Test # 1

CSCI-3400 – Spring 2023

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Instruction: Write clearly and give full justification to each question. Show all your MATHEMAICAL working.**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Name: \_\_\_\_\_\_\_\_Tyler Burleson\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**E#:\_\_\_\_\_\_\_\_\_\_\_E00672866\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |
| --- | --- | --- |
| Questions | Max Points | Earned Points |
| 1 | 10 |  |
| 2 | 10 |  |
| 3 | 10 |  |
| 4 | 10 |  |
| 5 | 10 |  |
| 6 | 10 |  |
| 7 | 10 |  |
| 8 | 10 |  |
| 9 | 10 |  |
| 10 | 10 |  |
| Total | **100** |  |

**IMPORTANT**

* You can use online references, but **write answers in your own words**!  **Cite any references used**. Answer the question in your own words, no credit will be given for answers copied from any source*.*
* No collaborating with other people; a **0** will be given if any collaboration evidence is found.
* I am looking for very specific, detailed, correct, and complete answers.
* Most answers found on the Internet (especially Wikipedia) are generic answers for people without any networking background and are not any of the above! Research each problem completely.
* Turn in a Word document or pdf into the Dropbox.

1. Explain, why the overall file distribution time is much less in P2P architecture. Use figure below:

To further understand the comparison between distribution time and user with P2P and Client servers we need to understand how they both work. A client server is a method where clients connect to a server containing the information they desire. This information could be text, videos, and other types of large files. The key part is that there is only a connection between the server and the client. The server can host multiple clients, but the clients are only connecting to one source containing their desired file. This means when a file is distributed it relies on the speed of the server’s upload and the client’s download.

Let's start by saying upload speed is not an issue, but download speed varies between clients. The number of clients will be denoted as “n”, upload speed from the server is denoted as “Us”, and the file size is denoted by “F bits.” Under perfect conditions the equation is represented as (n \* F/Us) sec…. equation - 1 , but with multiple users we have to take download speed into consideration. These download speeds will be denoted as “dn,” so the minimum download speed is represented as Dmin = min (d1, d2, d3, … ,dn). To download the file on the slowest client is now, (F/Dmin) sec …. equation-2. With the following observations we can now determine the time it would take the server to distribute the file to the clients. We’ll use (D c-s) ≥ max {n \* F/Us, F/Dmin}sec to represent the distribution time for client-server. This then means max {n \* F/ Us, F/Dmin} is our lower bound or the minimum time for the file to be distributed to all the clients. This now means that our time will scale up depending on how many users are trying to access at the same time, hence the linear correlation on the graph in the client-server section.

Now that we understand client-server we can now learn P2P and why it’s more effective. The key part that makes P2P stand out compared to client-servers is its scalability and no real need for a dedicated server. For my mathematical formulas we’ll assume there is still a server holding the original file and so are the other peers. This means the server starts our connection, but isn’t solely responsible for all communication. In P2P We have multiple links connecting each source to each other all together. Once again the number of clients is denoted as “n,” file size is “F bits,” server upload is “Us.” The new features are the connections between users, so now we also have to upload alongside the download. Client upload will be denoted as “Un” and download respectively as “dn.” Just like earlier we’ll keep the same equation for the conditions to upload from the server (F/Us) and the slowest client is still represented in our minimum download time (F/Dmin). Our new piece is the correlating upload speed from all other clients since we also have connections to them to assist in obtaining the entire file, this will be represented as Utotal = Us + U1 + U2 + … + Un. With all of our other peers our new distribution time will be no less that (n \* F/Utotal.) Now that all our equations have been established we can now calculate our minimum distribution time, (DP2P) ≥ max {F/ Us, F/Dmin, n \* F / Utotal}. Now if we compare the two minimum distributions we will notice that the more clients in the P2P architecture help keep the minimum time considerably lower than the client-server, since the more peers that have the same file will help distribute it to the desired client.

GeeksforGeeks. (2022, November 22). How P2P can scale by&nbsp;itself. GeeksforGeeks. Retrieved February 18, 2023, from https://www.geeksforgeeks.org/how-p2p-can-scale-by-itself/

Kurose, J. F., &amp; Ross, K. W. (2017). In Computer networking: A top-down approach (7th ed.). essay, Pearson.

1. Explain how packet-switched and circuit switching networking works. What are the advantages and disadvantages for each? Give example, and explain how either one is implemented in the example you have given.

There are two fundamental methods of transferring data across a network link and a network switch, these are called packet switching and circuit switching. Each one has its own characteristics and differences, so let’s start with packet switching.

In a network application, end systems send messages to each other. These can contain any type of information and perform all sorts of actions as well, but we need a method of connecting them to each other. In a packet switching environment we break up the information into groups called packets. These packets are then transmitted across the links till they reach their desired destination, but how are these packets managed?

Let’s start with some variables to represent each piece of the question. The source sending the packet will be represented as “L bits” being sent over a link with a transmission rate “R bit/sec”, these put together give us a time to transmit a packet “L/R.” Now that we have our equation we can start looking at pros and cons of packet switching. In a majority of packet switching environments they use “store-and-forward transmission.” This means that when a packet is being transmitted it won't be pushed to the next link until the entire packet has arrived. This is a pro, so that we can ensure all the information is intact before it begins transmitting, but becomes an issue for delay. Since the initial packets are stored in the buffer until they have all arrived we can run into queuing delay and packet loss. If the original packet is sent from the source at L/R then reaches the destination at L/R, but this rate will scale for each packet. This means when the 2nd packet is sent after the first packet has arrived it is now at time 2L/R, then 3L/R, and P\*L/R for each remaining packet. As these packets funnel into the buffer the system will run into queuing delay as each packet is waiting for its turn to be transmitted, and in worse scenarios if the buffer becomes too full any remaining packets coming in will be lost.

Our next method is called circuit switching, this is the method of sending information over an established link to the end source. A common example of circuit switching is a telephone line. When users want to communicate they reserve a set of circuits connecting them to the desired destination. A pro is that this connection then stays on the entire time while the transmission is going back and forth with each other and does not have to wait in between buffers like its packet switching counterpart.

A few issues with circuit switching is when we look at transmission speeds, since each network switch wants to be able to have multiple connections at the same time it has to divide its overall speed between each circuit. If the section has 4 circuits and an overall connection speed of 4mbps this then means one circuit will have a speed of 1mbps. Another issue also discussed is the waste of resources during “silent periods.” When there is no active transmission going on the circuit is sitting idle consuming resources and can’t be rededicated compared to packet switching.

Kurose, J. F., &amp; Ross, K. W. (2017). In Computer networking: A top-down approach (7th ed.). essay, Pearson.

1. Suppose users share a 10 Mbps link. Also suppose each user requires 300 kbps when transmitting, but each user transmits only 10 percent of the time.
2. When circuit switching is used, how many users can be supported?

300kbps = .3mbps  
10(Link capacity)/.3(transmission amount)=33.33…(users)  
**33 users can be supported**

1. For the remainder of the problem, suppose packet switching is used. Find the probability that a given user is transmitting.

Diagram

Description automatically generated with medium confidence(Symbols from WolframAlpha)

**Probability that a given user is transmitting = 0.0309032**

1. Suppose there are 100 users. Find the probability that at any given time, exactly *n*  users are transmitting simultaneously.

Text

Description automatically generated with low confidence (Symbols from WolframAlpha)

1. Answer the following questions with details.
2. Explain how queuing delay can affect the overall networking performance. Illustrate your answer with the help of a figure.

There are several types of delay in a network such as processing delay, queuing delay, transmission delay, and propagation delay. The one we’re concerned about is Queuing delay and how it can affect the network. Queuing Delay is the delay a network faces as its packets are funneled into a link waiting for transmission. Depending on the speed of the link a packet may be forced to wait microseconds to milliseconds to travel across the link.

A common term for this is called a bottleneck, this is when the throughput on the previous link is higher than the next one. This can cause back-ups depending on if there are no other routes possible for the packet to travel. Even if there are other routes possible if there are the same speed as the previous link it will still slow down the entire process. If they are faster than the slower link it will still cause back-up no matter where since the faster links will get congested if they are only used and the slower link can’t match the same output as the others.

Another harmful effect from queuing delay could be the potential of packet loss. If the buffer between each link is not capable of sending the packets out fast enough and the buffer becomes full the packets will be lost and dropped from the system.

A mathematical example of the harm of queuing delay is how the delay is relative to the packets. If 10 packets enter the queue at the same time the first packet will have 0 delay, but the last packet will have the delay of 9 other packets while it waits through the queue. That's why when calculating queuing delay one typically will take statistical measurements: average queuing delay, variance queuing delay, and probability that the queuing delay will exceed a specified value

Inspect the following illustration as further evidence:

A white board with writing on it

Description automatically generated with medium confidence

Kurose, J. F., &amp; Ross, K. W. (2017). In Computer networking: A top-down approach (7th ed.). essay, Pearson.

1. Equation given below gives the end-to-end delay of sending one packet over length *L* over *N* links of transmission rate *R.* Generalize the formula for sending P such packets back-to-back over *N* links.



N\*L/R is the first packet as it reaches its destination

N\*L/R +L/R is the second packet as it reaches the destination

This means the formula for sending P packets over N links N\*L/R+(P-1)\*L/R simplifies to:

(N+P-1)\*L/R

Kurose, J. F., &amp; Ross, K. W. (2017). In Computer networking: A top-down approach (7th ed.). essay, Pearson.

1. Sketch the operation of your protocol for the case of a student “A” appealing for a grade change at a University. Use the same designing concept as you have used in your homework problem. Explicitly state the assumptions made by your protocol.

Diagram

Description automatically generated

1. Explain the concept of packet loss, and how it is related to the four types of delays.

When discussing a queue it’s good to remember they contain finite limits to how many packets may be contained in the buffer at one time. This means if that buffer is allowed to reach capacity a packet will be lost or dropped. This packet can then be decided by the source end machine to either attempt to resend the same packet or allow it to drop and stop attempting to send it.

This buffer can fill up due to a few forms of delay, these include processing delay, queuing delay, transmission delay, and propagation delay. All these forms then add up together to become our full Delay also represented by “dnodal = dproc + dqueue + ddtrans + dprop ,” so how do each of these delays correlate to packet loss?

There are several features that can cause “Processing delay.” These are the features when examining a packet before it is transmitted to the desired destination. These include the time to examine the header and determine where to direct the packet, the time needed to check for bit-level errors that may have occurred during transmission, and other validation methods to ensure the packet has been received correctly before it can be allowed to be transmitted. All these validation add up time before the packet can even attempt to leave the buffer.

The next form of delay is queuing delay, this is the delay of packets sitting in queue waiting to be transmitted. What's difficult about this form is that it is relative for each packet in the queue. As the packets funnel into the system the first packet has 0 delay since there is nothing in front of it, but if 10 packets come in together the 10th packet has to deal with the delay of 9 packets while it waits to be transmitted.

Transmission delay is the next form of delay when considering packet loss. This is the delay caused by the speed of how long it takes packages to be pushed out into the link, or transmitted. The first piece is that all packets in the package must arrive before the system is allowed to try and transmit this packet length is denoted as L in our equation. The second piece is the transmission rate denoted as R. An example of R is the speed of the link in between routers. If the speed is 10 Mbps then R = 10 Mbps, if its 100 Mbps then R = 100 Mbps. We then calculate transmission delay with the equation L/R. and that delay is then added to our total delay.

The last delay is propagation delay, the first portion is when the bit has finally been pushed to the link and how far it must travel to reach its destination. This is denoted as *d* in our equation. The next portion is the propagation speed and it depends on the physical medium of the link such as twisted pairs compared to fiber optics, this is denoted as *s* in our equation. When calculating this delay we then follow *d/s* and add it to our delay system.

With these forms of delay added together we now have our total delay. The issue is when the rate of these delays add up higher than the rate of the buffer. If the buffer is unable to push out more packets than it receives it will finally over flow and result in packet loss.

Kurose, J. F., &amp; Ross, K. W. (2017).In Computer networking: A top-down approach (7th ed.). essay, Pearson.

1. In the internet protocol stack, explain functionality of each layer and give example of which protocol lives on each layers.

Let’s start with what our layers are for the internet protocol stack. This stack compared to the OSI model only contains 5 layers, these are the application layer, transport layer, network layer, link layer, and the physical layer. Each of these layers have their own special roles and protocols they use when we are transferring information from system to system, so let’s start with the application layer.

The application layer is the section used for network applications and their corresponding protocols. This layer is used alongside protocols like HTTP, SMTP, and FTP to exchange information or messages across other network applications in other end systems.

The next layer is called the transport layer. This section is dedicated to transporting application-layer messages to other application-layer endpoints. There are only two protocols in this layer and they are TCP and UDP. These protocols help deliver the messages across endpoints, with TCP the messages are broken into segments and provide flow control, congestion control, and reliability, but UDP will provide a connectionless service that does not provide reliability or other control functions like TCP. The packets in this layer are also considered as a segment.

The third layer is the Network layer and its functionality. This layer is responsible for moving datagrams from one host to another. This layer provides the segments from the transport-layer and a destination similar to a letter and delivers the segment to the destination host. This layer’s protocols are the IP protocol and other types of routing protocols to ensure the segments reach their destination.

The fourth layer is the Link layer and its protocols Ethernet, WiFi, and DOCSIS. This layer is used alongside the network layer to help ensure datagrams are passed to the correct node then back up to the network layer using one of the protocols like ethernet or WiFi. Finally the packets in this layer are also referred to as frames.

Finally the last layer is known as the Physical layer. The job of this layer is to work with the link layer to move the individual bits in the frame from one node to the next node. The protocols in this layer are similar to the link layer and vary across the medium of the link, such as fiber optics, and twisted pairs.

Kurose, J. F., &amp; Ross, K. W. (2017).In Computer networking: A top-down approach (7th ed.). essay, Pearson.

1. Explain in detail, how is P2P file distribution concept is used in Bit Torrent. Illustrate your answer with a figure.

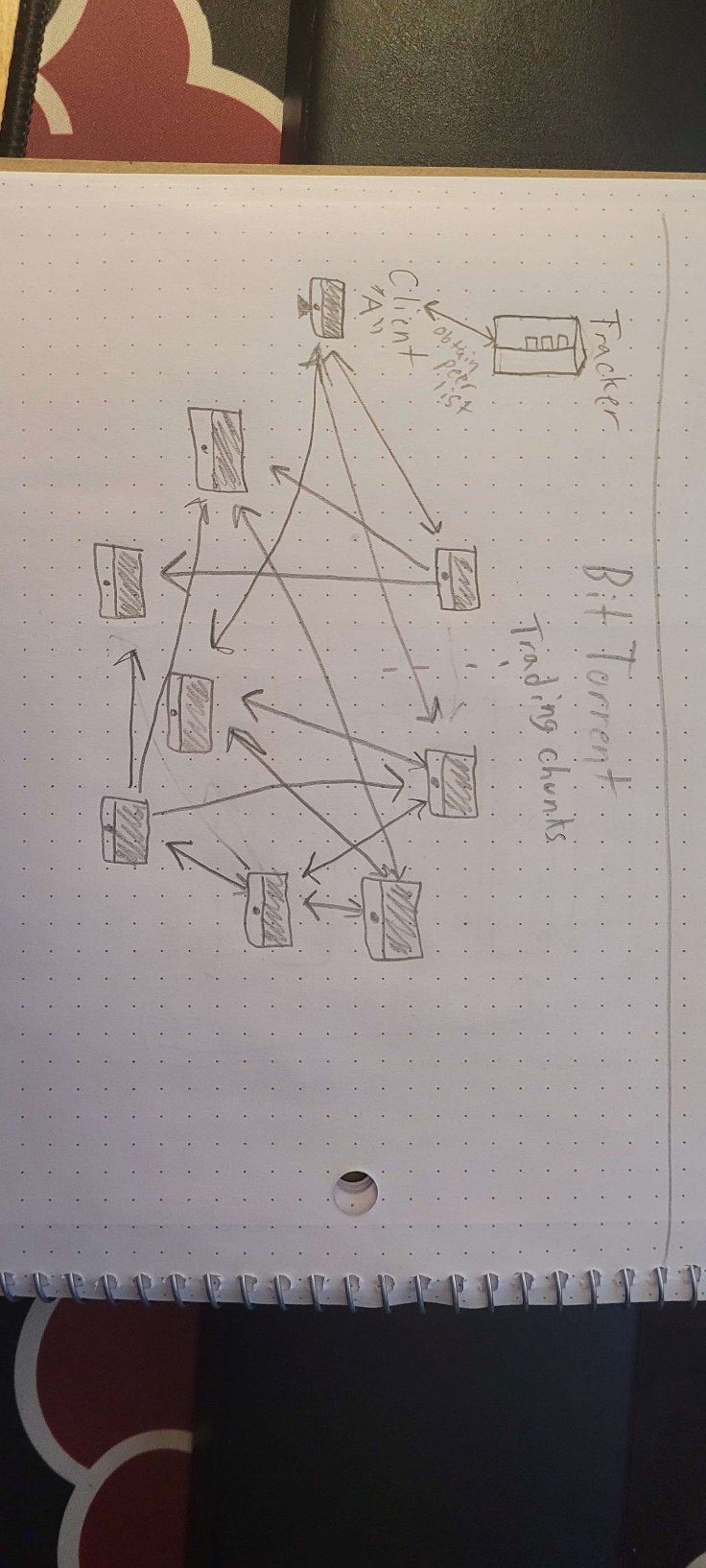
BitTorrent is a popular P2P protocol used for accelerated file distribution. This protocol is used to bring peers together to share a common file and work together to download it together also known as a torrent. Users join the torrent and communicate with each other about which piece of the file they currently have and work in groups to ensure the best speed and necessary portions or chunks are being traded.

Now that we have established the simple explanation of a Bittorrent let’s take a closer look at how it operates. When a peer joins a torrent it starts by registering itself with the tracker and periodically will update the system that it is still connected to the torrent. The tracker is used to keep track of all users and who is currently participating in the torrent. When a user joins a torrent it could have any number of peers from a small group less than 10 up to higher boundaries exceeding 1000 users. With this in mind we can remember that P2P architectures are scalable and ensure that the user won’t have the same wait time for a file compared to a client-server with the same large number of users.

When user “A” joins the torrent and has registered with the tracker they then gain a list of random peers to attempt a concurrent TCP connection with. These successful connections are also known as “neighboring peers.” As time passes user A’s neighboring peers may fluctuate as other peers leave and join the torrent. Now that user A has a stable connection to its peers they are ready to start downloading the file. User A’s machine will look at each subset of chunks from their peers and then issue a request for the chunks they do not possess. From there A will use a technique called “rarest first” to determine which chunks they don’t have are the rarest to then obtain those more quickly compared to others with multiple copies and attempt to equalize the copies of chunks in the torrent.

Next is the priority of connections, User A will then use the BitTorrent trading algorithm to decide this for their machine. User A will give priority to the neighbors that are supplying the fastest rates and then responds by sending chunks back to those top users, these peers are also known as “unchoked.” User A will then choose another random peer also said to be “optimistically unchoked,” if this random peer has a high enough rate they will be put in their top list. User A will then continue this cycle of changing between neighbors until they have fully downloaded the file and disconnects from the torrent.

Kurose, J. F., &amp; Ross, K. W. (2017). In Computer networking: A top-down approach (7th ed.). essay, Pearson.



1. Explain in detail the concept of DNS. Also explain the concept of iterated and recursive query, and the impact of either on the information retrieval process.

When identifying a host network there are two ways to do it, either by an IP address or by a hostname. Routers prefer a structured IP address, but humans prefer a more mnemonic way by using hostnames like “google.com.” To help combine the two we need a service to translate the two also known as DNS.

This is done in the following steps, the user enters the desired URL and their machine starts the client side DNS application. Next the browser extracts the host name and gives it to the DNS application. The client then sends a query that contains the hostname to the DNS server and receives a response containing an IP address. Finally the browser can create the TCP connection to the server using the IP address.

Now that we know how DNS works, what is an iterative and recursive query and what's their impact on the information retrieval process? The two common types of queries are iterative and recursive. These are the queries used when trying to fetch the IP address from the DNS server and it has to be redirected to another DNS server to attempt to find it.

During a recursive query the local DNS client sends its query asking for the IP address from cse.nyu.edu. When the original server does not contain the IP address it then sends the query to the next server dns.nyu.edu on its behalf. Once it finds the IP address it sends it back to cse.nyu.edu and that server returns the IP back to the requesting host.

Alternatively in an iterative query the DNS client sends the query to cse.nyu.edu, but returns without the IP address. Instead the server sends the connection to the next DNS server dns.nyu.edu and the DNS client then sends the query by itself directly to the new server instead of relying on the previous one to communicate on its behalf.

Kurose, J. F., &amp; Ross, K. W. (2017). In Computer networking: A top-down approach (7th ed.). essay, Pearson.

1. Explain in detail, what is the difference between persistent and non- persistence HTTP.

HTTP is one of the fundamental forms of information shared between a server and a client, but for this information to be shared there has to be a way for these two machines to communicate with each other. This is through a protocol called TCP, and has two methods of connections for HTTP. These are called “persistent connection” and “non-persistent connection.”

By default HTTP today uses persistent connections, but can be configured in clients and servers to use non-persistent connections. Let’s start with an assumption for a webpage, This page consists of one HTML file and 10 jpeg images that all live on the same server. The 1st step is for the client to establish a TCP connection to the server on port 80. Next the HTTP client sends an HTTP request to the server via its socket including the path name to the index file. The HTTP server then receives the request and retrieves the object from its storage and sends a response message back to the client. The HTTP server then tells TCP to close the connection, but waits till it receives a response message from the client that it has received the message intact. Once the HTTP client receives the response message the TCP connection ends. Now the client examines the object and finds the other 10 references to the JPEG images and repeats the previous steps for each JPEG object.

While understanding the process it’s easy to notice that there are several connections that have to be opened to obtain every object for the desired page. This means there has to be two RTT’s, “Round Trip Time,” plus the transmission time at the HTTP server. This can cause a lot of stress for the host server especially if it is managing multiple users at the same time.

When using a persistent connection the HTTP server leaves open the original TCP connection. While this connection is open subsequent requests and responses can now be sent over the same connection. Now the server and client can talk more effectively and request objects back-to-back without waiting for pending replies. Once a set amount of time has passed of inactivity the server will then terminate the TCP connection.

Now understanding how persistent connections and non-persistent connections work we can compare them. During a non-persistent connection the client has to open multiple connections to the same HTTP server to continue requests for separate objects referenced in the original file. Alternatively with a persistent connection the client only has to open a single connection and ask for back-to-back objects. While using a persistent connection it relieves stress from the server compared to a non-persistent connection as well. Finally a persistent connection does not have to wait for replies to pending requests, pipelining, while a non-persistent connection do.

Kurose, J. F., &amp; Ross, K. W. (2017). In Computer networking: A top-down approach (7th ed.). essay, Pearson.