# ICS343 Subjective Part Summary

Alfaifi, Ammar

## 1 TCP/IP Protocol Suit

- Transmission Control Protocol/Internet Protocol (TCP/IP) is a protocol suit, meaning a set of functionalities provided between each layer.
- It is built in layer structure where the each layer takes services from the one bellow it, and provides services to the above it.
- Application, Transport, Network, Data link, Physical
  - **Physical**: carries individual bits in a frame. communication between two devices in the this layer is still logical. under this layer is the transmission medium; carring electrical or optical signals. Logical unit is *bit*
  - **Data link**: Any protocol that takes datagram and carries it in the link is suffice for this layer. This layer takes datagram encapsulate it into a *frame*. hop-to-hop communication.
  - **Network**: is responsible for creating a connection between the source and destination computers. It's host-to-host communication and routing the packet to best path. It includes the main protocol, Internet Protocol (IP), which defines a packet called *datagram*.
  - IP is connectionless, that provides no flow control, no error control, and no congestion control services. It provides unicast (one-to-one), and multicast (one-to-many) routing protocols. This layer has auxiliary protocols: ICMP helps IP to report some problems when routing a packet, DHCP helps IP to get the network-layer address's for a host, ARP help IP to find the data-link layer address of a host or a router given the network-layer address.
  - **Transport**: the lgical connection is end-to-end. The packet is called a segment or user datagram. It has Transmission Control Protocol (TCP), which is a connection-oriented that established a logical connection before sending data. User Datagram Protocol (UDP), is a connectionless, which is simple and fast.
  - **Application**: Logical connection is end-to-end. They exchange *messages*. The communications is between two *processes*. Examples: HTTP, SMTP, FTP, SSH, DNS is used by other protocols to find the network-layer address of a host.
  - well-known ports : 0 - 1023, controlled by ICANN.
  - registered ports: 1024 - 49151, not controlled by ICANN, but can be registered.
  - dynamic ports: 49152 - 65535, as temporary or private ports.

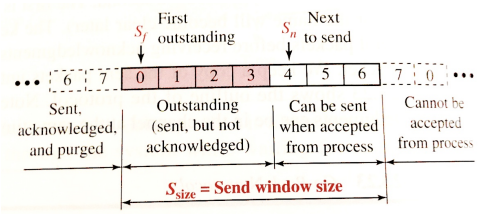
# 2 Transport-Layer Protocol

## 2.1 Stop-and-Wait Protocol

- Both sender and receiver use sliding window of size 1.
- sends one packet at a time, and waits for an ACK.
- every sent packet, a timer starts.
- ACK value is the sequence # of next packet.
- the sequence numbers are: 0, 1, 0, 1, ...

## 2.2 Go-Back-N Protocol (GBN)

- it tries to send many packets while waiting for an acknowledgment.
- ACK # is *cumulative*; the sequence # of next packet.
- window size up to $2^m - 1$.
- $S_f$ first outstanding, $S_n$, $S_{size}$
- from $S_f$ to $S_n - 1$: packets sent but not acknowledged.
- from $S_n$: packets to be sent
- window slides when an error-free ACK, $S_f \leq \text{ackNo} \leq S_n$, received.
- receive window is always of size 1.
- receive window, $R_n$, next packet expected.
- there is only one timer, for the first outstanding packet.
- at the timer expiration, *all* outstanding packets are resent.
- window is full if $S_n = (S_f + S_{size}) \mod 2^m$
- in receiver side, with error-free ACK, $S_f = \text{ackNo}$. if $\text{ackNo} = S_n$ stop timer.




## 2.3 Selective-Repeat Protocol

GBN is inefficient if network loses a lot of packets.

- sender window size is $2^{m-1}$, the receive window should equal the send window
- In reliable protocols, receiver never deliver packets out of order to the application layer.
- ackNo is the sequence # of the received packet; ackNo isn't cumulative. It does **not** mean all packets with sequence # less than ackNo are received.

### 2.4 Bidirectional Protocols: Piggybacking

The idea is that, we make the packets that carries data, to carry also the ackNo at the same time. So that we don't overwhelm the network with packets carring just ackNo.

## 3 HyperText Transfer Protocol

- The server uses the port 80, and 443 for secured one (HTTPS).
- form v1.1 persistent mode is the default.
- In the nonpersistent connection: the client opens a TCP connection and sends a request. The server responds and closes the connection. The client reads and colses the connection.
- Methods in the header message:
  - GET: requests a doc.
  - HEAD: requests info about a doc.
  - POST: sends info to the server.
  - PUT: sends a doc to the server.
- Some request headers:
  - user-agent: the client program.
  - accept: media formats client understands.
  - host: host and port number of client.
  - if-modified-since: last update of file; this tell the server to send the data only if it was modified since this data.
- some response headers:
  - server: info about the server.
  - last-modified: the date of last change.
  - content-type: media type.

## 4 Electronic Mail (Email)

- US: user agent, MTA: message transfer agent, MAA: message access agent.
- US, using MTA client, sends a msg to the MTA server then to a spool (a queue) then to MTA client in the server then to the MTA server of reciever then to the receiver UA mail box them to MAA server then to MAA client.
- it needs 2 UAs, 2 pairs of MTAs (slient & server), and a pair of MAAs.

## 4.1 Message Transfer Agent: SMTP

Some of Simple Mail Transfer Protocol (SMTP) commands:

- HELO: sender's host name
- MAIL FROM: sender of the msg
- RCPT TO: the recipient.
- DATA: message body.
- QUIT: ends the message

## 4.2 Message Access Agent

- Post Office Protocol (POP) is very simple to get the mails form the server mail boxes.
- Internet Mail Access Protocol (IMAP) it can check the header of a msg, search, or partially download a msg.

- Multipurpose Internet Mail Extension (MIME) is to help sending non-textual data.
- Base64 is one of the methods to encode data into ASCII characters. Base64 means $\log_2 64 = 6$ bits, converts each 6 bits into their ASCII chars.
- Note that 0 corresponds to **A**, so **a** is 26 and so on.
- the sequence: **A-Z**, **a-z**, **0-9**