Jordan Dube
HW #2
CMSC 440
QUESTIONS 4, 5, 7, 8, 9, 10

4.

a. http:/gaia.cs.umass.edu/cs453/index.html
    i. Combining strings found after the “GET” and “Host:”, prefaced by http:/
    ii. The host is the server that hosts the
b. 1.1
    i. Found after the “HTTP/” in the first line
c. It requests a persistent connection
    i. Found by looking at “Connection:keep-alive” in last line
d. I was not able to locate an IP address in the Request. The TCP protocol used only specifies ports, not IPs, which are handled by the transport layer. So the next step, the HTTP, definitely does not contain the IP address of either sender or receiver.
e. Mozilla/5.0 browser initiated the request (netscape) on a windows machine. This is needed because different browsers can run on different frameworks, which means that they handle the webpage differently, such as different display settings/loading processes. Due to this, the web server sends one of multiple versions of the webpage.
    i. Found after the second <cr><lf> pair, designated by “User-Agent:”

5.

a. Yes, it was able to find the document (i). It was provided at 12:39:45 GMT on Tue, 07 Mar 2008 (ii).
    i. This is seen by the “200 OK” found on the first line after “HTTP/1.1”
    ii. Found after the “Date” keyword after the first <cr><lf> pair
b. Sat, 10 Dec2005 18:27:46 GMT
    i. Found after the third <cr><lf> pair after “Last-Modified”
c. 3874 Bytes
    i. Found after 5th <cr><lf> pair, denoted by “Content-Length”. This field is in bytes because of the preceding <cr><lf> line: “Accept-Ranges: bytes”
d. The first 5 bytes returned are “<!doc”, and the connection is persistent.
    i. Found after the <cr><lf><cr><lf> portion, as this pattern separates the header from the content, and the content being returned is established as being in HTML format as shown by “Content-Type: text/html” in the line preceding the double <cr><lf>.
    ii. Found after the 8th <cr><lf> pair: “Connection: Keep-Alive”

7.

a. $t_{total} = 2 * RTT_0 + \sum_{i=1}^{n} RTT_i$
    i. The summation is to add up each successive visit to the DNS servers (up to visiting all of them), and the $2RTT_0$ is there because the first RTT_0 is used to establish a connection, and the second one is to actually transport the object.

8.

a. $t_{total} = 18 * RTT_0 + \sum_{i=1}^{n} RTT_i$

   i. The summation of RTT from 1 to n is still needed once due to DNS lookup, and the 18 RTT_0 is broken down: 2 RTT_0 to establish connection to get html base, 2 RTT_0 per each object retrieval to establish connection and get the object, so 2 + 8 * 2 = 18 RTT_0

b. $t_{total} = 6 * RTT_0 + \sum_{i=1}^{n} RTT_i$

   i. The summation of RTT from 1 to n is still needed once due to DNS lookup, and the 6 RTT_0 can be broken down: $2RTT_0$ to retrieve the base page, $2RTT_0$ to connect+retrieve the first 5 objects, and then $2RTT_0$ to get the remaining 3 objects in the same manner as the first 5 objects.

c. For case C, it is not clear whether the browser using persistent HTTP supports parallel connections or not, so both are provided below:

   i. With no parallel TCP connections:

   $$t_{total} = 10 * RTT_0 + \sum_{i=1}^{n} RTT_i$$

      a. Summation for DNS lookup, $2RTT_0$ for initial connection + base page retrieval, $8RTT_0$ for retrieving each of the 8 objects

   ii. With 5 parallel TCP connections (precedent taken from case B):

   $$t_{total} = 4 * RTT_0 + \sum_{i=1}^{n} RTT_i$$

      a. Summation for DNS lookup, $2RTT_0$ for initial connection + base page retrieval, $1RTT_0$ for retrieving first 5 objects, and $1RTT_0$ for the other 3 objects.

9.

a. $L = 850000\ bits$

$R = 15\ Mbps = 15000000\ bps$

$d_{trans} = \Delta = L/R = 850000/15000000 = 0.05667\ s$

$rate_{request} = \beta = 16rps$

$d_{avg\ access} = \Delta/(1 - \Delta\beta) = 0.05667\ s/(1 - 0.05667\ s * 16rps)$

$= 0.05667\ s/(1 - 0.9066r) = 0.05667\ s/0.09333r = 0.60714s/r$

$d_{internet} = 3s$

$t_{total\ avg\ response} = d_{internet} + d_{avg\ access} = 0.60714s/r + 3s = 3.60714s/r$

b. $rate_{miss} = 0.4$

$L = 850000\ bits$

$R = 15\ Mbps = 15000000\ bps$

$d_{trans} = \Delta = L/R = 850000/15000000 = 0.05667\ s$

$rate_{request} = \beta = rate_{miss} * 16rps = 0.4 * 16rps = 6.4rps$

$d_{avg\ access} = \Delta/(1 - \Delta\beta) = 0.05667\ s/(1 - 0.05667\ s * 6.4rps)$

$= 0.05667\ s/(1 - 0.3627r) = 0.05667\ s/0.6373r = 0.08891s/r$

$d_{internet} = 3s$

$t_{total\ avg\ response} = (1 - rate_{miss})(0) + rate_{miss} * (d_{internet} + d_{avg\ access})$

$= 0.4 * (0.08891s/r + 3s) = 0.4 * 3.08891s/r = 1.2356s/r$

10.

a. Parallel downloads via parallel instances of non-persistent HTTP

$m = 10m$

$R = 150bps$

$L_{data} = 100000\ bits$

$L_{control} = 200\ bits$

1 base object, 10 referenced objects

$length_{object} = 100\ Kbits = 100000\ bits$

$n_{connections} = 10$

$d_{prop} = m/s$

Assume speed of transfer allowed by the medium is $3 * 10^8 m/s$ (speed of light):

$d_{prop} = 10m/(3 * 10^8 m/s) = 3.3333 * 10^{-8} s$

To get the base page:

$t_1 = 3L_{control}/R + L_{data}/R + 4d_{prop} = 600/150 + 100000/150 + 4d_{prop}$

$= 100600/150 = 670.667s + 4d_{prop}$

To get the objects:

$t_2 = 3L_{control}/R/n_{connections} + L_{data}/R/n_{connections} + 4d_{prop} = 600/15 + 100000/15 + 4d_{prop}$

$= 100600/15 = 6706.667s + 4d_{prop}$

Technically t_2 is occurring once for each object, but since they are parallel, they can be represented as a single iteration of the above equation

$t_{total} = t_1 + t_2 = 670.667s + 4d_{prop} + 6706.667s + 4d_{prop}$

$= 7377.333s + 8d_{prop}$

Since the propagation delay is so small compared to the transmission delay, the propagation delay is negligible, so the total time to retrieve all objects using non

persistent parallel connections is 7377.333 seconds. Given this observation, it does make sense to use paralleldownloads via parallel instances of non-persistent HTTP

b. persistent HTTP without parallel connections

$m = 10m$

$R = 150bps$

$L_{data} = 100000\ bits$

$L_{control} = 200\ bits$

1 base object, 10 referenced objects

$length_{object} = 100\ Kbits = 100000\ bits$

$d_{prop} = m/s$

Assume speed of transfer allowed by the medium is $3 * 10^{8} m/s$ (speed of light):

$d_{prop} = 10m/(3 * 10^{8} m/s) = 3.3333 * 10^{-8} s$

To get the base page:

$t_1 = 3L_{control}/R + L_{data}/R + 4d_{prop} = 600/150 + 100000/150 + 4d_{prop}$

$= 100600/150 + 4d_{prop} = 670.667s + 4d_{prop}$

To get the 10 objects:

$t_2 = 10(L_{control}/R + L_{data}/R + 2d_{prop}) = 10 * (200/150 + 100000/200 + 2d_{prop})$

$= 10 * (100200/150 + 2d_{prop}) = 10 * 668 + 20d_{prop} = 6680 + 20d_{prop}$

$t_{total} = t_1 + t_2 = 6680 + 20d_{prop} + 670.667s + 4d_{prop} = 7350.6667s + 24d_{prop}$

Since the propagation delay is so small compared to the transmission delay, the propagation delay is negligible, so the total time to retrieve all objects using persistent HTTP without parallel connections is 7350.6667 seconds

c. From the examples above, we can see that we should not expect significant gains over the non-persistent case from the persistent case.