15-744 Computer Networks (Fall 2010)
Homework 1

Due: Sep. 27th, 2010, 3:00PM (in class)

Name:
Andrew ID:

September 28, 2010

1. Ethernet has a minimum packet size to guarantee that collisions are detected. This problem considers an Ethernet-like medium to explore how the minimum packet size is chosen.

   (a) How does a minimum packet size help detect collisions?

   Solution: A minimum packet size assures that a sender is transmitting for long enough to detect that its packet has collided with another host on the network. A host must be sending in order to detect a collision, because collisions are detected by comparing data transmitted from and received on the wire at the source.

   Therefore, a sender must be transmitting long enough to handle the worst case scenario that another host on the far end of the shared link begins transmitting right before it hears the bits from the other sender. The signal representing the collision will then have to propagate back down the wire back to the sender and reach it before it finishes transmitting.

   (b) Assume the speed of propagation through copper Ethernet wire is $2 * 10^8$ m/s . If the maximum size of your network is 500 m, and you transmit data at 100 Mbits/s what is the minimum packet size needed to detect collisions?

   Solution: The max one-way propagation time for this network is $\frac{500m}{2*10^8 m/s} = 2.5 * 10^{-6} s$
   Thus, min packet size is: $100 * 10^6 b/s * 2 * (2.5 * 10^{-6} s) = 500$ bits

   (c) Being a brilliant CMU graduate, you figure out how to create a Ethernet wire that carries data at the speed of light ($3 * 10^8$ m/s). What is the new minimum packet size for your network?

   Solution: $\frac{500\ m}{3*10^8\ m/s} = 1.66 * 10^{-6}\ s$
   $100 * 10^6\ b/s * 2 * (1.66 * 10^{-6}\ s) = 333\ bits$

   (d) Your boss wants you to upgrade the network from 100 Mbits/s to Gigabit (1000 Mbits/s). What are the two ways that you can change the network to handle this new speed.

   Solution: Increase the minimum packet size by 10 times the old value, or decrease the maximum network size by 10 times.

2. An Autonomous System (AS) claims that it "owns" an IP CIDR block such as 128.2.0.0/16 by advertising a path to 128.2.0.0/16 that has only the single number in its AS Path. For example, CMU's routers claim to own the block 128.2.0.0/16 by advertising a route to this block with AS Path 9. There is nothing, however, to prevent two different Autonomous Systems from both claiming to own the same IP address, although one of the routers must be mistaken.

   (a) Explain why even if this happens, the BGP protocol prevents cycles from forming in BGP routing tables.

   Solution: BGP is a path vector protocol and it sends the sequence of ASs for a route in route announcements. This mechanism prevents an AS from showing up multiple times in a route and thus the formation of loops.

   (b) Suppose that a router in AS 9 (Carnegie Mellon) mistakenly claims that it owns CIDR block 18.0.0.0/8 (M.I.T.) by advertising a route to 18.0.0.0/8 with AS Path 9. What are the consequences of this? Recall that one of the principles of BGP is that a router should only advertise the path that it considers the best path to a destination.

   Solution: Some traffic to MIT might be forwarded to CMU, where it is dropped.

   (c) Suppose AS 9 thinks that AS 3 drops too many packets. Using only BGP, is it possible for AS 9 to implement a policy stating that "traffic outbound from my AS should not cross AS 3?" Why or why not?

   Solution: Yes, assuming it has an alternative route. Prefer paths that don't contain AS 3. Of course, if the only path to a destination contains AS 3, it can not reach that destination without going through AS 3.

   (d) Now suppose AS 9 thinks that AS 3 generates a lot of illegal file sharing traffic. Using only BGP, is it possible for AS 9 to implement a policy stating that, "I don't want to carry traffic from 3 to my customers?" Why or why not? Assume that AS 9 does not want to deny transit to traffic from any other AS.

   Solution: Not in general. Traffic from a neighbor might be from both good ASs and AS 3. BGP can only accept all the traffic by advertising a route or deny all of it by not advertising.

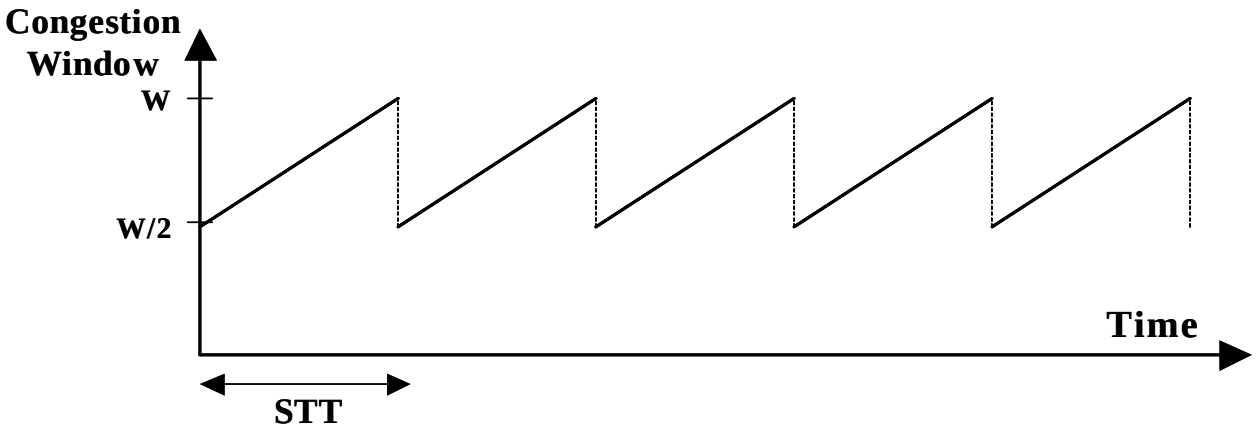

Figure 1: TCP sawtooth diagram

3. The picture above shows the famous TCP saw tooth behavior. We are assuming that fast retransmit and fast recovery always work, i.e. there are no timeouts and there is exactly one packet lost at the end of each "tooth". We are assuming that the flow control window is large and that the sender always has data to send, i.e. throughput will be determined by TCP congestion control.

   In the picture, W represents the congestion window size at which a congestion packet loss occurs (expressed in maximum transfer units). You can assume that W is large, so feel free to approximate (W-1) or (W+1) by W. STT represents the "saw tooth time" expressed in seconds.

   The aim of this exercise is to derive the average throughput of a TCP connection as a function of the roundtrip time (RTT), the maximum transfer unit (MTU), and the packet loss rate (PLR) for the connection. Please use the notation suggested by the figure, i.e. W and STT, as intermediate values if you need them.

   (a) Calculate the STT as a function of W, and the RTT. (Hint: the congestion window goes from W/2 to W in one STT, and remember the congestion window is increased by 1 MTU every RTT).

   > **Solution:** To calculate STT, we need to know the "slope" of the congestion window increase. For TCP, the congestion window is increased 1 MTU every RTT, so
   > STT = RTT * W / 2

   (b) How much data is sent in one STT? (Hint: how much data is sent each RTT?)

   > **Solution:** In one RTT, one congestion window worth of data is sent. There are W / 2 RTTs in STT, and the average size of the congestion window is 3 * W / 4, so
   > amount of data = (W / 2) * (3 * W / 4) * MTU

   (c) What is the average throughput of the connection?

   > **Solution:** The average throughput is the amount of data sent in STT divided by STT, so let us calculate those two entities.
   > How much data is sent in that amount of time? This means that the through put is:
   > T = (3 * W / 4) * MTU / RTT

   (d) What is the average packet loss rate? (Hint: How many losses occur per STT?)

Solution: Now, how much is W? Well, we lose one packet for every tooth in the picture, so the PLR is 1 divided by the number of packets sent in STT, which we already calculated above.
PLR = 1 / ((W / 2) * (3 * W / 4))

(e) What is the relationship between the throughput and the packet loss rate?

Solution: which allows us to calculate W as a function of PLR.
If we substitute W in T by this value, we get:
T = 1.22 * MTU / (RTT * sqrt(PLR))

4. In this problem, you will get experience using ethereal (or wireshark) to do real network packet analysis. Packet traces are useful for debugging and understanding the packet-level behavior of network protocols, among other things.

   Although many CMU machines may already have ethereal installed, you will need to find a Unix machine that you have admin access on in order to capture packets on that machine. Alternatively, you can download wireshark (http://www.wireshark.org) and install it on your local Unix machine. You may have to run the program with administrator privileges (e.g. sudo ethereal) to obtain access to the network interfaces. Please contact us if you have any problems finding a machine to run ethereal or wireshark on.

   In this problem, we would like you to use ethereal/wireshark to obtain TCP sequence and delay plots for the capture of a large file.

   We would like for you to do the following:

   - Run ethereal / wireshark and be able to capture network traffic.
   - Capture the download of any suitably large file. You may use any file, but the file should take at least 5 seconds to download.

   Based on the resulting packet capture:

   (a) Generate a TCP sequence plot based on the traffic generated by downloading the file. Highlight where losses occur during the transfer.

   (b) Generate a packet delay plot, showing the per-packet delay as a function of the sequence number. (Hint: Statistics *rightarrow* TCP Stream Graph *rightarrow* Round Trip Time Graph).