CS 640: Introduction to Computer Networks

Homework 1

Handed out: 09/27/2006, in class
Due: 10/04/2006, in class

Please type out or neatly print your answers. Add as much detail as you deem fit to your answers. For full credit you must explain how you obtained the answer and not just print the final result. Collaboration is discouraged strongly. Please try to work on your own. If you are stuck, you can discuss with your fellow students, but you are encouraged to talk to the course personnel first.

1. **Transmission overheads**

Consider sending a file of F Kbytes in the following settings. Setting 1 consists of two computers A and B, each equipped with a modem that is capable of sending/receiving at 33.3 kbps. For A to send a file to B, it must first establish a dial-up connection with B, which takes 30 seconds. It can then send the file in 128-byte packets, with a 1byte checksum attached to each packet. The propagation delay of the phone line is negligible. Assume that A and B are directly connected, i.e., there are no intermediate routers. Setting 2 consists of two computers C and D, connected by an established wireless connection that can transmit at 8 kbps through a satellite, with a 0.25s total propagation delay from C to D. In setting 2, files are transmitted without being split into packets. You may assume there are no errors during each transmission and you may ignore acknowledgments, i.e., consider only bits flowing from the sending computer to the receiving computer. Also note that 1 K byte = 1024 bytes and 1 kbps = 1000 bit/second and that 1 byte = 8 bits.

1.1. If F = 16, how long does it take to send the file from A to B?

1.2. How long does it take to send the file from C to D?

1.3. For F = 64, how long does it take to send the file from A to B?

1.4. For F = 64, how long does it take to send the file from C to D?

1.5. Which link is better with respect to total time? Briefly justify your answer.

2. **HTTP persistent connections**

2.1. Why is it difficult to implement persistent connections for CGI scripts and dynamic content in general?

In the most common implementation of persistent connections, the client (i.e., Web browser) opens n connections and requests all the objects (e.g., an image) of a page through these connections as opposed to opening a new connection for each object in the page. Typically, n = 5. The requests are scheduled across the n connections in a round-robin fashion.

2.2. Suppose you are downloading a page with m large objects, m > n, and a large number of small objects. Assume the client requests the objects in the order it finds them in the page. What problem does the above persistent connections scheme pose in this situation?

2.3. Describe a simple client side (i.e., in the browser) solution to address this problem and improve the overall response time for the Web page.

3. <u>**Traceroute**</u>

The program *traceroute* allows you to find out the path (i.e. sequence of routers) that a packet will follow to a specific destination. The routers along the path are often identified by name, which means that you can learn the identity of the various ISPs your packets travel through. For this problem, use *traceroute* to record the path taken by packets from a UW host to

- www.batlan.be
- www.chicagotribune.com
- www.washington.edu
- www.cmu.edu

Some traceroutes may not finish, since the destination may block ICMP packets. You should focus on the completed portion of the traceroute. See “man traceroute” for more information.

3.1. How many network hops to each host? Is the number of network hops indicative of the proximity of the host to UW? Explain your answer.

3.2. For www.batlan.be can you identify the countries which the network path traverses?

3.3. Is www.chicagotribune.com located in Chicago? Explain your answer.

4. <u>**Packets and Headers**</u>

Consider a message being transmitted from an application running on port 8080 on host S to port 19999 on host D. The transport protocol being used is TCP. The picture below shows one packet of the message at five distinct points along its path. Note that S and D are end hosts. B is a Layer-2 switch (or bridge). R1 and R2 are routers. Assign a number to each header in a packet: header 1 is physical layer, 2 is data link header,…, 5 is the application header.

```
I                                    VI
|                                     |
\/                                   /\
|____II_____III______IV_____V_____|
  S ------ B ------ R1 ------ R2 ------ D
```

An example of a “packet instance and header number” is (I, 2), where “2” represents the data link header, and “I” is the network location (in this case, the host S’s TCP/IP stack).

2.1 List all packet instances and header numbers where S’s IP address is written. Do the same for D’s IP. Do the same for S and D’s MAC addresses. Do the same for B’s MAC address.

2.2 List all packet instances and header numbers where TCP’s protocol identifier is written.

2.3 List all packet instances and header numbers where R1’s MAC address is written. Do the same for R2’s MAC address. Do the same for R1 and R2’s IP addresses.

2.4 List all packet instances and header numbers where the source port number 8080 is written. Do the same for the destination port 19999.

FAQ

See course web-page.