Homework2 Solutions

1. a. HTTP (Hypertext Transfer Protocol) is an application layer protocol for web communication, primarily used to transfer web content between servers and browsers.

   HTTP/1 uses a single connection for each request/response, leading to potential latency issues. HTTP/1 is the oldest and simplest protocol but lacks performance and efficiency compared to HTTP/2 and HTTP/3.
   HTTP/2 improves performance with multiplexing and header compression enhancing both speed and efficiency.
   HTTP/3, utilizing the QUIC protocol instead of TCP, excels in high-latency scenarios, enhancing performance and security with further efficiency improvements in modern network environments.

   b. A persistent HTTP connection (HTTP keep-alive) enables multiple requests and responses to use a single network connection, reducing connection setup overhead.
   It minimizes latency by avoiding the need to repeatedly establish connections. Fewer connections save server and network resources. It is ideal for loading web pages with multiple assets like images and scripts.
   In contrast, non-persistent (traditional) HTTP connections open a new connection for each request, leading to higher latency and less efficiency. Non-persistent connections work well for simpler, standalone requests.

   A persistent HTTP connection is ideal for complex web pages, streaming services, and interactive applications where keeping a connection alive reduces latency and enhances user experience.
   Non-persistent HTTP is suitable for simple web pages, resource-constrained servers, and security-sensitive transactions, where the overhead of connection reuse may not be necessary.

   c. HTTP treats each request as a separate event without remembering what happened before, keeping no memory on the server about past interactions.

   Advantages: It makes servers easier to design and handle many users. Stateless servers can serve lots of clients efficiently. Less risk of server issues due to memory problems.

   Disadvantages: It can be challenging to remember what users did in the past. Sometimes, methods like cookies are used, which can slow down requests. Managing state on the client side can be complex for developers, especially in complex applications.

   d. The request-response model in HTTP is a way for clients (e.g., web browsers) to ask for information or actions from servers (e.g., web servers) and receive a reply.
   HTTP Request Components: Method (e.g., GET, POST), URI (e.g., web address), HTTP version, Headers (optional), Body (optional)
   HTTP Response Components: Status code (e.g., 200 OK, 404 Not Found), HTTP version, Headers (optional), Body (optional)

2. a. A cookie in web technology is a small piece of data sent by a web server and stored in a user's browser for tracking and personalization.
   Cookies are limited in size, have expiration dates, and are sent with every request, while other methods like Web Storage and Cache API offer more storage capacity, manual data retrieval, and are better for structured or larger data.

   b. **First-Party Cookies:**
   Used by the website you're visiting.
   Example: Remembering your login.
   Lower privacy concerns because they're site-specific.
   Third-Party Cookies:
   Set by other domains, often for tracking.
   Example: Tracking your online behavior for ads.
   Higher privacy concerns as they can track you across websites without explicit consent.

   c. Web caching is like temporarily storing web content to make websites load faster.
   It is important because it speeds up page loading, reduces server work, saves bandwidth, and helps websites handle lots of users without crashing. This makes websites more reliable and cost-effective.

   d. In web development, browser caching stores web stuff (like images) on your computer to make websites load faster. Conditional GET requests help by checking if these stored things are still okay to use. This saves time and data because you only download what's changed, making websites quicker and using less internet.

3. a. DNS (Domain Name System) is like the internet's address book. Its main job is to change human-readable web addresses (like example.com) into computer-friendly numbers (like IP addresses) so that computers can find and communicate with each other on the internet. DNS also helps balance traffic, manage email, and ensure internet reliability.

   b. The DNS is a distributed hierarchical database. The root of this tree, we have the root DNS servers. The next layer we have the DNS servers that are responsible for all of the .com, .edu or .net domain names and these are known as the Top Level Domain or TLD servers. Then we have the authoritative name servers. These are the servers that have the ultimate responsibility, for resolving names within their domain. For example, wichita.edu, and nyu.edu.

   c. Authoritative DNS Servers hold official records for specific websites.
   Recursive DNS Servers help your computer find the right records by asking authoritative servers and caching answers.
   When you want to go to a website, your computer asks a recursive server, which checks its memory or asks others. It keeps asking until it gets the real answer from authoritative servers. This way, you can find websites using their names.

   d. Time-to-live (TTL) is a numerical value used for data validity and expiration. TTL tells how long a DNS record is valid, measured in seconds.
   It's vital for managing DNS caching and how quickly changes to DNS records spread across the internet.

Short TTLs mean quicker updates but more server load; longer TTLs reduce load but slow updates.

4. a. **Bandwidth Limitations:** One of the primary challenges in streaming video is the limitation of available bandwidth. Bandwidth determines how much data can be transmitted over the internet in a given amount of time. If a user's internet connection lacks sufficient bandwidth to support the video's required bitrates, it can result in buffering, lower video quality, or even playback interruptions.
Latency and Real-time Streaming: Latency, or the delay between when a video is encoded and when it is displayed on the viewer's screen, is a critical challenge, especially for real-time streaming applications like live broadcasts or video conferencing. High latency can cause synchronization issues and affect user engagement.

b. The role of a buffer in streaming video playback is crucial for ensuring a smooth and uninterrupted viewing experience.
The buffer in streaming video playback stores a portion of the video ahead of what's being watched. It compensates for network issues, ensuring smooth playback by preloading data and adapting to variable network speeds.

c. Dynamic Adaptive Streaming over HTTP (DASH) is a streaming protocol that divides video content into small segments, encodes them at multiple quality levels, and uses adaptive streaming to adjust video quality in real-time based on network conditions. DASH relies on standard HTTP for content delivery.
Traditional streaming methods often rely on a fixed bitrate, which can lead to buffering or low-quality playback on variable networks and may involve proprietary or non-standard protocols.
DASH allows for efficient use of bandwidth by only fetching the necessary segments, reducing wastage compared to traditional methods.
Its standardized, efficient, and flexible approach distinguishes it from traditional streaming methods.

d. Content distribution networks (CDNs) can adopt one of two server placement philosophies:
Server-Centric CDN: Content is distributed from centralized servers located strategically worldwide. These servers reduce latency and efficiently handle high traffic, making them ideal for large files.
Peer-to-Peer (P2P) CDN: Content is shared among decentralized end-user devices. Peers exchange content, reducing server load. P2P CDNs are cost-effective and resilient, often used for live streaming and large file distribution.