Nathan Tung
004-059-195

# CS118 Homework 2

1. 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; each object requires a request message to be sent (so 4 requests sent)**
   b. Two distinct Web pages (e.g. www.ucla.edu/research.html and www.ucla.edu/students.html) can be sent over the same persistent connection.
      →**True; since it's persistent, the connection doesn't close immediately, and these websites are on the same server**
   c. With nonpersistent connections between browser and origin server, it is possible for a single TCP segment two carry two distinct HTTP request messages.
      →**False; TCP segments in a nonpersistent connection carry a single request**
   d. The 'Date:' header in the HTTP response message indicates when the object in the response was last modified.
      →**False; the Date header indicates object's date of creation**
   e. HTTP response messages never have an empty message body.
      →**False; for example, error code 204 means no content was found**

2. Suppose you wanted to do a transaction from a remote client to a server as fast as possible. Would you use TCP or UDP? Why? (Book: Chapter 2, Question 6)
   **→I would use UDP, or User Datagram Protocol. While TCP is known to be more reliable and secure than UDP, which sometimes loses data, UDP is known for its speed. UDP is lightweight, connectionless, and fast. (On the other hand, TCP needs to do two roundtrips in order to both set up a connection and transfer messages.)**

3. 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 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 the DNS; the successive visits incur a RTT of RTT1,…,RTTn. Further suppose that the Web page associated with link contains exactly one object, consisting of a small amount of HTML text. Let RTT0 denote the RTT between the local host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the object? (Book: Chapter 2, Problem 7)
   **→As the problem states, the successive visits to accessing the n DNS servers incurs $RTT_1 + \ldots + RTT_n$**
   **Once we obtain the IP, we establish a TCP connection and request the single object. This requires $2*RTT_0$.**
   **In total, this results in a response time of $2RTT_0 + RTT_1 + \ldots + RTT_n$.**

4. Referring to the previous problem, suppose the HTML file references eight very small objects on the same server. Neglecting transmission times, how much time elapses with (Book: Chapter 2, Problem 7)
   a. Non-persistent HTTP with no parallel TCP connections?
      **→Obtaining IP address: $RTT_1 + \ldots + RTT_n$**
      **Downloading base HTML file: $2RTT_0$**
      **Downloading 8 objects serially: $8*2RTT_0$**
      **Total: $18RTT_0 + RTT_1 + \ldots + RTT_n$**

b. Non-persistent HTTP with the browser configured for 5 parallel connections?
**→ Obtaining IP address: $RTT_1 + \dots + RTT_n$**
**Downloading base HTML file: $2RTT_0$**
**Downloading 8 objects with 5 in parallel: $2*2RTT_0$**
**(since 8/5 = 2 when rounded up; first download 5, then download 3)**
**Total: $6RTT_0 + RTT_1 + \dots + RTT_n$**

c. Persistent HTTP?
**→ Obtaining IP address: $RTT_1 + \dots + RTT_n$**
**Downloading base HTML file: $2RTT_0$**
**(recall that the TCP does not have to be reestablished)**
**Downloading 8 objects (pipelined, since persistent HTTP): $RTT_0$**
**Total: $3RTT_0 + RTT_1 + \dots + RTT_n$**

5. Consider a short, ten-meter link over which a sender can transmit at a rate of 150 bits/sec in both directions. Suppose that packets containing data are 100,000 bits long, and packets containing only control (e.g. ACK or handshaking) are 200 bits long. Suppose that *N* parallel connections get *1/N* of the link bandwidth. Now consider the HTTP protocol and suppose that each downloaded object is 100Kbits long, and that the initial downloaded file contains 10 referenced objects from the same sender. Would parallel downloads via parallel instances of non-persistent HTTP make sense in this case? Now consider persistent HTTP. Do you except significant gains over the non-persistent case? Justify and explain your answer. (Book: Chapter 2, Problem 10)
**→Things to note:**
**-Downloaded objects are 100 Kbits, which fits into a data packet**
**-The connection of 150 bits/s, when shared by N=10, splits into 15 bits/s**
**-Speed of light is $3 \cdot 10^8 m/s$, so 10 meters takes a negligible $3.33 \cdot 10^{-8} s$**

**For non-persistent HTTP in parallel, we establish 3-way handshaking by transmitting the control packet 3 times along with the actual initial download at a normal 150 bits/s speed. The subsequent parallel downloads repeat the above at 15 bits/s.**

**Time = 3*200/150 + 100000/150 + 3*200/15 + 100000/15**
**Time = 7377.33 s**

**For persistent HTTP not in parallel, we establish 3-way handshaking and the initial download the same way as before. However, this time we are operating sequentially with persistent HTTP, so we download the 10 files each at a normal 150 bits/s. There is no need to continuously re-establish TCP connection.**

**Time = 3*200/150 + 100000/150 + 10*100000/150**
**Time = 7337.33 s**

**As the calculation above shows, persistent HTTP has a 40-second-leverage above its non-persistent parallel HTTP counterpart. While the gains are not exactly significant (approximately 0.5% speedup), persistence is often capable of delivery content faster while reducing CPU usage.**