1.
a.
142.250.217.142

b.
Something different cause now in local cache?
If it was in cache, same IP address, probably still in A address, maybe TTL different?
No authority or additional section

c.
19:04:33 (19:00:00 + 273 seconds), since that is the TTL

d.
Assuming related to Google's authoritative TTL, 55,416 seconds
Once Google.com expires from cache, we will have to hit Root DNS server -> .com name servers

08:23:36 (19:00:00 + 923 minutes, 36 seconds)

2.
a.
Alleviating the stress of the local DNS server, as it only gets contacted by at most one other DNS server (the Root DNS server)

By requiring each contacted DNS server to call the next DNS server more equally distributes the load.

b.
Alleviating the stress of the root DNS server, which has to always handle responses and call the next DNS server (when it isn't stored in cache). When there is a lot of people using the internet, the root DNS server may get overloaded.

Doesn't require name resolution, as any DNS server can respond with the translation or point local DNS server to the next server to ask.

3.
a.
Computer -> local DNS -> root DNS server -> local DNS -> .edu name server (TLD server) -> local DNS -> UCLA authoritative server -> local DNS -> Computer
5 ms + 60 ms + 60 ms + 60 ms = 185 ms

b.
Since the translation is now stored in local DNS server, just contact local DNS server from computer

5ms

c.
Computer -> local -> root -> local -> .com name server -> local -> Gradescope authoritative server -> local -> computer
5 ms + 60 ms + 60 ms + 60 ms = 185 ms

d.
Computer -> local -> .com name server -> local -> Google.com authoritative server -> local -> computer
5 ms + 60 ms + 60 ms = 125 ms

4.
a.
P2P architecture.
When there's a large group, maybe not all downloading the same files, P2P is far superior, as it is not limited by a fixed bandwidth. The more users, the more resources are required, however each user also acts as a seeder, meaning they can fulfill other users' download requests. This increases the overall bandwidth. This can scale infinitely, as more users join the torrent network.

b.
If there were disruptions, ie a file chunk being downloaded in a P2P stopped or failed, I'm not sure how well the trackers could manage this.
For client and server architectures, if a file download were to fail, the server could either listen for a failure or the client could re request the file. Very simple for

However, due to every peer being a client and server, if one client were to fail, other clients could fill-in and respond to file requests on their behalf.
For a client-server relationship, if the server were to fail, clients would not be able to request any file.

c.
For the files that are currently downloading, they will make it to their client. For any files that haven't been requested yet, even though we know what chunks we need, we need the tracker to tell us which peers have which file chunks. Assuming 1 or more chunks have not started downloading, the file-sharing process can not continue.

d.
Have multiple copies of the same tracker, also known as redundancy, which is a very common practice for servers, to prevent a single-point of failure.

5.

a.

1. Whenever a network becomes congested, say many users are using the same local access network, DASH can automatically drop the bitrate to ensure a smooth video experience. Decreases chance of packet loss.
2. Gives client ability to choose when to request a file chunk, preventing buffer overflow, aka packet loss
3. Enables companies to serve 100 of millions of users by allowing for video files to be stored across many servers, where the user can request from the server nearest them or that has high available bandwidth

b.

1. A single “megaserver” may act as a “single-point of failure”, if the server goes down, no client is able to request any video file. With multiple CDN servers, if one CDN server were to go down, clients that were requesting from that server can request from another one.
2. With many CDN servers, users have the freedom to request from the server closest to them or with the highest available bandwidth, increasing download speeds.
3. With a single server, there is limited bandwidth, especially at 100 of millions of users.

c.

Assuming nothing is stored in any DNS server cache
Assuming iterative query

- Computer -> Local DNS server, check if netflix.com/argo in cache
- Local DNS -> Root DNS server, check if in cache, nope check in TLD next
- Local DNS -> TLD .com server, check if in cache, nope check in authoritative Netflix server next
- Local DNS -> authoritative Netflix server, returns a .manifest file
- Local DNS -> computer, reads .manifest file, gives us a link to CDN server, us-west-content.netflix.com
- Local DNS -> us-west-content.netflix.com, request video and start downloading

For a socket, server has to upload AND user has to download, so total network usage is 2 * file size