Homework 3

Due: <u>Monday, March 1</u> at the beginning of class
CSE/EE 461, Winter 2004

1. IP Fragmentation (5 points). Peterson 4.7.

2. Sliding Window and Ethernet (5 points). Suppose a node is using sliding window to send a large to another node on the same Ethernet segment. Explain why nearly every packet send will result in a collision. How can the sending node avoid this problem?

3. Media Access (10 points). Your startup, skylab.com, has the brilliant idea to use rooftop satellite TV dishes as the basis of 2-way Internet service. Skylab's service is broadcast in the downward direction (all the ground stations can hear the satellite), and unicast in the upward direction. Ground stations can only talk to the satellite – they can't hear each other since the dishes are all on rooftops pointing upward! But, if more than one ground station transmits at the same time, the satellite will not be able to correctly receive the packet. None of the stations (ground or sky) can send and receive simultaneously. The propagation delay from ground to sky (and vice versa) is relatively large, about 200 milliseconds round trip. The bandwidth available is also large, around 100 Mb/s.

Your task is to design a media access protocol that efficiently arbitrates access to the shared media, enabling the ground stations and/or satellite to individually and/or collectively send at or near the full bandwidth available. You may assume end-to-end error recovery – no need to include hop-by-hop error recovery unless it will help performance for your design. You should not make assumptions about the traffic – the network needs to be designed to be efficient at sending traffic in any and all directions.

To get started, you may find it helpful to think about how this network compares to Ethernet and to wireless networks.

Many designs are possible. However, you should strive for a design that is as efficient as possible at statistically multiplexing the available bandwidth.

a) What protocol does your system use for communicating downward – from satellite to ground?

b) What protocol does your system use for communicating upward – from ground to satellite?

4. TCP Congestion Control (10 points). This question concerns the factors that affect TCP performance when the network bandwidth is not the bottleneck. We define the following constants:

RTT = round trip time, in seconds
MSS = TCP maximum segment (packet) size (in bytes)
P = packet loss rate ($0 \leq P \leq 1$)
S = file size in maximum-sized TCP segments
W = advertised receive window size (in bytes)

You should make the following assumptions:

- The path MTU is larger than the MSS. (That is, there are no ICMP errors or fragmentation.)
- The network bandwidth is large enough that the packet loss rate is independent of the behavior of the sending node (e.g., its packets are dropped because of the behavior of other nodes).
- The round trip time is independent of the packet size. (That is, propagation delay is much larger than transmission delay.)
- The round trip time is constant over time. (Actually, it will vary somewhat due to queuing delay as the network becomes congested, but that is a second-order effect.)

a) TCP small file performance. Devise an approximate formula, using the constants above and any others that you may find necessary or convenient, that specifies the latency of a small web transfer, from the time that the user clicks on a link in a web browser, until the last byte of the requested web page arrives at the client machine. (Note: the client web request fits into a single packet; the reply fits in S packets.) Be sure to explain your reasoning. You may find it helpful to draw a time-sequence diagram.

For this sub-problem, you should assume that P = 0 and W >= S*MSS (that is, TCP stays in slow start throughout the transfer). However, you should model as carefully as possible all other TCP effects on small file performance, including the initial three-way handshake, initial window size, slow start, etc. You do not need to model TCP's delayed acknowledgements (which are mentioned on p. 396 of the text, but not described in detail). Note that the TCP rule during slow start is to add an additional packet to the existing congestion window size whenever a successful acknowledgement is received.

b) TCP large file performance. Devise an approximate formula, again using the constants above and any others that you may find necessary or convenient, that specifies the steady-state bandwidth of a (very!) large file transfer, given a specified background loss rate. Be sure to explain your reasoning.

Note that in steady state, TCP will increase its window by one (maximum-sized) packet every time an entire window is successfully transferred, and decrease its window by half every time a packet is lost. You should assume TCP fast recovery – that is, that the loss event is immediately detected, and transfers are resumed without delay from that point. (Hint: the integral over one iteration of the TCP "sawtooth" defines how many packets are transferred between loss events, and therefore must equal the inverse of the loss rate.)