02. Application Layer

2.1 Principles of Network Applications

There are two main ways for applications to communicate across devices, client-to-server and peer-to-peer. Most web applications are client-to-server and are often viewed as more secure, as peer-to-peer leaks a lot of data about the other participant (such as IP address).

The transport layer protocol in use also has a significant meaning, because unlike UDP, TCP guarantees that all packages arrive at the destination. Which protocol we decide to use depends on what we're actually doing. It might be okay to lose a few packets when streaming a movie.

Each end system application communicates with the computer network through sockets. Sockets are interfaces that emit and receive messages. When my browser visits a webpage, it opens and writes the HTTP request to a network socket. The socket is bound to a port on the end system which is how the IP protocol is able to determine where the receiving packets are supposed to go.

2.2 The Web and HTTP

The HyperText Transfer Protocol is the web's application layer protcol. A web page consists of objects. An object being a file or asset that is addressable by a single URL. When the browser requests a webpage, it sends a HTTP GET request to the server which returns the document. Any linked objects are sent in individual requests after the initial document has been returned.

Because HTTP is stateless, it might be useful or necessary to send cookies alongside the HTML document. The HTTP protocol uses TCP for its transport layer protocol because you cannot afford losing parts of the HTTP response.

HTTP 1.1 and later support both persistent and non-persistent connections. Without persistent connections, each object requested by the client has to open and maintain a new TCP connection. By keeping the connection alive, the client may send multiple HTTP requests over the same TCP connection.

A web cache is typically a proxy server which caches HTTP responses from the server it's proxying in front of. This means quicker response times for consequent requests to

the same object. Typically the web cache is installed at the ISP level. A university may install a cache on its network and configure all browsers on campus to point to that cache, making consequent requests faster for everybody on campus.

2.3 Electronic Mail in the Internet

Simple Mail Transfer Protcol (SMTP) is a protcol for sending emails. The protocol uses 7-bit ASCII which makes it a little bit difficult to send unicode and more modern text.

The protocol works with my client sending a SMTP request to my mail server, which forwards the email to your mail server. Your mail client is then able to retrieve the email. Your client accesses the email using protocls such as POP3, IMAP or even HTTP.

2.4 DNS — The Internet’s Directory Service

Domain Name System (DNS) is the service that translates human-readable domain names into a server’s IP address. Querying an IP address through DNS works through a chain. My client asks the nearest DNS server if they have the domain cached. If it doesn’t, that DNS server will ask the next DNS server in line. DNS runs over UDP.

DNS is used in any service where domain name lookup is required, including email, webservers or FTP servers.

2.5 Peer-to-Peer File Distribution

Peer-to-Peer (P2P) networks are networks where all clients also act as servers, and clients communicate directly to eachother. As of today, the most popular P2P protocol is BitTorrent.

Upon connecting to a torrent, the client registers itself to the tracker and periodically informs the tracker that it’s still connected. The tracker returns a TCP connection which is shared among all the participants in the network.

2.6 Video Streaming and Content Distribution Networks

Today, video streaming has become a huge industry. HTTP streaming is one method to stream video across the web by simply storing the video at an HTTP server as an ordinary file. The problem with this approach is that all clients receive the same encoding which might not be optimal. A mobile user has different requirements than a desktop user.

Dynamic Adaptive Streaming over HTTP (DASH) is the go-to solution. In DASH, the video is encoded into several different versions, with each version having a different bitrate and quality level. Depending on the bandwidth of the client, an approriate version is selected.

To keep video streaming responsive with low latency, Content Distribution Networks (CDNs) are used. These are servers which are geographically located in different places of the world. The user will therefore get the video from the geographically closest CDN to keep latency low. It’d be a terrible experience to stream video from an American server in Australia.

2.7 Socket Programming: Creating Network Applications

Network applications are created by programming on network sockets. Sockets are streams where one system can send byte streams to another over a transport protocol such as TCP or UDP.

When developing a network application, the developer writes both the client and the server code. The developer can use an “open” and standardized protocol such as HTTP, or they can write their own properitary one. They also have to choose whether the protocol should run over TCP or UDP. If the developer uses a standardized open protocol, they should also use the correct port numbers defined in the RFCs for said protocol.

It’s important to note that the developer has full control over the application layer (duh, they’re writing it), but they have little control over the transport layer apart from choosing transport protocol. A server is analogous to a house, where the door is a contact point for clients. This means the server has to be running before any clients can connect.

When developing with UDP sockets, the client specifies the server IP and port number of the server process. With the client data, the IP and port the server should respond to is also provided (typically automatically by the operating system, not client application code).

When developing over TCP, the client and server first need to handshake and establish a connection. Unlike the UDP connection, the TCP server simply responds back into the socket established in the connection (instead of sending a new message over UDP).

The house anology is also slightly different for TCP, as establishing the handshake is more analogous to knocking on the door first.