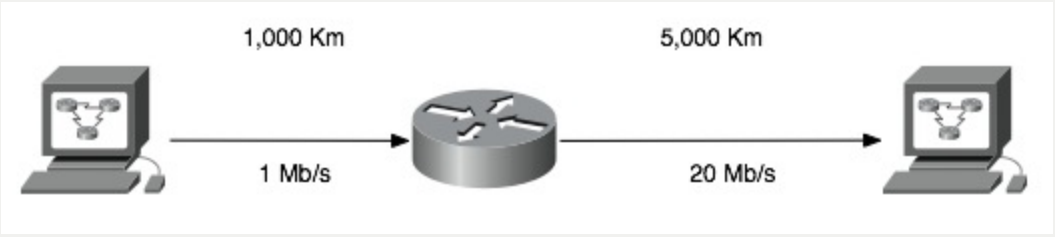
[WR] Workshop 1

Covered in	W2
Main topic	Application layer Overview
Status	Done

Question 1 - Latency through a Network

❓ Consider a packet flowing through the Internet over 2 hops (source -> router1 -> destination). Assume that it takes the router 1 msec to process a packet and determine the outgoing link. The simplified network is shown below:



Given a propagation speed of 2.5 x 10^8 m/s (slightly under the speed of light) and a packet data size of 1000 bytes, what is the end-to-end delay for a packet assuming no other traffic at the router?

What effect would other traffic at the router have?

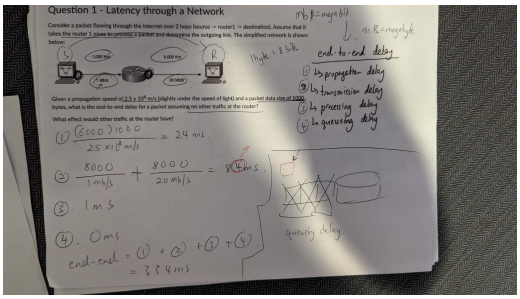
My Answer

There are 4 types of delay experienced by the packet: propagation delay, transmission delay, queuing delay and processing delay.

To calculate the total end-to-end delay, we can calculate the time taken for each type of delay

Workshop's notes

and sum them up.



1. Propagation delay:

   t1 = $\frac{distance}{speed} = \frac{(1{,}000+5{,}000)*1000m}{2.5*10^8 m/s} = 0.024s$

2. Transmission delay:
   t2 = $\frac{size}{bandwidth} = \frac{1000*8bits}{1*10^6 bps} + \frac{1000*8bits}{20*10^6 bps} = 0.0084s$

3. Queuing delay: assumed to be 0 in the question
4. Processing delay: assumed to be 1msec = 0.001s in the question

Total delay time = 0.024 + 0.0084 + 0.001 = 0.0334s = 33.4ms

Effects of other traffic at the router: if there is traffic at the router, the queuing delay will not be negligible. This is because the packet has to queue at the router to be processed, and time is spent waiting for other packets that arrive at the router first to be processed first. This will add to the total end-to-end delay of the packet.

Question 2 - Packet switching and Circuit switching (K&R)

❓ **Statistical multiplexing gains (see section 1.3 in the textbook)**

Assume that you are sharing a link with a bandwidth of 2 Mbps at home. Assume that each member of your family, when using the link, transmits continuously at 1 Mbps but each member transmits only 20 percent of the time.

1. If you set-up your network to circuit switching, how many family members can be supported simultaneously (remember, in circuit switching, resources are reserved for each user)?
2. Now, let's assume that you've re-designed the network to support packet switching and re-consider how may active (simultaneous users can be supported).

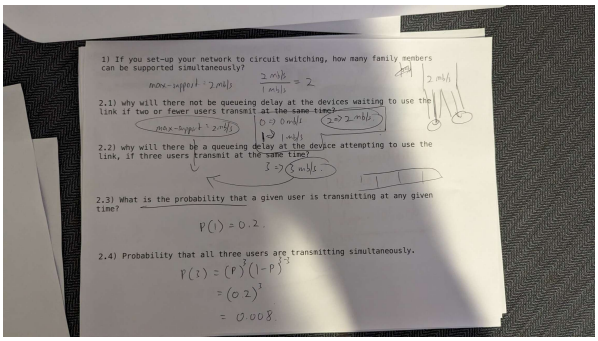
- Why will there not be a queuing delay at the devices waiting to use the link if two or fewer users transmit at the same time?Why will there be a queuing delay at the devices attempting to use the link, if ***three*** users transmit at the same time?What is the probability that a given user is transmitting at any given time?Now re-consider the situation of three users transmitting at the same time. First, find the probability that at any given time, all three users are transmitting simultaneously. Now find the fraction of time during which the queue grows.

My Answer

1. Number of family members can be supported simultaneously = 2Mbps / 1Mbps = 2
2. see below

I. There will not be queueing delay if there are 2 or fewer users transmitting at the same time because 2 users will only use up to 2Mbps which is within the bandwidth of the link. Each packet

Workshop's notes



Circuit switching

arriving at the router will be processed immediately without waiting for any other packet and will not cause a queue.

II. If there are 3 users transmitting at the same time, in total we need 3Mbps bandwidth to serve them all without delay. However, since the bandwidth is only 2Mbps, the router is working at the full capacity, new packets arriving at the router has to wait for the previous packet to be processed first, creating a queue. The queue will build up quickly and will cause significant queueing delay if 3 users continue using simultaneously for a long time.

III. p=0.2

IV. Probability that 3 users are transmitting at the same time = 0.2 * 0.2 * 0.2 = 0.008

Because the queue only grows when the 3 users are transmitting at the time, so the fraction of time during which the queue grows is equal to the probability that 3 users are transmitting at the same, which is 0.008 or 0.8%.

- resources are reserved —> guaranteed performance
- disadvantage: need to set up the circuit

Packet switching

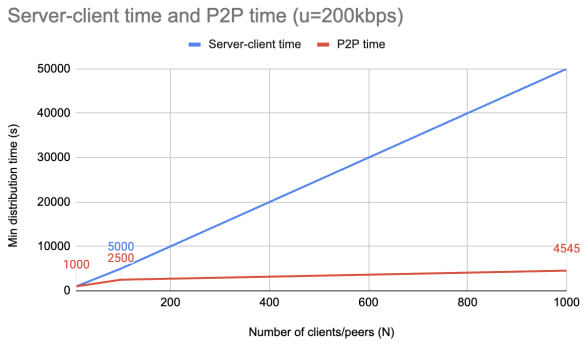
- used as needed
- good for bursty traffic
- disadvantage: when network is busy, the contention will occur, there will be delay, packet can be dropped or even lost

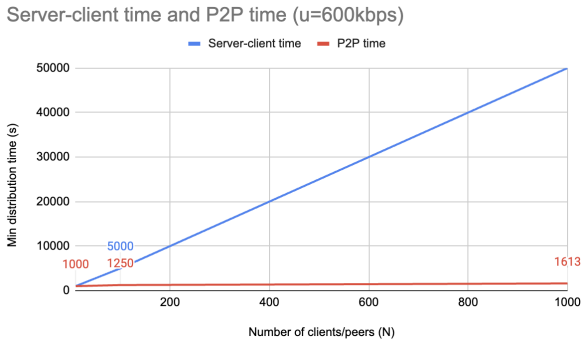
Question 3 - Peer to Peer (P2P) and Client Server (K&R)

❓ Consider distributing a file of 10 Gbits (10 x 10^9 bits - note we are making life easier by using base 10 here) to *N* peers. The server has an upload rate of 20 Mbps, and each peer has a download rate of 1Mbps and an upload rate of *u*. For N=10, 100 and 1,000 and u=200 Kbps, 600 Kbps, and 1Mbps, prepare a chart giving the minimum distribution time for each of the combinations of *N* and *u* for both client-server distribution and P2P distribution.

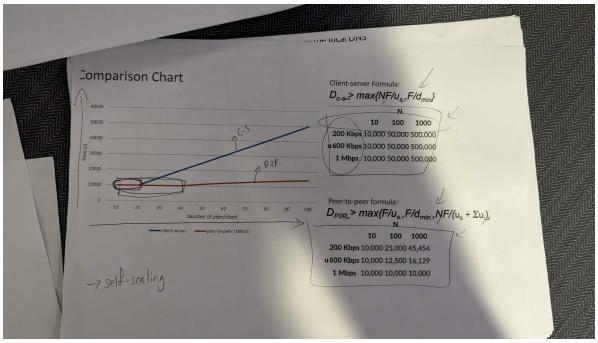
How close would this approximation be to bit torrent's performance?

My Answer





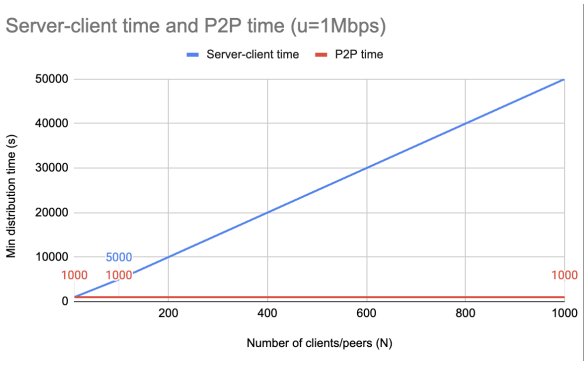
Workshop's notes



Distribution time = minimum time taken for the server to distribute N copies of file to N clients / peers

Assumptions:

- The internet core has abundant bandwidth
- The clients and server are only participating in this network
- All resources are fully devoted to distribute this file
- We only consider transmission delay only



It is hard to compare this approximation with Bit Torrent's performance because:

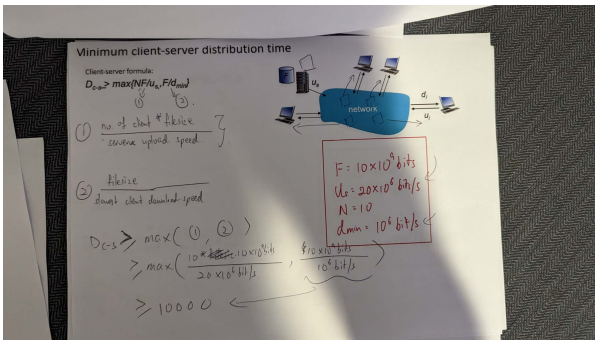
- for Bit Torrent, the speed is highly dependent on the number of peers and this number can vary depends on how many are available in the network
- the internet speed in each peer is different because everyone is sitting at their own home and have variable connection speed

Client-server:

1) no. of clients * file size / server upload speed

2) file size / slowest file download speed

Dc-s = max of 1) and 2) = 10,000



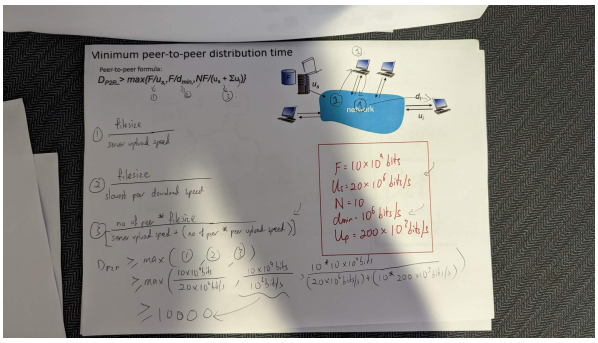
P2P:

1) file size / server upload speed

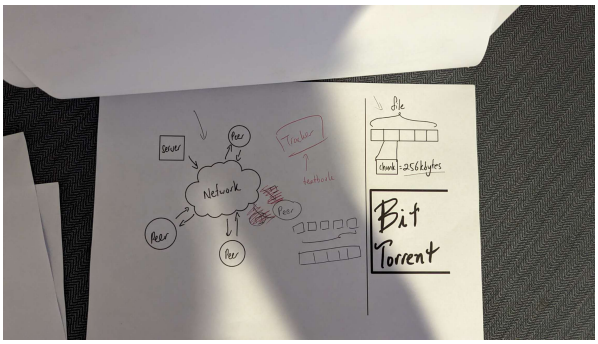
2) file size / slowest peer download speed

3) number of peers * file size / (server upload speed + sum of all peers' upload speed)

Dp2p = max of 1), 2), 3)



Bit Torren:



Question 4 - HTTP

❓ Why does HTTP specify a blank line between the headers and the entity body for requests and responses? Could HTTP have been designed without this blank line? Explain why or why not.

Is the content-length header necessary? Explain.

My Answer

HTTP specify a blank line between the headers and the entity body for requests and responses because this line is needed to separate the headers from the actual content of the entity body. If HTTP does not have the blank line, then both the servers and clients would not be able to tell when is the end of the header and the start of the message body.

HTTP could be designed without having the blank line, as long as it has an alternative way to separate the headers and the entity body. For example, instead of a blank line, it can use special

Workshop's notes

Some ways to replace the blank line:

- replace it with <html> tag —> however, images do not have html tag, will not work for images because HTTP request can be for images

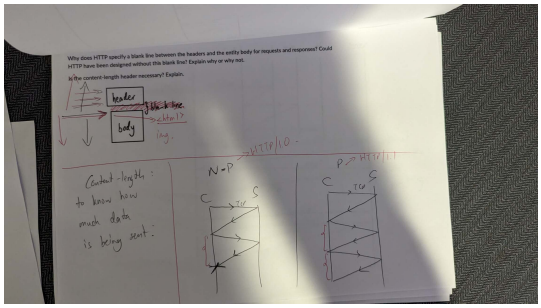
Content line: to know how much data is being sent

- For non-persisent connection: no need content line because, connection is only closed when the object has been fully retrieved

characters like —- or … or other character combination to mark the separate. However, the string chosen must be unique and must not appear in the headers at all, else it will be confused withe marker.

Content-length header is needed because it specifies the length of the content inside the entity body, serving as a check that the content received is exactly the size specified. It could also serve other functions such as when server has a limit on the memory space to store the content, the content-length can help with debugging if we can see that the client is trying to send a content higher than the allowed memory space reserved for that field in the server.

- For persistent connection: we need content line because client needs to know if the object has been fully retrieved



Question 5 - Caching and DNS

❓ Assume you are using persistent HTTP and request a web page that contains two images. Explain the events that must occur if:

1. The web page and images are cached in the web proxy cache and the domain is not cached in the local DNS
2. The web page is not cached; but the domain is cached in the local DNS
3. Neither the web page nor the domain is cached.

In case 3, how much time is likely to pass between the time the user clicks on the link until the web page is loaded? Assume the user is in Adelaide connected to the Internet through an ADSL modem and they are accessing a website in Sydney (about 1500 Kms away). Be creative in how you could work this out (or at least estimate it).

My Answer

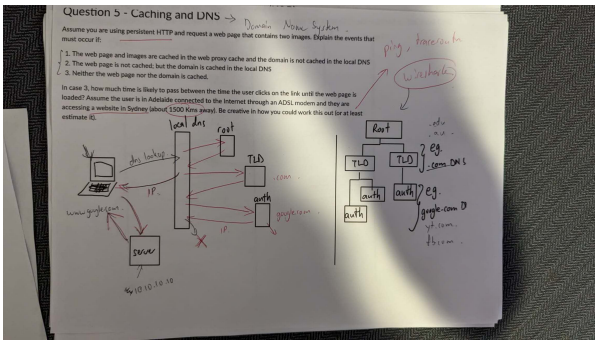
Case 1: when the web page and images are cached in the web proxy cache and the domain is not cached in the local DNS

1. User's browser sends the URL of the page to the web proxy server
2. Web proxy server retrieves the web page and images from its cache and return it to the user's browser for display

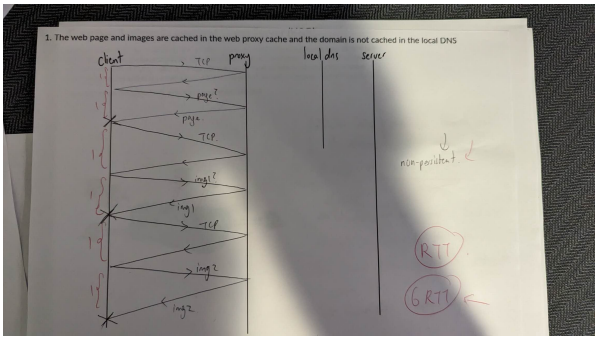
Case 2: The web page is not cached; but the domain is cached in the local DNS

1. User's browser sends the URL of the page to the web proxy server
2. Because this web page is not cached, the web proxy server will send the URL to the local DNS to request for the IP of the web page
3. The local DNS server retrieves from its cached the IP address of the server of the web page and returns to the web proxy server
4. The web proxy server establish a connection with the web server of the page
5. The web server returns to confirm the connection

Workshop's notes



Web page is cached, but the domain is not cached in local DNS with <u>non-persistent connection</u>



Flow:

- Client form a TCP connection with the proxy
- Send HTTP request to ask for the web page
- Proxy has a cache, proxy returns the web page
- TCP connection is closed
- Client forms a second TCP connection

6. The web proxy server requests for the web page
7. The web server returns the web page and the URL of the images
8. (Assuming the 2 images are stored in the same server) The web proxy server send 2 more requests to get the 2 images
9. The web server returns the 2 images
10. The web proxy server returns the web page and the 2 images to the user's browser

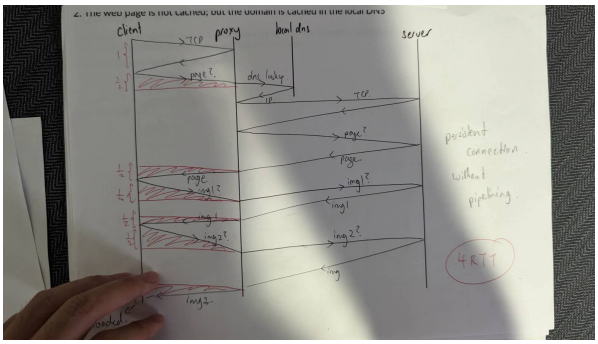
Case 3: Neither the web page nor the domain is cached.

Same as case 2, with step 3 changed to:

- Local DNS server sends the URL to the root DNS server
- Root DNS server extract out the prefix and returns the list of IP for TLD server responsible for this prefix
- Local DNS server sends the URL to one of the TLD servers above
- TLD server returns the IP of the authoritative DNS server
- Local DNS server sends the URL to the authoritative DNS server
- Authoritative DNS server returns the IP of the web server
- Local DNS server returns the IP of the web server to the web proxy
- Send HTTP request to ask for the image 1
- Proxy has a cache, proxy returns the image
- TCP connection is closed
- —> repeat for image 2

—> total 6 RTT

Web page not cached, but the domain is cached in local DNS <u>with persistent connection</u>



- Client form a TCP connection with the proxy
- Send HTTP request to ask for the web page
- Proxy does DNS look up to the nearest local DNS
- Local DNS return IP to proxy
- Proxy do TCP connection to origin server
- Proxy sends HTTP request to origin server

server

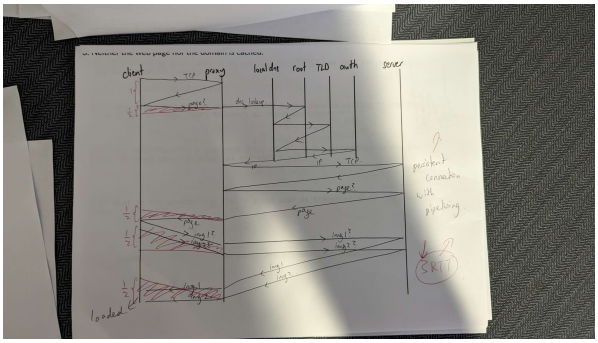
Estimation for website loading time:

- Assume the web proxy server and the local DNS server are located within proximity of the user's host machine, hence the transmission rate is negligible
- How to estimate the distance between the local DNS server with the root DNS, TLD server and authoritative DNS server? - unsure on this, to clarify in class
- After getting the IP, the speed of getting the website can be easily calculated by estimation
    - Assume the bandwidth in Adelaide is 25Mbps, file size is about 2Mb per image —> transmission delay is 0.08s per image. Assuming the connection is persistent with pipelining, the time taken for 2 images would be still around 0.08s. Also assume the size of the web page is small compared to the images.
    - Assume the propagation speed is 2.5 x 10^8 m/s, distance is 1500km —> propagation delay is 0.0075s
    - —> Total delay for file transfer (not including the DNS step) is

- Origin server returns the web page to proxy
- Proxy returns web page to client

—> total 4RTT (based on the client view, not counting the proxy view)

Neither the web page nor the domain is cached, using <u>persistent connection with pipelining</u>



- Client form a TCP connection with the proxy
- Send HTTP request to ask for the web page
- Proxy does DNS look up to the nearest local DNS
- Local DNS find root DNS → TLD → auth
- Local DNS returns IP to proxy
- Proxy forms TCP connection with server
- Proxy sends HTTP request to origin server to request for web page

about 0.08+0.0075 = 0.0875s

- …
- Client requests image 1, 2 at the same time (pipelining)
- Proxity

—> total 3 RTT

Estimate delay:

Use tools like ping / traceroute /wireshark to get the RTT, then multiply by the number of RTT

Can use wireshark to calculate exact - each RTT is the time between request and response

Question 6

❓ What topic(s) would you most like to review/discuss or have further examples of?

My Answer

I would like to have a summary of a big picture of the flows of data - including the web, HTTP and the DNS. Maybe a big diagram summarising the different components and knowledge would really help.

Workshop's notes

On a separate note, I would like to know if blockchain is a type of P2P architecture? And how the concepts we learnt in the class are applied to the blockchain technology.