Adam Alston

1. Steps:
    a. Alice's email client will collect all the email information typed by Alice. It will first contact DNS and query the IP address of the SMTP server from her ISP. The DNS replies with the IP address of the SMTP server to Alice's client. Then Alice's client will create a local TCP socket specifying the host and port number of the SMTP server which is on Alice's ISP.
    b. When the client creates this socket, the client's TCP establishes connection to server's TCP. After some TCP socket connection 'handshaking', the client will tell the SMTP server the information of sender and receiver.
    c. The SMTP server will confirm correct sender information, receiver information and receive the email. It will put the email into the queue of all the emails waiting to be sent.
    d. When Alice's email comes out of the queue, the server will get the receiver's domain name, split after the '@' of the receiver's email address. The SMTP server will first contact DNS and query the IP address of the receiver's SMTP server. The DNS will send reply with the IP address of receiver's SMTP server to Alice's SMTP server.
    e. Now Alice's SMTP server will connect on a specific port of receiver's SMTP server, notify of the sending activity, confirm the email account of the receiver and pass the email to receiver's SMTP server.
    f. At the other end, the receiver's email client will establish TCP socket, connect to the SMTP server, conduct 'handshaking' through TCP to connect with the email server and get email content. The SMTP protocol will transmit the message from transport layer to email application.

2.

a. The time of first 2 RTT with the main web:

T_0 = 2 * 50ms + 20000 bits / 100 Mbps = 100.2ms

The time of each of the first four parallel URLs:

Then, the network links speed is 100 Mbps / 4 = 25 Mbps each parallel connection

$$T_1 = 2 \times RTT_{bird} + \frac{10000}{100Mbps} = 40.1\text{ms}$$

$$T_2 = 2 \times RTT_{cat} + \frac{10000}{100Mbps} = 20.1$$

$$T_3 = 2 \times RTT_{dog} + \frac{15000}{100Mbps} = 100.15\text{ms}$$

$$T_4 = 2 \times RTT_{cat} + \frac{10000}{100Mbps} = 20.1$$

Because T_2 and T_4 have shortest time, these two parallel connections will start the rest of two URL connections separately.

$$T_5 = 2 \times RTT_{dog} + \frac{10000}{100Mbps} = 100.1\text{ms}$$

$$T_6 = 2 \times RTT_{dog} + \frac{5000}{100Mbps} = 100.05\text{ms}$$

The longest parallel connection = T_0 + T_2 + T_5 = 100.2 + 20.1 + 100.1 = 220.4ms

b. Time to get main page is the same as (a)

T_0 = 2 * 50ms + 20000 bits / 100 Mbps = 100.2ms

Time to get dog pages:

$$2 \times RTT_{dog} + \frac{10000+15000+5000}{100Mbps} = 100.3\text{ ms}$$

Time to get bird pages:

$$2 \times RTT_{bird} + \frac{10000}{100Mbps} = 40.1\text{ ms}$$

Time to get cat pages:

$$2 \times RTT_{cat} + \frac{10000+10000}{100Mbps} = 20.2\text{ ms}$$

Total time = 100.2ms + 100.3ms + 40.1ms + 20.2ms = 260.8ms

c. Parallel HTTP connection have the following benefits, which makes the use of parallel HTTP connection a good idea

- Reduced Latency: It means there is no waiting time. A transaction can start immediately instead of waiting for another transaction to be completed.
- Increased throughput: Since the waiting time has been reduced and the system can perform parallel transaction and it increases aggregate transaction throughput.
- Faster Page loading: It makes pages load faster. Composite pages consisting of embedded objects may load faster if they take advantage of the dead time and bandwidth limit of the single connection. These delays can be overlapped.

3.

a. T_{cache} = 10ms

T_{server} = 10ms + 1400ms + 2 * (40000 bits / 1.5 Mbps)ms + (40000 bits / 10 Mbps)ms

= 10ms + 1400ms + 2*27ms + 4ms = 1468ms

T_{mean} = p*T_{cache} + (1-p)*T_{server} = 10p + (1-p)1468 $\leq$ 450

p $\geq$ 0.7 So, the smallest hit ratio is 0.7

b. Invalid. Traffic intensity is too high from router 1 to router 2.

Traffic intensity = (40000 bits * 100)/1.5Mbps = $\frac{8}{3} > 1$

4.

a.

Name	Type
com	A
com	NS
catmyth.com	A
catmyth.com	NS
ads. catmyth.com	NS
ads. catmyth.com	A
www.ads.catmyth.com	A
www.catmyth.com	A
www2.catmyth.com	A
Biz	A
Biz	NS
cat.biz	A
cat.biz	NS
ads.cat.biz	A
ads.cat.biz	NS
www.ads.cat.biz	A
bird.biz	NS
bird.biz	A
ads.bird.biz	NS
ads.bird.biz	A
www.ads.bird.biz	A
www.cat.biz	A

b. If the local DNS server does not know the IP address, it will send 1 request and 2 responses.
If the DNS knows the IP address, it will only send 1 request and 1 response.
There are 8 times that the DNS does not know the IP address and 6 times the DNS does know the IP address.
Thus, the number of DNS query/reply message $(8 * 2) + 8 + (6 * 2) = 36$

5.

a. Bandwidth: $u_1, u_2, \ldots, u_N$

Total bandwidth: $u = u_1 + u_2 + \ldots + u_N$

Question says: $u_s \leq \frac{(u_1+u_2+\cdots+u_N)}{N}$

Size of a packet: $F(\frac{u_i}{u})$

Rate: $r_i = u_s(\frac{u_i}{u})$ -> $u_s = r_1 + r_2 + \ldots + u_N$

Overall rate: $r_i(N-1)$

$r_i(N-1)$ -> $u_s(\frac{u_i}{u})(N-1)$

$\text{Total rate} = r_i + \sum_{k=1}^{n} r_k = u_s$

Therefore time: $\frac{F(bits)}{u_s(rate)}$

b. Bandwidth: $u_1, u_2, \ldots, u_n$

Total bandwidth: $u = u_1 + u_2 + \cdots + u_n$

Question says: $u_s \geq \frac{(u_s+u_1+u_2+\cdots+u_N)}{N}$ -> $u_s \geq \frac{(u_s+u)}{N}$

Server rate: $r_i = \frac{u_i}{N-1}$

Peer forwarding: Nr_{N+1}

Combined rate: $r_1 + r_2 + \cdots + r_N + Nr_{N+1} = \frac{u_1+u_2+\ldots+u_N}{N-1} + u_s - \frac{u_1+u_2+\ldots+u_N}{N-1}$

$= \frac{u}{N-1} + u_s - \frac{u}{N-1} = u_s$

$Total\ rate = r_i + r_{N+1} + \sum_{k=1}^{n} r_k = \frac{u}{N-1} + \frac{(u_s - \frac{u}{N-1})}{N} = \frac{u_s+u}{N}$

Therefore time: $\frac{NF}{u_s+u}$

c. Minimum distribution time: $D_{P2P} \geq max\{\frac{F}{u_s}, \frac{NF}{u_s+u}\}$

Taking equations from (a) and (b): $D_{P2P} = max\{\frac{F}{u_s}, \frac{NF}{u_s+u_1+u_2+\cdots+u_n}\}$

6. Client-Server:

- The server must transmit one copy of the file to each of the N peers. Thus the server must transmit NF bits
- Since the server's upload rate is us, the time to distribute the file must be at least $\frac{NF}{u_s}$
- The peer with the lowest download rate cannot obtain all F bits of the file in less than $\frac{F}{d_{min}}$ seconds

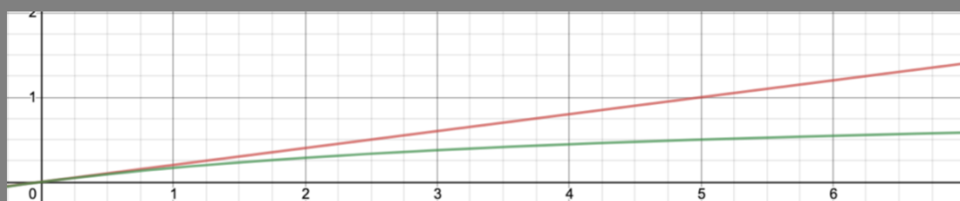
P2P:

- At the beginning of the distribution, only the server has the file. The minimum distribution time is at least $\frac{F}{u_s}$
- The peer with the lowest download rate cannot obtain all F bits of the file in less than $\frac{F}{d_{min}}$ seconds

Observe that the total upload capacity of the system as a whole is equal to the upload rate of the server plus the upload rates of each of the individual peers, that is, $u = u_1 + u_2 + \ldots + u_N$. The system must deliver (upload) F bits to each of the N peers, thus delivering a total of NF bits. This cannot be done at a rate faster than u. Thus, the minimum distribution time is also at least $\frac{NF}{u_s+u_1+u_2+\cdots+u_n}$ Simplify using $u = u_1 + u_2 + \ldots + u_N$

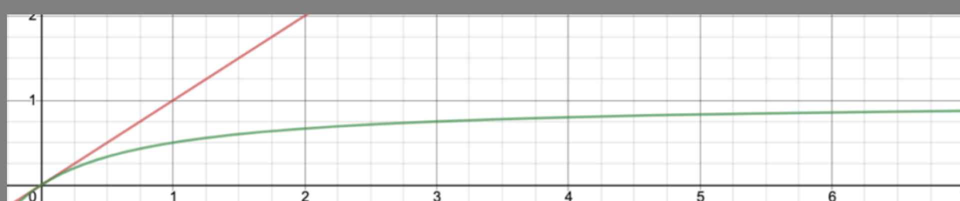
a. $D_{cs} \geq max\{\frac{NF}{u_s}, \frac{F}{d_{min}}\} \geq max\{\frac{N}{5}, \frac{1}{5}\} \geq \frac{N}{5}$

$D_{P2P} \geq max\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s+u_1+u_2+\cdots+u_n}\} \geq max\{\frac{1}{5}, \frac{F}{5u}, \frac{NF}{5u+u}\} = max\{\frac{1}{5}, \frac{1}{5}, \frac{N}{5+N}\}$



b. $D_{cs} \geq max\{\frac{NF}{u_s}, \frac{F}{d_{min}}\} = max\{N, 1\} \geq N$

$D_{P2P} \geq max\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s+u_1+u_2+\cdots+u_n}\} \geq max\{1, 1, \frac{NF}{2u}\} \geq max\{1, 1, \frac{N}{1+N}\}$

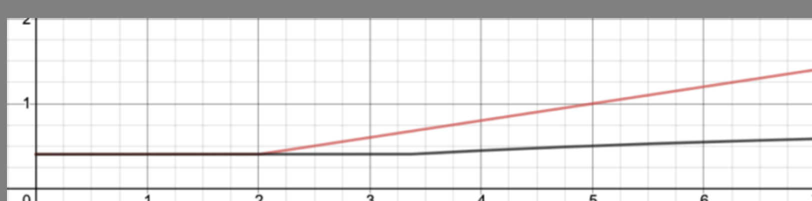


c. $D_{cs} \geq max\frac{NF}{u_s}, \frac{F}{d_{min}} = max\frac{N}{5}, \frac{F}{(u_s/2)} \geq max\frac{N}{5}, \frac{2F}{5u} \geq max\frac{N}{5}, \frac{2}{5} \geq \frac{N}{5}$

$D_{P2P} \geq max\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s+u_1+u_2+\cdots+u_n} \geq max\frac{1}{5}, \frac{2}{5}, \frac{NF}{5u+u} \geq max\frac{1}{5}, \frac{2}{5}, \frac{N}{6}$

$y = \left\{0 \leq x \leq 2: \frac{2}{5},\ x \geq 2: \frac{x}{5}\right\}$

$y = \left\{0 \leq x \leq \frac{10}{3}: \frac{2}{5}, x \geq \frac{10}{3}: \frac{x}{5+x}\right\}$



7.

a. Yes, Bob can download a copy that is shared by the swarm. Bob is a considered a leecher (peer who doesn't share). The seeders (peer who does share) share their files within their public user chosen file directory where files can be downloaded from other users like Bob. Bob can then communicate with the seeders machines and download the desired file that Bob is looking for due to the P2P communication features of the software BitTorrent. Thus, making Bob's claim possible.

b. Bob can do this by having more machines communicating with more seeders. Therefore, the more computers communicating with other seeders allows Bob to have a more efficient way of downloading files faster because each machine is downloading multiple files from several seeders. This gives Bob the ability to obtain more files quickly.
   Collaborators: Zihong Chen