HW2 Jed Alcantara G00846927

Monday, February 20, 2023 5:51 PM

1. [30 pt] Consider the network provided in Fig. 2.12 of the textbook. For simplicity, let's assume that the LAN delay is 0 and the Internet delay is 2 seconds. Assume that the average object size is 80Kb and there are 20 requests per second from the institution's browsers to the origin servers.
    a. Calculate the traffic intensity (also called link utilization) on the access link.
    b. Find the total response time. The total response time/delay is the sum of LAN delay, access delay and Internet delay. Here, the average access delay can be calculated as x/(1-y) where y is the traffic intensity you calculated above and x is the average time required to send one object over the access link.
    c. Now suppose a cache is installed in the institutional LAN with a hit rate of 0.5. Find the total response time.

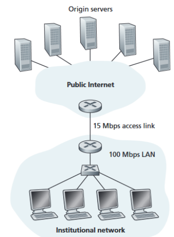

Figure 2.12 • Bottleneck between an institutional network and the Internet

A. traffic intensity $= \frac{\text{no. of request}}{s} \times \frac{\text{object size}}{\text{link rate}}$

$= \frac{20}{s} \times \frac{80 \times 10^3 b}{100 \times 10^6 b/s}$

$= \frac{1600 \times 10^3}{1 \times 10^8} = \frac{1.6 \times 10^6}{1 \times 10^8}$

$= 0.016 s = 16ms$

B. total response rate $= T_{LAN} + T_{int} + T_{ACC}$

$X = \frac{80 \times 10^3 b}{15 \times 10^6 b} = \frac{80 b}{15 \times 10^3 b/s} = 5.33 ms$

$T_{acc} = x/(1-y) = 5.33ms / (1 - 0.016) = 5.42ms$

$T_{Tot} = 2 + 0 + 0.00542 = 2.00542 s$

C. Cache hit rate $= \frac{20}{s} \times 0.5 = \frac{10}{s}$

traffic intensity $= \frac{10}{} \times \frac{80 \times 10^3}{100 \times 10^6} = 8ms$

$5.33ms / (1 - 0.008) = 5.38ms$

Total time $= 2 + 0 + 0.00538 = 2.00538 s$

2. [10 pt] What is the use of mail access protocols? Why can't we use SMTP to deliver the emails all the way to receiving user agent's computer?

Mail access protocols are a set of standards used by email clients to retrieve email messages from a mail server through a TCP/IP connection. SMTP cannot be used to deliver to the client directly because the sender may not be able to reach the client; therefore, the server acts as an intermediary much like a mail box. Consider examples like if Alice is sending to Bob. If Bob's server is unreachable, SMTP is responsible for returning the message to Alice's server.

3. [20 pt] Suppose Client A initiates an FTP session with Server S. At about the same time, Client B also initiates an FTP session with Server S. What are the possible source and destination port numbers for the following? (note that popular services such as FTP have preassigned port numbers https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers)
    a. The segments sent from A to S
    b. The segments sent from B to S
    c. The segments sent from S to A
    d. The segments sent from S to B
    e. If A and B are different hosts, is it possible that the source port number in the segments from A to S is the same as that from B to S?
    f. What if A and B are the same hosts, same question (e).

The source port number is between 1024 and 65535,0x400 and 0xFF00 respectively. Because the destination server is FTP, the destination port number is 21. It differentiates the clients by the port number of the source and their IP address.

a. The segment sent from A to S is src 8484 dest 21
b. The segment sent from B to S is src 43592 dest 21
c. The segment sent from S to A is src 21 dest 8484
d. The segment sent from S to B is src 21 dest 43592
e. Yes, it is possible that the source port number from A to S is the same as B to S since unique hosts would have similar port numbers but unique ip addresses.
f. No, if they are the same hosts, then the A and B will have the same port and ip address.

4. [20 pt] For the Internet checksum that we discussed in class (used in UDP and TCP), consider two bytes 01011100 and 01100101.
    a. Calculate the Internet checksum of the two bytes
    b. Show that the checksum captures all 1-bit errors (1 bit changed in either of the two bytes) using an example.
    c. Show that the checksum might not capture all 2-bit errors (1 bit changed in each of the two bytes) using an example.

a. checksum: 01011100
+01100101

sum 11000101
checksum 00111111

b. checksum: exp: 0110 0101
→ 1100 0001 err: 0110 0001

0110 0101
0101 1100

→ 10111101

Bits in blue are inverted but checksum is the same

C. checksum:

11000001

$$
\begin{array}{r}
1\,1\,1\,1 \\
01010100 \\
+\ 01101101 \\
\hline
11000001
\end{array}
$$