50.012 Networks (2020 Term 6)

Homework 2

Hand-out: 8 Oct

Due: 20 Oct 23:59

Name: ________________ Student ID: _________________

1. (Adapted from last year's midterm exam question) Consider a browser that wants to retrieve a Web document at a given URL. The IP address of the web server is initially unknown. Name the two application layer protocols used in this scenario and the corresponding underlying transport protocol that each application protocol typically uses. Explain briefly for each application layer protocol, why the specific underlying transport protocol is used.

Suggested solution:

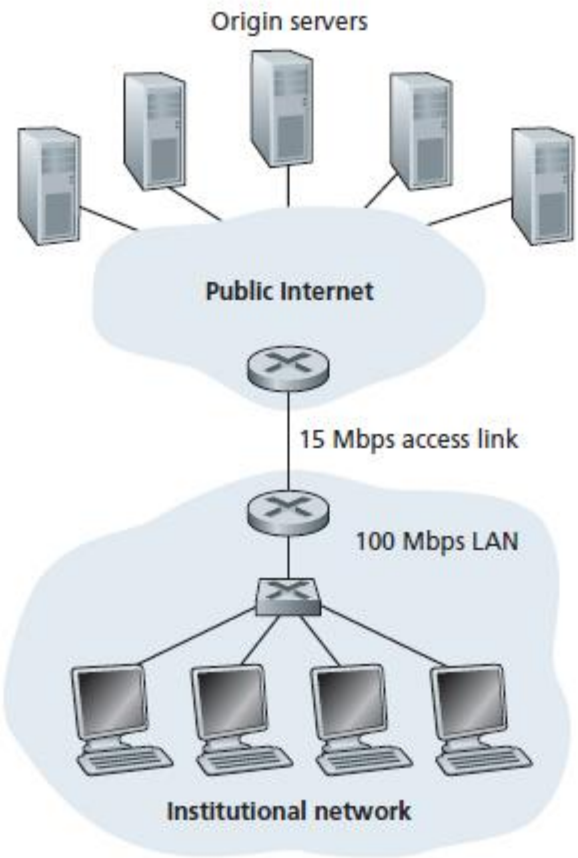
HTTP: TCP, reliable data transfer (NB: It is also correct if the answer is HTTP/3 uses UDP)

DNS: UDP, avoid extra delay due to connection setup / handshake.

2. (textbook problem chapter 2, problem 9) Consider the Figure below, for which there is an institutional network connected to the Internet. Suppose that the average object size is 850,000 bits and that the average request rate from the institution's browsers to the origin servers is 16 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 three seconds on average. Model the total average response time as the sum of the average access delay (that is, the delay from Internet router to institution router) 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.

2.1 Find the total average response time.

2.2 Now suppose a cache is installed in the institutional LAN. Suppose the miss rate is 0.4. Find the total average response time under this case (assuming the response time is zero if the request is satisfied by the cache, which happens with probability 0.6).



Suggested solution:

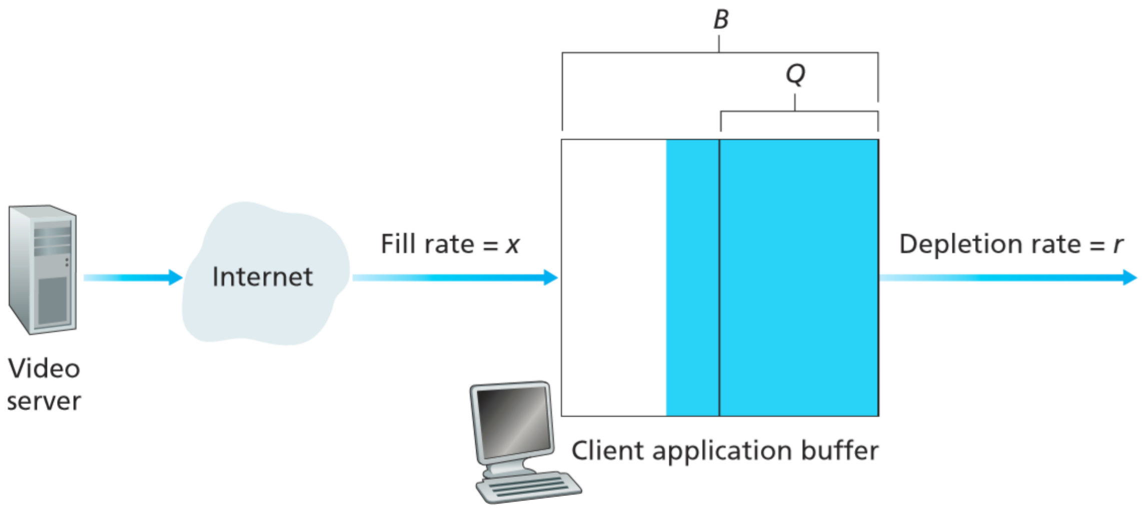
1) The time to transmit an object of size L over a link or rate R is L/R. The average time is the average size of the object divided by R:

   Δ = (850,000 bits)/(15,000,000 bits/sec) = 0.0567 sec

   The traffic intensity on the link is given by $\beta\Delta$=(16 requests/sec)(0.0567 sec/request) = 0.907. Thus, the average access delay is (0.0567 sec)/(1 - 0.907) $\approx$ 0.6 seconds. The total average response time is therefore 0.6 sec + 3 sec = 3.6 sec.

2) The traffic intensity on the access link is reduced by 60% since the 60% of the requests are satisfied within the institutional network. Thus the average access delay is (0.0567 sec)/[1 – (0.4)(0.907)] = 0.089 seconds. The response time is approximately zero if the request is satisfied by the cache (which happens with probability 0.6); the average response time is 0.089 sec + 3 sec = 3.089 sec for cache misses (which happens 40% of the time). So the average response time is (0.6)(0 sec) + (0.4)(3.089 sec) = 1.24 seconds. Thus the average response time is reduced from 3.6 sec to 1.24 sec.

3. Recall the simple model for HTTP streaming shown in the Figure below. Let B denote the size of the client's application buffer, and let Q denote the number of bits that must be buffered before the client application begins playout. Also, let r denote the video consumption rate. Assume that the server sends bits at a constant rate x whenever the client buffer is not full.



3.1 Suppose that x < r. In this case playout will alternate between periods of continuous playout and periods of freezing. Determine the length of each continuous playout and freezing period as a function of Q, r, and x.

3.2 Now suppose that x > r. At what time does the client application buffer become full?

Suggested solution:

1) During a playout period, the buffer starts with Q bits and decreases at rate r - x. Thus, after Q/(r - x) seconds after starting playback the buffer becomes empty. Thus, the continuous playout period is Q/(r - x) seconds. Once the buffer becomes empty, it fills at rate x for Q/x seconds, at which time it has Q bits and playback begins. Therefore, the freezing period is Q/x seconds.

2) Time until buffer has Q bits is Q/x seconds. Time to add additional B - Q bits is (B - Q)/(x - r) seconds. Thus the time until the application buffer becomes full is Q/x + (B - Q)/(x - r) seconds.

4. (Adapted from last year's midterm exam question) Consider distributing a file of $F = 6 \times 10^9$ bits to $N=100$ peers. The server has an upload rate of $u_s = 30$ Mbps, and each peer has a download rate of $d_i = 2$ Mbps and an upload rate of $u_i=1$ Mbps. Assume $1M = 10^6$. Calculate the minimum distribution time (i.e., to let every peer have a copy of the file) for:

4.1 the client-server distribution mode, and

4.2 the P2P distribution mode.

Suggested solution:

Client-server:

$\max(F * N / u_s , F/ d_i)=\max(6 \times 10^9 \times 100 / 30 \times 10^6 , 6 \times 10^9 / 2 \times 10^6)=20{,}000$ s

P2P:

$\max(F / u_s , F/ d_i , F * N / (u_s+N* u_i) = \max(6 \times 10^9 / 30 \times 10^6 , 6 \times 10^9 / 2 \times 10^6 , 6 \times 10^9 \times 100 / (30 + 100) \times 10^6) =60/13 * 10^3$ s

5. (last year's midterm exam question) Consider data communication over a link of RTT 100ms and transmission bandwidth 1Gbit/s. Assume $1G=10^9$. Consider a pipelined transport protocol that uses ACKs to decide if packets were received successfully. Answer the following three questions:

5.1 After the protocol has sent a packet, what is the minimum amount of time needed for the protocol to infer that the packet was lost?

5.2 If the protocol uses a window size of 6 packets (each of size 1000 bytes), what is the maximum achievable data throughput?

5.3 To fully use the transmission bandwidth, estimate the minimum window size (in bytes) needed.

Suggested solution:

1) RTT, or 100ms
2) 6000 * 8 /0.1s = 480K bps
3) ~12.5M Bytes

6. (optional, will not be graded) (textbook problem chapter 2, problem 24) Consider distributing a file of F bits to N peers using a P2P architecture. Assume a fluid model (i.e., a peer can immediately upload a bit it just download, without any delay). For simplicity assume that d_{min} is very large, so that peer download bandwidth is never a bottleneck.

6.1 Suppose that $u_s \leqslant (u_s + u_1 + ... + u_N)/N$. Specify a distribution scheme that has a distribution time of F/u_s. [Hint 1: consider breaking the file into N parts, according to the uploading capability of each peer. Hint 2: define $u = u_1 + u_2 + ... + u_N$ can help simplify your reasoning. Hint 3: you need to ensure each node (server and every peer) transmits according to their uplink rate, and each peer node receives the whole file after F/u_s]

6.2 Suppose that $u_s \geqslant (u_s + u_1 + ... + u_N)/N$. Specify a distribution scheme that has a distribution time of $NF/(u_s + u_1 + ... + u_N)$. [Hint 1: built upon your answer for the sub-question 6.1, and consider breaking the file into N+1 parts, where the more powerful server under this assumption handles the last part. Hint 2 & 3 are the same as above]

6.3 Conclude that the minimum distribution time is in general given by $\max\{F/u_s, NF/(u_s + u_1 + ... + u_N)\}$.

Suggested solution:

1. Define $u = u_1 + u_2 + ... + u_N$. By assumption $u_s <= (u_s + u)/N$ (Eq 1)

   Divide the file into N parts, with the i^{th} part having size $(u_i/u)F$. The server transmits the i^{th} part to peer i at rate $r_i = (u_i/u)u_s$. Note that $r_1 + r_2 + ... + r_N = u_s$, so that the aggregate server rate does not exceed the link rate of the server. Also have each peer i forward the bits it receives to each of the $N-1$ peers at rate r_i. The aggregate forwarding rate by peer i is $(N-1)r_i$. We have $(N-1)r_i = (N-1)(u_s u_i)/u <= u_i$, where the last inequality follows from Eq 1. Thus the aggregate forwarding rate of peer i is less than its link rate u_i.

   In this distribution scheme, peer i receives bits at an aggregate rate of
   $$r_i + \sum_{j \neq i} r_j = u_s$$
   Thus each peer receives the file in F/u_s.

2. Again define $u = u_1 + u_2 + ... + u_N$. By assumption $u_s >= (u_s + u)/N$ (Eq 2)

Let $r_i = u_i/(N-1)$ and $r_{N+1} = (u_s - u/(N-1))/N$

In this distribution scheme, the file is broken into $N+1$ parts. The server sends bits from the i^{th} part to the i^{th} peer ($i = 1, \ldots., N$) at rate r_i. Each peer i forwards the bits arriving at rate r_i to each of the other N-1 peers. Additionally, the server sends bits from the $(N+1)^{st}$ part at rate r_{N+1} to each of the N peers. The peers do not forward the bits from the $(N+1)^{st}$ part.

The aggregate send rate of the server is

$$r_1 + \ldots. + r_N + N\, r_{N+1} = u/(N-1) + u_s - u/(N-1) = u_s$$

Thus, the server's send rate does not exceed its link rate. The aggregate send rate of peer i is $(N-1)r_i = u_i$ Thus, each peer's send rate does not exceed its link rate.
In this distribution scheme, peer i receives bits at an aggregate rate of

$$r_i + r_{N+1} + \sum_{j \neq i} r_j = u/(N-1) + (u_s - u/(N-1))/N = (u_s + u)/N$$

Thus each peer receives the file in $NF/(u_s+u)$.

To see the size of the file part for $i = 1, \ldots., N+1$, let $\Delta = NF/(u_s+u)$ be the distribution time. For $i = 1, \ldots, N$, the i^{th} file part is $F_i = r_i\, \Delta$ bits. The $(N+1)^{st}$ file part is $F_{N+1} = r_{N+1}\, \Delta$ bits. It is straightforward to show that $F_1 + \ldots.. + F_{N+1} = F$.

3. We know from section 2.6 that

$$D_{P2P} >= max\{F/u_s, NF/(u_s + u)\}$$

Combining this with 1) and 2) above gives the desired result.