Chapter 3 **Siyuan Feng (516030910575)**
CS 391 Computer Networking ACM Class, Zhiyuan College, SJTU
Prof. **Yanmin Zhu** Due Date: October 25, 2018
TA. **Haobing Liu** Submit Date: October 25, 2018

P3

$$\begin{array}{cr} & 01010011 \\ + & 01100110 \\ + & 01110100 \\ \hline = & 100101101 \end{array}$$

Wrap around the extra bit.

$$\begin{array}{cr} & 00101101 \\ + & 1 \\ \hline = & 00101110 \end{array}$$

Then, invert all the bits to get the check sum. Hence, the check sum is 11010001.

The receiver adds the four words (the three original words and the checksum). If the sum contains a zero, the receiver knows there has been an error.

All one-bit errors will be detected. But two-bit error may be undetected. (e.g., if three bits convert to 01010010, 01100111 and 01110100)

P7

To best answer this question, consider why we needed sequence numbers in the first place. We saw that the sender needs sequence numbers so that the receiver can tell if a data packet is a duplicate of an already received data packet. In the case of ACKs, the sender does not need this info (i.e., a sequence number on an ACK) to tell detect a duplicate ACK. A duplicate ACK is obvious to the rdt3.0 receiver, since when it has received the original ACK it transitioned to the next state. The duplicate ACK is not the ACK that the sender needs and hence is ignored by the rdt3.0 sender.

P10

Since, knowing the maximum delay, we set a timeout at $WaitforNAKat0$ and $WaitforNAKat1$. If the sender dose not receive ACK or NAK within a certain time, we assume that the packet has lost and send the packet again.

P17

P33

Suppose that the sender sends packet $P1$ and retransmitted packet $P2$ with the timer for $P1$.Further more, after $P2$ sent away, the sender received the acknowledge for $P1$. However, the sender will take mistake this acknowledge for $P1$, and calculate a wrong SampleRTT.

P40

(a) TCP slowstarts at intervals $[1, 6]$ and $[23, 26]$.

(b) TCP congestion avoidance at intervals $[6, 16]$ and $[17, 22]$.

(c) After the 16^{th} round, packet loss by recognizing a triple duplicate ACK. If there is a timeout, the window size would reduce to 1.

(d) After the 22^{th} round, segment loss due to timeout, and window size would reduce to 1.

(e) The threshold is set to initially 32

(f) The threshold is set to half value of the congestion window when packets loss. When loss appeared at round 16, the congestion window size is 42 and the threshold is 21.

(g) When loss appeared at round 22, the congestion window size is 29 and the threshold is 14.

(h) Hence, packet 70 is in 7th round.

packet	round
1	1
2 - 3	2
4 - 7	3
8 - 15	4
16 - 31	5
32 - 63	6
64 - 95	7

(i) The new value of threshold and window will be 4 and 7.

(j) Threshold is 21 and window size is 1.

(k) Total number is 52.

round	packet number
17	1
18 - 3	18
19 - 7	19
20 - 15	20
21 - 31	16
22 - 63	21

P42

TCP uses pipeline method, that is sender is able to send multiply segment. The doubling of the timeout interval prevent a sender from retransmit too many packets.

P50

Time(msec)	Window Size of C1	Speed of C1 (wins / 0.05)	Window Size of C2	Speed of C2 (wins / 0.1)
0	10	200	10	100
50	5	100		100
100	2	40	5	50
150	1	20		50
200	1	20	2	20
250	1	20		20
300	1	20	1	10
350	2	40		10
400	1	20	1	10
450	2	40		10
500	1	20	1	10
550	2	40		10
...	...	...	...	...

No, in long time, sending rate of $C1$ is $(40 + 20 + 40 + 20) = 120$ and sending rate of $C2$ is $(10 + 10 + 10 + 10) = 40$