Networks, Security, and Privacy

158.235

A/Prof. Julian Jang-Jaccard

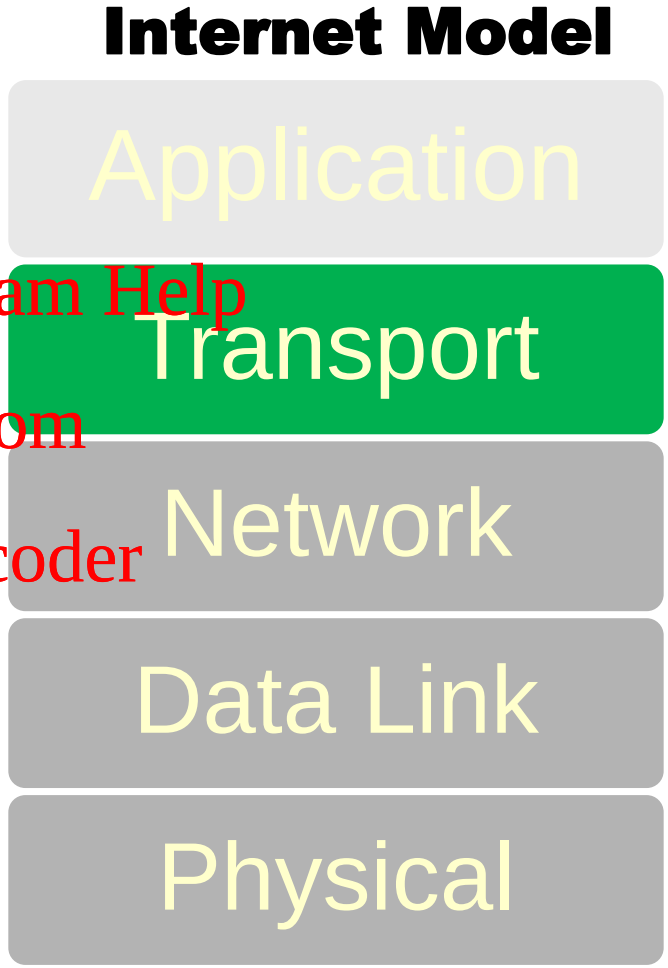
Massey University

Transport Layer

Reading: Chapter 5 in the prescribed textbook

Transport Layer

- Layer 4 in the Internet model
- Main function;
  - Links application and network layers
  - Responsible for segmentation and reassembly
  - Connection Management: end-to-end delivery of messages



Outline

- **Transport layer functions**
  - **Linking to the application layer**
  - **Segmentation**
  - **Connection Management**

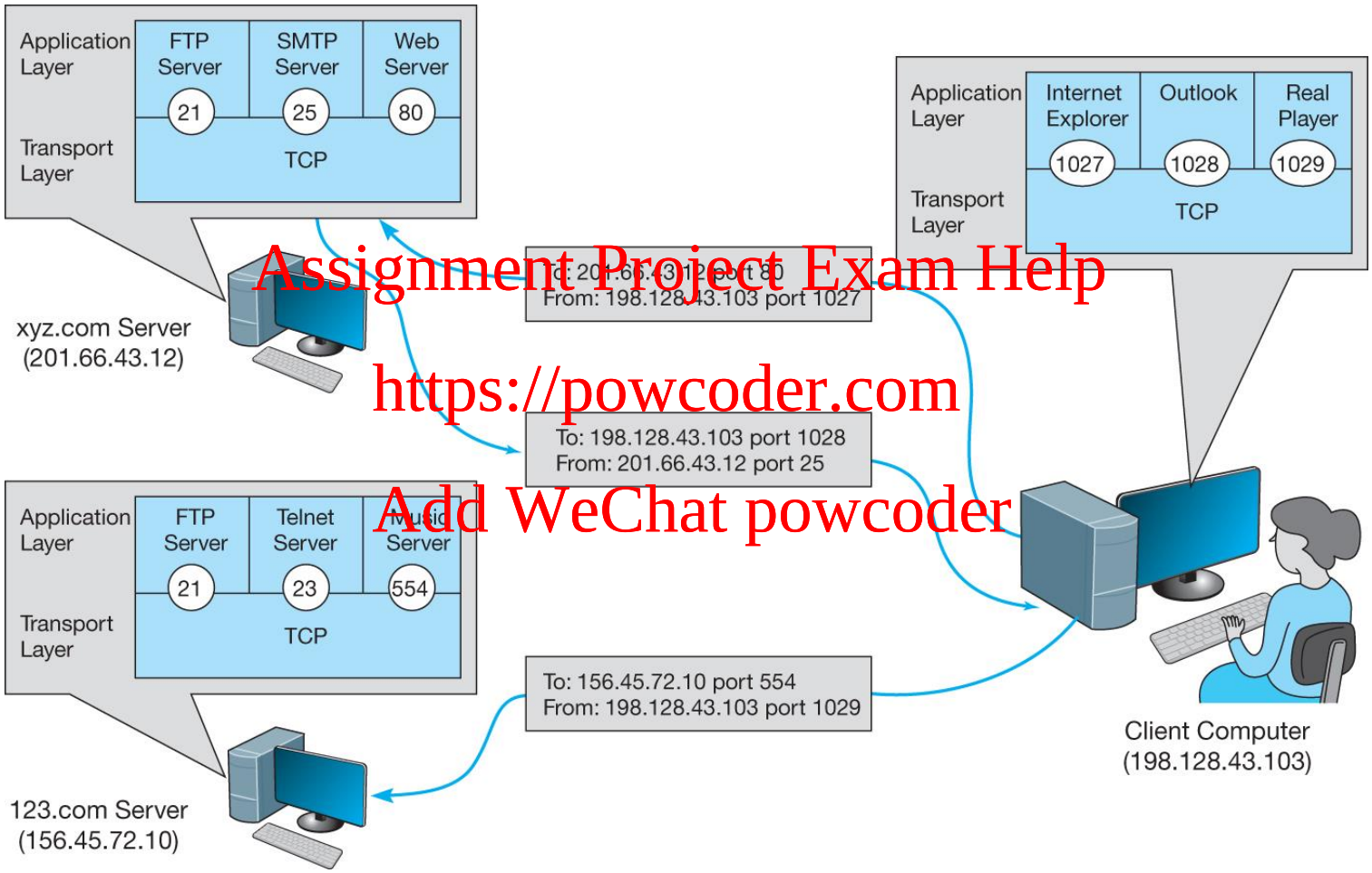
Linking to Application Layer

- **TCP may serve several Application Layer protocols at the same time**
- **Which application layer program to send a message to?**
- **Ports used to identify application (2-byte numbers)**

Linking to the application layer

- **Many source/destination ports follow standards**
  - **Common port standards**
    - **HTTP: TCP port 80**
    - **HTTPS: TCP port 443**
    - **FTP: TCP ports 20 and 21**
    - **SMTP: TCP port 25**
    - **IMAP: TCP port 143**
    - **POP3: TCP port 110 (more commonly TCP port 995 secure version)**
    - **DNS: TCP or UDP port 53 (most commonly UDP)**

Application Layer Services



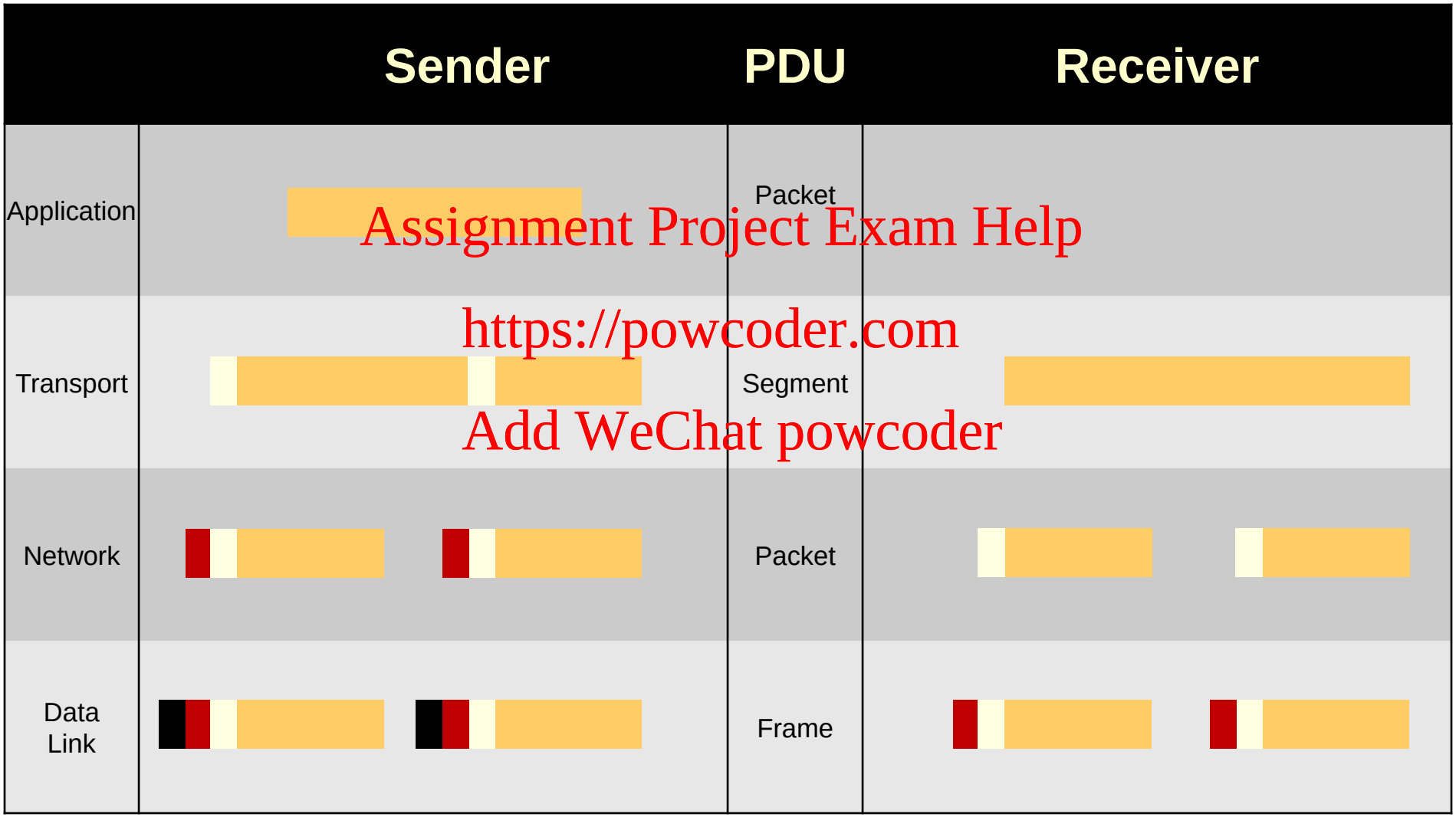
Outline

- **Transport layer functions**
  - **Linking to the application layer**
  - **Segmentation**
  - **Connection Management**

Segmentation

- **Segmenting**
  - **Breaking up large application data into smaller segments (and putting them back together)**
  - **Segments may be passed individually to application layer or after reassembly**
  - **How large are the segments?**
    - **Size depends on the network and data link layer protocols**
    - **Maximum Segment Size (MSS) is negotiated during TCP handshake**

Transport Layer Functions



Outline

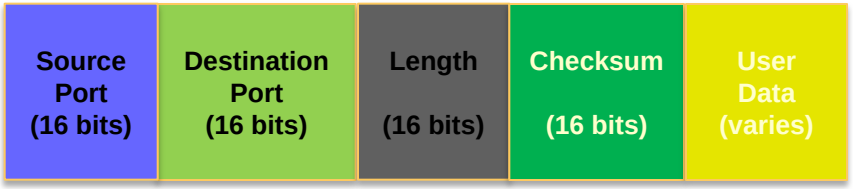
- **Transport layer functions**
  - **Linking to the application layer**
  - **Segmentation**
  - **Connection Management**
    - **Connectionless (UDP)**
    - **Connection-oriented (TCP)**
    - **Quality of Service (QoS)**

Connection Management

- ***Connectionless Routing*** **is provided by UDP**
  - **Sending packets individually without a virtual connection, emphasis on reduced latency over reliability**
  - **Each packet is sent independently of one another, and will be routed separately, following different routes and arriving at different times**
- ***Connection Oriented*** **is provided by TCP**
  - **Setting up a virtual connection, or a TCP connection for a reliable transmission**
    - **Packet deliveries are acknowledged**
    - **Used by HTTP, SMTP, FTP**
- **QoS Routing**
  - **A special kind connection oriented routing with priorities**

User Datagram Protocol (UDP)

- Operates at the transport layer
- PDU called a segment
- Used in time-sensitive situations, for control messages, or when reliability is handled by the application layer
- 32-64 bits (4-8 bytes) of overhead
  - Source port is optional in IPv4 and IPv6, Checksum is optional in IPv4

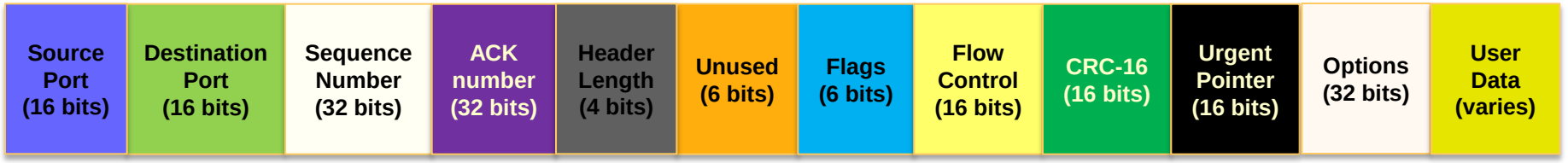


UDP - User Datagram Protocol

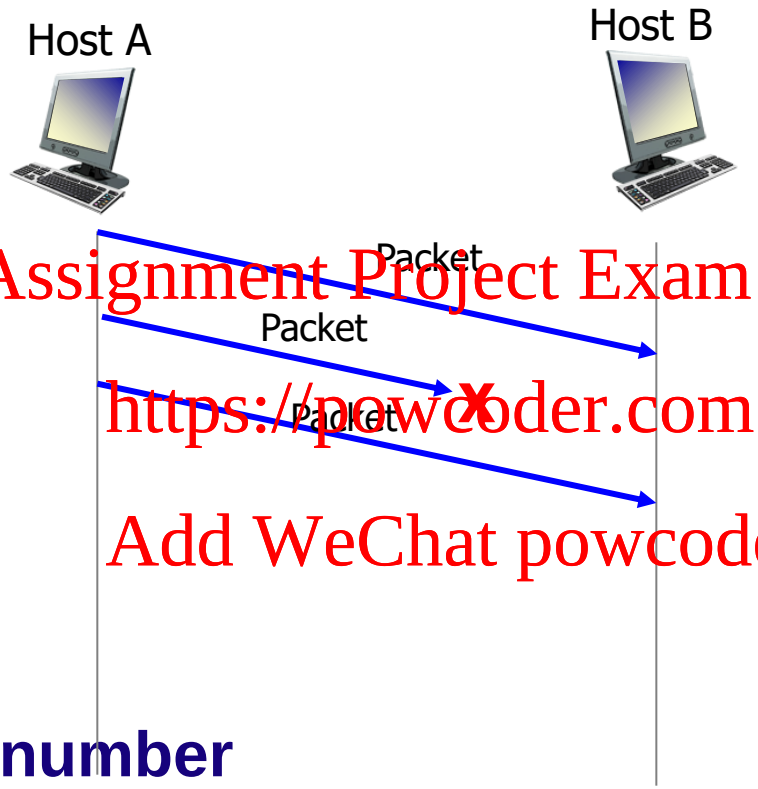
- "No frills", "bare bones" transport protocol
- "Best Effort" service
  - Can be lost or delivered out-of-order to app
- Connectionless messaging
  - No handshaking between UDP sender and receiver
  - Each UDP segment handled independently of others
- UDP: Efficiency before reliability
  - Used in time-sensitive situations, for control messages, or when reliability is handled by the application layer
  - Commonly used for application control messages that are usually small, such as DNS, DHCP, RIP and SNMP
  - Can also be used for applications where a packet can be lost, such as information rich video/audio

Transport Layer Protocols

- **Transmission Control Protocol (TCP)**
  - **Most common transport layer protocol**
  - **PDU called a segment**
  - **Used for reliable transmission of data**
  - **160 - 192 bits (20 -24 bytes) of overhead**
    - **Options field is not required**

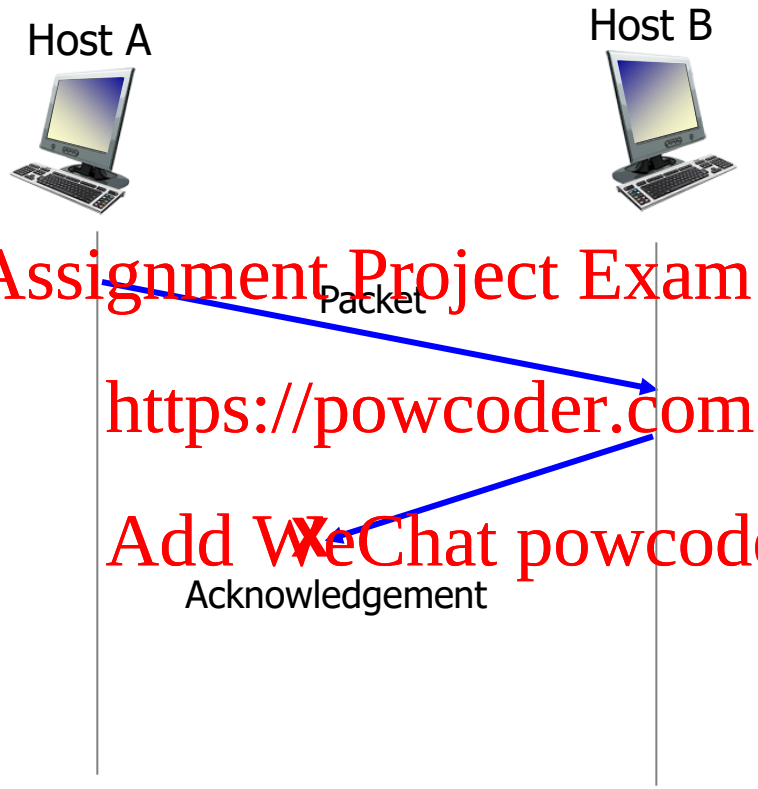


Reliable Data Transfer



- Sequence number
- Acknowledgment
- Retransmission

Reliable Data Transfer

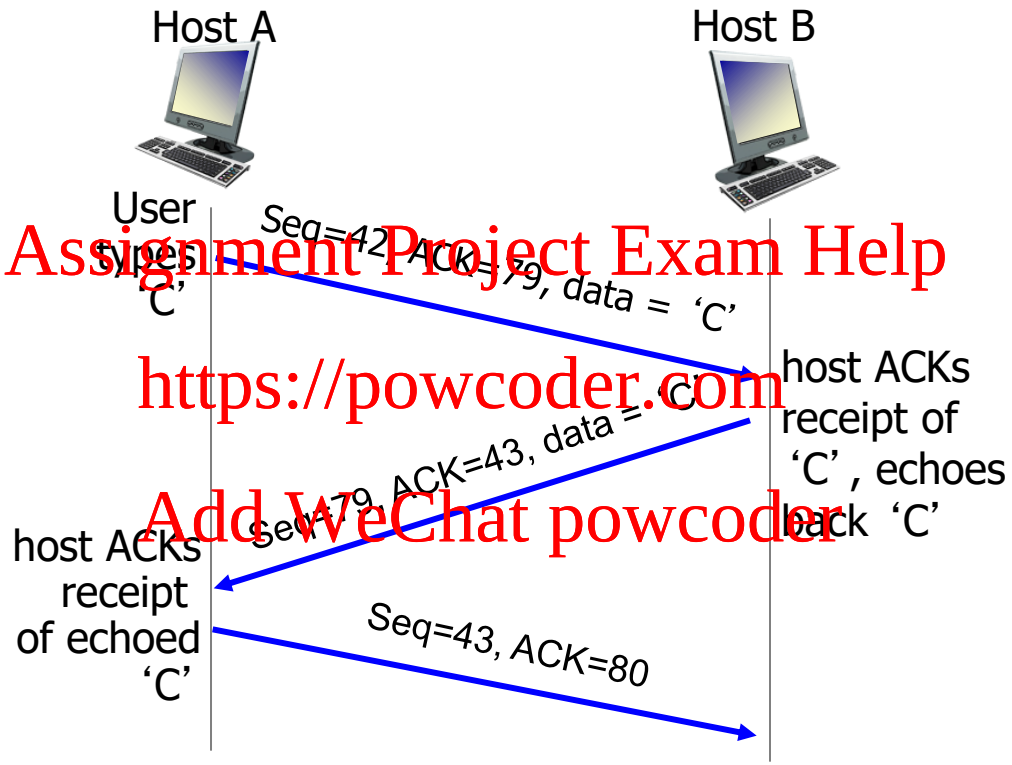


- Timer

Reliable Data Transfer

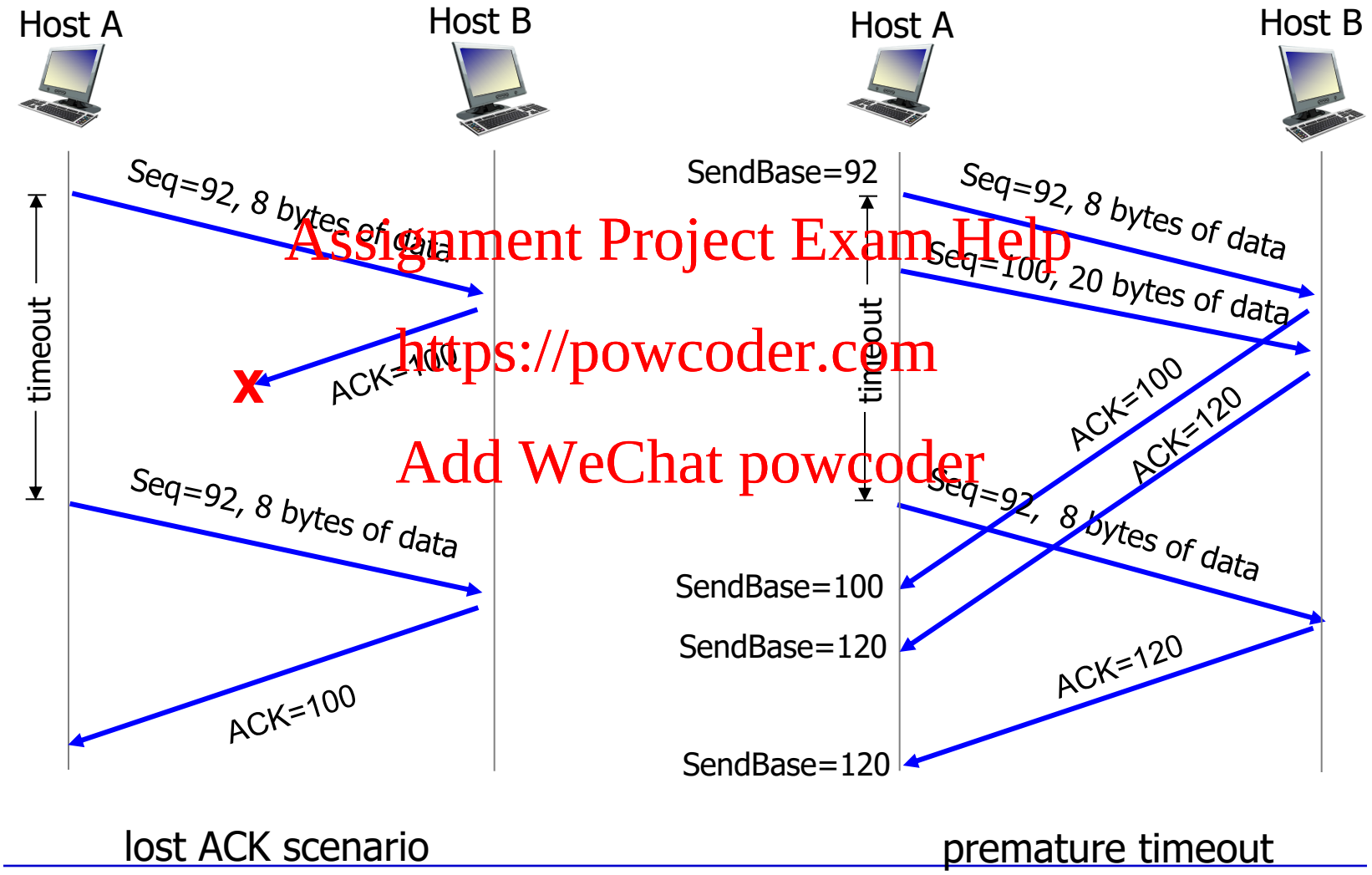
- **Sequence Numbers**
  - byte stream "number" of first byte in segment's data
- **Acknowledgement Numbers**
  - seq # of next byte expected from other side
  - cumulative ACK
- **Timer**
  - Ensure acknowledgement has received within the expected time frame
- **Retransmission**
  - Retransmit the data after timeout

SEQ and ACK

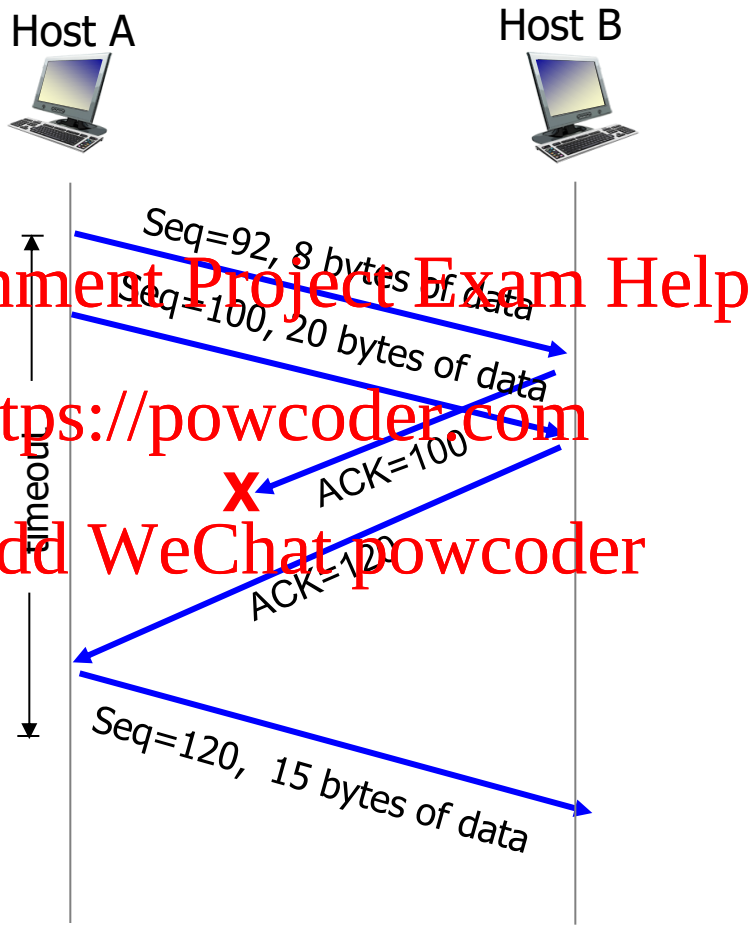


simple telnet scenario

Retransmission



Retransmission



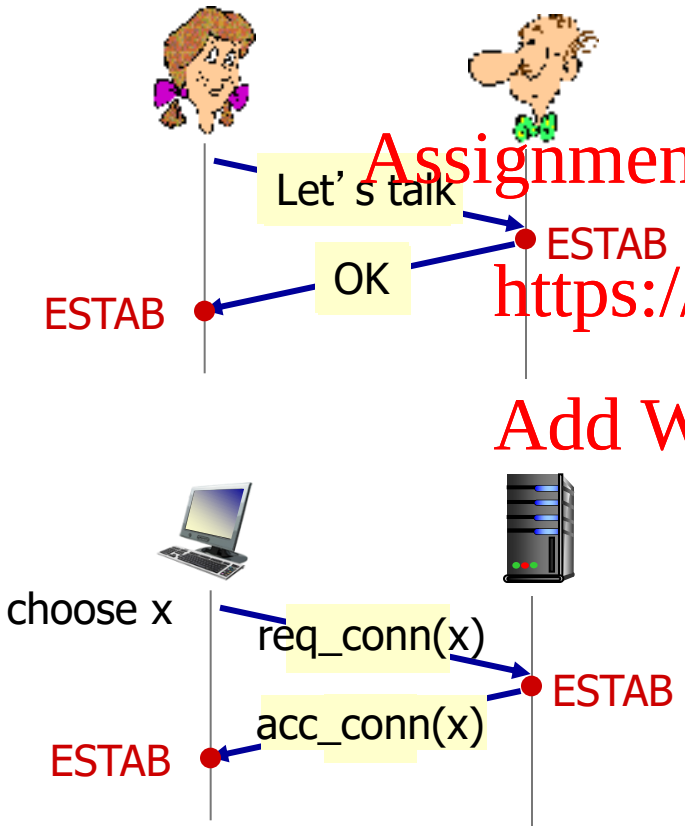
cumulative ACK

Connection Management

- **Before exchanging data, sender/receiver "handshake":**
  - **Agree to establish connection (each knowing the other willing to establish connection)**
  - **Agree on connection parameters**

Establish a connection

2-way handshake:

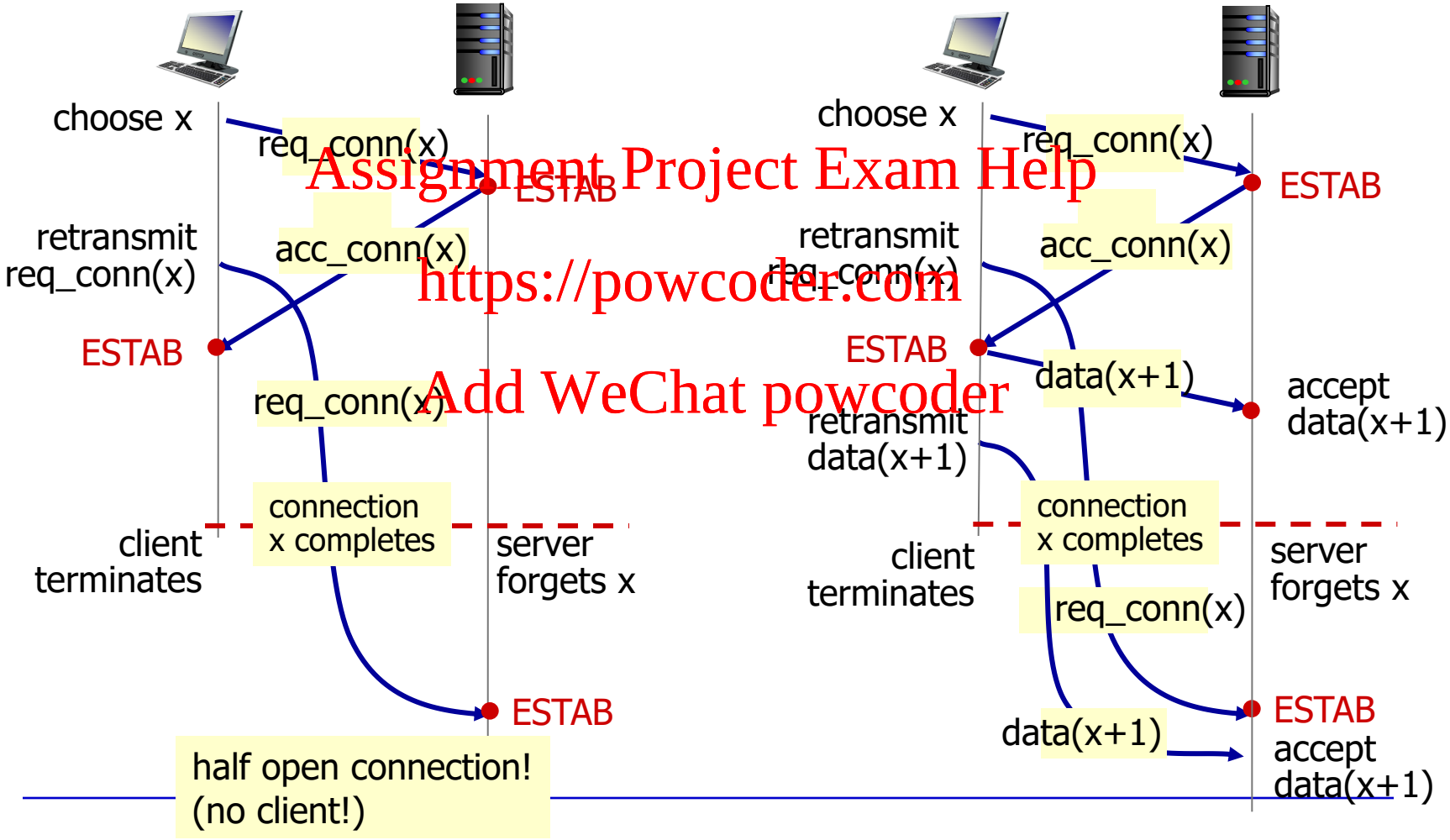


Q: **will 2-way handshake always work in network?**

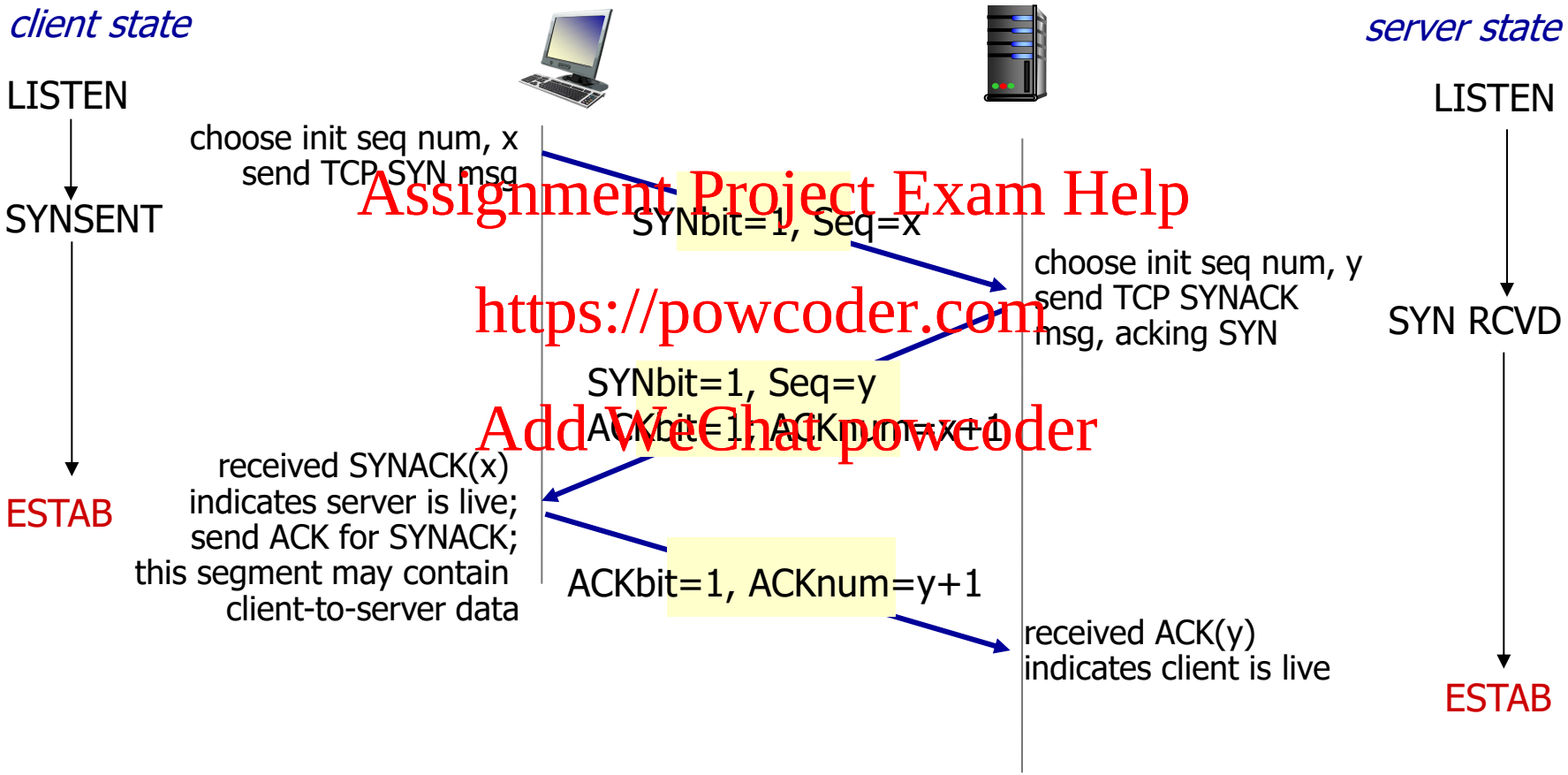
- **can't "see" other side**
- **variable delays**
- **retransmitted messages (e.g. req_conn(x)) due to message loss**
- **message reordering**

Establish a connection

2-way handshake failure scenarios:



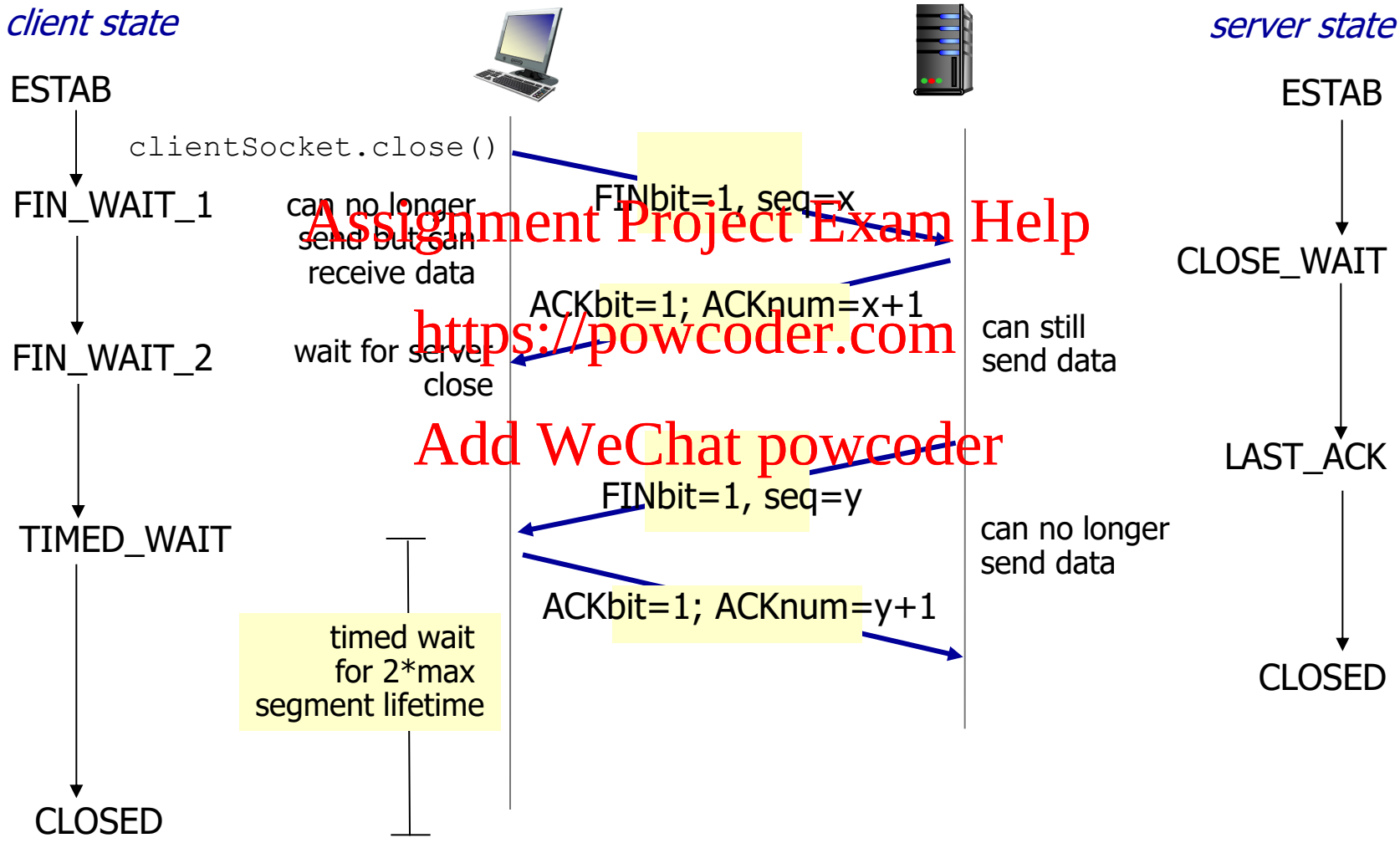
TCP3-way handshake



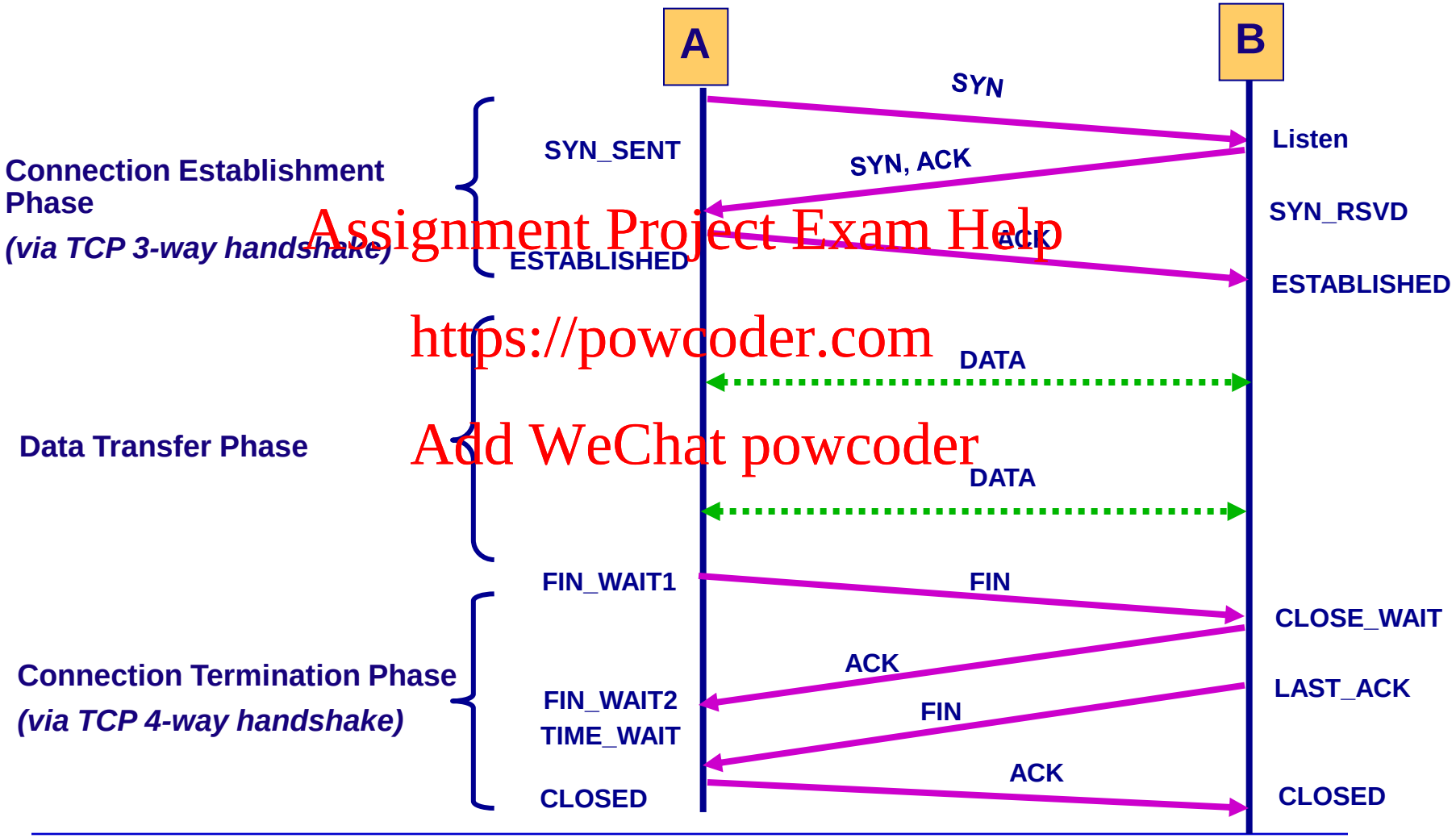
Closing a connection

- client, server each close their side of connection
  - send TCP segment with FIN bit = 1
- respond to received FIN with ACK
  - on receiving FIN, ACK can be combined with own FIN
- simultaneous FIN exchanges can be handled

TCP 4-way handshake



Setting up and Tearing down TCP Connections

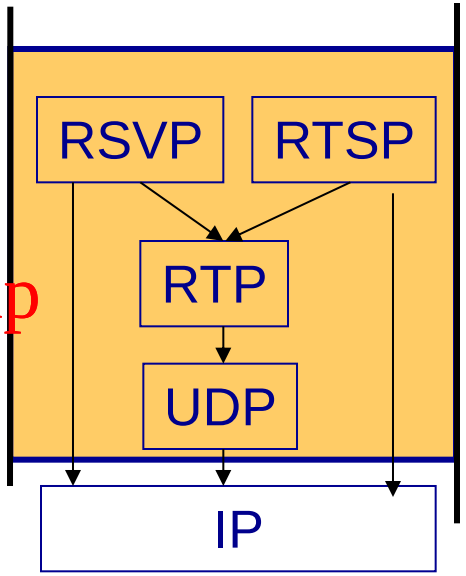


QoS - Quality of Service

- QoS defines and assigns priorities to “classes of service”
- Timeliness - timely delivery of packets
  - Packets be delivered within a certain period of time (to produce a smooth, continuous output)
  - Required by some applications, especially real time applications (e.g., voice and video frames)
- QoS routing
  - Defines classes of service, each with a different priority:
    - Real-time applications such as VoIP- highest
    - A graphical file for a Web page - a lower priority
    - E-mail - lowest (can wait a long time before delivery)

Protocols Supporting QoS

- **TCP/IP protocol suite**
  - **Resource Reservation Protocol (RSVP)**
    - **Sets up virtual circuits for general purpose real-time applications**
  - **Real-Time Streaming Protocol (RTSP)**
    - **Sets up virtual circuits for audio-video applications**
  - **Real-Time Transport Protocol (RTP)**
    - **Used after a virtual connection setup by RSVP or RTSP**
    - **Adds a sequence number and a timestamp for helping applications to synchronize delivery**
    - **Uses UDP (because of its small header) as transport**



END