1

(a) Circuit Switching vs. Packet Switching

Circuit Switching:

- **Pre-Allocated Capacity:** Circuit switching involves establishing a dedicated communication path between two endpoints for the duration of the session. This path reserves the full bandwidth of the connection, even if there is no data being transmitted at certain times.
- **Unused Capacity:** Once a circuit is established, the allocated bandwidth is reserved and cannot be used by other connections, leading to inefficiencies when the transmission is not continuous or fully utilizing the bandwidth.

Packet Switching:

- **Dynamic Capacity Allocation:** Packet switching divides data into packets, each of which can travel independently through the network. The network resources are not reserved for any single connection, allowing for more efficient use of the available bandwidth.
- **Improved Utilization:** Since packets from multiple connections can share the same network paths, packet switching allows for better utilization of the network capacity, minimizing the wasted bandwidth that is common in circuit switching.

Effectiveness of Packet Switching:

- **Efficiency:** Packet switching is more efficient as it allows the network to dynamically allocate resources based on demand, reducing idle times and improving overall utilization.
- **Scalability:** It is more scalable, handling a larger number of connections and varying data rates more effectively.
- **Flexibility:** Packet switching can easily adapt to changes in network topology and traffic patterns, providing robust performance even in complex and dynamic networks.

(b) Multi-Homing for Access ISPs

Concept: Multi-homing refers to the practice of connecting a network to multiple upstream providers (transit ISPs) to ensure redundancy, reliability, and potentially better performance.

Benefits:

- **Reliability:** If one connection fails, the access ISP can still maintain connectivity through another transit ISP, improving uptime and reliability for end-users.
- **Load Balancing:** Traffic can be distributed across multiple paths, reducing congestion and improving performance.
- **Redundancy:** Provides a backup in case of failure of a single upstream provider, ensuring continuous service availability.
- **Performance:** Potentially better routing paths and lower latency as traffic can take the best available route provided by multiple transit ISPs.

Conclusion: Introducing multi-homing for access ISPs is beneficial for users as it enhances reliability, performance, and overall network resilience.

(d) Socket API and Client-Server Communication

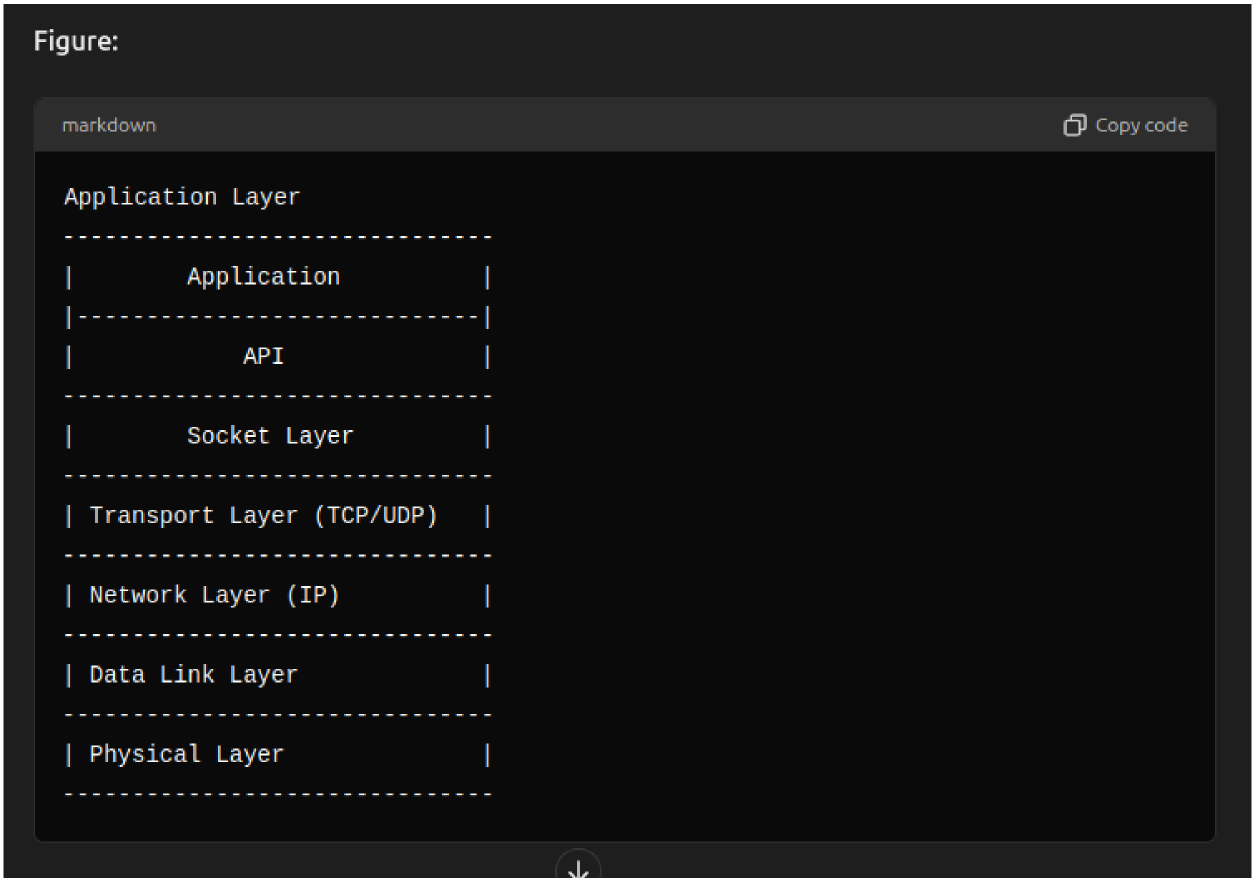


Figure:

```
Application Layer
-------------------------------
|        Application          |
|-----------------------------|
|            API              |
-------------------------------
|        Socket Layer         |
-------------------------------
| Transport Layer (TCP/UDP)   |
-------------------------------
| Network Layer (IP)          |
-------------------------------
| Data Link Layer             |
-------------------------------
| Physical Layer              |
-------------------------------
```

Parameters:

Application-Layer Side:

- **IP Address:** Specifies the destination address for communication.
- **Port Number:** Identifies the specific process or application on the host.

Transport-Layer Side:

- **Protocol:** Defines whether TCP or UDP is used.
- **Socket Descriptor:** A unique identifier for each socket.
- **Buffer Sizes:** Specifies the sizes of send and receive buffers.
- **Timeouts:** Configurations for connection timeouts and retransmission.

Client-Server Communication:

1. **Client Side:**

    - **Socket Creation:** `socket()`
    - **Connection:** `connect()`
    - **Data Transfer:** `send()`, `recv()`
    - **Disconnection:** `close()`
2. **Server Side:**
    - **Socket Creation:** `socket()`
    - **Binding:** `bind()`
    - **Listening:** `listen()`
    - **Accepting Connections:** `accept()`
    - **Data Transfer:** `send()`, `recv()`
    - **Disconnection:** `close()`

Principle: The client initiates a connection to the server using the socket API. The server listens for incoming connections and accepts them when they arrive. Once connected, data is exchanged between the client and server using the send and receive functions. Finally, the connection is terminated by closing the socket.

2

(a) Single TCP Connection in HTTP/1.0 and Solutions in HTTP/1.1 and HTTP/2.0

(i) Problem Encountered by Small Objects in HTTP/1.0:

- **Head-of-Line Blocking:** When a large video clip is transferred first over a single TCP connection, it causes delays for subsequent small objects. These smaller objects have to wait until the large video transfer completes, leading to inefficient utilization of the connection and increased page load times.

(ii) Addressing the Problem in HTTP/1.1:

- **Persistent Connections:** HTTP/1.1 introduced persistent connections, allowing multiple requests and responses to be sent over a single TCP connection without reopening new connections for each request.
- **Pipelining:** HTTP/1.1 also supports pipelining, where a client can send multiple requests without waiting for each response. However, responses must be sent back in order, which can still lead to head-of-line blocking.
- **Parallel Connections:** While HTTP/1.1 does not inherently open multiple parallel connections for a single request, it does allow browsers to open multiple connections (usually 6-8) to a server to download resources concurrently. This is a workaround rather than a protocol feature.

(iii) Frame-Interleaving in HTTP/2.0:

- **Multiplexing:** HTTP/2.0 uses multiplexing, allowing multiple requests and responses to be sent over a single TCP connection simultaneously. Frames from different streams (requests) are interleaved, eliminating head-of-line blocking.
- **Stream Prioritization:** HTTP/2.0 supports prioritization of streams, meaning important resources (like small objects) can be given priority over less critical ones (like large video clips).
- **Efficiency:** These features significantly improve the efficiency and speed of web page loading by making better use of available bandwidth and reducing latency.

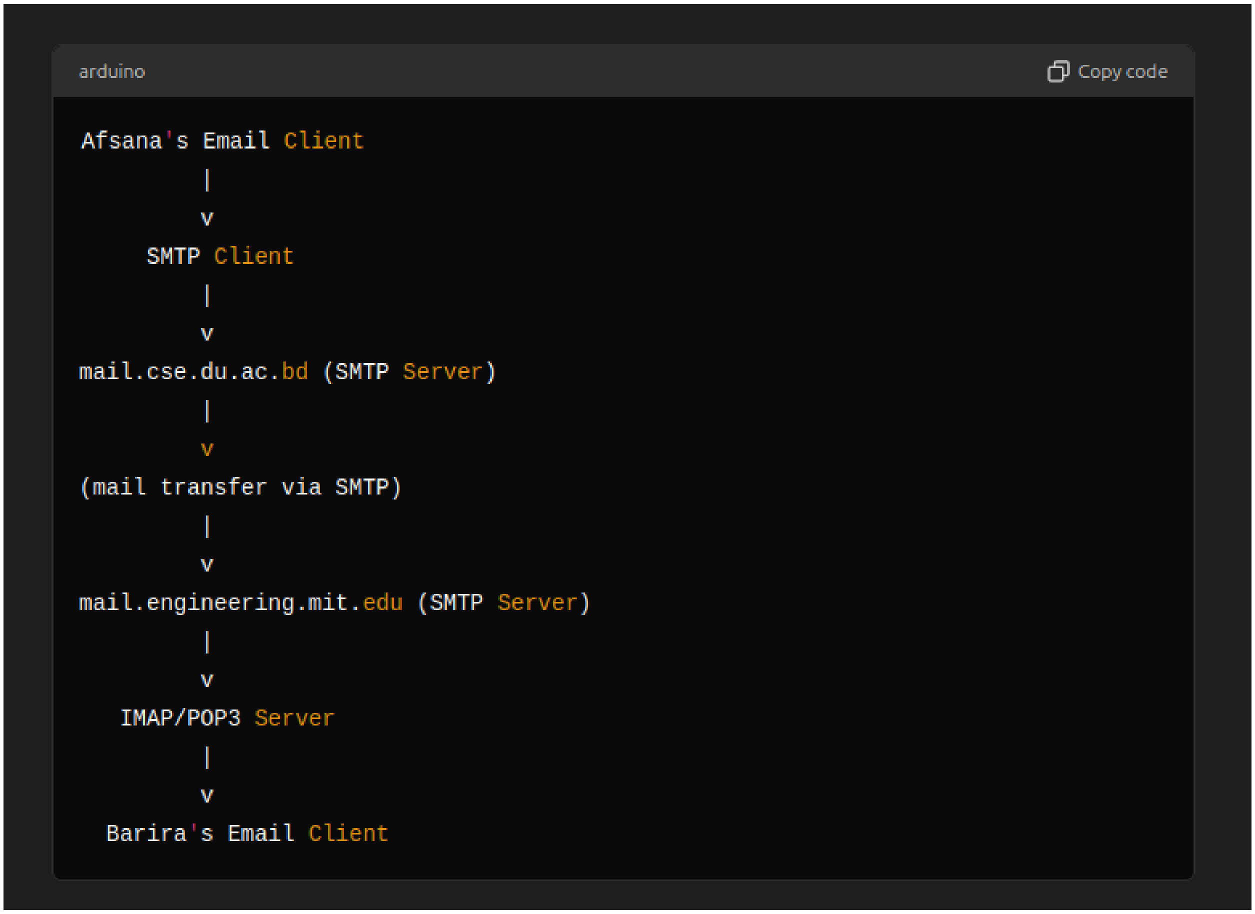
(b) Email Communication Between Afsana and Barira

Protocols Involved:

1. **SMTP (Simple Mail Transfer Protocol):** Used to send and transfer emails from Afsana's mail server to Barira's mail server.
2. **IMAP (Internet Message Access Protocol) or POP3 (Post Office Protocol):** Used by Barira to retrieve the email from her mail server.

Steps and Principles:

1. **Afsana Composing Email:**
    - **Mail Client:** Afsana uses her email client to compose a message.
    - **SMTP Client:** Her email client acts as an SMTP client to send the message to the mail server `mail.cse.du.ac.bd`.
2. **Sending Email:**
    - **SMTP Handshake:** `mail.cse.du.ac.bd` establishes an SMTP connection to `mail.engineering.mit.edu`.
    - **Message Transfer:** The email is transferred from `mail.cse.du.ac.bd` to `mail.engineering.mit.edu` using the SMTP protocol.
3. **Receiving Email:**
    - **IMAP/POP3 Client:** Barira uses her email client to connect to `mail.engineering.mit.edu`.
    - **Email Retrieval:** Her email client retrieves the email using IMAP or POP3.



```
Afsana's Email Client
         |
         v
     SMTP Client
         |
         v
mail.cse.du.ac.bd (SMTP Server)
         |
         v
(mail transfer via SMTP)
         |
         v
mail.engineering.mit.edu (SMTP Server)
         |
         v
   IMAP/POP3 Server
         |
         v
  Barira's Email Client
```

(c) Is FTP a Stateful Protocol?

Answer: Yes, FTP (File Transfer Protocol) is a stateful protocol.

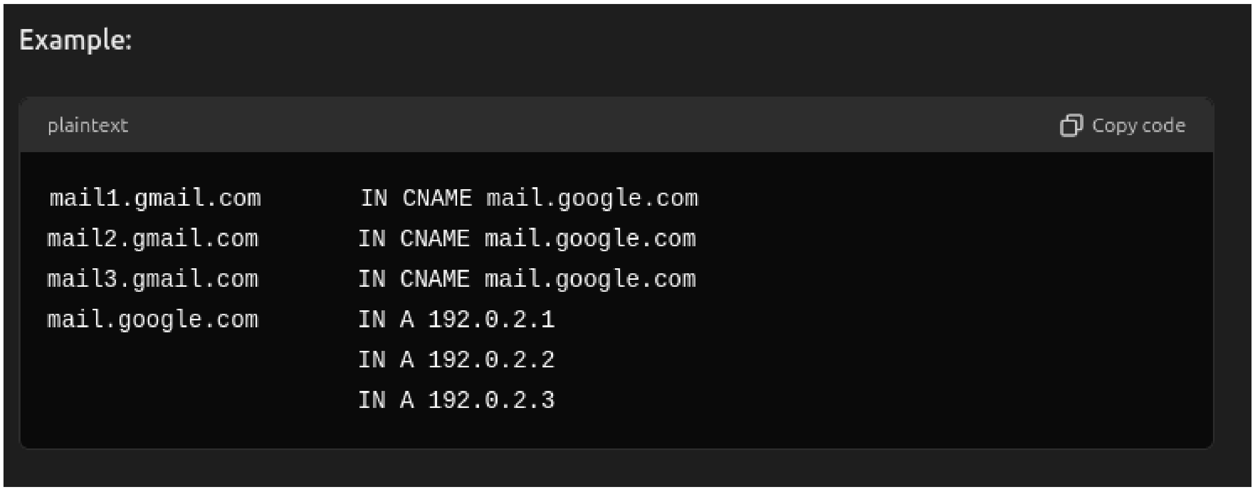
Justification:

- **Control Connection:** FTP uses a persistent control connection to maintain state information about the session, including user authentication, current working directory, and transfer mode.
- **State Management:** The server retains information about the client's state across multiple commands within the same session.
- **Commands Depend on State:** Many FTP commands rely on the state established by previous commands (e.g., `RETR` depends on the current directory set by `CWD`).

(d) How CNAME Records Help Distribute Loads Among Servers

Explanation: CNAME (Canonical Name) records can be used in the DNS system to distribute loads among multiple servers by pointing multiple domain names to a single canonical domain name. Here's how this helps:

1. **Alias Creation:** CNAME records create aliases for the canonical name. For example, `mail1.gmail.com`, `mail2.gmail.com`, and `mail3.gmail.com` can all be aliases for `mail.google.com`.
2. **Load Balancing:** The DNS can use these aliases to distribute requests among different IP addresses associated with the canonical name. This can be achieved through DNS round-robin or other load balancing techniques.
3. **Flexibility and Scalability:** By using CNAME records, traffic can be redirected to different servers without changing the user's requested domain. This allows easy scaling by adding more servers and updating DNS records accordingly.
4. **Redundancy and Failover:** If one server goes down, DNS can redirect traffic to other servers, providing redundancy and ensuring high availability.



Example:

```plaintext
mail1.gmail.com       IN CNAME mail.google.com
mail2.gmail.com       IN CNAME mail.google.com
mail3.gmail.com       IN CNAME mail.google.com
mail.google.com       IN A 192.0.2.1
                      IN A 192.0.2.2
                      IN A 192.0.2.3
```

3

(a) DASH (Dynamic Adaptive Streaming Over HTTP Protocol)

Working Principle:

- **Segmented Content:** DASH divides video content into small segments, each containing a few seconds of video.
- **Multiple Bitrates:** Each segment is encoded at multiple bitrates, allowing for different quality levels.
- **Manifest File:** A manifest file (MPD - Media Presentation Description) lists the available segments and bitrates.
- **Client-Side Adaptation:** The client monitors its network conditions and buffer status. Based on this, it selects and requests the next segment at the most appropriate bitrate.
- **HTTP Requests:** Segments are requested and delivered over standard HTTP, leveraging existing web infrastructure.

Advantages:

1. **Adaptability:** Dynamically adjusts video quality based on current network conditions and client capabilities, providing a smoother viewing experience.
2. **Buffer Management:** Reduces the likelihood of playback interruptions by managing buffer levels more effectively.
3. **Compatibility:** Uses standard HTTP, making it compatible with existing CDNs, caches, and firewalls.
4. **Scalability:** Handles a large number of clients efficiently since it leverages standard web protocols and infrastructure.
5. **Improved User Experience:** Provides higher quality video streams when bandwidth allows and gracefully degrades quality during network congestion.

(b) TCP Segments and Acknowledgements

Scenario:

- Host B receives two TCP segments from Host A.
- First segment sequence number: 1580
- Second segment sequence number: 3760

(i) Amount of Data Sent in the First Segment:

- The sequence number of the second segment is 3760.
- The sequence number of the first segment is 1580.
- Amount of data in the first segment = 3760 - 1580 = 2180 bytes.

(ii) Handling Lost Segments:

- If the first segment (sequence number 1580) is lost but the second segment (sequence number 3760) arrives at Host B:
- Host B will detect a gap in the received data.
- Host B will send an acknowledgement (ACK) for the last correctly received sequence number before the gap.

Acknowledgement Number:

- Since Host B did not receive the segment starting at sequence number 1580, it will send an ACK for sequence number 1580 (indicating it is still expecting this data).

Diagram:

scss

4

(a) Network Layer Activities: Data Plane vs. Control Plane

Data Plane:

- **Forwarding:** Actual movement of packets from an incoming interface to an outgoing interface on the router. It involves looking up the forwarding table and determining the next hop for the packet.
- **Packet Processing:** Involves handling the headers, possibly performing Network Address Translation (NAT), packet filtering, and quality of service (QoS) enforcement.

- **Traffic Shaping:** Implementing policies to control the volume of traffic being sent into the network, often to ensure fair bandwidth allocation and to prevent congestion.

Control Plane:

- **Routing:** Determining the best paths for data to travel across a network. This involves running routing protocols (like OSPF, BGP) to exchange routing information and build the routing table.
- **Network Management:** Involves configuring, monitoring, and managing network resources, such as IP address assignment, policy enforcement, and performance monitoring.
- **Signaling:** Setting up and maintaining state information needed to establish and manage end-to-end communication paths. This can include protocols for path setup in MPLS networks, for example.

(b) IP Address Assignment and NAT in Home Networks

IP Address Assignment:

- **ISP Assigned IP:** The ISP dynamically assigns a single public IP address to your wireless router via DHCP.
- **Internal IP Addresses:** The wireless router acts as a DHCP server for the home network, assigning private IP addresses (typically from the 192.168.x.x range) to the five hosts using 802.11 wireless connections.

NAT Usage:

- **Yes, the wireless router uses NAT (Network Address Translation).**
    - **Reason:** The router translates private IP addresses of the home devices to the single public IP address assigned by the ISP. This is necessary because the private IP addresses used within the home network are not routable on the public internet. NAT allows multiple devices to share a single public IP address, conserving the limited number of available public IP addresses

and providing a level of security by masking the internal network structure.

(c) IPv4 vs. IPv6 Header Fields

IPv4 Header Fields:

1. Version
2. Header Length
3. Type of Service
4. Total Length
5. Identification
6. Flags
7. Fragment Offset
8. Time to Live (TTL)
9. Protocol
10. Header Checksum
11. Source Address
12. Destination Address
13. Options (if any)

IPv6 Header Fields:

1. Version
2. Traffic Class
3. Flow Label
4. Payload Length
5. Next Header
6. Hop Limit
7. Source Address
8. Destination Address

Comparison and Effectiveness:

- **Simplified Header:** IPv6 has a simplified header with fewer fields, making packet processing more efficient.

- **Larger Address Space:** IPv6 addresses are 128 bits long, compared to IPv4's 32 bits, effectively eliminating the need for NAT due to the vast number of available addresses.
- **Flow Label:** A new field in IPv6, used to identify flows of packets needing special handling, such as real-time service.
- **Traffic Class:** Replaces the Type of Service field in IPv4, used for QoS management.

The new fields and the streamlined header in IPv6 improve routing efficiency, scalability, and support for new services like real-time data flows.

(d) Generalized Forwarding in Software-Defined Networking (SDN)

Generalized Forwarding:

- **Definition:** In SDN, generalized forwarding refers to the ability of the network to forward packets based on a broad range of criteria, not just the destination address. This can include source address, type of service, application type, or any other packet attributes.
- **Mechanism:** Uses flow tables maintained by switches that are populated by a centralized controller. The controller defines how packets should be forwarded based on higher-level policies.

Differences from Destination-Based Forwarding:

- **Traditional Destination-Based Forwarding:** Packets are forwarded solely based on the destination IP address and the routing table entries in routers.
- **Generalized Forwarding:** Decisions can be made based on various packet attributes and rules defined by a central controller, allowing for more complex and flexible forwarding policies.

(e) Differences Between RR and WFQ Packet Scheduling

Round Robin (RR):

- **Mechanism:** Packets are sent in a rotating, cyclic order from each queue.

- **Fairness:** Provides equal opportunity for each queue to send packets, but does not consider packet size or flow requirements.
- **Example:** If three flows have packets in their queues, each queue gets an equal turn to send a packet.

Weighted Fair Queueing (WFQ):

- **Mechanism:** Packets are scheduled based on weights assigned to each flow, which reflect the priority or share of the bandwidth.
- **Fairness:** Provides more precise control over bandwidth allocation by considering the weight and size of packets.
- **Example:** If three flows have weights 1, 2, and 3, respectively, the third flow gets three times the bandwidth of the first flow.

Case Where RR and WFQ Behave the Same:

- If all flows are assigned equal weights in WFQ, both RR and WFQ will behave the same.
- **Example:** Consider three flows with equal weights (1, 1, 1):
    - Both RR and WFQ will allow each flow to send packets in a round-robin manner, effectively giving each flow an equal share of the bandwidth.

By assigning equal weights to all flows in WFQ, the packet scheduling will mimic that of RR, where each flow gets an equal turn.

5

(a) Compare and Contrast Link State and Distance Vector Routing Algorithms

Link State Routing:

- **How it works:** Each router has a complete map of the network and calculates the shortest path to every other router.

- **Updates:** Routers broadcast their link state information to all other routers in the network.
- **Convergence:** Typically faster, as routers have a full view of the network.
- **Example Protocol:** OSPF (Open Shortest Path First).

Distance Vector Routing:

- **How it works:** Each router knows the distance to its immediate neighbors and shares this information with them.
- **Updates:** Routers periodically send their distance vectors to neighboring routers.
- **Convergence:** Slower, can suffer from routing loops and “count to infinity” problems.
- **Example Protocol:** RIP (Routing Information Protocol).

Comparison:

- **Convergence:** Link state converges faster; distance vector can be slower.
- **Complexity:** Link state is more complex and resource-intensive; distance vector is simpler and uses fewer resources.
- **Information:** Link state has a global view of the network; distance vector only has local information.

(b) Autonomous System (AS)

Definition: An Autonomous System (AS) is a group of IP networks operated by a single organization with a unified routing policy. Each AS is identified by a unique number called an Autonomous System Number (ASN).

Reducing Route Computation Complexity: Yes, it reduces complexity.

- **Reason:** The internet is divided into ASes, and each AS handles its internal routing. External routing between ASes is managed by protocols like BGP. This hierarchical structure simplifies routing by limiting the scope of route calculations to within an AS and between ASes, rather than across the entire internet.

(c) BGP Route Selection Process

When BGP has multiple routes to the same destination, it uses the following steps to select the best one:

1. **Highest Local Preference:**
   - Prefer the route with the highest local preference value.
2. **Shortest AS-PATH:**
   - Prefer the route with the shortest AS-PATH.
3. **Lowest Origin Type:**
   - Prefer the route with the lowest origin type (IGP < EGP < Incomplete).
4. **Lowest MED (Multi-Exit Discriminator):**
   - Prefer the route with the lowest MED value.

If routes are still equal after these steps, additional criteria like eBGP over iBGP, lowest IGP cost to the NEXT-HOP, and finally router ID can be used to break ties.