HW3

8. p73 In a scenario w/ high latency, such as satellite comms
Ex. A communication link between ground & satellite. When packet loss occurs in such setup, w/o buffering, all subsequent packets (though received) are discarded until the lost one is resent, causing substantial delays due to long trip time. By buffering out of order packets, delays are avoided and retransmits are limited to missing packets.

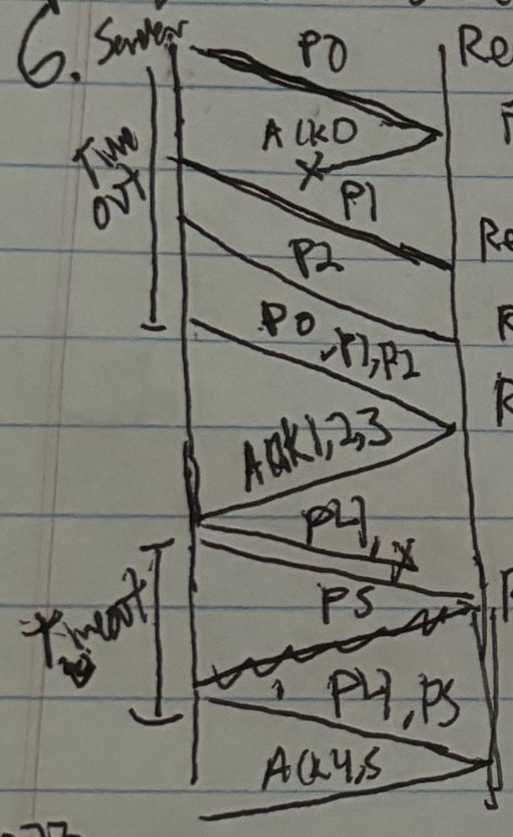
9. p74 A. Yes
B. NO

Homework 3

320

5. p15
RTT * 30ms = 0.03s
Transmission Rate(R) 10 Gbps = 10^{10} bps
Size of Data(L) 1500 bytes * 8 = 12000 bits = 1.2×10^4
Utilization = 0.98
~~Transport = L/R = 12×10^4 / 10×10^7 = .000012 ms~~
~~Beep~~ BDP = $10^{10} * 0.03 = 3 \times 10^8$
Data in transit: $0.98 * 3 \times 10^8 = 2.94 \times 10^8$
Window Size = $2.94 \times 10^8 / 1.2 \times 10^4 = 2.45 \times 10^4$
p19 Window size is 2.45×10^4 packets

6.



p22
7. N = window size = 4
Seq# range = 1024
A. Case 1: Rever is K & ACK all the k-1 packets, Sender window will be in range [K, K+N-1]. Seq # → [K, K+N-1]
Case 2: If sender window ~~range~~ is range = [k-N, k-1], sequence #. Then seq# in the sender's window at time t are in range of [K-N, K] → [k-4, K]
B. Ack Field will be [K-N, K-2]. The sender sent all the K-N packets ACK less than K-N-1 ACK.

HTTP Overv

Homework 3

1. PS
No, not with complete certainty. Checksum will detect if one error occurs, but if the sum of errors is equal to the checksum then it will not detect it.
Ex. If one of the bits was flipped from 0 to 1 in one of the packets, incrementing the sum by 1 & the opposite occurred in a subsequent packet, decreasing by 1 then the checksum will equal the sum but the information will be invalid

2. PG
1. Sender sends packet w/ seq # 1
2. Sender state changes to wait ACK or NAK 1
3. Recver receives the packet 1 & returns the ACK
4. The recver state is "wait for next packet"
5. Sender reves a CORRUPT ACK
6. Sender that reved corrupt ACK, it transitions the packet 1 again
7. Recver sends NACK since it rcved Seq#1 instead of packet 0
8. Sender sends packet w/ seq#1
9. Recver sends NAK (loop of 7.) (Creating a deadlock)

3. PG2
RDT3.0 is used to transport data from sender to reciever when premature timeouts occur. Sender will retransmit all of the following packets, including the current packet. This will generate a waiting time on the remaing packets until a particular (prior) packet is delivered.

4. PG3
If used UDP, congestion control would have to be handled at app level. There is not much that can be done at transport layer with UDP. With TCP, there is congestion control algorithms that can help. QUIC is an alternative that is used WITH UDP, on the app layer

HTTP Overview

HTTP response message
status line (protocol ⟶ HTTP/1.1 200