Homework 3

1. (a) The actual chain of delegation for www.cs.wisc.edu is listed in the following table.

Server Required	NS delegations to
a.root-servers.net	M3.NSTLD.COM. A3.NSTLD.COM. C3.NSTLD.COM. D3.NSTLD.COM. E3.NSTLD.COM. G3.NSTLD.COM. H3.NSTLD.COM. L3.NSTLD.COM.
M3.NSTLD.COM	DNS2.ITD.UMICH.edu. cs.wisc.edu. DNS.cs.wisc.edu. DNS2.cs.wisc.edu.
cs.wisc.edu	dns.cs.wisc.edu. hostmaster.cs.w

At the last step, a SOA is received indicating no new delegations can be obtained.

(b). The query chain of the address 128.105.175.44 is summarized in the following table:

Server Required	NS delegations to
a.root-servers.net	figwort.ARIN.NET. chia.ARIN.NET. dill.ARIN.NET. BASIL.ARIN.NET. henna.ARIN.NET. indigo.ARIN.NET. epazote.ARIN.NET.
figwort.ARIN.NET	dns.cs.wisc.edu. dns2.itd.umich.edu. cs.wisc.edu.
dns.cs.wisc.edu	bluebird.cs.wisc.edu

where the last request returns the server name of the IP, namely *bluebird.cs.wisc.edu*. For the IP 128.105.6.15, we can find the DNS name as *topaz.cs.wisc.edu*, and we omit the details here.

2

(a). According to the flag field, the packet 3,4,5 are the three way handshake.

Packet 3 is the SYN from the client (192.168.0.100)

Packet 4 is the SYN/ACK reply form the server (192.168.0.102)

Packet 5 is the clients ACK

(b).

Seq: 0x94F22EBE (Relative is 0)

The flag field of this segment has SYN set (0x02) which identifies it as SYN segment.

(c).
Sequence number of the SYNACK segment is 0x84CABEB3 (relative 0).

The acknowledgement number of this segment is 0x94F22EBF (relative 1) which is 1 more than the original sequence number sent by the client.

The flag field of this segment has both SYN and ACK flags set (0x12) which identifies it as SYNACK segment.

(d)
The receiver acknowledges 2 packets at a time. So for every 2 packets of data, there will be a single ACK send by the receiver acknowledging the 2 data packets.
12 acknowledges packet 9 and 10. 14 acknowledges 11 and 13.

(e)
4 requests.

Objects(size):	/ (12671)
	/gnu.css(1270)
	/graphics/gnu-head-sm.jpg(5286)
	/graphics/gnu-head-mini.png(438)

(f)
Based on the observation, it should be persistent HTTP connection, because for one TCP connection, it requests multiple objects

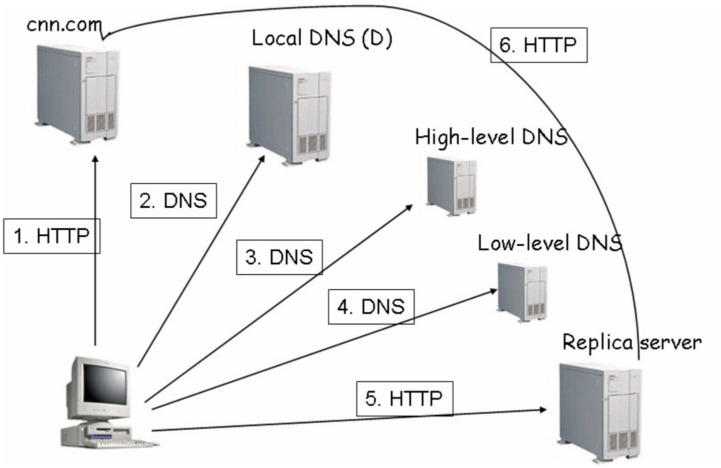
(g)
www.google.com
0.110624-0.066037
0.231138-0.154804
Average = 0.0604605 sec

www.gnu.org
5.752849-5.558597
5.791381-5.745246
5.895037-5.806331
5.855740-5.806383
Average = 0.0946125

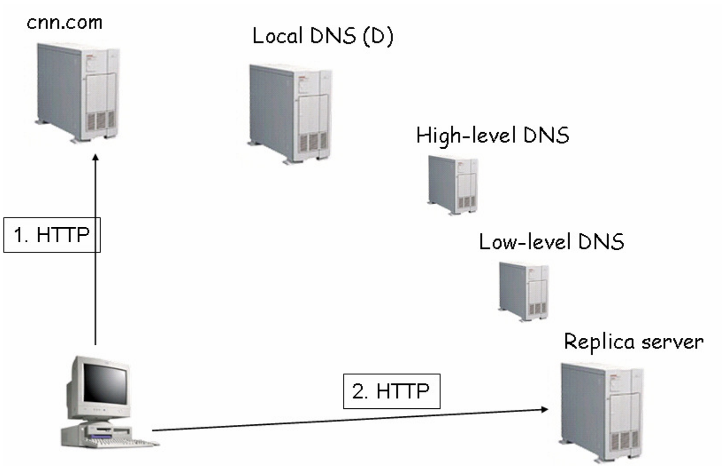
(h)
Packet 21 and 22 contains the DNS request and corresponding reply. From the packet, we can see that they are using UDP protocol in transport layer. The IP address of DNS server is 24.92.226.48. Packet 22 contains the DNS reply, and the IP address for www.gnu.org is 199.232.41.10 based on

the reply.

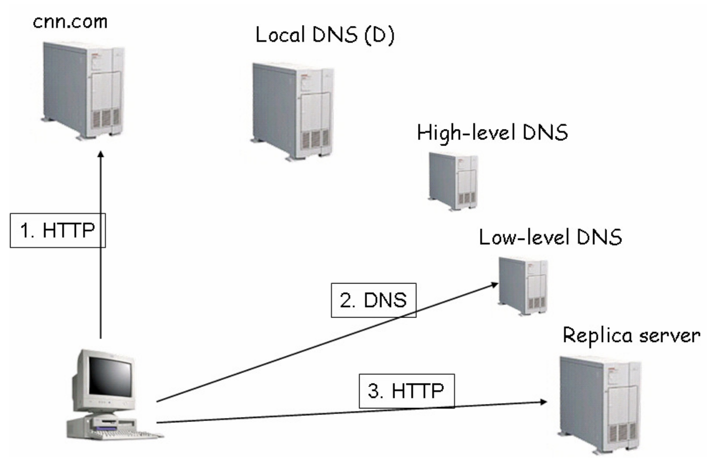
3. (a) The first time A requests index.html from CNN, it needs to get the IP from the local DNS server first. The sequence of the HTTP request and DNS request along the timeline can be drawn in the following figure, where the number represents the request sequence.



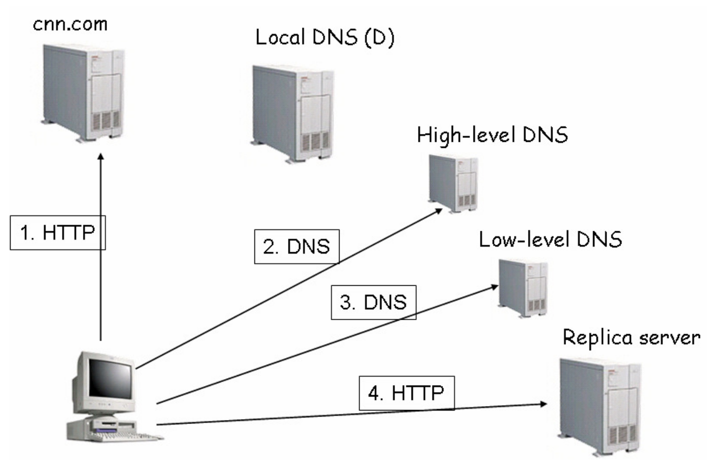
(b) After 10s, the host address is still stored, therefore no DNS needed. In this case, the request sequence can be represented in the following figure.



(c). After 15min, the local address lifetime exceeds the TTL, however, the low level DNS NS record is still effective. Thus, the request sequence can be drawn like following:



(d) After 45 min, the low level DNS NS record also expires, therefore, the request sequence would be



4. (a) Since UDP does not have congestion control, thus they could not lower the transmission rate when congestion occurred. On the other hand, TCP would lower its transmission rate. In this case, most of the bandwidth would be occupied by UDP, which would significantly worsen the TCP performance.

(b) The percentage of successful received packets gives the source the indication of current network congestion conditions. When the percentage is low, the source should lower its sending rate accordingly, and when the percentage is high, it can steadily increase its sending rate until the percentage starts dropping. This mechanism is very similar to that used in TCP like AIMD.

5. (a) <1> To fully utilize the link, we need to ensure that the average transfer amount equals bandwidth-delay product (BDP). Therefore, with ¾ multiplicative decreasing, the buffer size should satisfy 3(BDP+Q)/4>=BDP, thus the minimum buffer size is BDP/3.

<2> If the additive increase is 2 packets per RTT, and multiplicative decrease is still ½, then the buffer size needed would be BDP as in normal case. Essentially, the buffer size needed only relates to the multiplicative decreasing rate regardless the additive increasing rate.

(b) The proposed solutions can be analyzed respectively in the following:
<1>. If we just increase the buffer size to a huge amount, what we may have is a very slow network although with few packet loss. The first reason is that the increasing buffer size may result in long queues and hence long queuing delays during congestion time. Additionally, the transmission conflict would also increase during congestion time, and this could not be relieved by enlarging the router buffer size.

<2>. First of all, increasing the bandwidth is a very costly and inefficient solution. In most of the time, the utilization of the bandwidth is pretty low, and it would be unnecessary to avoid the congestion by large bandwidth. On the other hand, even when we increase the bandwidth, the processing speed and buffer size of routers are still limited, and could not avoid the congestion in nature. Therefore, this costly solution is inappropriate.

<3>. Restriction on the access link would also be ineffective since althougsh it may eliminate the packet loss, the transmission speed would also be largely limited during the normal time. Therefore, for the sake of overall efficiency, it is unwise to limit the bandwidth to avoid congestion.

6. (a) Since the total bandwidth is 50Mbps, originally each flow will be allocated 5Mbps. Thus the first five flows will be allocated 1Mbps, 2Mbps, 3Mbps, 4Mbps, and 5Mbps respectively, since they do not exceeds the limit. Then there would be 35Mbps left for 5 flows. Thus 6th and 7th flow can also be guaranteed. After that 22Mbps will be evenly allocated by the other 3 flows. So the final result would be

No.	1	2	3	4	5	6	7	8	9	10
Size(Mbps)	1	2	3	4	5	6	7	7.33	7.33	7.33

If the total bandwidth increases to 60Mbps, in the similar manner, we can calculate that the allocation would be like the following table:

No.	1	2	3	4	5	6	7	8	9	10
Size(Mbps)	1	2	3	4	5	6	7	8	9	10

which makes sense because currently the resource is enough for all the users.

(b). Suppose we have constant and equal flows for both A and B, and they arrived at the same time, but A was served first. Therefore, based on the WFQ policy, the first 6 packets would be A, B, A, B, A, A or A, B, A, A, B, A