Homework 5

1. 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 RTT1,...,RTTn. Further suppose that the Web page associated with the 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?

The total amount of time to get the IP address is

$RTT_1 + RTT_2 + \ldots + RTT_N$

Time taken to initiate TCP Connection: RTT_0

Time taken to receive object: RTT_0

So total time taken = 2RTT0 + RTT1 + RTT2+ … + RTTN

2. Referring to the problem above, suppose the HTML file references eight very small objects on the same server. Neglecting transmission times, how much time elapses with

(a)Non-persistent HTTP with no parallel TCP connections?

The total amount of time to get the IP address is

$RTT_1 + RTT_2 + \ldots + RTT_N$

Time to receive main html: $2RTT_0$

Time to receive 8 referenced objects: $8 * 2RTT_0$

So total time taken = $16RTT_0 + 2RTT_0 + RTT_1 + RTT_2 + \ldots + RTT_N$

$= 18RTT_0 + RTT_1 + RTT_2 + \ldots + RTT_N$

(b) Non-persistent HTTP with the browser configured for 5 parallel connections?

The total amount of time to get the IP address is

$RTT_1 + RTT_2 + \ldots + RTT_N$

Time to receive main html: $2RTT_0$

Time to receive 8 referenced objects with 5 parallel conn: $2 * 2RTT_0$

So total time taken = $2 * 2\ RTT_0 + 2\ RTT_0 + RTT_1 + RTT_2 + \ldots + RTT_N$

$= 6\ RTT_0 + RTT_1 + RTT_2 + \ldots + RTT_N$

(c) Persistent HTTP?

The total amount of time to get the IP address is

$RTT_1 + RTT_2 + \ldots + RTT_N$

Time to receive main html: $2\ RTT_0$

Time to receive 8 referenced objects with persistent: RTT_0

So total time taken = $RTT_0 + 2RTT_0 + RTT_1 + RTT_2 + \ldots + RTT_N$

$= 3RTT_0 + RTT_1 + RTT_2 + \ldots + RTT_N$

3. **In this problem, we use the useful dig tool available on Unix and Linux hosts to explore the hierarchy of DNS servers. Recall that 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, and answer the following question. 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 cs.binghamton.edu by using dig. List each intermediate name server contacted.**

i) dig +norecurse @b.root-servers.net any cs.binghamton.edu

```
;; AUTHORITY SECTION:
edu.                    172800  IN      NS      c.edu-servers.net.
edu.                    172800  IN      NS      d.edu-servers.net.
edu.                    172800  IN      NS      f.edu-servers.net.
edu.                    172800  IN      NS      a.edu-servers.net.
edu.                    172800  IN      NS      l.edu-servers.net.
edu.                    172800  IN      NS      g.edu-servers.net.
```

ii) dig +norecurse @c.edu-servers.net any cs.binghamton.edu

```
;; AUTHORITY SECTION:
binghamton.edu.         172800  IN      NS      bingnet2.cc.binghamton.edu.
binghamton.edu.         172800  IN      NS      bingnet1.cc.binghamton.edu.
binghamton.edu.         172800  IN      NS      bingnet4.cc.binghamton.edu.
```

iii) dig +norecurse @bingnet2.cc.binghamton.edu any cs.binghamton.edu

```
;; ANSWER SECTION:
cs.binghamton.edu.      3600    IN      SOA     dns1.cs.binghamton.edu. admin.cs.binghamton.edu. 2017050200 1200 300 1209600 3600
cs.binghamton.edu.      3600    IN      NS      bingnet3.cc.binghamton.edu.
cs.binghamton.edu.      3600    IN      NS      bingnet1.cc.binghamton.edu.
cs.binghamton.edu.      3600    IN      NS      dns1.cs.binghamton.edu.
cs.binghamton.edu.      3600    IN      NS      dns2.cs.binghamton.edu.
cs.binghamton.edu.      3600    IN      NS      bingnet2.cc.binghamton.edu.
cs.binghamton.edu.      3600    IN      MX      10 mail3.cs.binghamton.edu.
cs.binghamton.edu.      3600    IN      A       128.226.118.14
cs.binghamton.edu.      3600    IN      A       128.226.118.15
cs.binghamton.edu.      3600    IN      A       128.226.118.12
cs.binghamton.edu.      3600    IN      A       128.226.118.13
cs.binghamton.edu.      3600    IN      TXT     "v=spf1 a:mailers.cs.binghamton.edu -all"
```

4.

(a) In P2P file sharing systems, peers need to locate which other peers have a copy of the desired content. Explain and compare the approaches used in Napster and Gnutella for content discovery.

Napster	**Gnutella**
It involves a central server.	There are no central servers involved.
The entire state or location of the data is managed by the central server.	The state is managed by the nodes instead of central servers
Single point of failure	Highly decentralized
The user queries the centralized server for the location of the data.	The user queries nearby nodes for the location of particular data.
Can scale but hard to maintain state	Cannot scale as queries are flooded

(b)Consider a BitTorrent system with a total of N peers. Each peer has an uploading link with bandwidth capacity u and a downloading link with bandwidth capacity d much greater than u. Among all peers, S of them have finished downloading and voluntarily offer to upload to others.

What is the total aggregated uploading bandwidth within the system?

The total aggregate uploading bandwidth within the system: S * u

For peers that are still downloading, what is the best average downloading throughput that can be achieved?

$$= S * u/ (N – S) * d$$

5. Consider the following text messaging procedure:
Step 1. Alice computes the MD5 digest of message M, call it D.
Step 2. Alice encrypts D using her private key, call the result D'.
Step 3. Alice produces M' = M || D', where notation || stands for concatenation.
Step 4. Alice compress M' by zip, call the result Z.
Step 5. Alice encrypts Z using DES with a randomly selected key K, call the result Z'.
Step 6. Alice encrypts K using Bob's public key, call the result K'.
Step 7. Alice sends K' || Z' to Bob.

(a) (5 pts) How does Bob read the message?

i) Bob separates K' and Z' from K' || Z'
ii) Bob decrypts K' to key K using his private key.
iii) Then Bob decrypts Z' to Z using the Key K generated from above step.
iv) Bob decompress Z to form M'
v) Bob separates M' to form M and D'
vi) Bob reads the message M

(b) (5 pts) How does Bob authenticate the message?

i) Bob tries to decrypt D’ using Alice public key.

ii) If it is successful, It authenticates the message.

(c) (5 pts) How is message integrity ensured?

i) Bob calculates MD5 hash D’’ from the message M

ii) Bob decrypts D’ to D from previous step using Alice public key

iii) Bob checks if D’’ = D. if both are equal, then message integrity is ensured.