Problem 1

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 Jan 25, 2023 at 19:00:00, 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 9.10.6 <<>> google.com A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32000
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 4

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

;; ANSWER SECTION:
google.com.             273     IN      A       142.250.217.142

;; AUTHORITY SECTION:
google.com.             55416   IN      NS      ns4.google.com.
google.com.             55416   IN      NS      ns2.google.com.
google.com.             55416   IN      NS      ns1.google.com.
google.com.             55416   IN      NS      ns3.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.         145523  IN      A       216.239.32.10
ns2.google.com.         215985  IN      A       216.239.34.10
ns3.google.com.         215985  IN      A       216.239.36.10
ns4.google.com.         215985  IN      A       216.239.38.10

;; Query time: 5 msec
;; SERVER: 128.97.128.1#53(128.97.128.1)
;; WHEN: Wed Jan 25 19:00:00 2023
;; MSG SIZE  rcvd: 180
```

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

(b) If you issue the same command 2 minute later, how would "ANSWER SECTION" look like?

(c) When would be the earliest (absolute) time the caching resolver 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 caching resolver would contact one of the `.com` name servers?

Problem 2

Suppose that you're tasked with setting up a DNS infrastructure for a large organization. Your colleague suggests using DNS servers that perform recursive queries, while you consider implementing DNS servers that perform iterative queries.

(a) List 2 potential advantages of recursive queries over iterative queries.

(b) List 2 potential advantages of iterative queries over recursive queries.

Problem 3

Suppose your computer is connected to a WiFi network, which gives you the IP address of the local DNS server; however, the DNS Server was just rebooted and its cache is completely empty.

Suppose that the RTT between your computer and the local DNS server is 5ms, and the RTT between the local DNS server and *any* other DNS server is 60ms. Assume the iterated query is used and all responses have TTL of 5 hours.

(a) If you try to visit cs.ucla.edu, what would be the minimum amount of time that you need to wait before the web browser is able to initiate connection to the web server of UCLA CS? (Assume the ucla.edu name server is the authoritative DNS server for cs.ucla.edu)

(b) Using the similar assumption as in part(a), if you try to visit bruinlearn.ucla.edu one minute later, what would be the minimum waiting time?

(c) If you try to visit gradescope.com one minute later, what would be the minimum waiting time? (Assume the gradescope.com name server is the authoritative DNS server for gradescope.com)

(d) Using the similar assumption as in part(c), if you try to visit google.com one minute later, what would be the minimum waiting time?

Problem 4

Recall BitTorrent from lecture. BitTorrent is a popular Peer-to-Peer (P2P) file-sharing application that divides files into small chunks and distributes the downloading tasks among clients. To download a file, a user first retrieves a ".torrent" file, which contains metadata about the desired file, including the addresses of "trackers." These trackers keep track of the peers participating in sharing that particular file. Once connected, the user's client downloads chunks from other peers and simultaneously offers the chunks it has already downloaded for others to retrieve.

(a) Consider a traditional centralized file-sharing architecture where a single server hosts files for clients to download and a fixed bandwidth is used regardless of number of clients.

Compare this with a decentralized Peer-to-Peer (P2P) architecture like BitTorrent. Which architecture can offer faster average download speed when a large group of clients simultaneously need to download the same files? Please briefly explain your reasoning.

(b) Continuing on part(a). Which architecture provides greater resilience against disruptions or failures? Please briefly explain your reasoning.

(c) For BitTorrent, suppose that some clients are sharing a single tracker for a file. What problem may emerge if the tracker becomes unavailable after some of clients have already established peer-to-peer connection and started file sharing? Can the file-sharing process continue? Please briefly explain your reasoning.

(d) Can you think of some potential solutions to handle tracker failure like the scenario in part (c)?

Problem 4 continued on next page. . .

Problem 5

DASH is a modern, adaptive bit-rate streaming protocol. Instead of streaming a video as a continuous flow, DASH breaks the content into a sequence of small chunks, each representing a short interval of playback time. The video is encoded at multiple bit rates, and the client selects the appropriate bit rate for streaming based on network conditions, ensuring smooth playback even with fluctuating bandwidth.

(a) What are some advantages of using an adaptive streaming protocol like DASH compared to a traditional download-and-play scheme with a fixed bitrate? Please list at least two and briefly explain your answer.

(b) Now consider the integration of DASH with Content Distribution Network (CDN). Compare the benefits of streaming DASH content through a CDN versus serving it from a centralized "mega-server". Please briefly discuss at least two points.

(c) Suppose that you are using Netflix to watch your favorite movie, Argo (assume the video url is netflix.com/argo, and the best CDN is us-west-content.netflix.com), briefly explain how to find the best CDN server using DNS step-by-step (in bulletpoints).