Carnegie Mellon
Computer Science Department.
15-744 Fall 2009 Problem Set 1

This problem set has ten questions. Answer them as clearly and concisely as possible. You may discuss ideas with others in the class, but your solutions and writeup must be your own. If you do discuss at length with others, please mention in your solution for the problem who you collaborated with. Do not look at anyone else's solutions or copy them from anywhere.

This assignment is due by in class on 10/5.

A Background: Ethernet

1. This is a background question about ethernet that you should be able to answer.

   Suppose you set up a 100 Mbps Ethernet with a maximum packet size of 1500 bytes and a diameter of 1 km. You measure the worst case RTT in your Ethernet and find it to be 20.48 usec. Also, the resistance of the wire is 50 ohms.

   You have heard that when using Ethernet you should make sure that all packets have a certain minimum packet size. You want to avoid unnecessary overhead (stuffing packets to reach the minimum packet size) by choosing a small minimum packet size. How would you choose the minimum packet size?

B TCP Timeout Estimation: Now and Then

2. Suppose that TCP is consistently observing RTTs of 1.0 second, with a mean deviation of 0.1 second. Suddenly, the RTT jumps to 5.0 seconds with no deviation. Compare the behavior of the original RTO estimator with the Jacobson/Karels algorithm:

   (a) How many timeouts are encountered with each algorithm?

   (b) What is the largest value of RTO calcuated?

   Assume that α, the srtt EWMA parameter, $= \frac{1}{8}$, and that γ, the sdev EWMA parameter, $= \frac{1}{4}$.

   [This is based on problem 5.29 from Peterson & Davie, 3rd edition]

C TCP Congestion Control

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.

3. 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 → TCP Stream Graph → Round Trip Time Graph).

D Buffers, Losses, and TCP

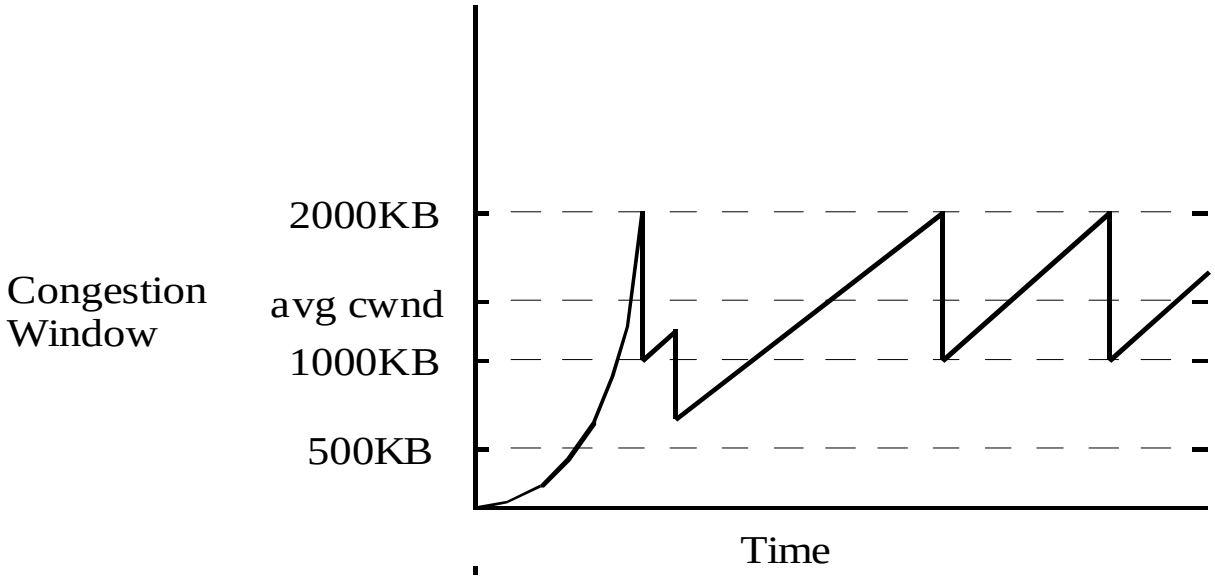
Harry Bovik is given the responsibility of configuring the packet queuing component of a new router. The link speed of the router is 100 Mbit/s and he expects the average Internet round-trip time of connections through the router to be 80ms. Harry realizes that he needs to size the buffers appropriately.

You should assume the following:

- You're dealing with exactly one TCP connection.
- The source is a long-running TCP connection implementing additive-increase (increase window size by 1 packet after an entire window has been transmitted) and multiplicative-decrease (factor-of-two window reduction on congestion).
- The advertised window is always much larger than the congestion window.
- The loss recovery is perfect and has no impact on performance.
- The overhead due to headers can be ignored.
  Harry argues that because the average RTT is 80ms, the average one-way delay is 40ms. Therefore, the amount of buffering he needs for high link utilization is 100 Mbit/s * 40 ms or 500 KBytes.

4. Approximately what bandwidth will TCP achieve with this buffering? (6 points)

Harry raises the buffering in the router to 1000KBytes in hope of fully utilizing his link. After Harry installs the router, he notices that packets are dropped due to both corruption and congestion. Therefore, he decides to make some observations about the observed TCP performance. He makes a transfer to some distant site and records the congestion window whenever it changes. The RTT to this distant site was exactly 100ms throughout the transfer. Below is the plot of the congestion window over time.



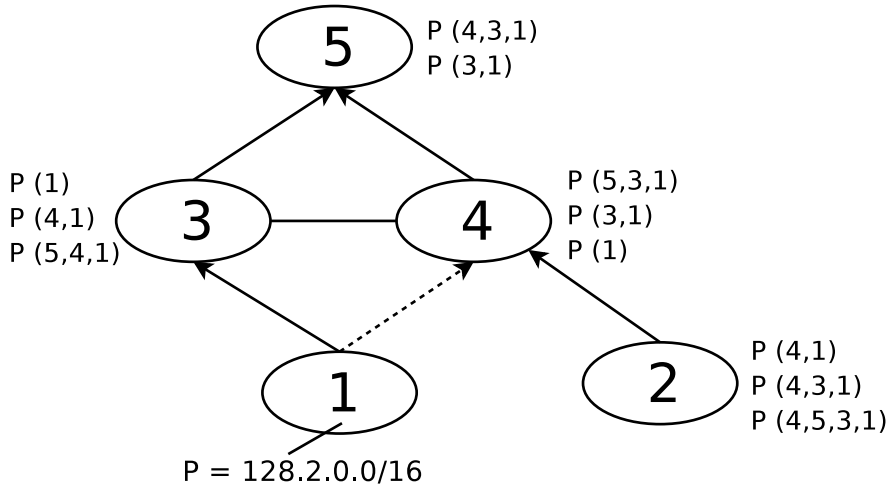
5. Assuming that Harry is using something like TCP NewReno or SACK, mark an X on the Time axis of the congestion window plot whenever a loss event occurred. (4 points)

6. Circle the losses (X's from the previous question) that you think are likely due to corruption and not congestion. (3 points)

7. Highlight the periods of time during which the link is fully utilized **by drawing a bold line with bars on the end below the time axis**. (3 points)

E BGP

8. Consider the AS graph below in parts (a) and (d):



The vertices are inidividual ASes and edges are links between them.

(a) Now suppose the arrows represent customer-provider relationships where the customer points to its provider. An edge without arrows represents a link between peers. The dashed edge is a backup link. Suppose all ASes follow the export rules proposed by Gao and Rexford. List the routes in each AS's local-pref list shown above that can never be received by that AS.

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

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

(d) **Bonus question**: Shown next to each AS is its local-pref ranking of possible routes to P, a prefix in AS 1. Ignore the arrows for now and consider all edges equally valid. Is this AS graph gauranteed to converge? Why or why not? (If you read Gao and Rexford early because it was still on the syllabus, this question is for you!)