Homework 2 assignment for ECE374

Posted: 02/20/15

Due: 02/27/15

Note: *In all written assignments, please show as much of your work as you can. Even if you get a wrong answer, you can get partial credit if you show your work.* If you make a mistake, it will also help the grader show you where you made a mistake.

Problem 1: (10 Points)

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 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, consisting of a small amount of HTML text. Let RTT_0 denote the RTT between the local host and the server containing the object.

a. 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?

Now assume the HTML file references seven very small objects on the same server. Neglecting transmission times, how much time elapses with

b. Non-persistent HTTP with no parallel TCP connections?
c. Non-persistent HTTP with the browser configured for 5 parallel connections?
d. Persistent HTTP?

Solution:

a. *The total amount of time to get the IP address is* $RTT_1 + RTT_2 + \cdots + RTT_n$*. Once the IP address is known,* RTT_O *elapses to set up the TCP connection and another* RTT_O *elapses to request and receive the small object. The total response time is* $2RTT_o + RTT_1 + RTT_2 + \cdots + RTT_n$
b. $RTT_1+\ldots+RTT_n+2RTT_O+7*2RTT_O = 16* RTT_O+ RTT_1+\ldots+RTT_n$.
c. $RTT_1+\ldots+RTT_n+2RTT_O+2*2RTT_O = 6* RTT_O+ RTT_1+\ldots+RTT_n$.
d. $RTT_1+\ldots+RTT_n+2RTT_O+RTT_O = 3* RTT_O+ RTT_1+\ldots+RTT_n$..

Problem 2: (20 Points)

Consider the scenario shown in Figure 1 in which a server is connected to a router by a 100Mbps link with a 50ms propagation delay. Initially this router is also connected to two routers, each over a 25Mbps link with a 200ms propagation delay. A 1Gbps link connects a host and a cache (if present) to each of these routers and we assume that this link has 0 propagation delay. All packets in the network are 20,000 bits long.

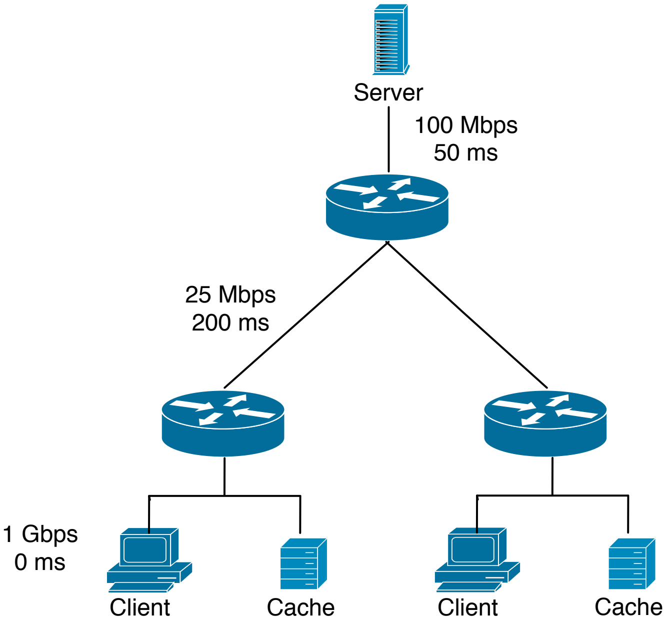


Figure 1

a. What is the end-to-end delay from when a packet is transmitted by the server to when it is received by the client? In this case, we assume there are no caches, there's no queuing delay at the routers, and the packet processing delays at routers and nodes are all 0.
b. Here we assume that client hosts send requests for files directly to the server (caches are not used or off in this case). What is the maximum rate at which the server can deliver data to a single client if we assume no other clients are making requests?
c. Again we assume only one active client but in this case the caches are on and behave like HTTP caches. A client's HTTP GET is always first directed to its local cache. 60% of the requests can be satisfied by the local cache. What is the average rate at which the client can receive data in this case?
d. What is the average end-to-end delay for the case that 60% of the requests can be satisfied by the local cache?

Solution:

a. *Answer: If all packets are 20,000 bits long it takes 200 usec to send the packet over the 100Mbps link, 800 usec to send over the 25Mbps link, and 20 usec to send over the 1Gbps link. Sum of the three-link transmission is 1020 usec. Thus, the total end-to-end delay is 251.02 msec.*
b. *Answer: Server can send at the max of the bottleneck link: 25Mbps.*
c. *Answer: We assume that requests are serially satisfied. 40% of the requests can be delivered at 25Mbps and 60% at 1Gbps. So the average rate is 610Mbps.*
d. *Answer: 40% of the requests have a delay of 251.02 msec and 60% of the requests have a delay of 20 usec. So the average delay is (150.61 + 12) 162.61 msec.*

Problem 3: (15 Points)

In this problem, we use the useful dig tool available on Unix and Linux hosts to explore the hierarchy of DNS servers. Recall that in slide 67 in Chapter 2, a DNS server higher in the DNS hierarchy delegates a DNS query to a DNS server lower in the hierarchy, by sending back to the DNS client the name of that lower-level DNS server. First read the man page for dig (e.g., http://linux.die.net/man/1/dig) , and then answer the following questions.

a. Starting with a root DNS server (from one of the root servers [a-m].root-servers.net], initiate a sequence of queries for the IP address for your department's Web server (www.ecs.umass.edu) by using *dig*. Show the list of the names of DNS servers in the delegation chain in answering your query.
b. Repeat part a) for several popular Web sites, such as google.com, yahoo.com, or amazon.com.

Back up your answers with screen shots that show the results of your *dig* queries.

Solution:

a)

The following delegation chain is used for gaia.cs.umass.edu
a.root-servers.net
E.GTLD-SERVERS.NET
ns1.umass.edu(authoritative)

First command:

```
dig +norecurse @a.root-servers.net any gaia.cs.umass.edu
```

```
;; AUTHORITY SECTION:
edu.            172800  IN    NS    E.GTLD-SERVERS.NET.
edu.            172800  IN    NS    A.GTLD-SERVERS.NET.
edu.            172800  IN    NS    G3.NSTLD.COM.
edu.            172800  IN    NS    D.GTLD-SERVERS.NET.
edu.            172800  IN    NS    H3.NSTLD.COM.
edu.            172800  IN    NS    L3.NSTLD.COM.
edu.            172800  IN    NS    M3.NSTLD.COM.
edu.            172800  IN    NS    C.GTLD-SERVERS.NET.
```

Among all returned edu DNS servers, we send a query to the first one.

```
dig +norecurse @E.GTLD-SERVERS.NET any gaia.cs.umass.edu
```

```
umass.edu.      172800  IN    NS    ns1.umass.edu.
umass.edu.      172800  IN    NS    ns2.umass.edu.
umass.edu.      172800  IN    NS    ns3.umass.edu.
```

Among all three returned authoritative DNS servers, we send a query to the first one.

```
dig +norecurse @ns1.umass.edu any gaia.cs.umass.edu

gaia.cs.umass.edu.     21600   IN     A      128.119.245.12
```

b) The answer for google.com could be:
a.root-servers.net
E.GTLD-SERVERS.NET
ns1.google.com(authoritative)

Problem 4: (25 Points)

Consider distributing a file of *F* = 25 Gbits to *N* peers. The server has an upload rate of u_s = 30 Mbps, and each peer has a download rate of d_i = 2 Mbps and an upload rate of *u*.

a. For *N* = 10, 100, and 1,000 and *u* = 300 Kbps, 700 Kbps, and 2 Mbps, prepare a chart giving the minimum distribution time for each of the combination of *N* and *u* for both client-server distribution and P2P distribution.

Client Server:

		N		
		10	**100**	**1000**
u	**300 Kbps**			
	700 Kbps			
	2 Mbps			

Peer-to-Peer:

		N		
		10	**100**	**1000**
u	**300 Kbps**			
	700 Kbps			
	2 Mbps			

b. Assuming you could provide as many servers as you like, what would be the maximum number of servers you would need to reduce the client-server distribution time to its minimum for the cases of *N* = 10, 100, and 1,000? What would the minimum distribution times be in this case?

N		
10	**100**	**1000**

Solution:

a. *For calculating the minimum distribution time for client-server distribution, we use the following formula:*

$D_{cs} = max \{NF/u_s, F/d_{min}\}$

Where, $F = 25$ *Gbits* $= 25 * 1024$ *Mbits* $= 25{,}600$ *Mbits*
$u_s = 30$ *Mbps*
$d_{min} = d_i = 2$ *Mbps*

Note, 300Kbps = 300/1024 Mbps.

Client Server

	10	100	1000
300	12800	85333.33333	853333.3333
700	12800	85333.33333	853333.3333
2000	12800	85333.33333	853333.3333

Peer to Peer
Similarly, for calculating the minimum distribution time for P2P distribution, we use the following formula:

$$D_{P2P} = max\{F/u_s, F/d_{min}, NF/(u_s + \sum_{i=1}^{N} u_i)\}$$

	10	100	1000
300	12800	43172.59552	79264.63474
700	86687.83069	87174.56616	87268.59322
2000	87278.23592	87278.81088	87278.92121

b.

N		
10	**100**	**1000**
10	100	1000

Problem 5: (10 Points)

a. Suppose that Alice wants to send an email message to Bob. This will involve four entities: Alice's mail client (for email composition and sending), Alice's outgoing mail server, Bob's incoming mail server, and Bob's mail client (for email retrieval and viewing). Between which of these four entities does the SMTP protocol operate? What about the IMAP protocol?

b. How does SMTP mark the end of a message body? How about HTTP? Can HTTP use the same method as SMTP to mark the end of a message body? Explain.

Solution:

a. *Answer: SMTP runs between Alices mail client and her server, and also (separately) between her server and Bob's server. IMAP runs between Bob's server and his mail client to retrieve messages from Bob's server.*

b. SMTP uses a line containing only a period to mark the end of a message body. HTTP uses "Content-Length header field" to indicate the length of a message body. No, HTTP cannot use the method used by SMTP, because HTTP message could be binary data, whereas in SMTP, the message body must be in 7-bit ASCII format.

Problem 6 (20 Points): Sliding Window Protocols

a. Consider the sliding window protocol in Figure 2 to the right. Does this figure indicate that Go-Back-N is being used, Selective Repeat is being used, or there is not enough information to tell? Explain your answer briefly.

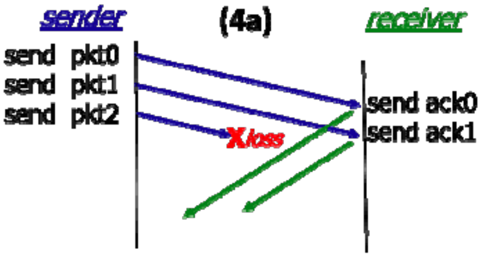


Figure 2

b. Consider the sliding window protocol in Figure 3. Does this figure indicate that Go-Back-N is being used, Selective Repeat is being used, or there is not enough information to tell? Explain your answer briefly..

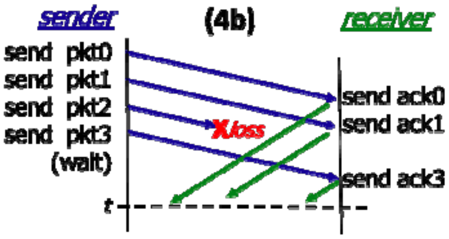


Figure 3

c. Consider Figure 3 again. Suppose the sender and receiver windows are of size N = 5 and suppose the sequence number space goes from 0 to 15. Show the position of the sender and receiver windows over this sequence number space at time t (the horizontal dashed line).

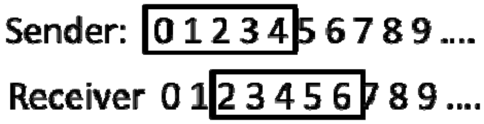
d. Suppose that it takes 1 ms to send a packet, with a 10 ms one-way propagation delay between the sender and receiver. The sliding windows size is again N = 4. What is the channel/link utilization?

Solution:

a. *Answer: there is not enough information to tell, since both GBN and SR will individually ACK each of the first two messages as they are received correctly.*

b. *Answer: This must be the SR protocol since pkt 3 is acked even though pkt 2 was lost. GBN uses cumulative ACKs and so would not generate an ACK 3 if pkt 2 was missing*

c. *Answer:*



Sender: 0 1 2 3 4 5 6 7 8 9 ....

Receiver 0 1 2 3 4 5 6 7 8 9 ....

d. *Answer: the utilization is 4/(1+20) or 0.19.*