June Kim 301096198

1. (16 points) A system uses statistical time division multiplexing, there are **50 channels** (50 different signals) being combined into one multiplexed signal. Each channel carries frames that contain 400 control bits and 1600 data bits (a total of **2000 bits**). Each channel supports a data rate of **100Kbits per second** (1Kbit = 1000 bits). Assume that **each channel is used an average of 60% of the time**. Assume that we wish to limit the average line **utilization to 95% of the capacity** of the line (the line carrying the multiplexed signals).

What is the capacity of the line required to carry the multiplexed signal (in bits per second, **bps**)?

> total_capacity * utilization_limit = limited_capacity
> limited_capacity >= total_multiplexed_throughput
> total_multiplexed_throughput = all_channels_throughput
> all_channels_throughput = sum(individual_channels_throughput)
> sum(individual_channels_throughput) = number_of_channels * average_channel_throughput
> average_channel_throughput = average_channel_utilization * average_channel_capacity
> average_channel_capacity = data_in_Kbits_per_second * 1000
>
> data_in_Kbits_per_second = 100
>
> average_channel_capacity = 100,000 bits/second
> average_channel_utilization = 60%
>
> average_channel_throughput = 60% * 100,000 bits/second = 60,000 bits/second
> number_of_channels = 50
>
> sum(individual_channels_throughput) = 50 * 60,000 bits/second = 3,000,000 bits/second
> all_channels_throughput = 3,000,000 bits/second
> total_multiplexed_throughput = 3,000,000 bits/second
>
> limited_capacity >= 3,000,000 bits/second
>
> total_capacity = 3,000,000 bits/second / 95% = 3,157,895 bits/second

The minimum capacity needed to cap the utilization of the line to 95% is 3,157,895 bits per second.

2. Consider a T3-type transmission line with a capacity of **44.736 Mbps**. **Two percent** of the capacity of the line is used for carrying control messages that coordinate the sharing of the line. If this line were shared between several users using **synchronous TDM, how many users** would the line support? Assume that each user accesses the line using

a) (3 points) **64 Kbps** voice frequency lines

44.736 - 2% = 43.84128 Mbps available for line data
43.84128 Mbps = 43.84128 Mbps * 1000Kbps/Mbps = 43,841.28 Kbps (error)
43,841.82 Kbps / 64Kbps = 685.02

The line would be able to support up to 685 users simultaneously.

b) (1 points) **1.5 Mbps** cable modem

43.84128 Mbps available for line data / 1.5 Mbps/user = 28.63

The line would be able to support up to 28 users simultaneously.

Next consider the line is shared using **asynchronous TDM**. The percent of line capacity reserved for synchronization/overhead increased to **4%**. Each of the users accessing the line **use an average of 15%** of the time available to them.

c) (5 points) How many users would the line support assuming each user accesses the line using a **1.5 Mbps cable modem**?

44.736 - 4% = 42.94656 Mbps available for line data
1.5Mbps * 15% = throughput_per_user = 0.225 Mbps
capacity/throughput_per_user = user_capacity
42.94656 Mbps / 0.225 Mbps = 190.8736

The line has a capacity of 190 cable modem users via asynchronous TDM.

d) (5 points) Suppose there are **120 users**, using cable model connections at the same time. Find the probability that at any given time, exactly **n users are transmitting simultaneously**. (Hint: use the binomial distribution).

P(X = n) = $\binom{120}{n}$ * (0.15)^n * (0.85)^(120-n)

e) (3 points) Assume that the T3 line in the problem is used within a **packet switched** network. Would you use **Asynchronous or Statistical TDM**? Why?

Statistical TDM. Because it can handle 20% more channels than asynchronous TDM if each source broadcasts 80% of the time. Also, statistical TDM is most useful in systems where sources do not broadcast continuously, such as a packet switched network.

f) (3 points) Assume that the T3 line in the problem is used within a connection oriented **circuit switched** network. Would you use **Asynchronous or Statistical TDM**? Why?

Asynchronous TDM. Because Asynchronous TDM is more likely to preserve the order of packets. Also, in a circuit switched network, utilization is lower and the higher overhead costs of statistical TDM can be foregone.

3. As an example of a communication protocol in the application layer consider DNS.

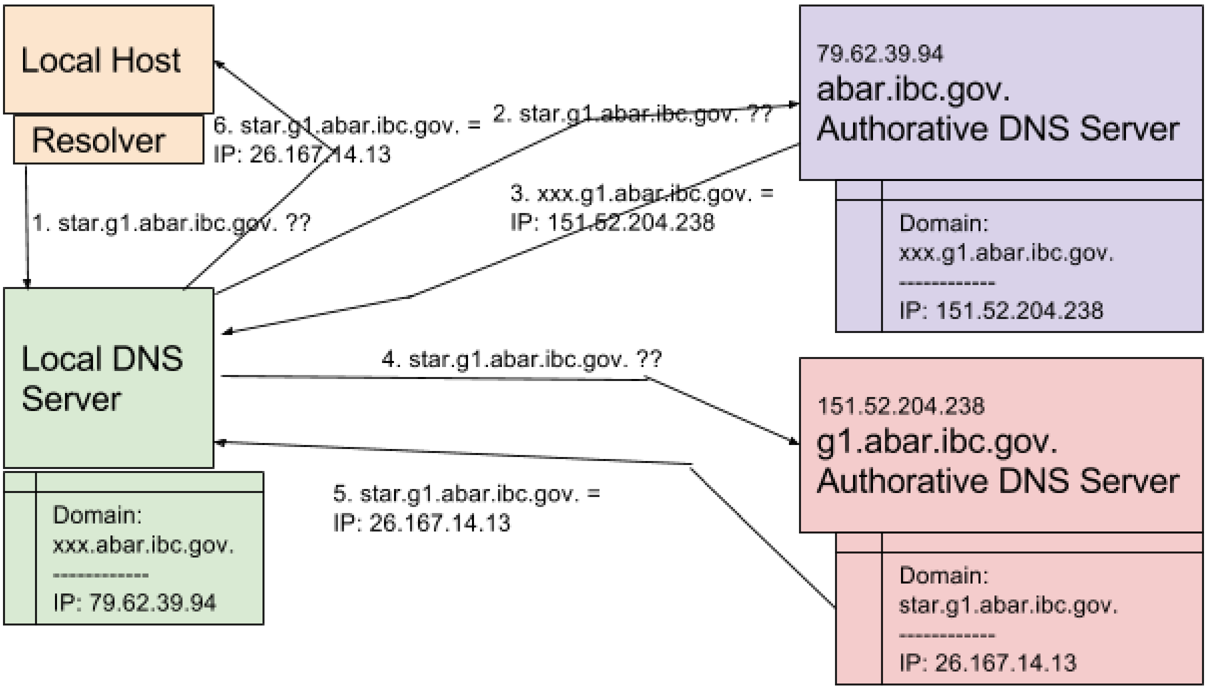
a) [2 point] What does DNS stand for?

Domain Name System

b) [4 points] State two basic uses for DNS?

Assigning domain names to IP addresses in a given network and returning IP addresses to queries for a given domain name.

c) [10 points] **Annotated diagram**



1. The local host attempts to connect to star.g1.abar.ibc.gov. However, the ip address for this domain name does not yet exist on the local host's cache. Unable to directly access IP, it asks the local DNS server about the IP address of the said name.
2. The local DNS server checks its own local cache to check whether the queried domain name is cached in its cache. But it finds no direct IP address to the exact name. So it finds the closest match to the name, which is the suffix xxx.abar.ibc.gov, which is in the cache. There are no closer matches than xxx.abar.ibc.gov. So it fetches the IP address for that suffix, and requests the IP address of the full name to that IP address.This IP address points to the authoritative DNS server for that suffix.
3. The authorative DNS server for the prefix abar.ibc.gov. does not have the address for the full name, either. So it returns the closest thing, which is xxx.g1.abar.ibc.gov.
4. The local server takes the IP for the more specific authorative DNS server for the prefix of g1.abar.ibc.gov. and requests the IP address of the full name yet again.
5. The g1.abar.ibc.gov. Authorative DNS server has the IP address for the full domain name, so it returns the IP address of the full name.
6. The local DNS server finally returns the IP address associated with the full domain name to the local host's resolver, which then is able to make requests directly to the website, star.g1.abar.ibc.gov.

4. Suppose two hosts A and B are directly connected. The distance between the two hosts is 4km. The direct connection can support 100 Mbps (1 Mbit = 2^20 bits), and the propagation speed through the connection is 2*10^8 meters per second. A user has a 3 GB (3*2^30 bits) file to transfer across the connection.

a) (2 points) How long does it take for any given bit in the message to travel from A to B? In other words, what is the propagation delay?

> d_prop = distance/propagation_speed
> distance = 4km
> propagation_speed = 2*10^8 m/s = 200,000 km/s
> d_prop = 4/200,000 = 0.00002 = 0.02 ms

b) (2 points) What is the maximum number of bits that have left A before the first bit arrives at B? This number of bits is called the bandwidth delay product.

> bandwidth_delay_product = (d_prop)* transmission_rate
> transmission_rate = 100Mbps = 100 Mbit/sec * 2^20 bits / Mbit = 209715200 bits/sec
> d_prop = 0.00002s
> bandwidth_delay_product = 104,857,600 bits/sec * 0.00002 sec = 2,097.152 bits

The bandwidth delay product is 2,097.

c) (2 points)We can think of each bit occupying a piece of the link. If the bandwidth delay product tells us that there are 100 bits in the transmission medium then we can think of each bit occupying 1/100 of the length of the transmission media. What is the width of each bit (the length of the transmission medium it fills)?

(This question is badly worded because it describes the width as the length) The the length of the transmission medium it fills represents the inverse of the transmission capacity of the system. That is, higher the capacity, the less time it takes for each bit to be represented. This system includes the physical medium and the transmission capacity.

d) (2 points) Assume that the file is sent as a single message (not broken into packets). What is the transmission time of the file?

> file_size = 3 GB = 3*2^30 bits = 3221225472 bits
> transmission_capacity = 100Mbps = 100 * 2^20 bits/Mbit = 104857600 bits/s
> d_data = 3221225472 bits / 104857600 bits/sec = 30.72s

Now consider that the connection between A and B is through a packet switched network. Message segmentation is the process of breaking our file (or any other message) into smaller sections and sending each of those sections in a separate packet. Assume that each link in the packet switched network has a **capacity of 100Mbps**, the **propagation delay on each link is 0.002s**, the **processing delay on each link is 0.003s**, there are no queuing delays, and the packet is transmitted by A and by **9 additional hosts** as it travels through the packet switched network to B.

e) (5 points) Consider sending the file through the packet switched network without message segmentation. Assume that each of the intermediate hosts is a store and forward node. The time taken by processing and queuing at each node is 0.002s. How long does it take to send the file from A to B?

> d_end_to_end = {d_proc + d_data + d_head + d_prop} * N_trans
> d_proc = 0.002s
> d_prop = 0.003s
> d_data = 30.72s
> d_head = 0
> N_trans = 9
> d_end_to_end = {0.002s + 30.72s + 0s + 0.003s} * 9 = 276.525 s

f) (8 points) Assume that the file is segmented into packets containing 12000 bits of data each. Each packet has a header of 200 bits. If a packet is partially full of data the remainder of the data field is filled with zeros before the resulting full size packet is transmitted. How long does it take for the file to be transmitted through the network (assume no queuing delays).

> d_end_to_end = {d_proc + d_head + d_data} * {N_trans + N_packets - 1} + d_prop * N_trans
> d_proc = 0.002s
> d_prop = 0.003s
> d_head = header_size / transmission_capacity = 200 / 104857600 bits/s = 0.00000190734 s
> d_data = packet_size / transmission_capacity = 12000 / 104857600 bits/s =0.00011444091s
> N_trans = 9
> N_packets = ceiling(file_size / packet_size) = ceiling(3221225472 bits / 12000 bits) = 268436
> d_end_to_end = { 0.002s + 0.00000190734s + 0.00011444091s} * {9 + 268435} + 0.003s * 9
> = 568.15s

g) (6 points) What is the optimal packet size to transmit this file through this network?

Deriving with respect to the packet size, the end-to-end delay, while holding all else constant,

d_end_to_end = {d_proc + d_head + d_data} * {N_trans + N_packets - 1} + d_prop * N_trans
d_end_to_end = { 0.002s + 0.00000190734 s + file_size / N_packets / 104857600 } * {N_trans + N_packets - 1} + 0.003s* 9

Constants cancel

$\delta d_{end-to-end}/\delta N_{packet}$ = $\delta/\delta N_{packet}$({ 0.002 + 0.00000190734 + 3221225472/ 104857600 / N_packets } * {N_packets + 8})

Distributing

$\delta d_{end-to-end}/\delta N_{packet}$ = $\delta/\delta N_{packet}$(N_packets * { 0.00200190734 + 30.72 / N_packets } + 8 * { 0.00200190734 + 30.72 / N_packets })

= $\delta/\delta N_{packet}$(N_packets * { 0.00200190734 + 30.72 / N_packets) +
8 * $\delta/\delta N_{packet}$(0.00200190734 + 30.72 / N_packets }
= 0.00200190734 + 8 * (30.72) * (-1) * N_packets ^ (-2)
= 0.00200190734 - 245.76 / (N_packets ^ 2)

Setting the derivative to 0,

0.00200190734 = 245.76 / (N_packets ^ 2)
N_packets = 350.3754

Rounding, the optimal number of packets is 350.

5. Consider the HTTP protocol. HTTP is the protocol the protocol used for sending the contents of web pages between hosts, from a web server to an agent (client like Firefox or explorer). HTTP is also used by agents to make requests to web servers for particular web pages. HTTP communications, both requests and replies travel through TCP connections. The packets sent back and forth between an agent requesting a web page and the web server receiving the web page were captured using the packet sniffer Wireshark. The data captured is provided for you in the files httpsamplepackets.out and httpsamplelist.out. The httpsamplelist.out file provides a list of packets that were transmitted when the web page http://web.wt.net/~cbenton/relativity.htm was requested using the Firefox browser on a windows XP machine. The file httpsamplepackets.out file contains the contents of the http packets chosen from this list. Based on the information in these two files answer each of the following questions. In each case explain how the contents of the files supports the answer you have given.

a) (4 points) What is the difference between a **basic HTTP GET** request and a **conditional HTTP get** request? When is each type of request used?

> A basic HTTP GET requests a document file in whole, whereas a conditional HTTP GET request signals to the server about the existence of a cache in the client. If the cache is valid - that is, if the client has a copy of the document, then the server will return either the document or nil, depending on if the document has changed since it has been cached. This date is signaled in the If-Modified-Since header.

b) (6 points) What is different about the responses to a basic HTTP GET command and to a conditional HTTP GET request? What information does each type of response return?

> A response to a basic GET always returns an HTTP document upon success, whereas a conditional HTTP GET request could send a 304 Not Modified in place of 200 OK. This 304 signals to the client that the cached version is equivalent to the one on the server. The 200 OK signals that the request has been successfully acknowledged and the correct document returned.

Would you expect a different response to the conditional get if the web page had been modified between the two requests?

> If the server had updated the content of the document between the basic and conditional GET, then the server would return a 200 OK along with the document instead of 304 Not Modified.

c) (2 points) What version of HTTP is the Firefox browser running? What version of HTTP is the queried web server running?

> Firefox is running HTTP/1.0 and the web server is running HTTP/1.1.

d) (1 points) When was the webpage being requested last modified on the web server?

> Sun, 12 Oct 1997 16:57:30 GMT\r\n

e) (1 points) Were the packets carrying the HTTP GET responses fragmented during transmission?

> No.

f) (2 points) What was the IP address of the computer running Firefox?

> 142.58.61.8

What was the IP address of the web server it queried?

206.123.129.3

Now consider the file httpsamples2.out. This file contains a query made to a more complicated web page including multiple objects. Based on the contents of the httpsamples2.out file answer the following questions:

g) (3 points) Were persistent or non-persistent connections used to download the webpage information from the server?

Persistent. The HTTP header includes:
Connection: Keep-Alive\r\n
Keep-Alive: 300\r\n

This is indicative of a persistent connection. In frame 24 onwards, it is clear that multiple requests have been acknowledged without SYN/ACK pairs.

h) (2 points) Was pipelining used to download the webpage information from the server?

Multiple HTTP requests were sent on a single TCP connection without waiting for the corresponding responses. In frames 21 to 23, three GET requests were sent in sequence without waiting for the response for the first one.