

**SYSC 5201 Fall 2012 Assignment #2**

1.

Suppose users share a 3 Mbps link. Also suppose each user requires 150 kbps when transmitting, but each user transmits only 10 percent of the time. (See the discussion of statistical multiplexing in Section 1.3.)

- a. When circuit switching is used, how many users can be supported?
- b. For the remainder of this problem, suppose packet switching is used. Find the probability that a given user is transmitting.
- c. Suppose there are 120 users. Find the probability that at any given time, exactly  $n$  users are transmitting simultaneously. (*Hint:* Use the binomial distribution.)
- d. Find the probability that there are 21 or more users transmitting simultaneously. (*Hint:* Use Central Limit Theorem to estimate the final number.)

Solution:

a) 20 users can be supported.

b)  $p = 0.1$ .

c)  $\binom{120}{n} p^n (1-p)^{120-n}$ .

d)  $1 - \sum_{n=0}^{20} \binom{120}{n} p^n (1-p)^{120-n}$ .

We use the central limit theorem to approximate this probability. Let  $X_j$  be independent random variables such that  $P(X_j = 1) = p$ .

$$P(\text{“21 or more users”}) = 1 - P\left(\sum_{j=1}^{120} X_j \leq 21\right)$$

$$P\left(\sum_{j=1}^{120} X_j \leq 21\right) = P\left(\frac{\sum_{j=1}^{120} X_j - 12}{\sqrt{120 \cdot 0.1 \cdot 0.9}} \leq \frac{9}{\sqrt{120 \cdot 0.1 \cdot 0.9}}\right)$$

$$\approx P\left(Z \leq \frac{9}{3.286}\right) = P(Z \leq 2.74)$$

$$= 0.997$$

when  $Z$  is a standard normal r.v. Thus  $P(\text{“21 or more users”}) \approx 0.003$ .

2.

Obtain the HTTP/1.1 specification (RFC 2616). Answer the following questions:

- a. Explain the mechanism used for signaling between the client and server to indicate that a persistent connection is being closed. Can the client, the server, or both signal the close of a connection?
- b. What encryption services are provided by HTTP?
- c. Can a client open three or more simultaneous connections with a given server?
- d. Either a server or a client may close a transport connection between them if either one detects the connection has been idle for some time. Is it possible that one side starts closing a connection while the other side is transmitting data via this connection? Explain.

Solution:

- a) Persistent connections are discussed in section 8 of RFC 2616 (the real goal of this question was to get you to retrieve and read an RFC). Sections 8.1.2 and 8.1.2.1 of the RFC indicate that either the client or the server can indicate to the other that it is going to close the persistent connection. It does so by including the including the connection-token "close" in the Connection-header field of the http request/reply.
- b) HTTP does not provide any encryption services.
- c) (From RFC 2616) "Clients that use persistent connections should limit the number of simultaneous connections that they maintain to a given server. A single-user client SHOULD NOT maintain more than 2 connections with any server or proxy."
- d) Yes. (From RFC 2616) "A client might have started to send a new request at the same time that the server has decided to close the "idle" connection. From the server's point of view, the connection is being closed while it was idle, but from the client's point of view, a request is in progress."

3.

Suppose you can access the caches in the local DNS servers of your department. Can you propose a way to roughly determine the Web servers (outside your department) that are most popular among the users in your department? Explain.

Solution:

We can periodically take a snapshot of the DNS caches in the local DNS servers. The Web server that appears most frequently in the DNS caches is the most popular server. This is because if more users are interested in a Web server, then DNS requests for that server are more frequently sent by users. Thus, that Web server will appear in the DNS caches more frequently.

For a complete measurement study, see:

Craig E. Wills, Mikhail Mikhailov, Hao Shang

"Inferring Relative Popularity of Internet Applications by Actively Querying DNS Caches", in IMC'03, October 27-29, 2003, Miami Beach, Florida, USA

4.

Consider distributing a file of  $F$  bits to  $N$  peers using a P2P architecture. Assume a fluid model. For simplicity assume that  $d_{\min}$  is very large, so that peer download bandwidth is never a bottleneck.

- Suppose that  $u_s \leq (u_s + u_1 + \dots + u_N)/N$ . Specify a distribution scheme that has a distribution time of  $F/u_s$ .
- Suppose that  $u_s \geq (u_s + u_1 + \dots + u_N)/N$ . Specify a distribution scheme that has a distribution time of  $NF/(u_s + u_1 + \dots + u_N)$ .
- Conclude that the minimum distribution time is in general given by  $\max\{F/u_s, NF/(u_s + u_1 + \dots + u_N)\}$ .

Solution:

- a) Define  $u = u_1 + u_2 + \dots + u_N$ . By assumption

$$u_s \leq (u_s + u)/N \quad \text{Equation 1}$$

Divide the file into  $N$  parts, with the  $i^{\text{th}}$  part having size  $(u_i/u)F$ . The server transmits the  $i^{\text{th}}$  part to peer  $i$  at rate  $r_i = (u_i/u)u_s$ . Note that  $r_1 + r_2 + \dots + r_N = u_s$ , 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

$$(N-1)r_i = (N-1)(u_s u_i)/u \leq u_i,$$

where the last inequality follows from Equation 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

$$r_i + \sum_{j > i} r_j = u_s$$

Thus each peer receives the file in  $F/u_s$ .

- b) Again define  $u = u_1 + u_2 + \dots + u_N$ . By assumption

$$u_s \geq (u_s + u)/N \quad \text{Equation 2}$$

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  $i^{\text{th}}$  part to the  $i^{\text{th}}$  peer ( $i = 1, \dots, N$ ) 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  $(N+1)^{\text{st}}$  part at rate  $r_{N+1}$  to each of the  $N$  peers. The peers do not forward the bits from the  $(N+1)^{\text{st}}$  part.

The aggregate send rate of the server is

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

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

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

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

(For simplicity, we neglected to specify the size of the file part for  $i = 1, \dots, N+1$ . We now provide that here. Let  $\Delta = (u_s+u)/N$  be the distribution time. For  $i = 1, \dots, N$ , the  $i^{\text{th}}$  file part is  $F_i = r_i \Delta$  bits. The  $(N+1)^{\text{st}}$  file part is  $F_{N+1} = r_{N+1} \Delta$  bits. It is straightforward to show that  $F_1 + \dots + F_{N+1} = F$ .)

c) The solution to this part is similar to that of 17 (c). We know from section 2.6 that

$$D_{p2p} \geq \max\{F/u_s, NF/(u_s + u)\}$$

Combining this with (a) and (b) gives the desired result.

5.

In the circular DHT example in Section 2.6.2, suppose that a new peer 6 wants to join the DHT and peer 6 initially only knows peer 15's IP address.

What steps are taken? (See Slide 72 in Chapter 2)

Solution:

Peer 6 would first send peer 15 a message, saying "what will be peer 6's predecessor and successor?" This message gets forwarded through the DHT until it reaches peer 5, who realizes that it will be 6's predecessor and that its current successor, peer 8, will become 6's successor. Next, peer 5 sends this predecessor and successor information back to 6. Peer 6 can now join the DHT by making peer 8 its successor and by notifying peer 5 that it should change its immediate

successor to 6.

6.

In this problem, we are interested in finding out the efficiency of a BitTorrent-like P2P file sharing system. Consider two peers Bob and Alice. They join a torrent with  $M$  peers in total (including Bob and Alice) that are sharing a file consisting of  $N$  chunks. Assume that at a particular time  $t$ , the chunks that a peer has are uniformly at random chosen from all  $N$  chunks, and no peer has all  $N$  chunks. Answer the following questions.

- What is the probability that Bob has all the chunks that Alice has, given that the numbers of chunks that Bob and Alice have are denoted by  $n_b$  and  $n_a$ ?
- Remove part of the conditioning in part a) to find out the probability that Bob has all the chunks that Alice has, given that Alice has  $n_a$  chunks?
- Suppose that each peer in BitTorrent has 5 neighbors. What is the probability that Bob has data that is of interest to at least one of his five neighbors?

Solution:

a).

Note that we assume  $n_b \geq n_a$ .

$\frac{C(N - n_a, n_b - n_a)}{C(N, n_b)}$ , where  $C(N, n)$  is the notation for combination, which means the number of

ways of choosing  $n$  out of  $N$ .

$$\text{b). } p(n_a) = \sum_{n_b=n_a}^{N-1} \frac{1}{N} \frac{C(N - n_a, n_b - n_a)}{C(N, n_b)}.$$

$$\text{c). prob} = 1 - \left( \sum_{n_a=0}^{N-1} \frac{1}{N} p(n_a) \right)^5.$$

For a complete analysis, see:

Donyu Qiu and R. Srikant.

Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks.

ACM Sigcomm 2004, Portland, Oregon, USA