

CS 164 COMPUTER NETWORKS

Assignment 4 -- Answers

- Answer the following 4 questions from your textbook. Make sure your calculations are organized; provide explanations where needed.
- Breakdown of points will be 25% for each question.
- Reading Assignment: Complete reading of Chapter 2, and read Chapter 4 through 4.1 to 4.3.

1. Exercise #40

Hosts sharing the same address will be considered to be the same host by all other hosts. Unless the conflicting hosts coordinate the activities of their higher level protocols, it is likely that higher level protocol messages with otherwise identical demultiplexing information from both hosts will be interleaved and result in communication breakdown.

2. Exercise #47

Many possible scenarios are valid for this example. One possible solution is as follows: The probability of four collisions appears to be quite low. Events are listed in order of occurrence.

A attempts to transmit; discovers line is busy and waits.

B attempts to transmit; discovers line is busy and waits.

C attempts to transmit; discovers line is busy and waits.

D finishes; A, B, and C all detect this, and attempt to transmit, and collide. A chooses random backoff interval $k(A)=1$, B chooses $k(B)=1$, and C chooses $k(C)=1$.

One slot time later A, B, and C all attempt to retransmit, and again collide. A chooses $k(A)=2$, B chooses $k(B)=3$, and C chooses $k(C)=1$.

One slot time later C attempts to transmit, and succeeds. While it transmits, A and B both attempt to retransmit but discover the line is busy and wait. C finishes; A and B attempt to retransmit and a third collision occurs. A and B back off and (since we require a fourth collision) once again happen to choose the same $k < 8$. A and B collide for the fourth time; this time A chooses $k(A)=15$ and B chooses $k(B)=14$.

14 slot times later, B transmits. While B is transmitting, A attempts to transmit but sees the line is busy, and waits for B to finish.

3. Exercise #57

At 4Mbps it takes 2ms to send a packet. A single active host would transmit for 2000 μ s and then be idle for 200 μ s as the token went around; this yields an efficiency of $2000/(2000+200)= 91\%$. Note that, because the time needed to transmit a packet exceeds the ring latency, immediate and delayed release here are the same.

At 100Mbps it takes $82 \mu\text{s}$ to send a packet. A single host would send for $82 \mu\text{s}$, then wait a total of $200 \mu\text{s}$ from the start time for the packet to come round, then release the token and wait another $200 \mu\text{s}$ for the token to come back. Efficiency is thus $82/400 = 20\%$. With many hosts, each station would transmit about $200 \mu\text{s}$ apart, due to the wait for the delayed token release, for an efficiency of $82/200 = 40\%$.

4. Exercise #4

Consider the first network. Packets have room for $1024 - 14 - 20 = 990$ bytes of IP-level data; because 990 is not a multiple of 8 each fragment can contain at most $8 * \lfloor 990 / 8 \rfloor = 984$ bytes. We need to transfer $2048 + 20 = 2068$ bytes of such data. This would be fragmented into fragments of size 984, 984, and 100.

Over the second network (which by the way has an illegally small MTU for IP), the 100-byte packet would be unfragmented, but the 984-data-byte packet would be fragmented as follows. The network+IP headers total 28 bytes, leaving $984 - 28 = 956$ bytes for IP-level data. Again rounding down to the nearest multiple of 8, each fragment could contain 952 bytes of IP-level data. 956 bytes of such data would become fragments with data sizes 476, 476, and 4.