Problem 1

Suppose that you walked into Boelter Hall and get connected to CSD WiFi network, which automatically gave you IP address of the local DNS server. Suppose the local DNS server has just rebooted and its cache is completely empty; RTT between your computer and the local DNS server is 10ms and RTT between the caching resolver and any authoritative name server is 50ms; all responses have TTL 12 hours. Suppose the DNS lookup follows iterative searching.

(a) If you try to go to `ucla.edu`, what would be minimum amount of time you will need to wait before your web browser will be able to initiate connect to the UCLA's web server? Suppose the location of UCLA's web server is delegated to ucla.edu name server.

(b) What would be the time, if a minute later you will decide to go to `ccle.ucla.edu`? Suppose `ccle.ucla.edu` happen to be delegated to ucla.edu name server.

Answer:

From the question,
RTT between computer and local DNS server is 10ms.
RTT between the caching resolver and any authoritative name server is 50ms
All response have TTL 12 hours.

a)
RTT from computer -> local DNS server: 10ms
RTT from local DNS server -> Root DNS server : 50ms
RTT from local DNS server -> .edu TLD server: 50ms
RTT from local DNS server -> ucla.edu authoritative server 50ms
Minimum amount of time: 160ms

b)
This time, local DNS server has already cached IP address of ucla.edu, it does not need query Root DNS server.
RTT from computer -> local DNS server: 10ms
RTT from local DNS server -> ucla.edu authoritative server: 50ms
RTT from local DNS server -> ccle.ucla.edu authoritative server: 50ms
Mimimum amount of time: 110ms

Problem 2

Suppose you have a new computer just set up. `dig` is one of the most useful DNS lookup tool. You can check out the manual of `dig` at `http://linux.die.net/man/1/dig`. A typical invocation of `dig` looks like: `dig @server name type`.

Suppose that on April 14, 2021 at 15:35:21, you have issued "`dig google.com A`" to get an IPv4 address for `google.com` domain from your caching resolver and got the following result:

```
; <<>> DiG 9.8.3-P1 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17779
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 4

;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             139     IN      A       172.217.4.142

;; AUTHORITY SECTION:
google.com.             12412   IN      NS      ns4.google.com.
google.com.             12412   IN      NS      ns2.google.com.
google.com.             12412   IN      NS      ns1.google.com.
google.com.             12412   IN      NS      ns3.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.         13462   IN      A       216.239.32.10
ns2.google.com.         13462   IN      A       216.239.34.10
ns3.google.com.         13462   IN      A       216.239.36.10
ns4.google.com.         13462   IN      A       216.239.38.10

;; Query time: 81 msec
;; SERVER: 128.97.128.1#53(128.97.128.1)
;; WHEN: Wed Apr 14 15:35:21 2021
;; MSG SIZE rcvd: 180
```

(a) What is the discovered IPv4 address of `google.com` domain?

(b) If you issue the same command 1 minute later, how would "ANSWER SECTION" look like?

(c) If the client keeps issuing `dig google.com A` every second, when would be the earliest (absolute) time the local DNS server would contact one of the `google.com` name servers again?

(d) If the client keeps issuing `dig google.com A` every second, when would be the earliest (absolute) time the local DNS server would contact one of the `.com` name servers?

Problem 2 continued on next page...

Answer:

a) IPv4 address of google.com : 172.217.4.142

b) The answer section would basically the same, except for the numerical value 139. It will be replaced by 79.
This value is TTL value, which is the time to live of the resource record; it determines when a resources should be removed from a cache. (P132 textbook).
Therefore, as a minute (60 seconds) has passed, 139-60=79 sec.
The TTL of the packet is 60s less. As below:

;; ANSWER SECTION:
google.com 79 IN A 172.217.4.142

c) The earliest (absolute) time the local DNS server would contact google.com name servers again would be 139 seconds after Wed Apr 14 15:35:21 2021 .
I.e, Apr 14 15:37:40 2021

d) The earliest (absolute) time the local DNS server would contact one of the .com name servers would be 12412 seconds from Wed Apr 14 15:35:21 2021.

Problem 3 Textbook p149.

Besides network-related considerations such as delay, loss, and bandwidth performance, there are other important factors that go into designing a CDN server selection strategy. What are they?

Answer:

1. ISP delivery cost.
LSP (logistic service provider).
The content requested by the user have to cross multiple links and ISPs to reach the client system. So it will have delivery cost, the content providers need to pay the ISPs for sending content.
So it needs to be minimized to reduce waste, save costs.

2. Geographically distance.
If the users are far to the servers (e.g., oversea) , it may increase the latency to access the content from the server. I.e., minimize the latency caused by geographical distance.

3. Throughput
The content requested by the subscriber from CDN have to cross multiple links and ISPs to reach the client system. If the throughput provided by one of the links is less than consume rate, it may cause delay.

4. failure: If one of the links may not be able to send content to the user, the data may be lost

Problem 4

If the TCP server were to support n simultaneous connections, each from a different client host, how many sockets would the TCP server need? How many sockets would a UDP server need? Why?

Answer:

TCP server need n+1 socket.
Reason: TCP has a welcoming socket, and it creates a new socket with each simultaneous connections, so the total socket TCP needs is n+1.

UDP server need only one socket.
Reason:
UDP server does not have welcoming socket, it send all traffic into the same one socket, so it does not need multiple sockets.

Problem 5

Please answer the following questions on UDP.

(a) How does a UDP receiving host demultiplex incoming packets to the right application process?

(b) Please describe two scenarios where the UDP segment checksum field value still matches with the computed checksum of received segment but the UDP segment is corrupted.

Answer:

A) UDP has no connection protocol and no state. the transport layer introduced the UDP protocol. In order to identify each application, the UDP protocol defines a port, each application on the same host needs to specify a unique port number, and stipulates that packets transmitted over the network must be accompanied by port number information.

And there are source port# and Destination port# in the header of UDP segment. With these two port number information, UDP is able to send packet to the right application process.

b) checksums can detect most common errors, but not all of them.

- if the packet is altered such that the sum of all the data as 16 bit values remains constant, the checksum will not detect the error.
- If swapping two bits in the same index of different 16-bit data, since the checksum may still be the same as before swapping, the error will not be detected. For example, If the data 0...010001, 0...000001, the most least significant numbers are swapper, the checksum will remains the same.