THE UNIVERSITY OF SYDNEY

Tutorial
ELEC3506/9506
Communication Networks

School of Electrical and Information Engineering
The University of Sydney

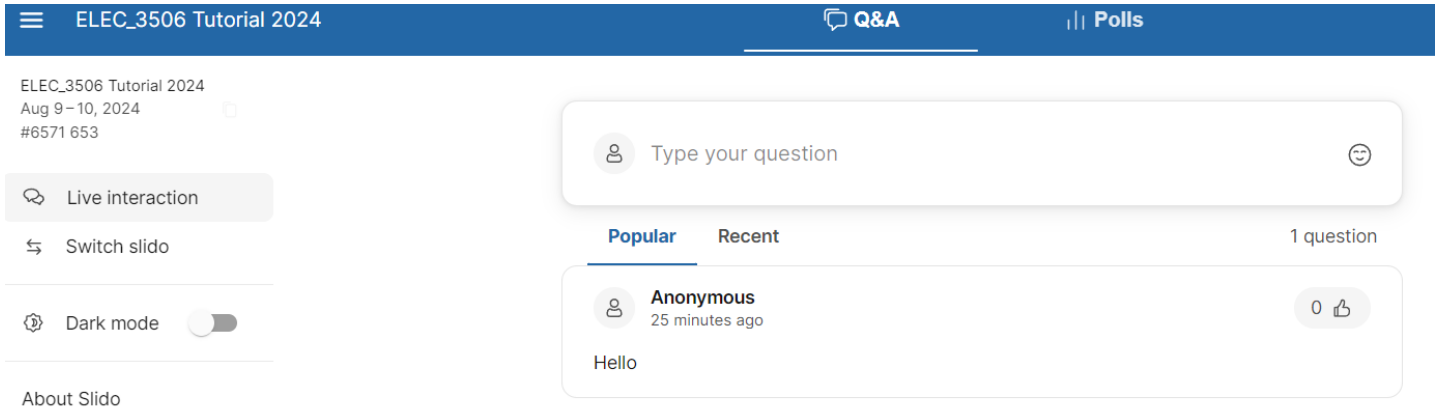
Dr. Shuvashis Saha
shuvashis.saha@sydney.edu.au

Tutorial 06

THE UNIVERSITY OF SYDNEY

Live Q&A Sessions

Join the Session: Scan the QR code



Ask Questions: Type your questions anytime during the tutorial.

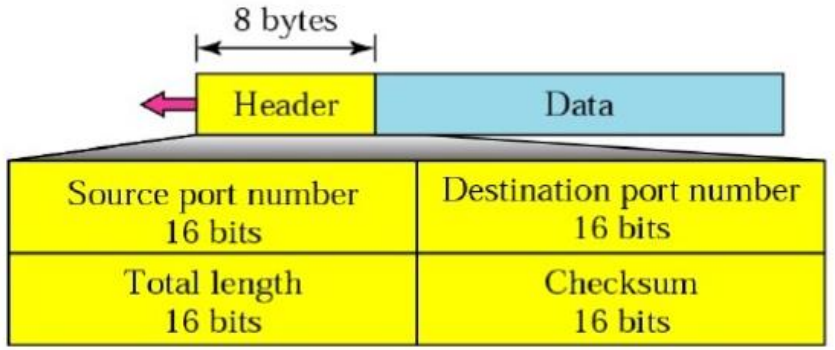
Q1. In a case where reliability is not of importance, UDP would make a good transport protocol. Give examples of specific cases.

User Datagram Protocol (UDP)

- **UDP** is a **connectionless** and **unreliable transport protocol** that offers process-to-process delivery with limited error checking.
  - **Connectionless**: The segments are sent to the destination host without any prior establishment of the connection between communicating hosts.
  - **Unreliable transport protocol**: UDP does not-perform proper error and flow control and thus, does not guarantee about the proper delivery of segments at the destination.

Why UDP?

- Processes that require simple request-response communication with little concern for flow and error control. (e.g., Live video streaming)
- **Small delay**: no need for connection establishment.
- **Simple**: no connection state at sender, receiver.
- **Low overhead**: small segment header.



User datagram format

Q1. In a case where reliability is not of importance, UDP would make a good transport protocol. Give examples of specific cases.

Applications (cases) use UDP

- Used for simple request response communication when size of data is less and hence there is lesser concern about flow and error control.

- **Applications where we send a small message and need a fast service:**
  - Domain Name System (DNS)
  - Simple Network Management Protocol (SNMP)
  - Dynamic Host Configuration Protocol (DHCP)

- **Applications where occasional packet losses are acceptable:**
  - Real-time video and audio streaming
  - Online games (Games that don't care if you get every update)

- **Applications where the error checking is performed at the application layer:**
  - Some VPN systems, e.g., OpenVPN

- **Multicast applications where you want to send the same data to multiple hosts.**
  - **Traceroute** (computer network diagnostic commands for displaying possible routes (paths) and measuring transit delays of packets)

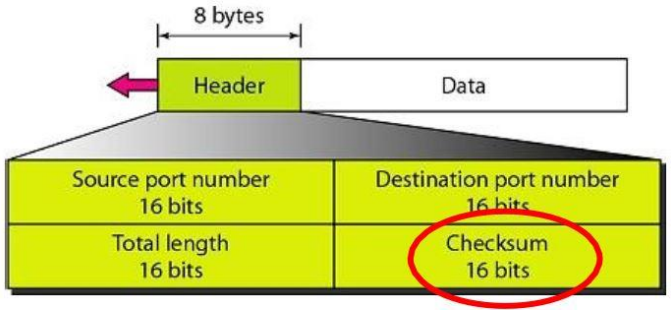
Q2. Are both UDP and IP unreliable to the same degree? Why or why not?

Both UDP and IP are connectionless and unreliable protocol

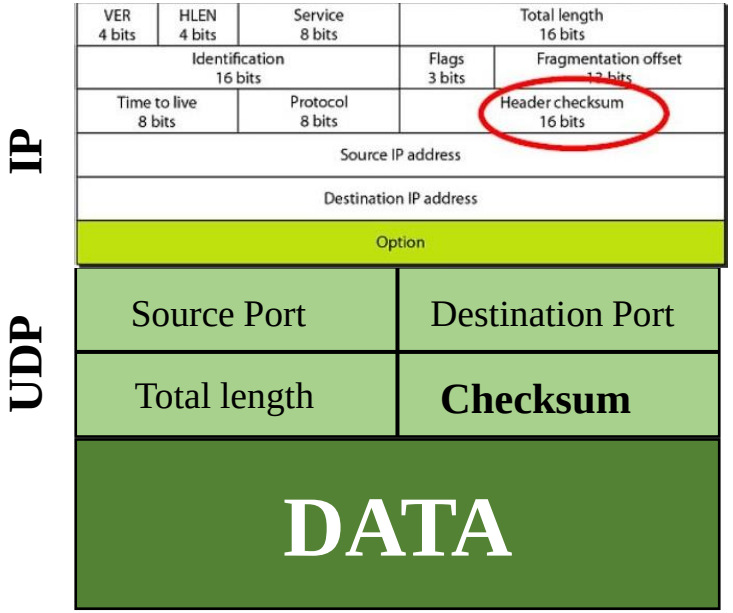
- **Connectionless** means each datagram is sent to the destination host without any prior establishment of the connection between communicating hosts.
- **Unreliable** means it does not guarantee about the successfully delivery of the message.

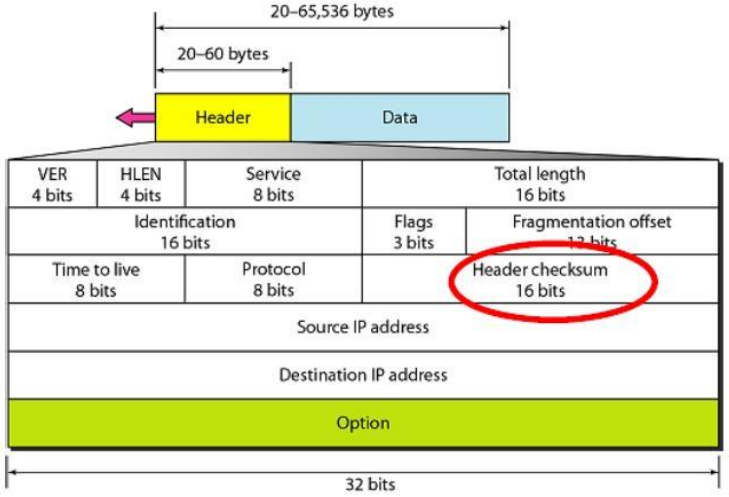
Difference between UDP and IP

- **IP** only calculates a checksum for the IP header and not for the data.
- **UDP** calculates a checksum for the entire datagram



UDP Datagram Format





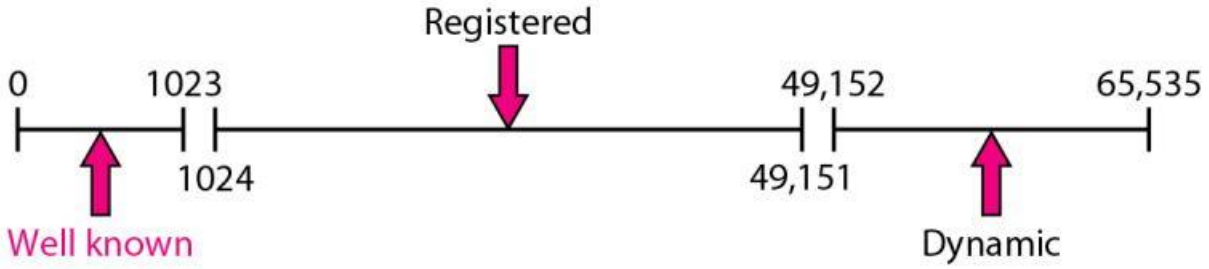
IP Datagram Format

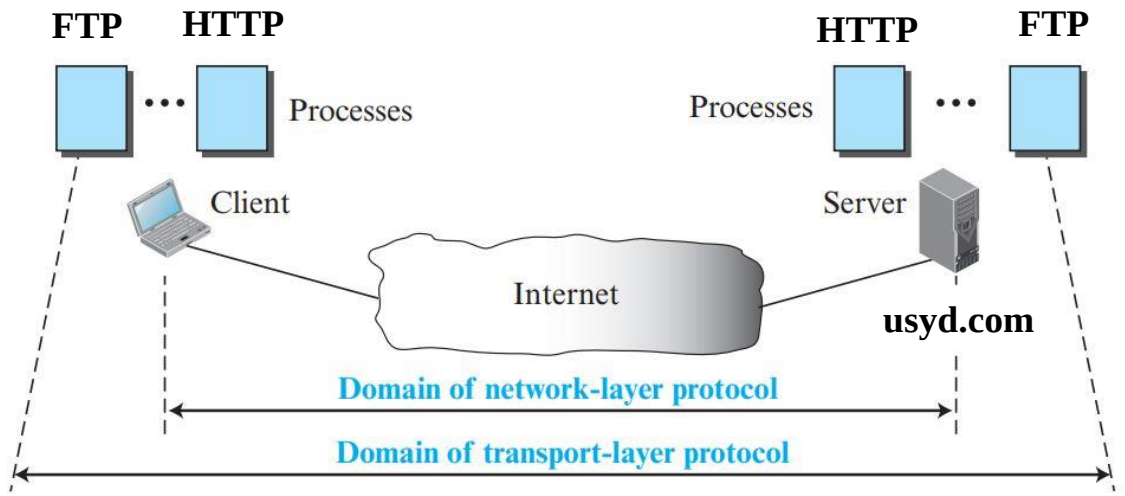
❑ **Therefore, in terms of data integrity, UDP is more reliable.**

Q3. Do port addresses need to be unique? Why or why not? Why are port addresses shorter than IP addresses?

Transport Layer Addressing

- Client and server may run multiple programs at the same time
- For communication, we must define:
  - Local host and remote host (**IP Addresses**)
  - Local process and remote process (**Port Number**)
- **Port number**: 16-bit integer between 0 and 65,535
- **Client port number**: ephemeral (short-lived)
- **Server port number**: well-known





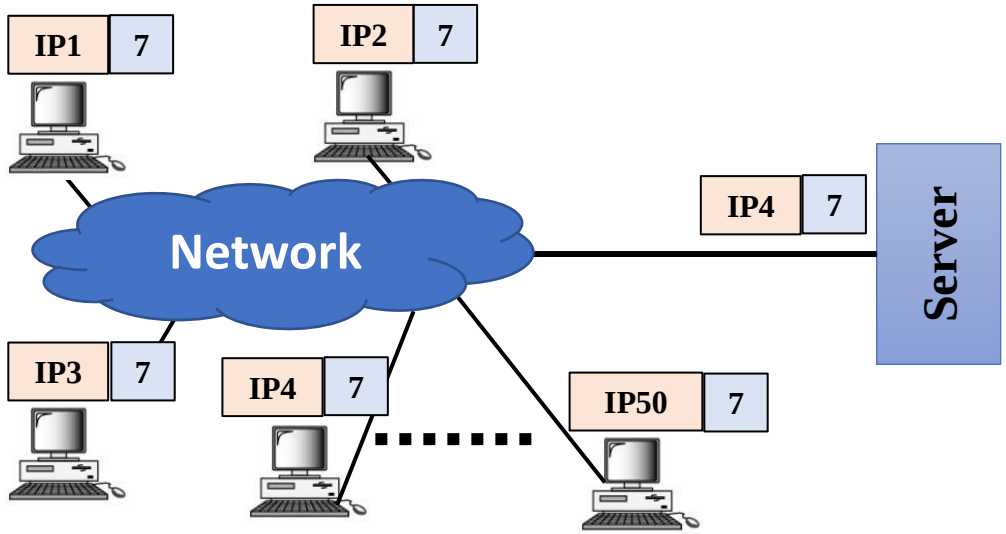
Well Known Port Numbers

Port Number	Application Process Protocol
20	FTP data transfer
21	FTP Control
23	Telnet
25	SMTP
53	DNS
80	HTTP

Q3. Do port addresses need to be unique? Why or why not? Why are port addresses shorter than IP addresses?

Do port addresses need to be unique?

- **Port addresses do not need to be universally unique** - *as long as each IP address/port address pair uniquely identify a particular process running on a particular host.*
- **Example**: a network consisting of 50 hosts, each running echo server software. Each server uses the well known port number 7, but the IP address, together with the port number of 7, **uniquely identify a particular server program on a particular host.**



Why are port addresses shorter than IP addresses?

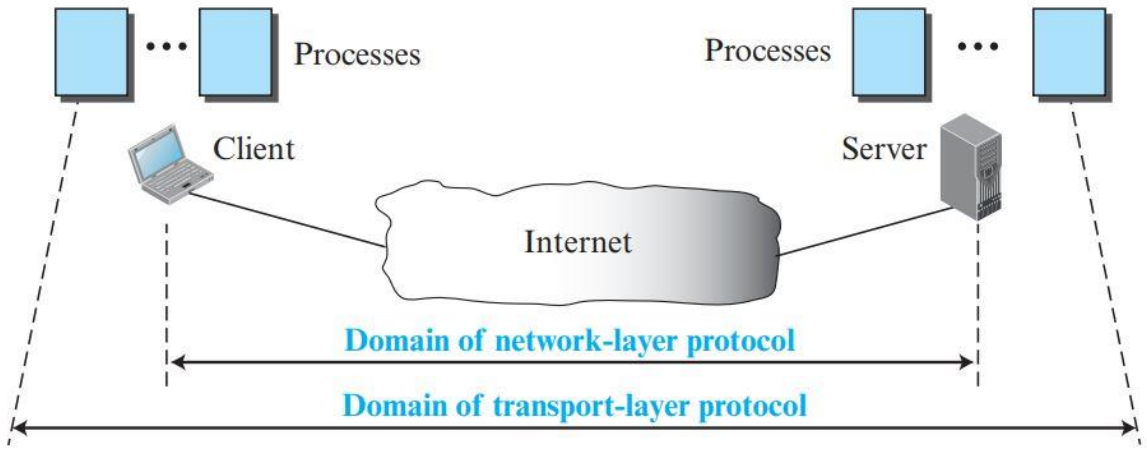
- Port addresses are shorter than IP addresses because their domain, a single system, is smaller than the domain of IP addresses, all systems on the Internet.

- The destination IP address defines the host among the different hosts in the world.
- After the host has been selected, the **port number** defines one of the processes on this particular host.

Q4. Distinguish between network layer and transport layer services.

Network Layer Vs. Transport Layer

- Network-layer protocol provides logical communication between hosts.
- Transport-layer protocol provides logical communication between processes running on different hosts.



A network-layer protocol can deliver the message only to the destination computer.

A transport-layer protocol is responsible for delivery of the message to the appropriate process.

Network Layer uses Logical Address (IP)

Transport Layer uses Port Address

Q4. Distinguish between network layer and transport layer services.

Network Layer Services

- **Packetizing**: Encapsulating the payload (data received from upper layer) and decapsulating the payload from the network-layer at the destination.
- **Moving Packets**: Network layer delivers individual packets from the source host to the destination host
- **Logical (IP) Addressing**: For a packet being sent to another network, logical addressing system is required to distinguish the source and destination systems.
  - The network layer adds a header to the packet coming from the upper layer and includes the IP address.
- **Routing:** The network layer is responsible for routing the packet from its source to the destination**.** The network layer is responsible for finding the best one among these possible.
- **Error Control (Header):** This error control may prevent any changes or corruptions in the header of the datagram.

Transport Layer Services

1. Moving Message: Delivers message from one process to another**.**

2. Port Addressing: It is necessary to identify the desired process out of many processes. For this, the transport layer header include a port address in each segment.

3. Segmentation and Reassembly: A message is divided into segments by the transport layer with each segment being given a sequence number. These sequence numbers enable the destination transport layer to reassemble the segments in exact order as they were sent by the sender.

4. Flow Control: The-transport layer is responsible for controlling the flow of data such that no sending process should send segments at a rate faster than the receiving process can process.

5. Error Control: The transport layer provides process-to-process error control rather than across a single link as provided by the data link layer.

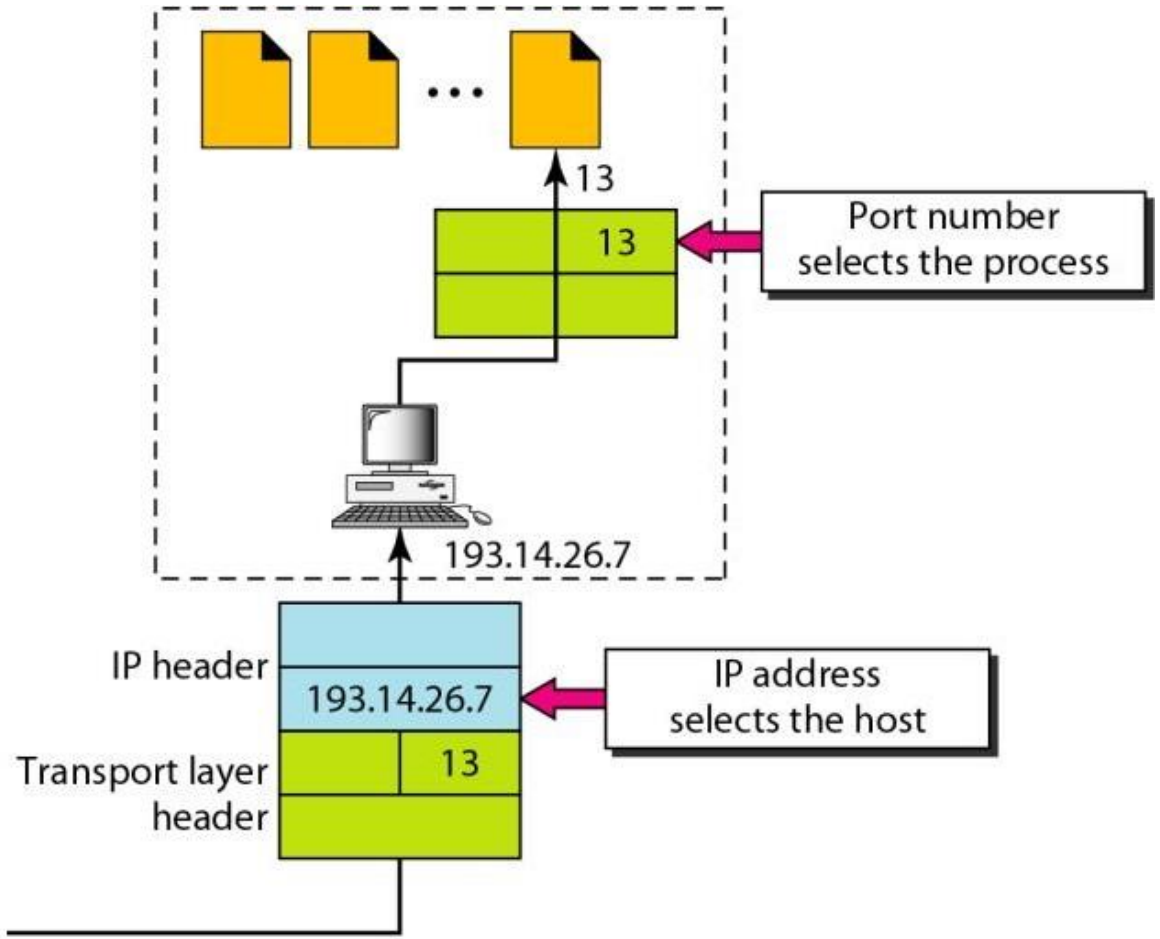
Q5. Distinguish between an IP address, a Port address and a Socket address.

IP Address

- The IP address defines the sender and receiver at the network layer and is used to deliver messages across multiple networks. IPv4: 32 bit and IPv6: 128 bit.
- For example, in Internet, each host connected to the Internet is assigned a 32-bit IP address and no two hosts connected to the Internet can have the same IP address.

Port address

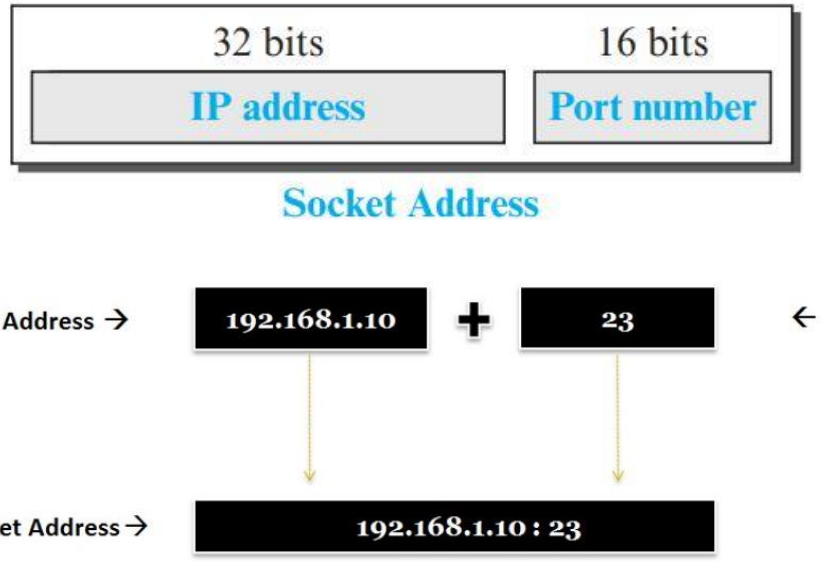
- Transport layer concerns with Port address.
- The port address (service-point) identifies the application process on the station.
- Since multiple processes may be running simultaneously on the host machine, there should be some means to identify the process to which data is to be communicated.

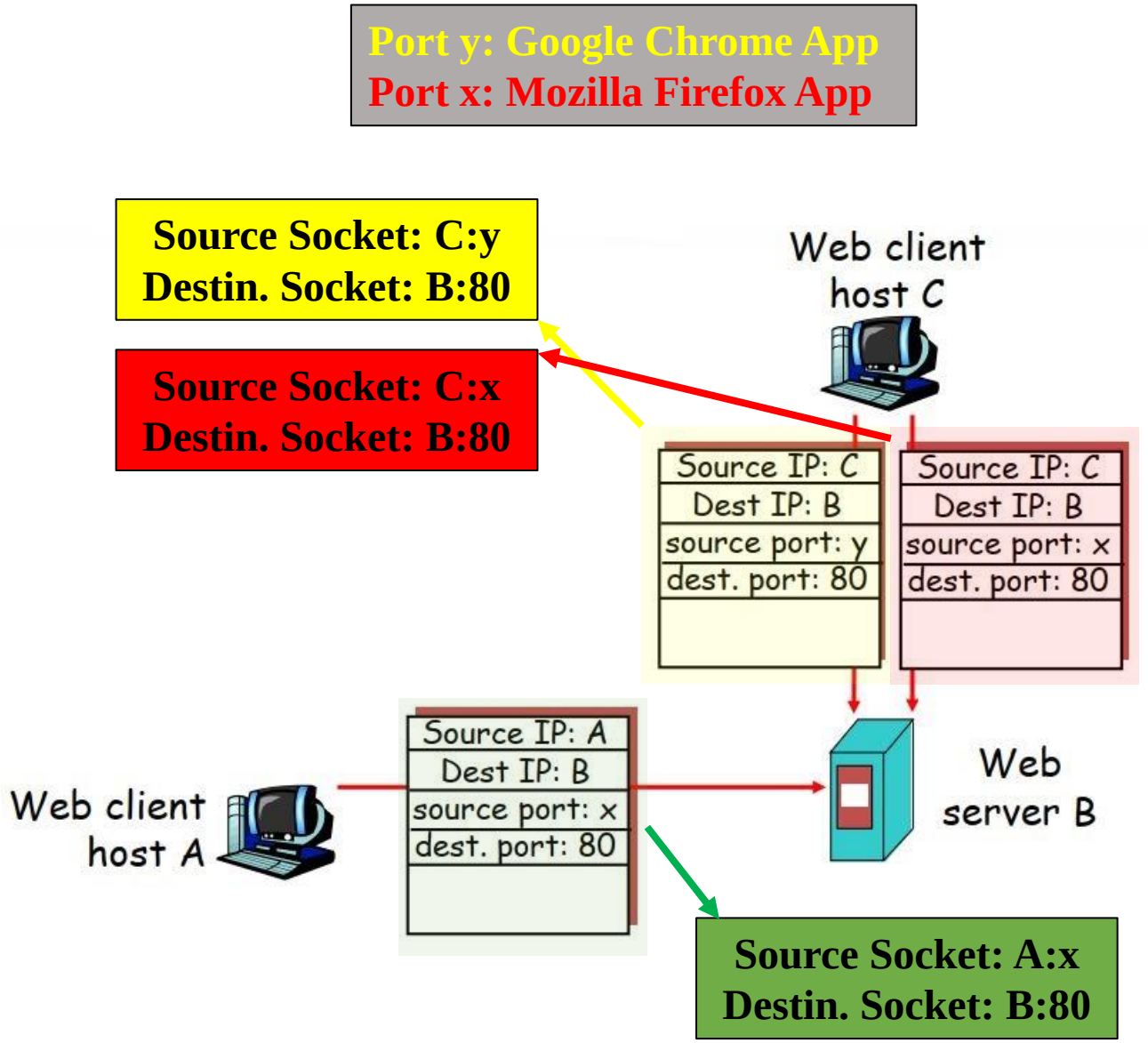


Q5. Distinguish between an IP address, a Port address and a Socket address.

Socket address

- **Socket address**: Combination of **IP address** and **port number**.
- Uniquely defines the client/server process.
- To use services of the transport layer, we need a pair of socket addresses: client and server socket addresses.

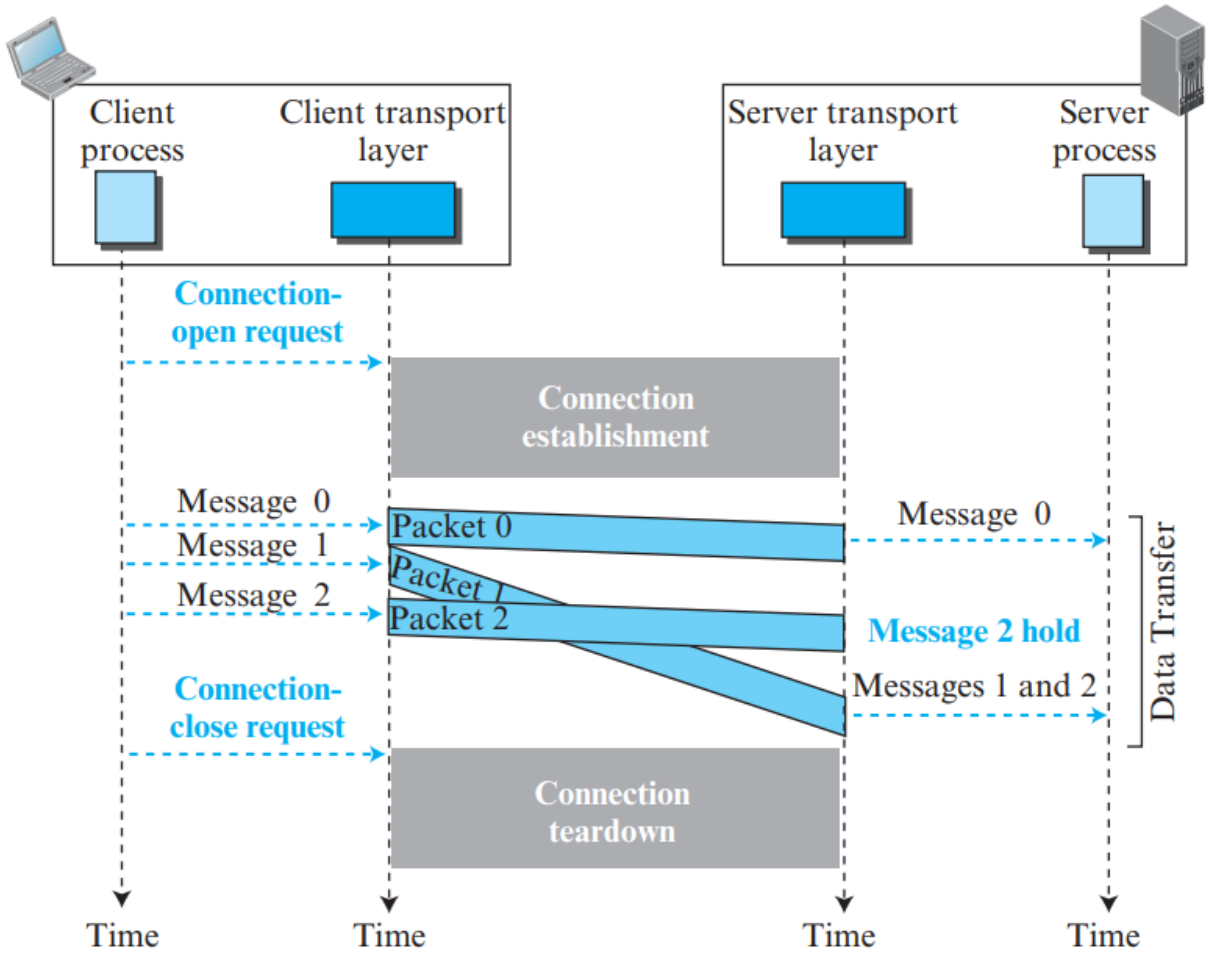




Q6. Distinguish between connection oriented and connection-less services.

Connection oriented service

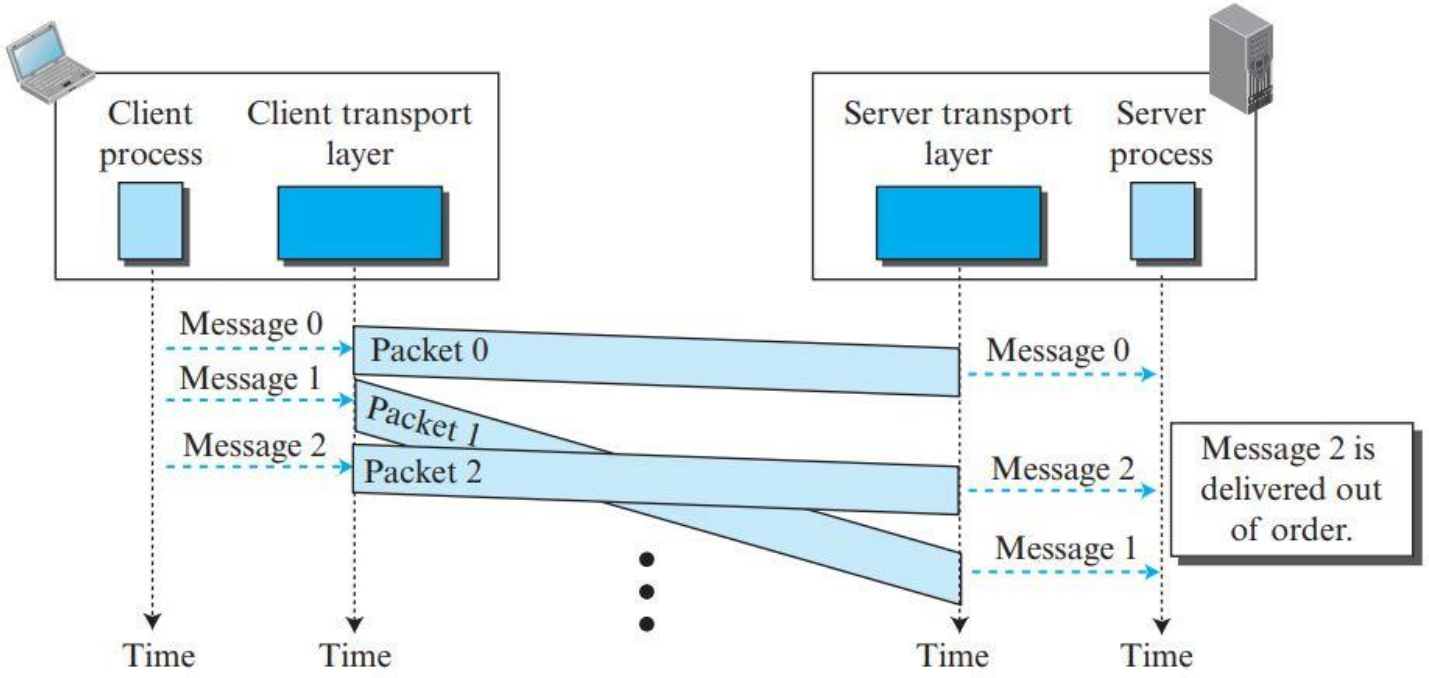
- A logical connection is first established between sender and receiver.
- The data exchange can only happen after the connection establishment.
- All packets follow the same path
- Data transferred /delivered in sequence.
- Communication has three phases:
  - Connection establishment,
  - Data transfer,
  - Connection termination



Q6. Distinguish between connection oriented and connection-less services.

Connection-less service

- No need for connection establishment or release.
- Each packet is independent from every other packet and can take different path.
- Data transferred /delivered does not follow any sequence.
- The packets may be delayed or lost or may arrive out of sequence.
- Faster process but unreliable.



Q7. Distinguish between reliable and unreliable services.

- **Reliable Delivery**: Data sent from a source across the communication system must be delivered only to the intended destination.

Reliable Transport layer protocol provides	An unreliable transport layer service provides
Delivery confirmation: **acknowledge mechanism**	No delivery confirmation
Flow control	No flow control
Error control	No (Limited) error control
Slower and complex service but secured.	Faster and less complex service
Example: Transmission Control Protocol (TCP)	Example: User Datagram Protocol (UDP)

Q8. What are the three stages of connection-oriented transmission?

- A connection-oriented transport protocol establishes a logical path between the source and destination. TCP is connection-oriented transport layer protocol.

- Connection-oriented transmission requires three phases:

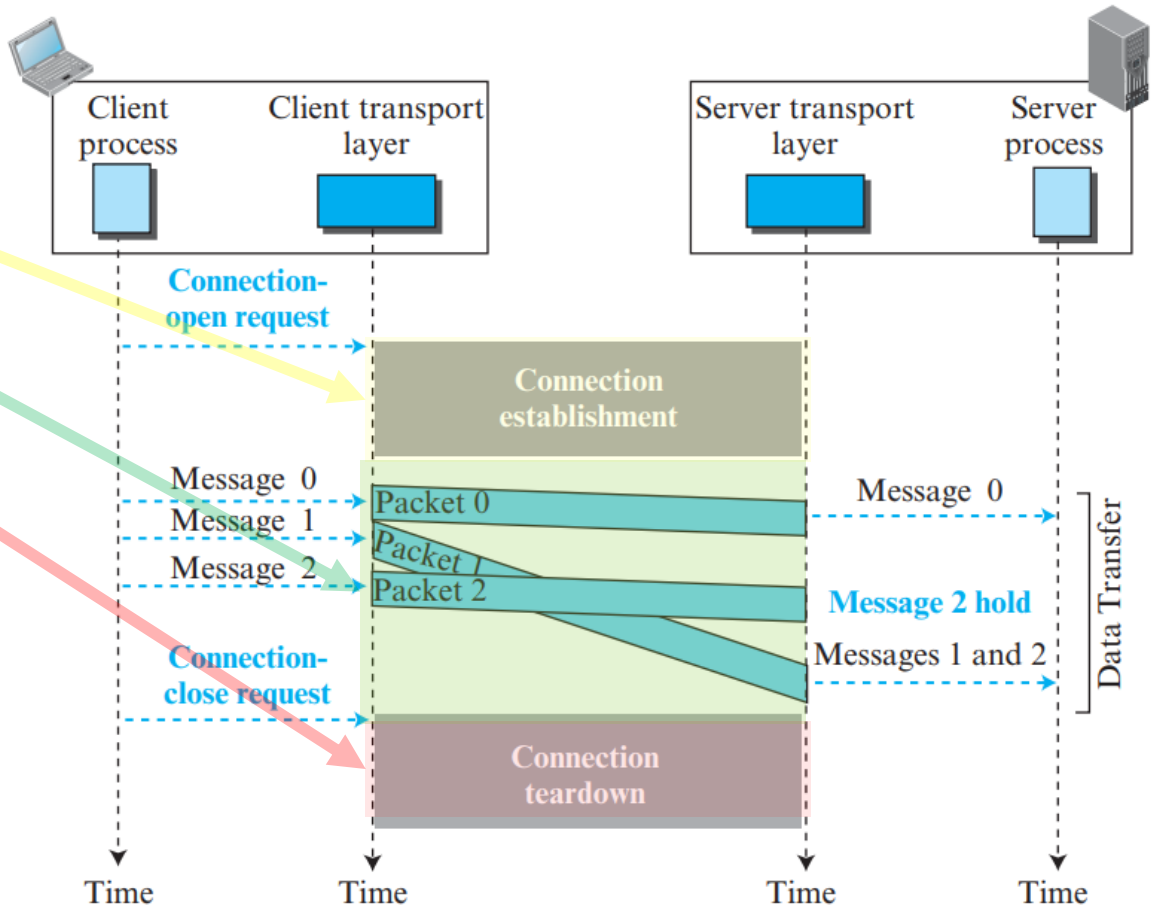
I. Connection establishment

II. Data transfer

III. Connection termination

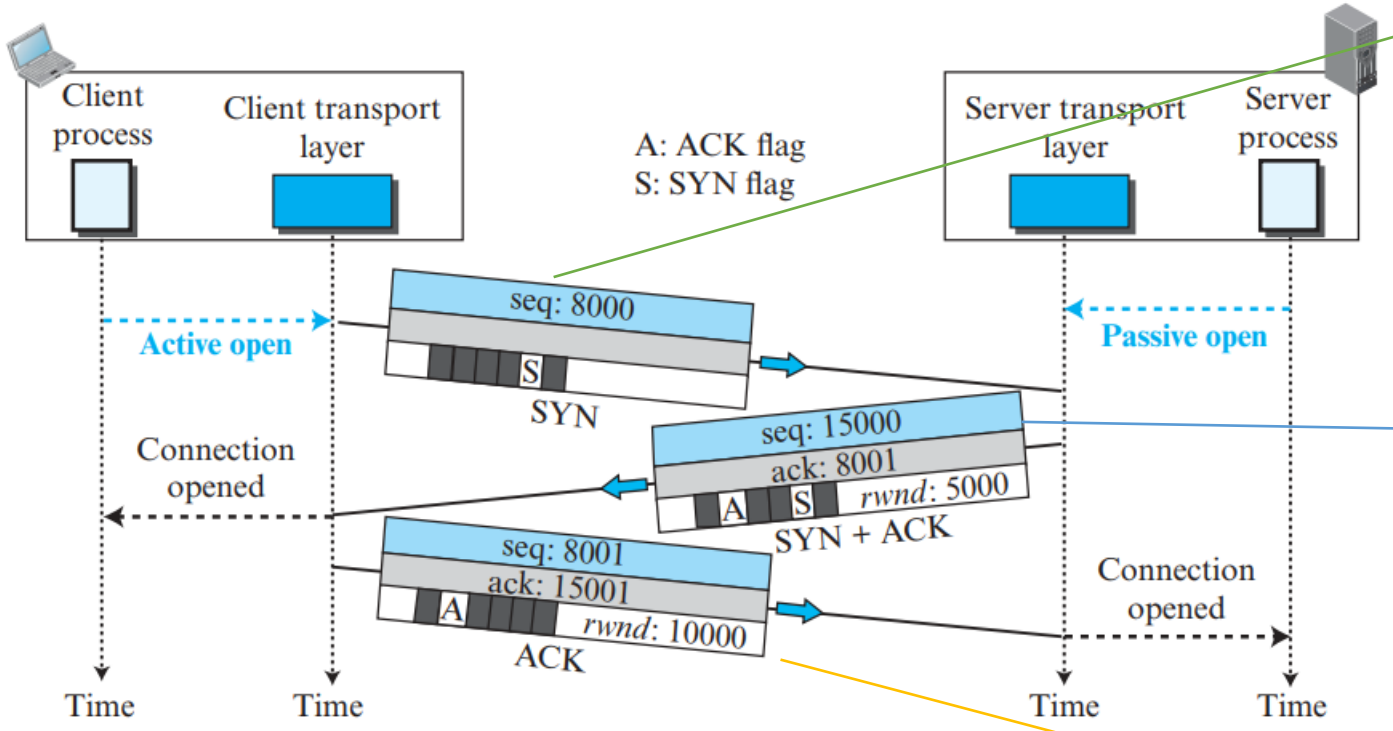
- The connection establishment in TCP is called three-way handshaking.

- Once a connection is established, data may flow back and forth until the connection is closed.



Q9. Describe the three-way handshake used for TCP connection establishment.

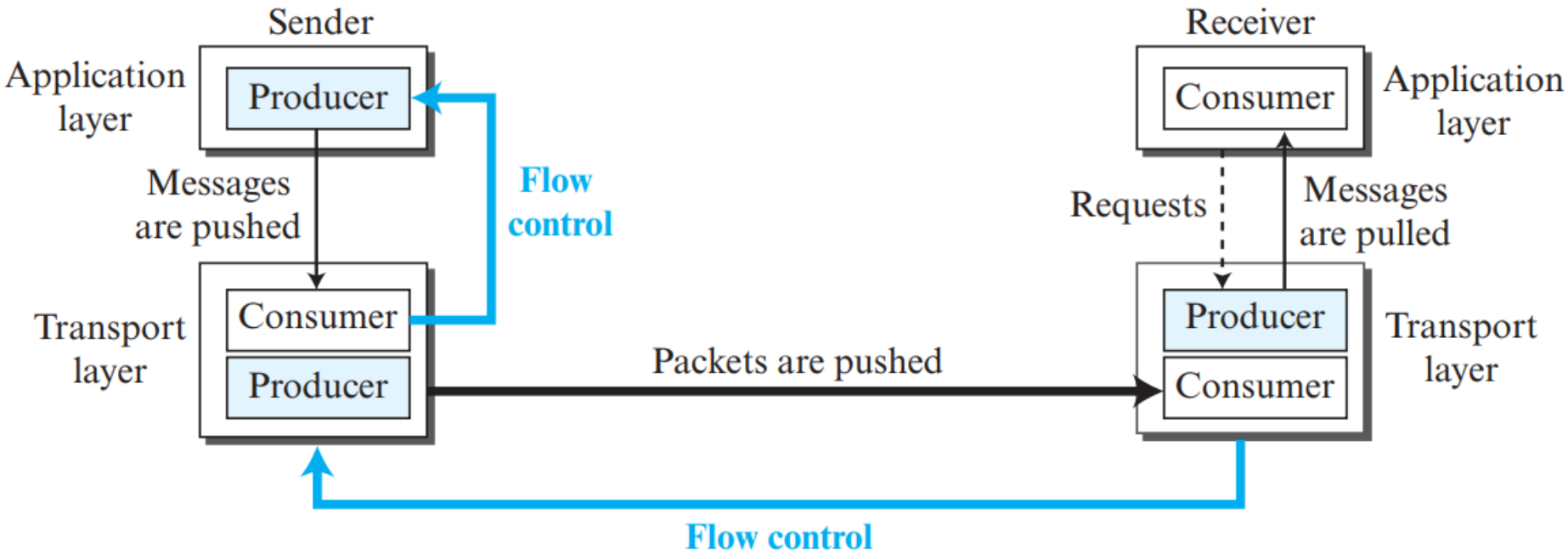
Figure *Connection establishment using three-way handshaking*



- Connection Request (SYN): The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for the synchronization of sequence numbers.
- Connection Confirmation (SYN + ACK): The servers send the second segment, a SYN+ACK segment. It serves to establish communication in the other direction and acknowledgment of the first SYN segment.
- Acknowledgment of confirmation (ACK): The client sends the third segment, an ACK segment. It serves to acknowledge the receipt of the second segment

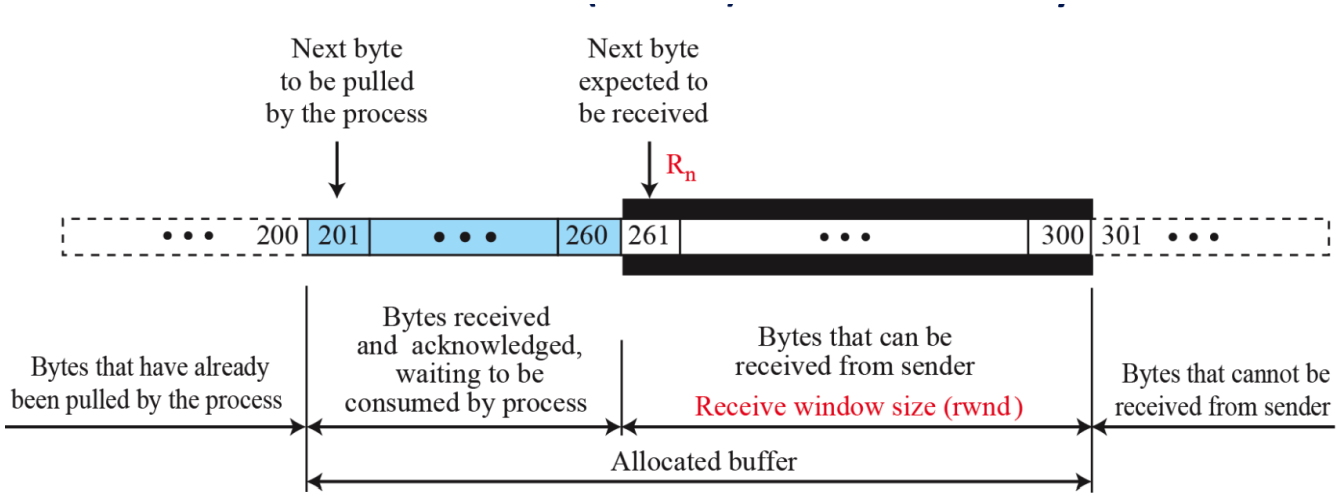
Q10. What is Flow Control?

- Whenever an entity produces items and another entity consumes them, there should be a balance between production and consumption rates. If the items are produced faster than they can be consumed, the consumer can be overwhelmed and may need to discard some items.



Q10. What is Flow Control?

- Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for an acknowledgement.
- This is done to prevent the receiver from being overwhelmed with data.



a. Receive window and allocated buffer

- The receiver controls the amount of data that can be transmitted via the received window size (rwnd) and sender obeys this.

rwnd = buffer size – number of waiting bytes to be pulled

- Rwnd gets smaller when more bytes arrive from the sender.
- It becomes bigger when more bytes are processed

Q11. Distinguish between the flow control provided by the data link layer and the transport layer?.

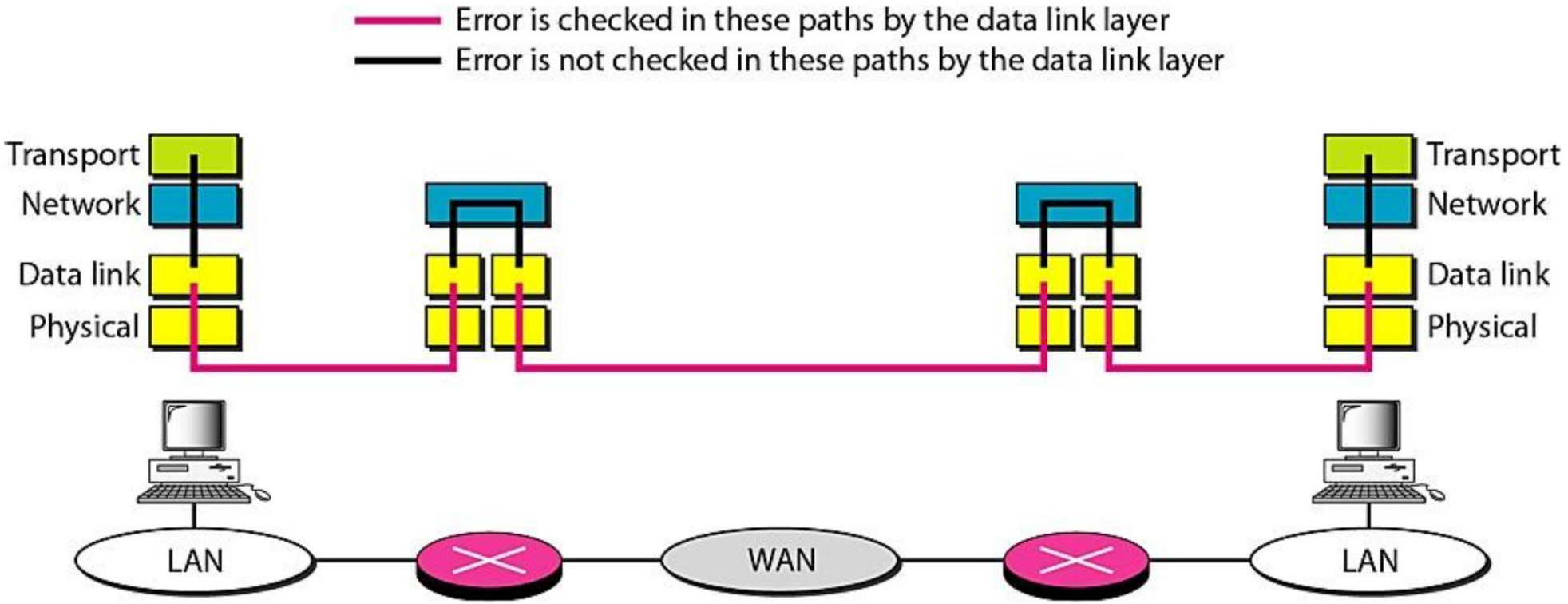
- The sliding window of TCP is byte-oriented; while the one in the data link layer is frame-oriented.
- TCP sliding window is of variable size; while the one in the data link layer is of fixed size.

Q12. How does TCP provide error control?

- TCP provides reliability by using error control. The Error control is used for:
  - Detecting corrupted segments,
  - Detecting lost segments,
  - Detecting out-of-order segments,
  - Detecting duplicated segments, and
  - Correcting errors after they are detected
- The error control is achieved through the use of
  - checksum,
  - acknowledgement,
  - and time-out.

Q13. Distinguish between the error control provided by the data link layer and the transport layer?.

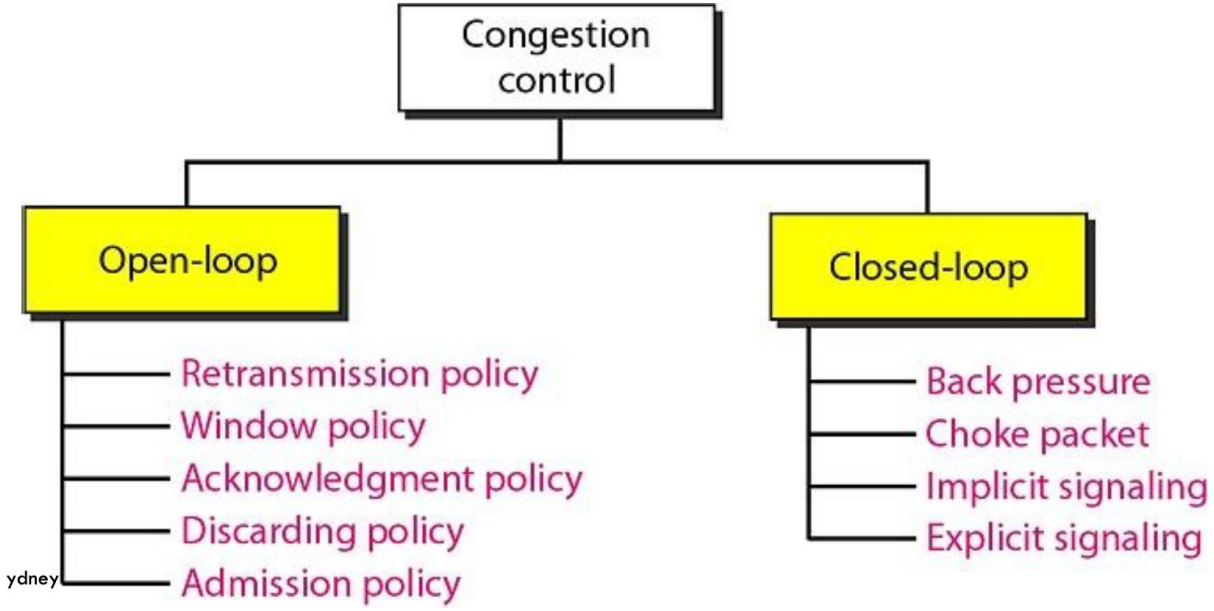
- Reliability at the data link layer is between two nodes; we need reliability between two ends.
- Because the network layer in the internet is unreliable, we need to implement reliability at the transport layer



Q14. What is congestion control and how does TCP provide congestion control?

- Congestion control refers to techniques and mechanism that can either prevent congestion, before it happens, or remove congestion, after it has happened.

- Open-loop: Prevent congestion before it happens
- Closed-loop: Alleviate congestion after it happens



- TCP uses implicit signalling to adjust the congestion window size.

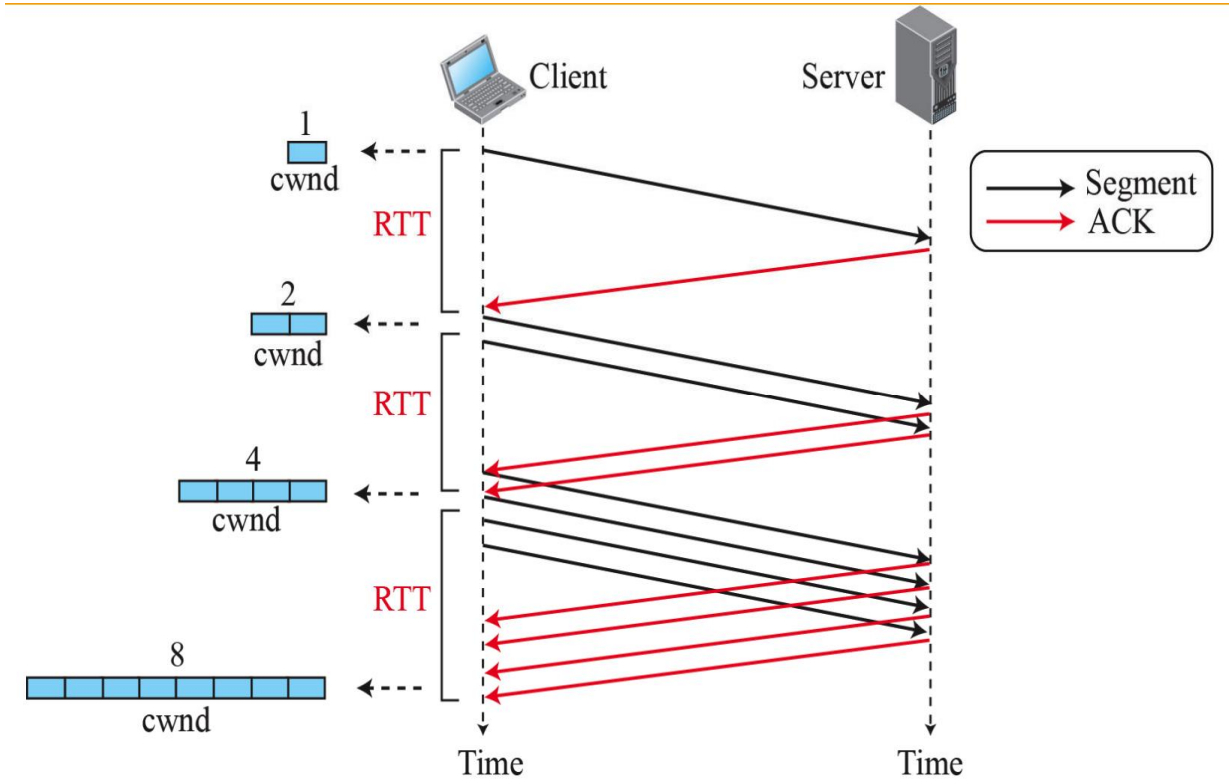
Q15 Discuss the three phases of TCP congestion control policy.

- Slow Start: Exponential Increase
- Congestion avoidance: Additive Increase
- Congestion Detection: Multiplicative Decrease

Q15 Discuss the three phases of TCP congestion control policy.

Phase 1 – Slow-start: Exponential Increase

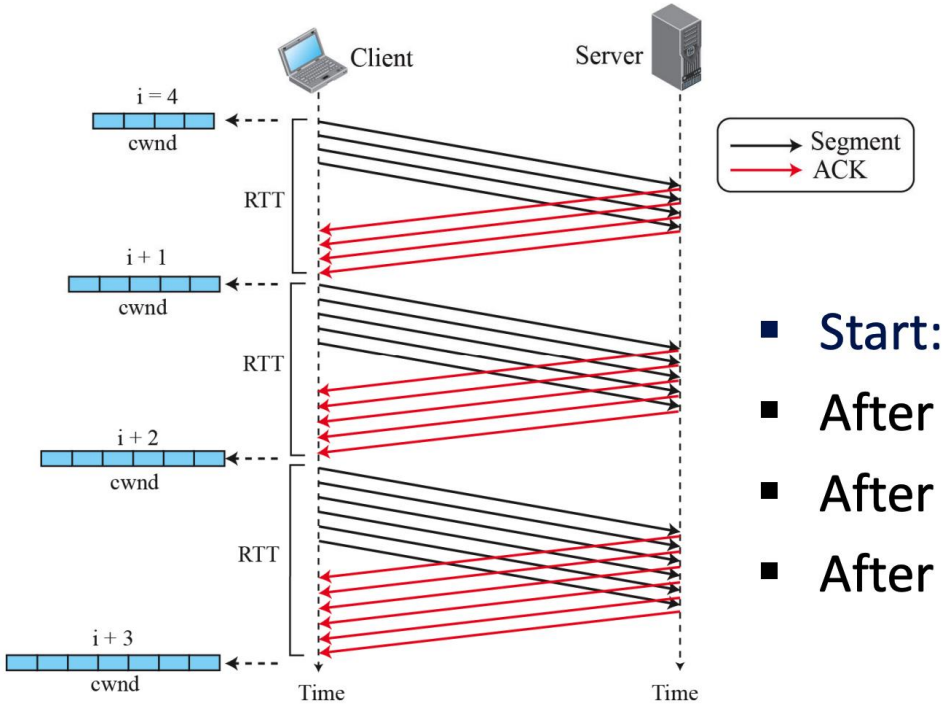
- The size of the congestion window (cwnd) starts with one maximum segment size.
- If an ACK arrives, cwnd = cwnd*2, cwnd increases exponentially in terms of round-trip times (RTT)
- The sender keeps track of a variable named sshtresh (slow-start threshold). When cwnd reaches the threshold, slow start ends and move to phase 2.



Q15 Discuss the three phases of TCP congestion control policy.

Phase 2 – Congestion avoidance: Additive Increase

- After the sender has received acknowledgements for a complete window size of segments,
- cwnd = cwnd + 1, cwnd increases linearly in terms of round-trip times (RTT)
- This phase continues, until a congestion occur.
- If congestion occurs move to phase 3.

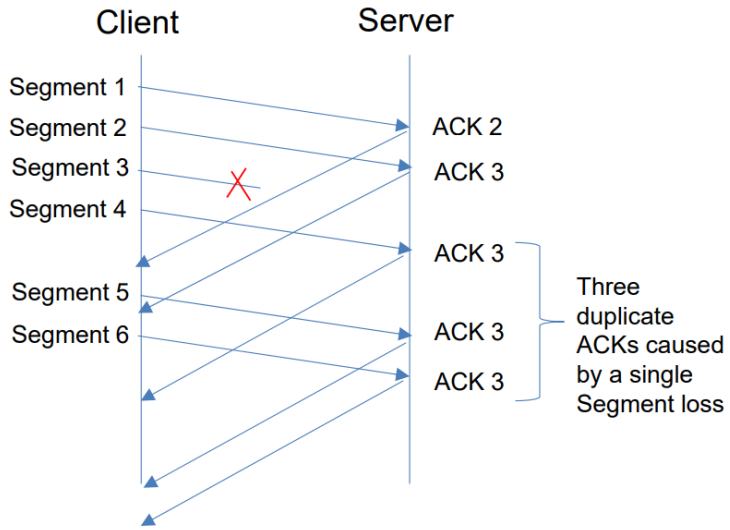


- Start: *cwnd=i*
- After 1 RTT: *cwnd=i*+1
- After 2 RTT: *cwnd=i*+2
- After 3 RTT: *cwnd=i*+3

Q15 Discuss the three phases of TCP congestion control policy.

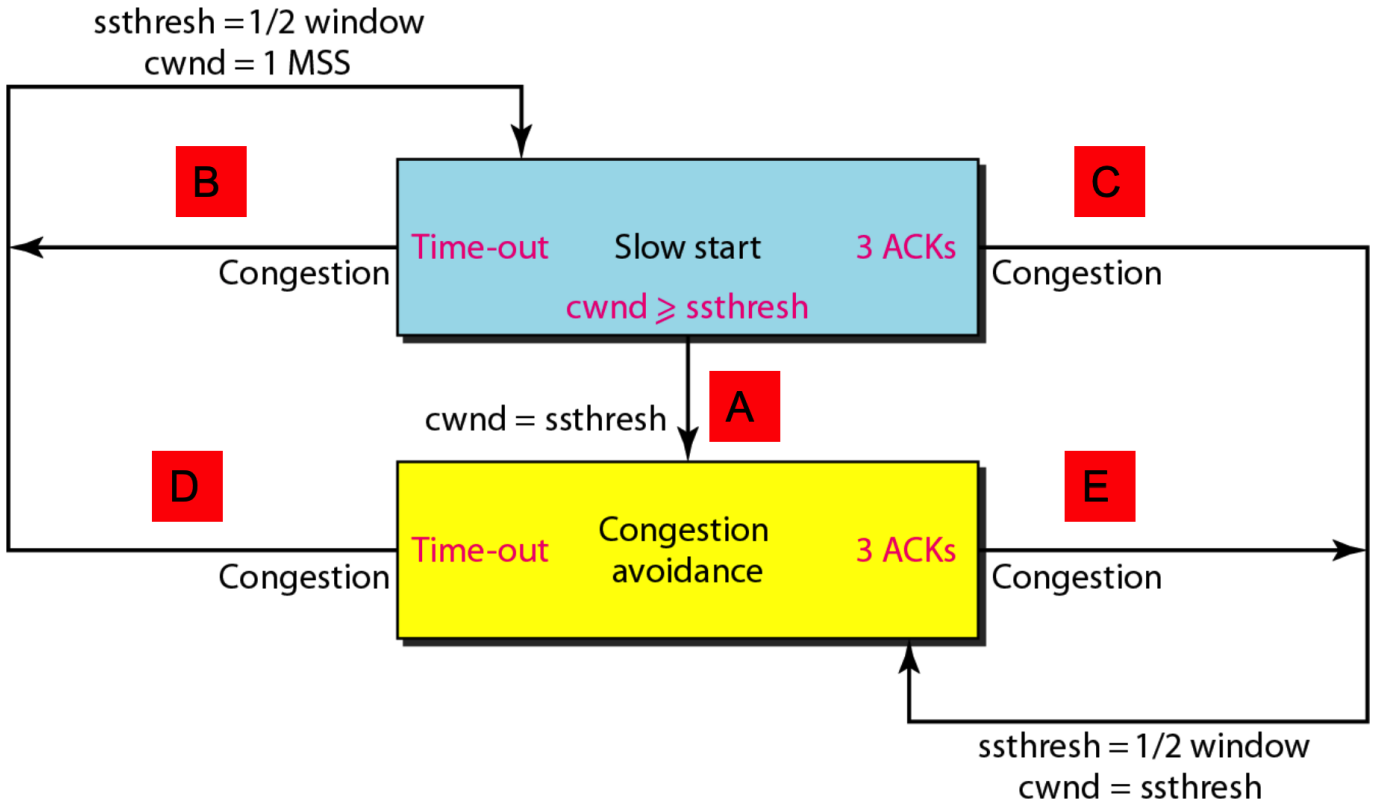
Phase 3 – Congestion Detection: Multiplicative Decrease

- If congestion is detected, the congestion window size must be decreased.
- Congestion can be detected by: time-out or three ACKs.
- If time-out occurs (no ACKs received), there is a stronger possibility of congestion and thus TCP reacts strongly:
  - Sets the value of sshtresh to one-half of the current window size. (sshtresh = ½ cwnd).
  - Sets cwnd to one.
  - Restarts from phase 1 (slow-start)
- If three ACKs are received (some ACK is missing), TCP has a weaker reaction:
  - Sets the value of sshtresh to one-half of the current window size. (sshtresh = ½ cwnd).
  - Sets cwnd to the value of sshtresh (cwnd = sshtresh).
  - Restarts from phase 2 (congestion avoidance)



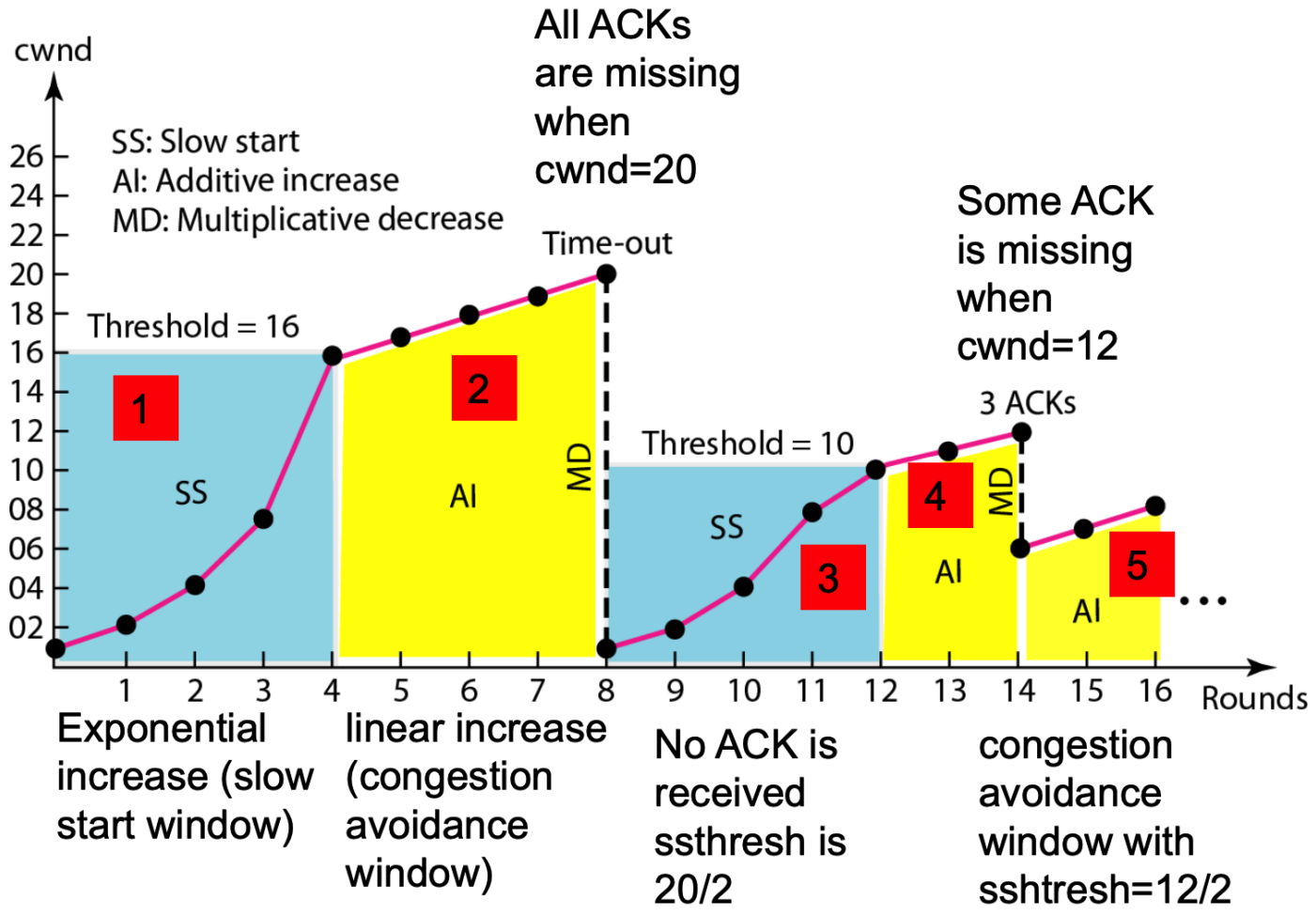
Q15 Discuss the three phases of TCP congestion control policy.

TCP Congestion Control Example



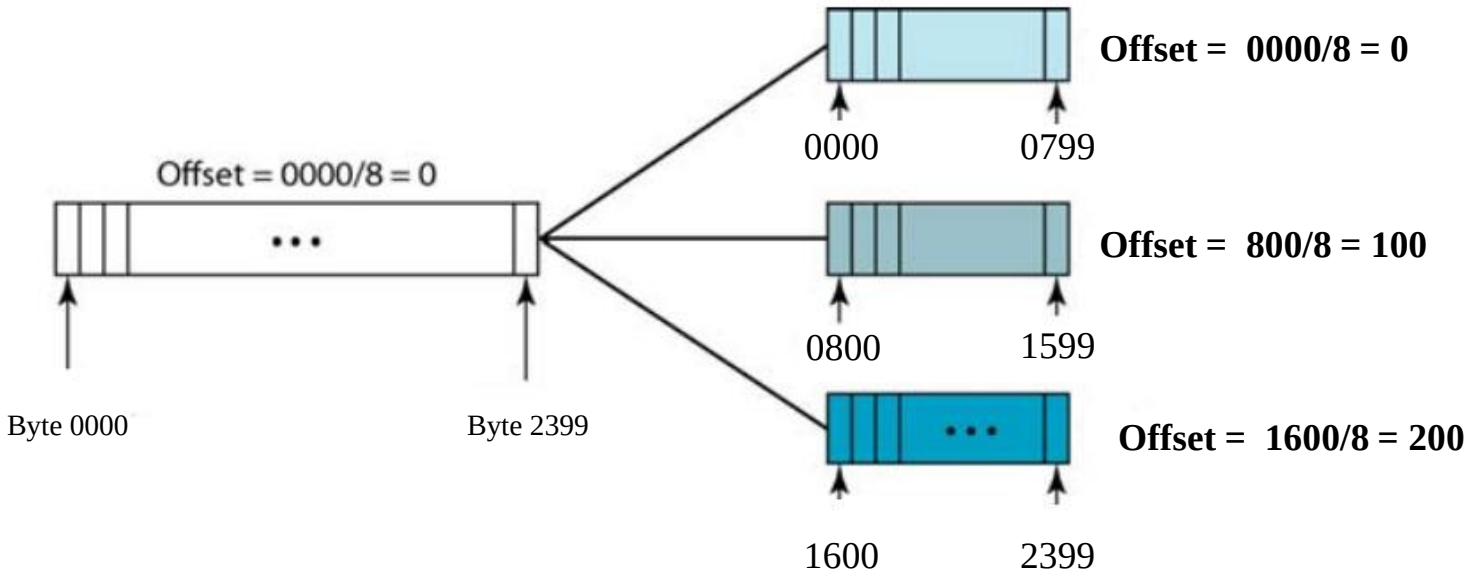
Q15 Discuss the three phases of TCP congestion control policy.

TCP Congestion Control Example



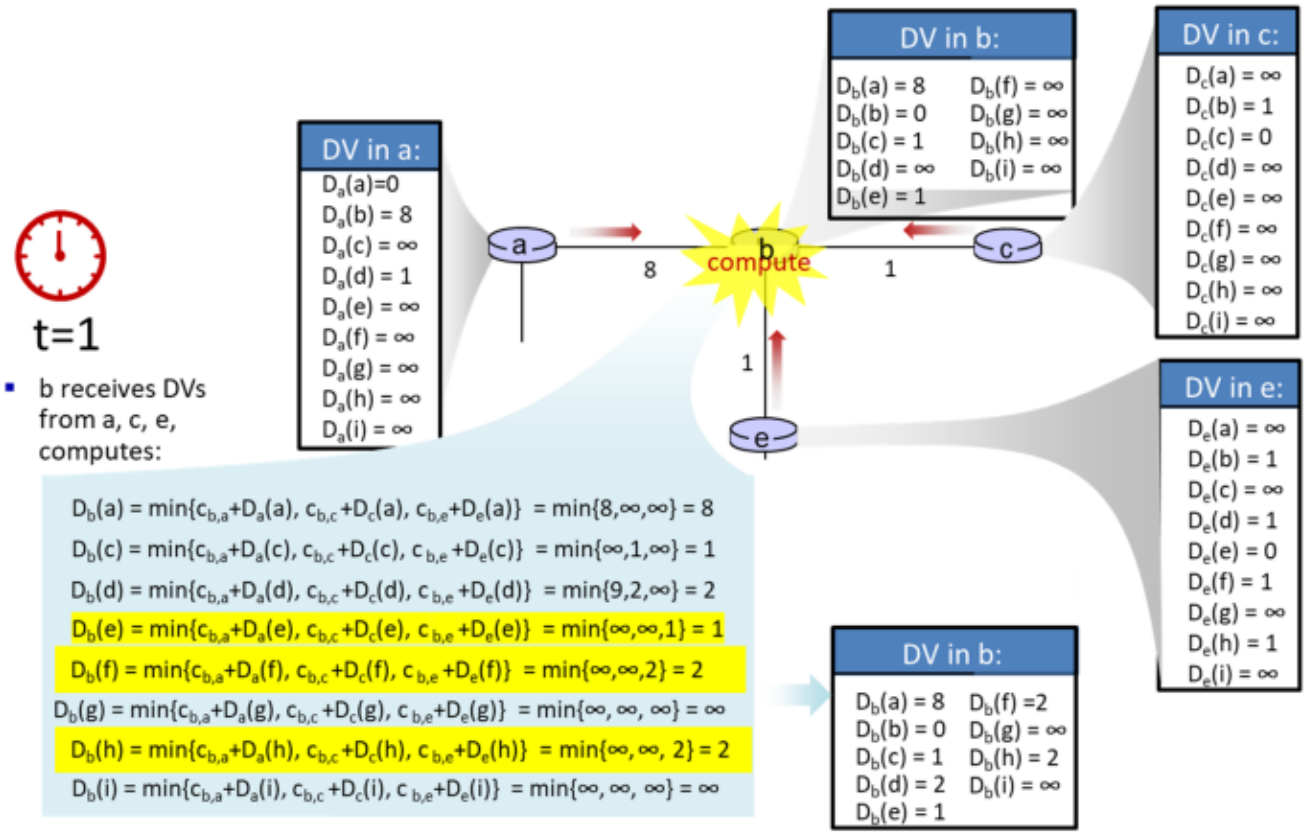
Q16. Consider now a IP data frame size of 2400 bytes to be fragmented into 3 section to support TCP networks. What will be the value of Fragmentation Offset Flag for the 3 sections?

- The fragmentation offset flag is not a value in an IP data frame. It is a bit within the IP header that indicates whether the fragment is the first fragment of a fragmented IP datagram.



- Thus, the fragmentation offsets are:
  - First fragment: 0
  - Second fragment: 100
  - Third fragment: 200

Q17. Consider the example in lecture 6, where the one highlighted in yellow is the portion of the DV table of b that was not sent to e. Please list the portion of the DV table of b not sent to c. **Explain why.**



The optimum route in yellow is NOT sent to **e by b** as optimum route via **e else send to e.**

The portion of the DV table of **b not sent to c**

The optimum route

$D_b(c) = \min\{c_{b,a}+D_a(c), c_{b,c}+D_c(c), c_{b,e}+D_e(c)\} = \min\{\infty,1,\infty\} = 1$

The route via c

$D_b(d) = \min\{c_{b,a}+D_a(d), c_{b,c}+D_c(d), c_{b,e}+D_e(d)\} = \min\{9,2,\infty\} = 2$