Austin Melendez
302368430
CSC138 - Section 03

Homework 2

1)

a) The client-server architecture maintains a permanent IP address with always-on host capability. This type of architecture is often used in data centers, which house a large number of hosts and are often used to create a powerful virtual server.

b) The peer-to-peer architecture does not maintain a permanent IP address because as users download the files they become another source for a different user to download from. The means peers are intermittently connected and change IP addresses. In this model peers request service from other peers, who then provide service in return to new peers.

c) In our day-to-day application usage we use client-server architecture more often, because large businesses and applications are able to pay for and maintain the infrastructure needed. This architecture type also provides more security. Most social networks, email, websites, video games, video streaming, and voice-over IP apps use this. However, if the size of the project needs to be extremely scalable at a minimal cost, peer-to-peer architecture can be used. Applications like torrenting services or file sharing services use peer-to-peer because there is no need for reliance on dedicated servers reducing cost while still allowing for millions of users.

2) Within the same host (a single machine), two processes communicate using the inter-process communication enabled by the rules that are governed by the operating system. When two processes are on different hosts (between two machines) they need to communicate by exchanging messages across a network. A sending process creates and sends messages into the network, where a receiving process receives these messages and possibly responds by sending messages back. If messages need to be sent both ways a socket can be used, the Application Programming Interface (API) is the interface between the application layer and the transport layer within a host.

3) Based on the constraints given I would choose the UDP protocol to provide the absolute fastest and lowest latency connection between users. Hopefully the speed of communication makes up for the relative unreliability of the signals, but because there is going to be such a large amount of data sent between players we should prioritize speed to make the fastest "real-time" updates to each player's screen.

4) The server knows how to tell the difference between the two applications due to the port numbers. Although the IP address of the machine will be the same, each host process is dedicated to a specific port number. The port number is what identifies the process or application that created the process.

5) $F = 10$ Gbits $N = 100$ peers

Upload $\underset{\text{server}}{u_s} = 20$ Mbps Download $\overset{\text{client}}{d_i} = 3$ Mbps

Upload $u = 500$ Kbps or 800 Kbps

a) Client-Server

$$D_{c\text{-}s} \geq \max \{ NF/u_s, \; F/d_{min} \}$$

$$D_{c\text{-}s} \geq \max \left\{ \frac{10000\,\text{Mbit} \cdot 100}{20\,\text{Mbps}}, \; \frac{10000\,\text{Mbit}}{3\,\text{Mbps}} \right\}$$

$$\max \{ 50000_{sec}, \; 3333.3\bar{3}\,sec \} = \boxed{50000_{sec}}$$

b) Peer-to-peer

$$D_{p\text{-}p} \geq \max \{ F/u_s, \; F/d_{min}, \; NF/(u_s + \sum_{i=1}^{N} u_i) \}$$

$$D_{p\text{-}p} \geq \max \left\{ \frac{10000\,\text{Mbits}}{20\,\text{Mbps}}, \; \frac{10000\,\text{Mbits}}{3\,\text{Mbps}}, \; \frac{10000\,\text{mbits} \cdot 100}{(20\text{Mbps} + .5\text{Mbps}(100))} \right\}$$

$$\max \{ 500\,sec, \; 3333.\overline{33}\,sec, \; 14285.71429\,sec \}$$

$$\boxed{= 14{,}285.71\,sec}$$

$$D_{p\text{-}p} \geq \max \left\{ 500\,sec, \; 3333.\bar{3}\,sec, \; \frac{10000\,\text{mbits} \cdot 100}{(20\text{Mbps} + .8\text{Mbps}(100))} \right\}$$

$$\max \{ 500\,sec, \; 3333.\bar{3}\,sec, \; 10000\,sec \}$$

$$\boxed{= 10{,}000\,sec}$$

(a) AVG response = access delay + internet delay

$$= \frac{\Delta}{(1-\Delta\beta)} + \text{internet delay}$$

$$= \frac{(23\ \text{req/sec})(.06\ \text{Mbps/req})}{1.5\ \text{Mbps}} = 0.92 \quad \text{intensity}$$

$\frac{La}{R} \to 1$ queueing delay large!

AVG = minutes + 2 sec

Δ = avg time $\frac{.06}{1.5}$ = .04 sec

β = arrival rate = 23 req/sec

$$= \frac{\Delta}{(1-\Delta\beta)} + 2\text{sec}$$

$$= \frac{.04}{(1-.04(23))} + 2\text{ sec} =$$

$$= .5\text{ sec} + 2\text{ sec} \quad \boxed{= 2.5\text{ sec}}$$

b) Miss rate = 0.4

= 0.6 · (server delay) + 0.4 (cache delay)

= 0.6 · (2.5 sec) + 0.4 (0.1 sec)

$$\boxed{= 1.54\text{ sec}}$$

7) When you first join a torrent, the new peer gets registered with the tracker. The tracker randomly selects a subset of peers from the set of participating peers. Using the IP addresses of this subset, the new user attempts to establish TCP connections with all the peers on the list. Successful TCP connections are called “neighboring peers”. At any given time each peer will

have a subset of chunks from the file, so the new user will request a list of the chunks the neighbors have, then issue requests for the chunks the new user does not currently have. Periodically the new user will also choose a random new trending partner, and if the speed is faster than that of the top four uploaders, they will continue trading with each other.

8) DNS uses UDP instead of TCP because it favors speed over absolute reliability. The speed and efficiency allows for the quick delivery of DNS queries and responses and the processing of small but time-sensitive data transfers. DNS query and response messages don't require the constant end-to-end connection of TCP, and instead want maximum speed so hosts and DNS servers can communicate to resolve hostnames quickly.