Homework 2 Solutions

1. (6 points) Answer the following:
    a. Identify the architecture where a server with permanent IP address has always-on host capability to provide service. Provide an example of the application of this model.

       Answer: Client-Server model. Web server.

    b. Identify the architecture where a server with temporary IP address host file services does not have the always-on-host capability to share files. Provide an examples application of this model.

       Answer: P2P model. Bittorrent

    c. Which architecture do you find ideal in your day-to-day application usage? Give some examples.

       Answer: Client-server model, social media, whatsapp,

2. (4 points) Differentiate between Inter-process communication (IPC) in multi-machines versus the single machine using examples? What programming interface do you use for IPC between two hosts?
   Answer: IPC is used between multiple machines in networked environment such as client-server model. IPC is used in single machine for one process to communicate with another, such as reader writer problems. Sockets are used as programming interface.

3. (5 points) Imagine you are designing a communication system for a real-time online gaming application. Players need to exchange information about their current positions, actions, and updates in the game environment. Additionally, the system should prioritize low latency to ensure a smooth gaming experience.

   Based on the given scenario, analyze the requirements of the online gaming application and propose the most suitable transport protocol for the communication between players. Justify your choice by considering factors such as reliability, latency, and the nature of the data being transmitted.

   Answer: UDP transport protocol. The application requires speed over reliability. The reliability can be provided by other protocol. TCP is very heavy and will not be suitable for this reason. UDP also ensures low latency. Real-time communication requires UDP protocol for faster communication.

4. (5 points) Suppose that you have two browser applications open and active at the same time and that both applications are accessing the same server to retrieve HTTP documents at the same time. How does the server know how to tell the difference between the two applications (to send the correct document(s) to the correct application)?

Answer: A client application generates an ephemeral port number for every TCP connection it sets up. An HTTP request connection is uniquely specified by the five parameters: (TCP, client IP address, ephemeral port #, server IP address, 80). The two applications in the above situations will have different ephemeral port #s and will thus be distinguishable to the server.

5. (10 points) Suppose that we want to distribute a file with a size of $F = 10$ Gbits to $N = 100$ clients/peers. The server supports an upload rate of $u_s = 20$ Mbps while each client/peer has a download rate of $d_i = 3$ Mbps and an upload rate of u, where $u = 500$ Kbps or 800 Kbps. Show your calculations.

   a. Calculate the minimum distribution time for a client-server distribution using the two values of u given above.

      Answer: For calculating minimum distribution time for client-server distribution, we use the following formula, $D_{c\text{-}s} = max\ \{NF/u_s, F/d_{min}\}$

      NF/u_s=(100×10×1024^3)(20×1024^2)=51200

      F/d_{min}=(10×1024^3 / 3×1024^2)=3413.33

      For u=300 Kbps or 700 Kbps, the minimum distribution time remains the same.

      $D_{c\text{-}s}$ =max{NF/u_s,F/d_{min} }=max{51200,3413}=51200.

   b. Calculate the minimum distribution time for a peer-to-peer distribution using the two values of u given above.

      Answer: For calculating the minimum distribution time for P2P distribution, we use the following formula: $D_{p2p} = max\ (NF/u_s, F/d_{min}, NF/u_s + \sum_{i=1}^{N} u_i)$

      $U = 500\ Kbps$

      $F/u_s = (10 \times 1024^3) / (20 \times 1024^2) = 512$

      $F/d_{min} = (10 \times 1024^3 / 3 \times 1024^2) = 3413.33$

      $NF/(u_s + \sum_{i=1}^{N} u_i) = (100 \times 10 \times 1024^3) / ((20 \times 1024)^2 + (100 \times 500 \times 1024)) = 14877.6$

      $D_{p2p} = max\ (F/u_s, D/d_{min}, NF/u_s + \sum_{i=1}^{N} u_i) = max\ (512, 3413.33, 14877.6) = 14877.6$

      $U = 800\ kbps$

      $F/u_s = (10 \times 1024^3) / (20 \times 1024^2) = 512$

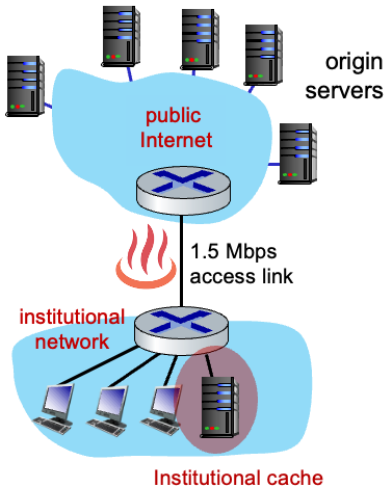
      $F/d_{min} = (10 \times 1024^3 / 3 \times 1024^2) = 3413.33$

      $NF/u_s = (100 \times 10 \times 1024^3) / (20 \times 1024^2 + 100 \times 800 \times 1024) = 10435.66$

$D_{p2p} = max\ (\ F/u_s,\ D/d_{min},\ NF/u_s + \sum_{i=1}^{N} u_i) = max\ (512,3413.33,10435) = 10435.66$

Note: Your answers will be considered correct if you use 1000 instead of 1024 to convert kilobits into bits.

6. (10 Points) Consider the following simplified network diagram where there is an institutional network connected to the Internet:



Suppose that the average object size is 60,000 bits and that the average request rate from the institution's browsers to the origin server is 23 requests per second. Also, suppose that the amount of time it takes from when the router on the Internet side of the access link forwards an HTTP request until it receives the response is two seconds on average. Model the total average response time as the sum of the average access delay and the average Internet delay. For the average access delay, use $\Delta/(1-\Delta\beta)$, where Δ is the average time required to send an object over the access link and β is the arrival rate of objects to the access link. You can assume that the HTTP request messages are negligibly small and thus create no realized traffic on the network or the access link. Show your calculations.

a. Find the total average response time.

Answer:

Δ = transmission delay = $60000 / 1.5 * 10^6 = 0.04$

β = arrival rate = 23 requests/ sec

Average access delay is given by $\Delta/(1-\Delta\beta)$,

$\Delta/(1-\Delta\beta)$, = 0.04 / (1 – (0.04 * 23))

= 0.04/ (1 – 0.92)

= 0.04 / 0.08 = 0.5 sec

Total delay = Internet delay + average access delay

$= 2 + 0.5$

$= 2.5$ sec

b. Now, suppose a cache is installed in the institutional LAN (see figure). Suppose the miss rate is 0.4. Find the total response time.

Answer:

Since the miss rate is 0.4, assume web-cache delay is 0.

Web-cache delay = 0.4 (time of response) = 0.4 * 0 = 0

Rest of the response takes 70% , for which we use $\Delta/(1 - 0.6 * (\Delta\beta))$.

Average access delay = $\Delta/(1 - 0.6 * (\Delta\beta))$

= 0.04/ (1 – (0.6*0.92))

= 0.04 / 0.448 = 0.089

Total delay = Internet delay + average access delay

= 2 + 0.089

= 2.089 sec

7. (5 points) Suppose that you join BitTorrent as a new peer without possessing any chunks. Unfortunately, you cannot become a top-4 uploader for any of your peers since you do not have anything to upload. Describe how you will be able to get your first chunk. Be specific.

Answer: You will get your first chunk as a result of being selected by one of your neighbors as a which is called “optimistic unchoke” for sending out chunks to you. Recall that a peer periodically selects one of its neighbors at random as a peer for uploading irrespective of whether this neighbor is uploading data to it or not.

8. (5 points) Identify at least one reason why DNS uses UDP instead of TCP for its query and response messages.

Answer: One key reason is that DNS queries and responses are relatively small and can fit within a single UDP packet. UDP is a connectionless protocol, meaning it does not establish a persistent connection before transmitting data. In the case of DNS, this allows for faster communication since there is less overhead associated with connection establishment and teardown compared to TCP.

DNS transactions are typically lightweight, involving a single query and a corresponding response. If DNS were to use TCP for every query, the additional connection setup and teardown processes could introduce unnecessary latency, especially for quick, simple queries where the benefits of a reliable, connection-oriented protocol like TCP may not be essential.