Homework 3

Due: 30 October 2006

Problem 1

The window-scale option of TCP (see the notes, page VI-51[1]) provides a means for supporting a large advertised window. PAWS (see page VI-54 provides a means for extending the 32-bit sequence number of TCP.

a. Suppose we have a one-gigabit/second terrestrial network (see page VI-48). What is the maximum throughput we can attain if we use 16 bits for the window size? Explain.

b. With the window-scale option we can have windows up to $2^{30} - 1$ in length (to keep things simple, let's say the maximum window size is 1 gigabyte). With a gigabyte window, what's the maximum throughput we can attain over the network of part 1?

c. With 32-bit sequence numbers and no PAWS, what problems will we have if we have sustained maximum-speed transfers using the parameters of part 2? Explain, showing an example.

d. Explain how PAWS would solve these problems.

Problem 2

Suppose we have a ten-gigabit/second network with a roundtrip time of 100 milliseconds. The maximum segment size (MSS) is 1500 bytes.

a. How long will it take for a TCP connection to reach maximum speed after starting in slow-start mode?

b. Assume there is some probability that a packet is lost in transmission, even though there is no congestion. Give a scenario in which the congestion window becomes 1 MSS (i.e., 1500 bytes) and TCP begins growing the window linearly rather than exponentially (i.e., it's not in slow-start mode).

[1] VI-51 refers to the lecture slides from spring 2006: lecture 6, slide 51.

c. How long will it take the TCP connection to achieve maximum speed after the scenario of part 2, assuming no further packet loss?

d. Suppose (exactly) every 100th packet is dropped. Approximate the throughput of TCP on this network.

Problem 3

We have learned that TCP provides a "reliable" byte stream. Consider a simple client/server pair. The client has 1 kilobyte of data it wishes to send to the server. It calls `connect`, `write`, and `close` appropriately, and each returns without error. Explain how:

a. The server might not receive the data.

b. The server might receive corrupted data.

You should assume this is the only TCP connection between the two hosts, ever.

Problem 4

a. Why is it not sufficient to have all TCP connections start with an initial sequence number of zero?

b. Why does UDP exist? Would it not have been enough to just let user processes send raw IP Packets?

c. Web servers are optimized to handle thousands of connections per second. How can there be more than one simultaneous connection to the server when the server is listening on only one port?

d. Both UDP and TCP use port numbers to identify the destination entity when delivering a message. Give two reasons for why these protocols invented a new abstract ID (port numbers), instead of using process IDs, which already existed when these protocols were adapted.