Group 10

Introduction to Computer Networks

Group Activity #03

Assignment Solutions

SWE3022 - Sungkyunkwan University

1. Network Application Architecture

A. Explain the client-server architecture. Suggest two example applications that use a client-server architecture.

Client-Server Architecture is a network paradigm in which services are provided by a central server and requested by multiple clients. The server is an always-on host with a permanent IP address, usually located in data centers for scalability and reliability. Clients, on the other hand, contact and communicate with the server to request resources or services. Unlike servers, clients may be intermittently connected to the network and often use dynamic IP addresses. Importantly, clients do not communicate directly with each other; instead, all communication goes through the server.

Examples are HTTP for web browsing and IMAP for Email services.

2. Dynamic Adaptive Streaming over HTTP (DASH)

A. Explain DASH in server and client's side.

Server Side:

- Divides video files into multiple chunks (segments)
- Each chunk is stored and encoded at different bit rates/quality levels
- Creates a manifest file that provides URLs for different chunks and their encoding rates

3. Domain Name System (DNS)

A. Explain the Domain Name System (DNS).

Domain Name System(DNS) is a fundamental Internet service that translates human-readable names into IP addresses, which are used by hosts and routers to deliver data. Technically, DNS is a distributed database implemented in a hierarchy of many name servers. It operates as an application-layer protocol, where hosts and name servers communicate to resolve names into addresses (or vice versa). This allows users to use simple names while the underlying network uses numerical IP addresses for routing.

4. Socket Programming

D. Make the description of how the following functions are used in C socket

- The **socket** function creates a new socket, which becomes the endpoint for communication. It returns non-negative socket descriptor on success, -1 on failure.
- The **connect** function is used by a client to initiate a connection to a server by specifying the server's IP address and port number.
- The **bind** function assigns a local IP address and port number to a socket, usually on the server side, so that incoming requests can be directed properly.
- The **listen** function places the socket into a passive, listening state, which prepares the server to accept incoming connection requests.
- The **accept** function acknowledges an incoming client request and creates a new socket dedicated to communication with that client. The original listening socket remains available to wait for additional connections. It returns new non-negative socket descriptor