CECS 474 - Homework 3

Solution

Created by: Haixia Peng

Problem 1:

```
   1 0 1 1 0 1 0 0
+  0 1 1 0 1 1 1 0
=1 0 0 1 0 0 0 1 0
+  0 0 0 0 0 0 0 1
=  0 0 1 0 0 0 1 1
```

Thus, the one's complement of the sum = 11011100.

Problem 2:

DATA Packet size: 840bytes=6720bits

ACK packet size 40bytes=320bits

P = propagation delay = 8ms

C= link data rate

Assume no transmission errors, negligible framing overhead**,** and negligible processing time p.

a) With the Stop-and-go protocol:

T = d + 2P + a + p = d + 2P + a (p is the processing time, assumed negligible)

T: Time to send a data frame and receive the corresponding ACK

d: Time to transmit a data frame

P: Time for the frame to propagate from the sender to the receiver

a: Time for the receiver to transmit an ACK

$$\text{Throughput} = \frac{1}{T} \text{ (packet/second)}$$

$$\text{Efficiency} = \frac{d}{T} \text{ (no units)}$$

C	d	a	P	T	Throughput	Efficiency
1kbps	6.72sec	320msec	8ms	7.056 sec	0.142pkts/s	95%

100kbps	67.2 msec	3.2 msec	8ms	86.4 msec	11.6 pkts/s	77.8%
10Mbps	0.672 msec	0.032msec	8ms	16.7 msec	59.9 pkts/s	4.0%
1Gbps	6.72 µsec	0.3 µsec	8ms	16 msec	62.5 pkts/s	0.042%

b) With a sliding window protocol under the same assumptions, assume the window size is W

$$\text{Efficiency} = \min\left[\left(\frac{W\times d}{T}\right), 1\right]$$

(the throughput cannot be larger than C (C and throughput should be in the same units).

C	Window size	T	Throughput	Efficiency
1Gbps	10	16 msec	625 pkts/s	0.42%
1Gbps	100	16 msec	6.25 kpkts/s	4.2%
1Gbps	1000	16 msec	62.5 kpkts/s	42%
1Gbps	10000	16 msec	148.8 kpkts/s	100%

The ideal window is: $W_i = \frac{T}{d} = 2381$ frames

Problem 3:

No, the receiver cannot be absolutely certain that no bit errors have occurred. This is because of the manner in which the checksum for the packet is calculated. If the corresponding bits (that would be added together) of two 16-bit words in the packet were 0 and 1 then even if these get flipped to 1 and 0 respectively, the sum still remains the same. Hence, the 1s complement the receiver calculates will also be the same. This means the checksum will verify even if there was transmission error.

Problem 4:

Assume the processing delays and queuing delays for both ends are negligible.

Given: Propagation speed in coaxial cable is $2\cdot 10^8\, m/\sec$

a) The one way propagation delay, P is:

$$P = \frac{5Km}{2\cdot 10^5\, Km/\sec} = 25\mu\sec$$

The transmission delays for ACK, a and for data packet, d are the same:

$$a = d = \frac{1500 bits}{10 \cdot 10^6 bits/\sec} = 150 \mu\sec$$

$$T = d + 2P + a = 0.35 \text{ ms}$$

$$\text{Throughput} = \frac{1}{T} = 3077 \text{ packets/s}$$

b) With no errors on the links, the throughput for the 5 links in series (corresponding to 5 independent ABP protocols) is $\frac{1}{T} = 3077$ packets/s, because the first packet will be completely received by the first router after P+L/C and could then be sent immediately to the second router that would receive it after 2P+2L/C. If we have 5 links, we have a sender, a receiver and 4 routers in the middle. Hence the first packet would be received at 5P+5L/C. The second packet could leave the sender at T (assuming that the first packet left at t=0) and would arrive at T+P+L/C just in time at the first router to be sent to the second router (since all the links are identical). Hence the second packet would arrive at the receiver exactly T seconds later. Hence the throughput is exactly 1 packet every T second.

When p = 0.01, each link's throughput will be multiplied with (1-p) which is the probability that no errors occur for a PDU and its ACK. Now the end-to-end throughput is just like before the same as the throughput on one link and hence:

$$\text{Throughput}_{\text{end-to-end}} = \frac{(1-p)}{T} \approx 3046 \, packets/s \quad \text{when } p = 0.01.$$

For a 25 Km link, the round trip delay for sending one PDU and receiving its ACK is:

$$T25Km = a + d + 2P25Km = 150\mu\sec + 150\mu\sec + \frac{2 \cdot 25 Km}{2 \cdot 10^5 Km} = 550 \mu\sec$$

Hence if no error occurs:

$$\text{Throughput}25Km = \frac{1}{T_{25Km}} \approx 1818 \, Packets/s \, .$$

Comparing Throughput25Km with Throughputend-to-end, we can see that multiple ABP in series provides a better throughput than a long ABP as long as each segment is identical (i.e., same length and same C).

Problem 5:

a. True. Suppose the sender has a window size of 3 and sends packets 1, 2, 3 at t0. At t1 (t1 > t0), the receiver ACKS 1, 2, 3. At t2 (t2 > t1), the sender times out and resends 1, 2, 3. At t3, the receiver receives the duplicates and re-acknowledges 1, 2, 3. At t4, the sender receives the ACKs that the receiver sent at t1 and advances its window to 4, 5, 6. At t5, the sender receives the ACKs 1, 2, 3 the receiver sent at t2. These ACKs are outside its window.
b. True. By essentially the same scenario as in (a).
c. True.
d. True. Note that with a window size of 1, SR, GBN, and the alternating bit protocol are functionally equivalent. The window size of 1 precludes the possibility of out-of-order packets (within the window). A cumulative ACK is just an ordinary ACK in this situation, since it can only refer to the single packet within the window.

Problem 6:

a) Consider sending an application message over a transport protocol. With TCP, the application writes data to the connection send buffer and TCP will grab bytes without necessarily putting a single message in the TCP segment; TCP may put more or less than a single message in a segment. UDP, on the other hand, encapsulates in a segment whatever the application gives it; so that, if the application gives UDP an application message, this message will be the payload of the UDP segment. Thus, with UDP, an application has more control of what data is sent in a segment.

b) With TCP, due to flow control and congestion control, there may be significant delay from the time when an application writes data to its send buffer until when the data is given to the network layer. UDP does not have delays due to flow control and congestion control.

Problem 7:

There are 2^{32} = 4,294,967,296 possible sequence numbers.

a) The sequence number does not increment by one with each segment. Rather, it increments by the number of bytes of data sent. So the size of the MSS is irrelevant -- the maximum size file that can be sent from A to B is simply the number of bytes representable by 2^{32} = 4.29 Gbytes.

b) The number of segments is $\left\lceil \frac{2^{32}}{536} \right\rceil = 8{,}012{,}999$. 66 bytes of header get added to each segment giving a total of 528,857,934 bytes of header. The total number of bytes transmitted is $2^{32} + 528{,}857{,}934 = 4.824 \times 10^{9}$ bytes. Thus it would take 249 seconds to transmit the file over a 155~Mbps link.

Problem 8:

a) It takes 1 RTT to increase CongWin to 7 MSS; 2 RTTs to increase to 8 MSS; 3 RTTs to increase to 9 MSS; 4 RTTs to increase to 10 MSS; 5 RTTs to increase to 11 MSS; 6 RTTs to increase to 12 MSS.

b) In the first RTT 6 MSS was sent; in the second RTT 7 MSS was sent; in the third RTT 8 MSS was sent; in the fourth RTT 9 MSS was sent; in the fifth RTT, 10 MSS was sent; and in the sixth RTT, 11 MSS was sent. Thus, up to time 6 RTT,

6+7+8+9+10+11 = 51 MSS were sent. Thus, we can say that the average throughput up to time 6 RTT was (51 MSS)/(6 RTT) = 8.5 MSS/RTT.