Fundamentals of Networks

Assignment 4

Q1: Simple, compare between each of the following?

- **TCP and UDP protocols :**

Transmission control protocol (TCP)	User datagram protocol (UDP)
TCP is a connection-oriented protocol. Connection-orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.	UDP is the Datagram oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, and terminating a connection. UDP is efficient for broadcast and multicast type of network transmission.
TCP is reliable as it guarantees the delivery of data to the destination router.	The delivery of data to the destination cannot be guaranteed in UDP.
TCP provides extensive error checking mechanisms. It is because it provides flow control and acknowledgement of data.	UDP has only the basic error checking mechanism using checksums.
Acknowledgement segment is present.	No acknowledgement segment.

- **Network and transport layers**
- **Network layer**
  - it delivers message in from of packets from source to destination.
  - It works on transport layer and data link layer.
  - It's path is connection oriented.
  - It is communication between hosts.
- **Transport Layer**
  - It delivers complete message from source to destination.
  - It works on network layer and session layer.
  - It can be both connectionless or connection oriented.
  - It is communication between processes.

- **RDT 2.1 and RDT 2.2**

RDT 2.1

Sender:

1- seq # added to pkt
2- two seq. #'s (0,1) will suffice.
3- must check if received ACK/NAK corrupted
4- twice as many states (state must "remember" whether "current" pkt has 0 or 1 seq. #)

Receiver:

1- must check if received packet is duplicate (state indicates whether 0 or 1 is expected pkt seq #)
2- receiver can *not* know if its last ACK/NAK received OK at sender

RDT 2.2

1- same functionality as rdt2.1, using ACKs only
2- instead of NAK, receiver sends ACK for last pkt received OK (receiver must *explicitly* include seq # of pkt being ACKed)
3- duplicate ACK at sender results in same action as NAK: *retransmit current pkt*

Pipelined protocols and Stop and wait protocols

key	Stop and Wait protocol	Pipelined protocol
Mechanism	In Stop and Wait protocol, sender sends single frame and waits for acknowledgment from the receiver.	In Pipelined protocol, sender sends multiple frames at a time and retransmits the damged frames.
Efficiency	Stop and Wait protocol is less efficient.	Sliding Window protocol is more efficient than Stop and Wait protocol.
Window Size	Sender's window size in Stop and Wait protocol is 1.	Sender's window size in Sliding Window protocol varies from 1 to n.
Sorting	Sorting of frames is not needed.	Sorting of frames helps increasing the efficiency of the protocol.

Q2. Write the definition of each TCP header filed.

1- Source Port: Identifies the port number of a source application program.
2- Destination Port: Identifies the port number of a destination application program.
3- Sequence Number: Specifies the sequence number of the first byte of data in this segment.
4- Acknowledgment Number: Identifies the position of the highest byte received.
5- Data Offset: Specifies the offset of data portion of the segment.
6- Reserved: Reserved for future use.
7- Code: URG: Urgent pointer field is valid.
   - ACK: Acknowledgement field is valid.
   - PSH: Segment requests a PUSH.
   - RTS: Resets the connection.
   - SYN: Synchronizes the sequence numbers.
   - FIN: Sender has reached the end of its byte stream.
8- Window: Specifies the amount of data the destination is willing to accept.
9- Checksum: Verifies the integrity of the segment header and data.

10- Urgent Pointer: Indicates data that is to be delivered as quickly as possible. This pointer specifies the position where urgent data ends.

11- Options (variable length)

Q3. State and example to clarify the bad performance of RDT 3.0

rdt3.0 works, but performance stinks

- rdt3.0 works, but performance stinks
- ex: 1 Gbps link, 15 ms prop. delay, 8000 bit packet:

$$d_{trans} = \frac{L}{R} = \frac{8000\text{bits}}{10^9\,\text{bps}} = 8\,\text{microseconds}$$

- U_{sender}: utilization – fraction of time sender busy sending

$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

- 1KB pkt every 30 msec -> 33kB/sec thruput over 1 Gbps link
- network protocol limits use of physical resources!