networks called **subnets**.

1.5.2. Why Subnet?

1. **Efficient IP Address Usage**: Prevents waste of IP addresses by allocating them more precisely to subnets based on need.
2. **Improved Network Performance**: Reduces congestion by limiting broadcast traffic within each subnet.
3. **Enhanced Security**: Isolates sensitive parts of a network from others.
4. **Simplifies Management**: Breaks down a large, complex network into smaller, manageable pieces.

1.5.3. What Is a Subnet Mask?

A **subnet mask** is a 32-bit number that defines how an IP address is divided into the **network** and **host** portions. It helps routers and devices identify which part of the IP address refers to the **network** and which part identifies the **host**.
Subnet Mask Representation
A subnet mask is written in dotted decimal notation, like an IP address.
Example: 255.255.255.0
In binary:
11111111.11111111.11111111.00000000
The **1s** represent the network portion, and the **0s** represent the host portion.

1.5.4. How Subnet Masks Work

- **Network Portion**:

The subnet mask tells devices which bits of the IP address belong to the **network portion**.
For example, with a subnet mask of 255.255.255.0, the **first 24 bits** (3 octets) of an IP address belong to the network.

- **Host Portion**:

  The remaining bits (after the network portion) are used for the **host portion**, which identifies devices within the network.
  Example:
  IP Address: 192.168.1.10
  Subnet Mask: 255.255.255.0
  In binary:
  IP Address: 11000000.10101000.00000001.00001010
  Subnet Mask: 11111111.11111111.11111111.00000000
  The first 24 bits (192.168.1) are the **network portion**, and the last 8 bits (.10) are the **host portion**.

1.5.5. Subnetting Example

Let's break down a real-world subnetting scenario.
Scenario: You have a network 192.168.1.0/24 and want to divide it into 4 subnets.

- **Original Subnet Mask**:

  /24 → 255.255.255.0
  Network bits: 24, Host bits: 8
- **Dividing into 4 Subnets**:

  You need to "borrow" 2 bits from the **host portion** to create 4 subnets, as 2 bits allow 4 combinations (2^2 = 4).
- **New Subnet Mask**:

  /26 → 255.255.255.192
  Network bits: 26, Host bits: 6
- **Resulting Subnets**: Each subnet has:

  64 addresses (2^6 = 64), with 62 usable IPs (excluding the network and broadcast addresses).
- **Subnet Breakdown:**
  - ✓ **Subnet 1**: 192.168.1.0/26 → Usable IPs: 192.168.1.1 to 192.168.1.62
  - ✓ **Subnet 2**: 192.168.1.64/26 → Usable IPs: 192.168.1.65 to 192.168.1.126
  - ✓ **Subnet 3**: 192.168.1.128/26 → Usable IPs: 192.168.1.129 to 192.168.1.190
  - ✓ **Subnet 4**: 192.168.1.192/26 → Usable IPs: 192.168.1.193 to 192.168.1.254

Lecture 2
Routing Tables and Routing Protocols

2.1.The Role of Routing Tables

1.5.6. What is a Routing Table?

Definition: A routing table is a set of rules, often in the form of a table, that is used to determine where data packets traveling over an IP network should be forwarded.

1.5.7. Basic Structure of a Routing Table:

Network Destination: The destination IP address or network.
Netmask: Identifies the network portion of the destination IP.
Gateway: Next-hop router address (used when the destination is not in the local network).
Interface: The local interface through which the packet is sent.
Metric: Used for selecting the best route (lower is better).

1.5.8. Example Routing Table (Windows example using route print):

```
Network Destination     Netmask          Gateway          Interface
    Metric
0.0.0.0                 0.0.0.0           192.168.1.1     192.168.1.10       1
192.168.1.0             255.255.255.0       0.0.0.0       192.168.1.10       1
```

1.5.9. Columns in a Routing Table

1. **Network Destination**:

   This column lists the **network addresses** for which the routing table has specific information.
   Each row represents a route to a particular network or range of IP addresses.
   Example:
   192.168.1.0 → The network 192.168.1.0/24.
   0.0.0.0 → The **default route** (used when no specific route matches).
2. **Netmask**:

The **subnet mask** defines how many bits of the IP address belong to the network portion.
It determines the size of the network or range of IPs covered by the route.
Example:

255.255.255.0 → A /24 subnet containing 256 addresses (192.168.1.0 to 192.168.1.255).

3. **Gateway**:

   The **next hop** IP address, which indicates where packets should be forwarded.
   If the destination is not directly reachable, the router sends the packet to this gateway for further forwarding.
   Example:
   192.168.1.1 → The gateway for packets destined for the listed network.

4. **Interface**:

   The **local network interface** (e.g., Ethernet, Wi-Fi) through which packets should be sent.
   Example:
   192.168.1.100 → The local IP of the network interface on the router.
5. **Metric**:

   The **cost** associated with using this route, often based on hop count, bandwidth, or delay.
   Lower metric values indicate preferred routes.
   Example:
   Metric 1 → Directly connected network (preferred).
   Metric 10 → Less preferred route, perhaps involving multiple hops.

1.5.10. Rows in a Routing Table

Each row in the table represents a specific route to a destination. Let's examine common rows and their meaning:

1. **Default Route (Gateway of Last Resort)**:

   2. Destination: 0.0.0.0
   3. Netmask: 0.0.0.0
   4. Gateway: The IP of the default gateway (e.g., 192.168.1.1).
   5. Interface: The local interface (e.g., 192.168.1.100).
   6. Metric: A relatively high value, as this is used only when no other specific routes match.
   7. Purpose: Forwards packets to an external network when no other routes apply.

2. **Directly Connected Networks**:

   8. These rows define networks directly attached to the router.
   9. Example:

   Destination: 192.168.1.0
   Netmask: 255.255.255.0
   Gateway: On-link (indicating the network is directly accessible without needing a gateway).
   Interface: The IP of the router's connected interface.

Metric: Typically 1 (most preferred route).

3. **Host-Specific Routes**:

- Specific entries for individual IP addresses.
- Example:

Destination: 192.168.1.10
Netmask: 255.255.255.255 (indicates a single host).
Gateway: The gateway to reach that specific host.
Purpose: Used when routing to a specific device.

4. **Loopback Route**:

- **Destination**: 127.0.0.0
- **Netmask**: 255.0.0.0
- **Gateway**: On-link
- **Interface**: 127.0.0.1 (loopback interface).
- **Purpose**: Handles traffic destined for the local device itself.

5. **Multicast and Broadcast Routes**:

- Routes for multicast or broadcast traffic.
- Example:

Destination: 224.0.0.0
Netmask: 240.0.0.0 (multicast range).
Purpose: Supports multicast applications like video conferencing or streaming.

6. **Interpreting a Sample Routing Table**

Network Destination	**Netmask**	**Gateway**	**Interface**	**Metric**
0.0.0.0	0.0.0.0	192.168.1.1	192.168.1.100	10
192.168.1.0	255.255.255.0	On-link	192.168.1.100	1
192.168.1.10	255.255.255.255	192.168.1.1	192.168.1.100	5
127.0.0.0	255.0.0.0	On-link	127.0.0.1	1

1. **Row 1 (Default Route)**:
   - All traffic for unknown networks is sent to 192.168.1.1 via the interface 192.168.1.100. This is the default gateway.
2. **Row 2 (Directly Connected Network)**:
   - Traffic for the 192.168.1.0/24 network is sent directly via the interface 192.168.1.100.
3. **Row 3 (Host-Specific Route)**:
   - Packets destined for 192.168.1.10 are forwarded to 192.168.1.1. This could represent a manually added route for a specific host.

4. **Row 4 (Loopback Route)**:
    - Packets for 127.0.0.0/8 are directed to the loopback interface (127.0.0.1) for local communication.

2.2. Introduction to Routing Protocols

- **Routing Protocols**:
    - A **routing protocol** is a set of rules used by routers to determine the best path for data to travel across a network. These protocols allow routers to communicate and share information about network topology.
    - Routing protocols are essential for the efficient operation of large networks, ensuring that data packets are delivered quickly and reliably.
- **Types of Routing Protocols**: Routing protocols can be categorized based on different factors. Today, we'll discuss:

a. **Interior vs. Exterior Gateway Protocols**

b. **Distance-Vector, Link-State, and Path-Vector Protocols**

2.2.1. Interior vs. Exterior Gateway Protocols

- **Interior Gateway Protocols (IGP)**:
    - **Definition**: IGPs are used within an organization or a single autonomous system (AS).
    - **Purpose**: These protocols help routers inside an AS share routing information and select the best paths for data within the internal network.
    - **Common IGPs**:
        - **RIP (Routing Information Protocol)**: A distance-vector protocol that uses hop count as the metric to determine the best path.
        - **OSPF (Open Shortest Path First)**: A link-state protocol that uses a cost metric and provides more accurate routing than RIP.
        - **EIGRP (Enhanced Interior Gateway Routing Protocol)**: A hybrid protocol that combines aspects of both distance-vector and link-state protocols.
- **Exterior Gateway Protocols (EGP)**:
    - **Definition**: EGPs are used to route data between different autonomous systems (ASes), often across the internet.
    - **Purpose**: These protocols help in the exchange of routing information between different networks or ASes, which are managed by different organizations.
    - **Common EGPs**:

        - **BGP (Border Gateway Protocol)**: The primary EGP used on the internet. BGP uses path-vector routing and is highly scalable, making it suitable for large-scale inter-AS routing.

2.2.2 Classification of Routing Protocols

Routing protocols can be further classified into three main types based on how they determine the best path: **Distance-vector**, **Link-state**, and **Path-vector**.

2.2.2.1. Distance-Vector Routing Protocols

- **Definition**: Distance-vector protocols determine the best path based on the number of hops (distance) to the destination.
- **Working Principle**:
    - Each router maintains a table (distance vector) that lists the distance (or metric) to every other router in the network.
    - Routers periodically exchange this information with their neighbors. They update their routing tables based on the information they receive.
- **Example: RIP (Routing Information Protocol)**:
    - RIP uses the number of hops as a metric. The maximum hop count is 15, meaning any destination more than 15 hops away is considered unreachable.
    - **Advantages**: Simple to configure and implement.
    - **Disadvantages**: Slow convergence (can take a long time to adapt to network changes) and scalability issues in larger networks.

2.2.2.2. Link-State Routing Protocols

- **Definition**: Link-state protocols determine the best path by examining the complete network topology and selecting the shortest path.
- **Working Principle**:
    - Routers send updates (link-state advertisements, or LSAs) about the state of their links (interface status, cost, etc.) to all other routers in the network.
    - Each router constructs a complete map of the network using the LSAs and computes the shortest path to each destination using algorithms like Dijkstra's algorithm.
- **Example: OSPF (Open Shortest Path First)**:
    - OSPF is the most widely used link-state protocol. It divides the network into areas and uses cost as the metric for path selection.
    - **Advantages**: Faster convergence compared to distance-vector protocols and scalability for large networks.

    - **Disadvantages**: More complex to configure and maintain due to the need for a complete network topology.

2.2.2.3. Path-Vector Routing Protocols

- **Definition**: Path-vector protocols are similar to distance-vector protocols, but they include the complete path to a destination, not just the distance.
- **Working Principle**:
    - Routers exchange path information, which includes the list of routers (path) that a packet will traverse. This allows the router to make decisions based on the entire path, not just the distance.
- **Example: BGP (Border Gateway Protocol)**:
    - BGP is the most popular path-vector protocol used in inter-AS routing, particularly on the internet. BGP routers share path information along with the AS path, which allows them to avoid routing loops and select optimal paths.
    - **Advantages**: Extremely scalable and adaptable to large networks.
    - **Disadvantages**: Complex to configure and manage.

Summary of Routing Protocols

Let's quickly summarize the key points:

- **Interior Gateway Protocols (IGP)** are used within a single autonomous system, while **Exterior Gateway Protocols (EGP)** are used to connect different autonomous systems.
- **Distance-Vector Protocols** (e.g., RIP) are simple but slower to converge and less scalable.
- **Link-State Protocols** (e.g., OSPF) provide faster convergence and scalability by maintaining a complete network topology.
- **Path-Vector Protocols** (e.g., BGP) are used for large-scale inter-AS routing and provide more control over routing decisions but are more complex.

Lecture 3: Distance-Vector Routing Protocols (RIP)

Learning Objectives:

By the end of this lecture, students should be able to:

1. Understand the fundamentals of Distance-Vector Routing Protocols.
2. Explain how the Routing Information Protocol (RIP) works.
3. Differentiate between RIP v1 and RIP v2.
4. Identify scenarios where RIP is appropriate for use in networks.
5. Comprehend the Bellman-Ford algorithm and its role in RIP.

3.1. Introduction to Distance-Vector Routing Protocols

3.1.1. What is Distance-Vector Routing?

- Distance-Vector Routing is one of the two primary types of routing protocols (the other being Link-State Routing).
- It determines the best path based on the number of hops.
- Each router shares its routing table with its neighbors at regular intervals.

3.1.2. Characteristics of Distance-Vector Routing

- Uses Bellman-Ford algorithm for path calculation.
- Relies on periodic updates.
- Does not have a complete map of the network—only knows routes via neighbors.
- Prone to routing loops but uses mechanisms like split horizon and route poisoning to prevent them.

3.1.3. Bellman-Ford Algorithm in RIP

- Used to compute the shortest path in distance-vector protocols.
- Each router maintains a table containing the distance to every destination.
- Algorithm iteratively updates the table by exchanging information with neighbors.
- Steps:
  1. Each router initializes its routing table with known distances.
  2. Routers exchange tables with neighbors at regular intervals.
  3. If a router receives a better path (lower hop count), it updates its table.

4. Process continues until the network stabilizes (convergence).

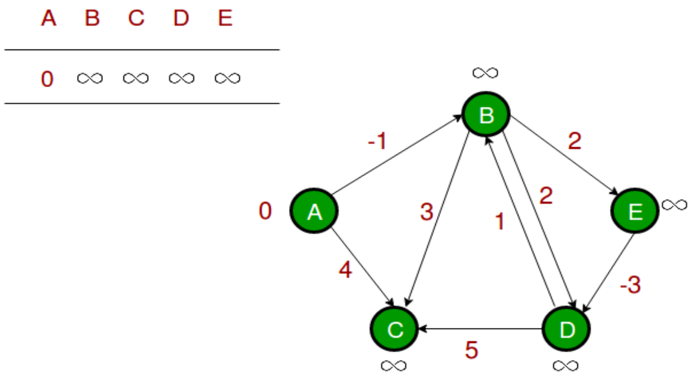
The **Bellman-Ford Algorithm** is used to find the shortest paths from a single source vertex to all other vertices in a graph. If a **negative weight cycle** exists, the algorithm detects it instead of computing shortest paths.

Steps of the Algorithm

1. **Initialization:**
   - Set the distance of the source vertex to 0 and all other vertices to ∞ (infinity).
2. **Relaxation (|V| - 1 times):**
   - Repeat **|V| - 1** times (where |V| is the number of vertices):
     - For each edge **(u, v)** with weight **w**, update the distance if a shorter path is found:
       - dist[v] > dist[u] + w → update dist[v] = dist[u] + w
   - This ensures that the shortest path with at most **|V| - 1 edges** is found.
3. **Negative Cycle Detection:**
   - If after the above steps, any edge **(u, v)** can still be relaxed, then the graph contains a **negative weight cycle**.

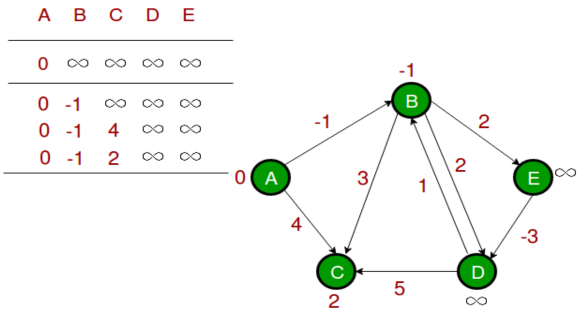
Example Execution

- Suppose we have **5 vertices** and a directed graph with weighted edges.



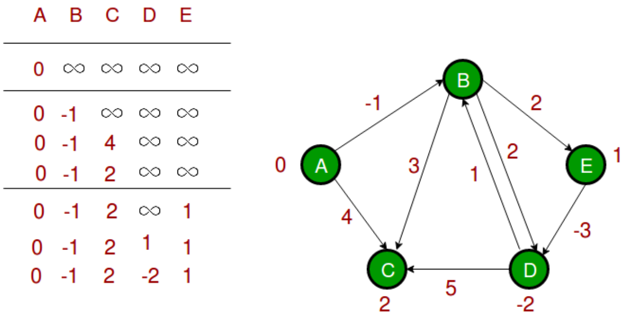
A	B	C	D	E
0	∞	∞	∞	∞

- The algorithm **updates distances in multiple iterations**.
- After the first pass, all shortest paths with **at most one edge** are found.



A	B	C	D	E
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	2	∞	∞

- The second pass finds paths with **at most two edges**, and so on.



A	B	C	D	E
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	2	∞	∞
0	-1	2	∞	1
0	-1	2	1	1
0	-1	2	-2	1

- If no further updates occur after **|V| - 1 iterations**, the algorithm terminates.

3.2. Routing Information Protocol (RIP) Overview

3.2.1. What is RIP?

- RIP is a Distance-Vector Routing Protocol.
- One of the oldest routing protocols, standardized in RFC 1058.
- Uses **hop count** as the metric for selecting the best route.
- Maximum hop count is **15** (16 is considered unreachable).

3.2.2. Features of RIP

- Periodic updates (every 30 seconds by default).
- Entire routing table is sent to neighboring routers.
- Uses UDP port 520 for communication.
- Implements techniques like **split horizon** and **route poisoning** to prevent routing loops.
- Simple to configure but not scalable for large networks.

3.3. Working of RIP

3.3.1. Hop Count Metric

- Each router counts the number of routers (hops) between itself and the destination.

- The shortest path is determined based on the lowest hop count.

3.3.2. Periodic Updates

- Every 30 seconds, routers exchange their routing tables.
- Ensure all routers have updated information about network topology.

3.3.3. Split Horizon

- Prevents routing loops by ensuring that a router does not advertise a route back to the router from which it learned that route.

3.4. Route Poisoning and Hold-Down Timers

- **Route Poisoning:** When a route becomes unavailable, it is advertised with a hop count of 16 (unreachable).
- **Hold-Down Timer:** Prevents immediate acceptance of potentially bad updates by waiting for a specific period before accepting a new route.

3.5. RIP Convergence Time

- RIP has slow convergence time due to periodic updates and hop count limitations.
- Can be improved by reducing update intervals or using triggered updates.

3.6. Differences Between RIP v1 and RIP v2 (20 minutes)

Feature	**RIP v1**	**RIP v2**
Routing Type	Distance-Vector	Distance-Vector
Classful/Classless	Classful (does not support CIDR)	Classless (supports CIDR)
Authentication	No authentication	Supports authentication
Subnet Mask Transmission	Does not send subnet mask	Includes subnet mask information
Multicast/Broadcast Updates	Broadcasts updates to 255.255.255.255	Uses Multicast (224.0.0.9)

3.6.1. RIP v1 Limitations

- Does not support Variable Length Subnet Masking (VLSM).
- Causes excessive traffic due to broadcast updates.
- Security risks due to lack of authentication.

3.6.2. RIP v2 Enhancements

- Supports subnet masks, allowing VLSM.
- Uses multicast for updates, reducing unnecessary network traffic.

- Implements authentication for secure routing updates.

3.7. When to Use RIP?

3.7.1. Appropriate Use Cases

- Small networks with simple routing requirements.
- Environments with low administrative overhead.
- Networks that do not require fast convergence.

3.7.2. When NOT to Use RIP

- Large-scale networks where scalability is a concern.
- Networks requiring rapid convergence (e.g., real-time applications).
- Environments where security is a high priority.

3.8. Summary

- RIP is a Distance-Vector Protocol using hop count as a metric.
- RIP v1 is classful; RIP v2 is classless with additional features.
- RIP is suitable for small networks but not for large-scale implementations.
- Techniques like split horizon and route poisoning help prevent routing loops.
- The Bellman-Ford algorithm is fundamental to RIP's route calculations.