2nd Lt David Crow
ENG/20M

CSCE 560 Homework / Wireshark Lab 2
Chapter 2 – Application Layer
Fall 18

Assigned: **Monday, 15 Oct**
Due: **Wednesday, 24 Oct, 1400**

Problem 1. Chapter 2, R.1
List five nonproprietary Internet applications and the application-layer protocols that they use.

Among others,

1. The World Wide Web uses HTTP,
2. Email uses SMTP or POP3,
3. Remote login uses Telnet or SSH,
4. File transfer uses FTP, and
5. Video conferencing uses RTP.

Problem 2. Chapter 2, R.3
For a communication session between a pair of processes, which process is the client and which is the server?

The server is the process that waits to be requested and that sends files. The client is the process that sends requests and waits for files.

For processes capable of both requesting and responding, when process A asks process B for a specific file, process A is the client and process B is the server; when process A sends a specific file to process B, process A is the server and process B is the client. Thus, both processes are each a client and a server.

Problem 3. Chapter 2, R.5
What information is used by a process running on one host to identify a process running on another host?

A host needs the other host's IP address and [socket] port number to identify a process.

Problem 4. Chapter 2, R.10
What is meant by a handshaking protocol?

Effectively, a client and a server introduce themselves before requesting and receiving information. TCP uses a "three-way handshake." It's performed as follows:

1. The client sends a small packet to the server.
2. The server sends a small acknowledgement packet to the client.
3. The client sends an acknowledgement packet (yes, to the server's acknowledgement packet) to the server.

Problem 5. Chapter 2, R.12
Consider an e-commerce site that wants to keep a purchase record for each of its customers. Describe how this can be done with cookies. [Describe how the cookies are created and exchanged between the computers. Discuss how and which headers are modified.]

Let's make up an e-commerce site. We'll call it *Amazon.* If this fictional company wants to track Susan's purchases, its servers will include in the HTTP response message a header line for Susan's cookie. This line will assign a unique value to Susan; let's use 1234. In the HTTP response, then, we'll find a header line that might say $Set - cookie$: 1234.

Susan's client will, upon receipt of the response message, add to its cookie file a cookie for Amazon with a value of 1234. Future HTTP requests from Susan's client to Amazon will include a header line that says $Cookie$: 1234. Amazon's servers can then cross-reference this cookie with Amazon's backend database of cookies for any information concerning Susan.

Of course, if Amazon ever needs to change Susan's cookie, its servers can send a new $Set - cookie$ header line in some future HTTP response.

Problem 6. Chapter 2, R.16
Suppose Alice with a Web-based e-mail account (such as Yahoo! mail or Hotmail) sends a message to Bob, who accesses his mail from his mail server using POP3. Discuss how the message gets from Alice's host to Bob's host. Be sure to list the series of application-layer protocols that are used to move the message between the two hosts.

Alice logs into her web-based email and sends an email to her mail server using HTTP. After the mail server receives Alice's message, it forwards the message to Bob's mail server using SMTP. At that point, Bob can access his email from his mail server using POP3.

Problem 7. Chapter 2, R.18
From a user's perspective, what is the difference between the *download-and-delete mode* and the *download-and-keep mode* in POP3? [How do these modes affect the user?]

Download-and-delete effectively splits emails over all machines a user accesses his email from. Once an email is downloaded, it no longer exists on the server; it only exists on the machine it was downloaded to. In other words, if Bob downloads an email on his work computer using download-and-delete, he will not be able to access the same email on his home computer.

Contrast this with download-and-keep. In this mode, Bob can download his email to multiple computers; emails are not deleted from the server after they are downloaded.

For users who *never* use more than one computer, download-and-delete is sufficient. For everyone else, download-and-keep ensures its users can access emails from wherever they please.

Problem 8. Chapter 2, P1
True or false:

a. A user requests a Web page that consists of some text and three images. For this page the client will send one request message and receive four response messages.

   False – the first response contains a reference to each of the three images; the client must then request each image.

b. Two distinct Web pages (for example, www.mit.edu/research.html and www.mit.edu/students.html) can be sent over the same persistent connection.

   True.

c. With nonpersistent connections between browser and origin server, it is possible for a single TCP segment to carry two distinct HTTP request messages.

   False – each nonpersistent connection is closed after one HTTP request and one HTTP response.

d. The `Date:` header in the HTTP response message indicates when the object in the response was last modified.

   False – the `Date:` line indicates when the the HTTP response was created and sent by the server.

Problem 9. Chapter 2, P7 **(this problem has been modified)**
Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so that a DNS lookup is necessary to obtain the IP address. Suppose that *n* DNS servers are visited before your host receives the IP address from DNS; the successive visits incur an RTT of $RTT_1, ..., RTT_n$. Further suppose that the Web page associated with the link contains exactly one

object, a small amount of HTML text. Let RTT_0 denote the RTT between the local host and the server containing the object. Assuming a transmission time of t_1 for the object, how much time elapses from when the client clicks on the link until the client receives the object?

The sum of all round-trip times to get the destination's IP address is $RTT_1 + RTT_2 \dots + RTT_n = \sum_{i=1}^{n} RTT_i$.

The host then contacts the destination and sets up a TCP connection before it can receive the HTML file; this process takes $2 * RTT_0$ in total.

Because we must also add the transmission time for the packet, our total time is $2 * RTT_0 + t_1 + \sum_{i=1}^{n} RTT_i$. ✓

Problem 10. Chapter 2, P9

Consider Figure 2.12, for which there is an institutional network connected to the Internet. Suppose that the average object size is 850,000 bits and that the average request rate from the institution's browsers to the origin servers is 16 requests per second. Also suppose that the amount of time it takes from when the router on the Internet side of the access link forwards an HTTP request until it receives the response is three seconds on average (see Section 2.2.5). Model the total average response time as the sum of the average access delay (that is, the delay from Internet router to institution router), and the average Internet delay. For the average access delay, use $\Delta/(1 - \Delta\beta)$, where Δ is the average time required to send an object over the access link and β is the arrival rate of objects to the access link. [You may assume the response time for a cache hit (i.e., LAN delay) is 0 seconds.]

a. Find the total average response time.

Average time required: $\Delta = \frac{850000\ bits}{15*10^6 \frac{bits}{second}} = 0.057 \frac{seconds}{request}$

Average access delay: $\frac{\Delta}{1-\Delta\beta} = \frac{0.057 \frac{seconds}{request}}{1-(16 \frac{requests}{second})(0.057 \frac{seconds}{request})} = 0.61\ seconds$

Average response time: $delay_{access} + delay_{internet}$
$= 0.61\ seconds + 3.00\ seconds = 3.61\ seconds$

b. Now suppose a cache is installed in the institutional LAN. Suppose the **miss** rate is 0.4. Find the total response time.

Average access delay: $\frac{\Delta}{1-\Delta\beta} = \frac{0.057 \frac{seconds}{request}}{1-40\%*(16 \frac{requests}{second})(0.057 \frac{seconds}{request})} = 0.090\ seconds$

Average response time (cache misses): $delay_{access} + delay_{internet}$
$= 0.090\ seconds + 3.00\ seconds = 3.09\ seconds$

Total average response time: $time_{cache-misses} + time_{cache-hits}$
$= 40\% * 3.09\ seconds + 60\% * 0\ seconds = 1.24\ seconds$ ✓

Problem 11. Chapter 2, Supplemental Question 1
What is the difference between persistent HTTP with pipelining and persistent HTTP without pipelining? [What event causes the requests to be sent?] Which of the two is used by HTTP/1.1?

> Persistent HTTP with pipelining allows a client to send over the same connection multiple requests without waiting for any request's response message. In other words, the client can just send a bunch of requests back-to-back. In the same vein, the server can send responses back-to-back.
>
> Contrast this with persistent HTTP without pipelining. In this mode, a client must wait for the server's response before said client can send a subsequent request (of course, the client doesn't need to wait for a response when sending its first request). ✓
>
> HTTP/1.1 employs persistent connections with pipelining.

Problem 12. Chapter 2, Supplemental Question 2
Why is it said that FTP sends control information "out-of-band?"

> FTP actually uses two different TCP connections to transfer files. One connection – the control connection – is used for sending and responding to requests. The other connection – the data connection – is used for actually transferring information. In other words, a client sends a request over a control connection; the server then sends a response along the same control. The data itself moves over the data connection. ✓
>
> Because the control information (the request, the response, acknowledgements, abort messages, etc) are not sent on the same connection as the data, it's said that the control information is sent *out-of-band*.

Wireshark Lab

Complete the lab in `02 - Wireshark_HTTP.pdf`.

> Okay.

The last page of this lab instructs you to go to http://www.motobit.com/util/base64-decoder-encoder.asp in order to decode a string. Another good website is http://www.opinionatedgeek.com/dotnet/tools/base64decode/; paste the string in the box then click the Decode button.