1) A system uses statistical time division multiplexing, there are 80 channels (80 different signals) being combined into one multiplexed signal. Each channel carries frames that contain 45 bytes of header and 1155 bytes of data (a total of 1200 bytes). Each channel supports a maximum data rate of 2.5MB per second (1KB = 1024 bytes, 1MB = 1024 KB, 1GB = 1024 MB). Assume that the average data rate passing through each channel is 0.5 MB/s. Assume that the bandwidth capacity of the line carrying the multiplexed signals is 50 MB/s.

a) [3 points] What is the average percentage of the time each channel is used?

Since each channel supports a maximum data rate of 2.5 MB/s, and the average data rate passing through each channel is 0.5 MB/s

The average percentage of the time each channel is used = average data rate passing through each channel / maximum data rate of each channel * 100% = (0.5 MB/s) / (2.5 MB/s) * 100% = 20%

The average percentage of the time each line is used is 20%

b) [3 points] What fraction of the information being sent is overhead?

The overhead in Statistical TDM is the header in each frame, so the fraction of overhead in the information being sent = size of header / size of frame = 45 bytes / 1200 bytes = 0.0375 = 3.75%.

c) [3 points] What is the average input bit rate summed over all signals?

Average data transmission rate of all signals = average data rate passing through each channel * number of channels = 0.5 MB/(s*channels) * 80 channels = 40 MB/s.

But the data rate counts only data bits and the bit rate counts all bit being sent.

Average input bit rate = Average data rate * (bytes in a packet / data bytes in packet) = 40MB/s * (1200/1155) = 41.56MB/s

d) [3 points] What is the average data transmission rate summed over all signals? (data only)

80 channels at an average data rate of 0.5 MB/(s*channel)

Number of channels * average data rate per channel = 80 channels * 0.5MB/(s*channel) = 40MB

e) [3 points] What percentage of the line's capacity will be utilized?

The information being sent through the line is arriving at an average rate of 41.56MBps

Rate / capacity of line = fraction of lines capacity used = 41.56 MBps / 50 MBps = 0.83

0.83 converted to a percentage is 83%

2) Consider a transmission line with a capacity of 180 Mbps (1Mbps = 2^{20} bits per second).

a) [4 points] How many users would the line support assuming each user has a 8 Mbps connection and uses synchronous TDM to access the line? Assume that only 95% of the capacity of the line may be used for user connections.

The size of the utilized part = line capacity * percentage of the capacity of the line be used = 180 Mbps * 95% = 171 Mpbs

Number of users the line support = FLOOR(the size of the utilized part / each user's connection) = FLOOR(171 Mbps / 8 Mbps) = FLOOR(21.375) = 21

b) [4 points] Suppose that each user only used their 8 Mbps connection 28% of the time, 95% of the capacity of the line may be used for user connections and synchronous TDM is being used. How many users can the line support for these assumptions?

The line would still support 21 lines as in a) because with synchronous TDM slots will be transmitted empty or partially full if a full slot of data is not available.

c) [4 points] Suppose that each user only used their 8 Mbps connection 28% of the time, 92% of the capacity of the may be used for user connections and statistical TDM is being used. How many users can the line support for these assumptions?

老师の方法:

Use of lines 28% (lines operational 28% of time) 8Mbps * 0.28 = 2.24Mbps

Number of lines = FLOOR(portion of bit rate of line available / average bit rate per line)

= FLOOR((fraction of line used * line's bit rate) / (bit rate per connection * fraction of connection bandwidth used))

= FLOOR{(180 * 0.92) / (0.28*8Mbps)} = 73 lines

我的方法:

The size of the utilized part = line capacity * percentage of the line of the capacity been used = 180Mbps * 92% = 165.6 Mbps

If all user use 100% of connection, the number of users x_0 = the size of the utilized part / user connection = 165.6 Mbps / 8Mbps = 20.7

Since each user only use 28% of the time, Statistic TDM can handle 1 - 28% = 72% more users.

Suppose Statistic TDM can handle x more users, then

the number of users in 29% connection / percentage of connection time = the extra number of users Statistic TDM can handle / percentage pf un-connection time

So, $20.7/28\% = x/72\%$, $x \approx 53.23$

Thus, the total number of users supported = the number of users in 28% connection + the extra number of users Statistic TDM can handle = $20.7 + 53.23 = 73.93$

Since the number of users must be an integer, and the line cannot support "half user", the number should be rounding down. So the number of users would the line support is 73.

d) [7 points] Suppose there are 80 users, using 3 Mbps connections at the same time. Each user uses their connection 41% of the time. Find the probability that at any given time, exactly 18 users are transmitting simultaneously. (Hint: use the binomial distribution).

Suppose the total number of users is $m = 80$

The number of users are transmitting simultaneously is $n = 16$

The probability that any particular user is transmitting at a given time is $p = 0.41$

The probability that any particular user is not transmitting at a given time is $1 - p = 0.59$

To determine a joint probability we multiply the probabilities of the individual events

If n stations are transmitting when $80 - n$ stations are not transmitting

So the probability that n stations are transmitting is $p^n(1-p)^{(80-n)}$

But there are many ways that n stations can be transmitting, next count those ways

There are $\binom{80}{18}$ ways to select n stations from 80 stations. So this is the number of different ways that n stations can be transmitting.

The probability of exactly 18 users are transmitting simultaneously = $\binom{m}{n} * p^n * (1-p)^{m-n}$ = $\binom{80}{18} * 0.41^{18} * (1-0.41)^{(80-18)} \approx 0.000236$

e) [3 points] Assume that the transmission line in the problem is used within a packet switched network. Would you use Synchronous or Statistical TDM? Why?

If the line were part of a packet switched network, then all users could send packets of data whenever they had such packets to send. No capacity is reserved for any particular communication so any available capacity can be used by any user.

This implies that the line can support the number of users that will on average produce enough packets to saturate(充满) the capacity of the line. Statistical TDM would be the choice in this case as it allows users that are using only part of the capacity of their connection to share the available bandwidth regardless of which of the conversations are currently using bandwidth and which are not transmitting. No time is reserved for conversations that are not transmitting during their slot, that time can be used for other conversations.

f) [3 points] Assume that the transmission line in the problem is used within a connection oriented circuit switched network. Would you use Synchronous or Statistical TDM? Why?

If the line were part of a circuit switched network, then each user would establish a connection. That connection would reserve enough of the T3 line's capacity to carry a continuous data flow at the maximum rate for the connection. That portion of the T3 line's capacity would then be unavailable to other users until the connection was terminated. Therefore, asynchronous TDM would be chosen. It would not be possible for more users than supported by synchronous TDM because only that number of connections could be made.

(asynchronous TDM = statistical TDM)

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

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

Domain Name Service or Domain Name System

b) [8 points] State four basic uses for DNS?

- Finding the IP address associated with a particular domain name (like addraig.cs.sfu.ca, note this is NOT an HTTP URL but a DNS domain name)
- Finding the domain name associated with a particular IP address
- Distributing the Domain name database across many servers throughout the Internet.
- Distributing the responsibility for the administration of parts of the domain name data base between a number of designated authorities.
- Distribution of load between replicated servers (multiple hosts holding identical servers for a particular service like a company's web site) used to distribute the a sites load.
- aliasing(别名), permitting multiple aliases (additional domain names) for the machine with a particular domain name. You might want multiple domain names for one machine if that machine were running multiple servers and you wanted the domain name to indicate the service that was being accessed.
- Aliasing of mail servers

c) A DNS resolver on your local host makes a query to the local DNS server for the IP address of siamese.cat.pet.bc.ca. The DNS server will make iterative queries. The DNS server has recently been initialized (cache empty at initialization, except for information about root servers) and has made only one query before the query for siamese.cat.pet.bc.ca is made. This one previous query was for the IP address of collie.dog.pet.bc.ca

i. [4 points] After the query for collie.dog.pet.bc.ca. what DNS name server records are stored in the cache (ignore information about root servers)?

.ca.

.bc.ca.

.pet.bc.ca.

.dog.pet.bc.ca.

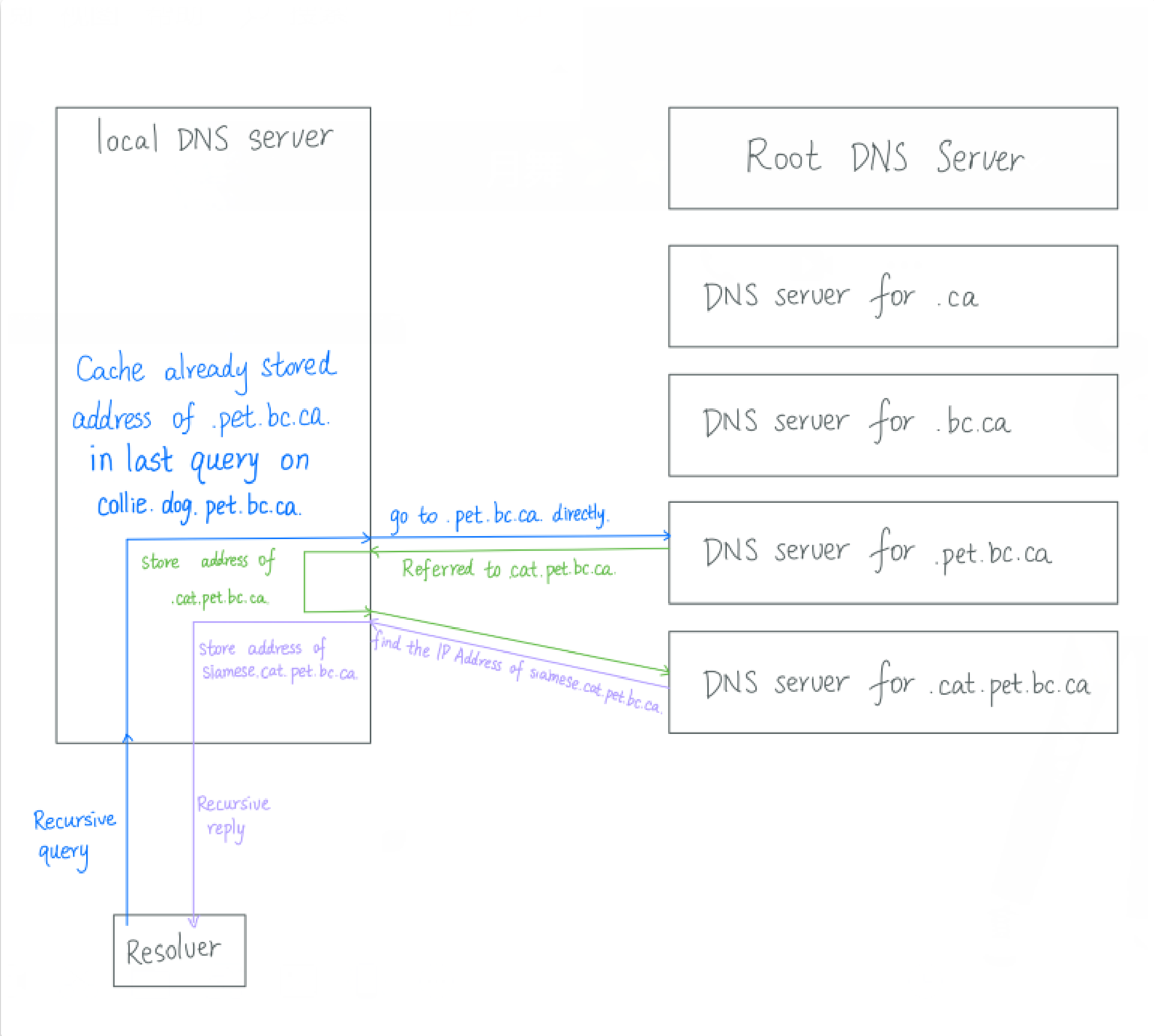
ii. [3 points] Now consider asking for the IP address of siamese.cat.pet.bc.ca. Which DNS server would the first query be sent to? Why?

The names and addresses of the domain .pet.bc.ca. are in the cache of the server so .pet.bc.ca. will be in the cache which will not be found, then it will look for cat.pet.bc.ca in the cache and it will not be found, then it will look for pet.bc.ca. in the cache. It will find pet.bc.ca. and will therefore make the next query to that server.

iii. [2 points] Now consider asking for the IP address of siamese.cat.pet.bc.ca. When making a query to .pet.bc.ca. what would be the DNS record being requested in the query?

The address (record) for siamese.cat.pet.bc.ca (record is not required)

iv. [6 points] Draw an annotated diagram to help you explain how the query for the address of siamese.cat.pet.bc.ca. is executed. Show the Resolver, the local DNS server and all DNS servers queried in your diagram. Show all information travelling between the resolver and the DNS servers in your diagram. Be sure to indicate on your diagram what information is requested and returned in each query. State any assumptions you make. Assume the local DNS server makes iterative queries



4) Suppose two hosts A and B a directly connected. The propagation speed through the direct connection is $2 * 10^8$ **meters per second. The distance between the two hosts is** 5000 **m. The transmission rate supported by the direct connection is 24 Mbps (1 Mbit =** 2^{20} **bits). A user has a 3 Gb (**$3 * 2^{30}$ **bits) file to transfer across the connection.**

a) [1 points] What is the propagation delay?

the propagation delay = distance between the two hosts / propagation velocity = $5000m/(2 * 10^8 m/s)$ = $0.000025s$ = $2.5 * 10^{-5}s$

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.

The first bit left A at time = 0s, and according to part a), it arrives at B at time = $2.5 * 10^{-5}s$, so the time duration is $2.5 * 10^{-5}s$.

The maximum number of bits that have left A before the first bit arrives at B = transmission rate * time duration = $24Mbps * 2^{20} bits/Mbit * 2.5 * 10^{-5} s$ = $0.0006 * 2^{20} bits$ = $1.2288 * 2^{9} bits$ = $629 bits$

Since $1.2288 * 2^{9} bits$ < 3 Gb = $3 * 2^{30}$ bits, the file does not finish transmitting. Thus, the maximum number of bits that have left A before the first bit arrives at B is 629

c) [1 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)?

the width of each bit = distance between two host / bandwidth delay product = $5000m/629bits$ = $7.95m/bits$

So the width of each bit is $50m$

Now consider that the connection between A and B is through a packet switched network. Message segmentation is the process of 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 90Mbps, the propagation delay on each link is 0.004s, 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.

d) [5 points] Consider sending the file through the packet switched network without message segmentation (as a single packet). Assume that each of the intermediate hosts are store and forward nodes. How long does it take to send the file from A to B?

transmission delay on each link = length of the packet / transmission speed = $(3 * 2^{30} bits)/(90Mbps * 2^{20} bits/Mbits)$ = $(1/15) * 2^{9} s$ = $1.0667 * 2^{5} s$ = $34.1333s$

delay on each link = processing delay + propagation delay + transmission delay = $0.003s + 0.004s + 34.1333s = 34.1403s$

There are 9 hosts between A and B, so there are total $9 + 1 + 1 - 1$ = 10 links.

Thus, end to end delay = delay on each link * number of links = $34.1403s * 10$ = $341.403s$

e) [6 points] Assume that the file is segmented into packets containing 12000 bits 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).

bits of data in each packet = packet size - bits of header = $12000bits - 200bits = 11800bits$

number of packets the file will divided into = file size / bits of data in each packet = $3 * 2^{30}bits/11800bits = 272985.209492$

Since the number of packets should be an integer, the last packet will contain data that not fully filled the packet.

So there are totally 272986 packets.

Then, the transmission delay on each link for each packet = packet size / transmission speed = $12000bits/(90Mbps * 2^{20}bits/Mbits) = 25/196608 = 0.000127s$

At each store and forward node and at the destination, the number of times the packet is transmitted is N=10.

The delay for datagram packet switching is

$$d_{endtoend} = (d_{proc} + d_{prop}) * (N_{trans}) + (d_{data} + d_{head}) * \{N_{pack} + (N_{pack} - 1)\}$$

= $(0.003 + 0.004) * (10) + 0.000127 * \{272986 + (10 - 1)\}$

= $0.07 + 34.67$

= 34.74

f) [0 points] What is the optimal packet size to transmit this file through this network. (will not be graded but may appear on a quiz or final)

Suppose the optimal packet size is x bits.

bits of data in each packet = packet size - bits of header = x bits - 200 bits.

number of packets the file will divided into = file size / bits of data in each packet = $3 * 2^{30}$ bits / (x bits - 200 bits)

The transmission delay on each link for each packet = packet size / transmission speed = x bits / (90Mbps * 2^{20} bits/Mbits).

Thus, the end to end delay

= (processing delay + propagation delay + transmission delay) * (number of packets + number of links - 1)

= $(0.003 + 0.004 + x/(90 * 2^{20})) * (3 * 2^{30}/(x - 200) + 10 - 1)$

= $0.1 * 2^{-20} + 2^9 x/(15(x - 200)) + (0.021 * 2^{30})/(x - 200) + 0.063$

Let $u = x - 200$,

= $0.1 * 2^{-20} * (u + 200) + 2^9(u + 200)/(15u) + (0.021 * 2^{30})/u + 0.063$

= $0.1 * 2^{-20}u + (0.021 * 2^{30} + 200 * 2^9/15)/u + 20 * 2^{-20} + 0.063$

According to Inequality of arithmetic and geometric means, when

$0.1 * 2^{-20}u = (0.021 * 2^{30} + 200 * 2^9/15)/u$,

the equation get the minimum value, which is the minimum end to end delay in this case.

We get $u = 15378899.9355$

Since the number of bits should be an integer, $u = 15378899$ or $u = 15378900$

When $u = 15378899$, the end to end delay = $2.99631145009s$

When $u = 15378900$, the end to end delay = $2.99631145009s$

Hence, we can choose either $u = 15378899$ or $u = 15378900$

Then, the optimal packet size $x = u + 200$, so the optimal packet size can be either 15379099 bits or 15379100 bits.

5) Consider the HTTP protocol. HTTP is 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 web pages and the web server receiving the web page were captured using the packet sniffer Wireshark. All the provided files provide a list of packets that were transmitted when web pages were requested using the chrome browser on a windows 10 machine.

You can download Wireshark and open the pcapng data files supplied with this problem. This will allow you to open the pcanpng files. These files contain a great deal of information about each packet. I will discuss some of the things you can do with Wireshark at the beginning of the class on May 27. You can capture your own packets, but please be aware that many license agreements specify you will not capture packets from the application, and many experiments using Wireshark will flag you to network admins of the networks you work on as a potential hacker. You do not need to capture any packets to complete this problem. Please use the supplied Wireshark files.

Summary data was captured and is provided for you in the files HTTP2020summary.pcapng and HTTP2020conversation.pcapng. The summary file contains the HTTP packets only. The conversation file contains both the HTTP packets and the TCP packets used to transmit the data sent by the server in response the HTTP requests. Both files include both the initial request for one or more web pages and the responses to those requests.

Based on the information in these 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? Give an example packet from one of the data files for at least one and if possible both cases. Explain briefly how you found the packets and what items within the packets tell you if the packet is executing a GET or a conditional GET.

A GET request asks the server send the desired object

A conditional GET request asks the server to send the desired object only if the copy of the object on the server is newer than the copy of the object already on the host.

Only a conditional GET will contain an 'if modified since' directive

Frame (7, 18, 33) in the file http2020conversation.pcapng is a conditional GET

Frame (49, 60, 79, 80, 107, 113, 114) is a GET (not conditional)

Packets were found by looking for labeled by Wireshark as HTTP GET frames. Then by looking in the frame from the 'if modified since' directive to determine if the GET was a conditional GET (only conditional gets contain this directive(指令))

A conditional HTTP GET request includes an additional directive in the header section. For the conditional HTTP GET request in frame 18 the directive is

```
IF-Modified-Since: Wed, 27 May 2020 06:40:36 GMT\r\n
```

b) [4 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? Would you expect a different response to the conditional get if the web page had been modified between the two requests?

When a basic GET request is made, the object being requested is returned in the entity body of the HTTP response. The entity body is sent in the TCP packets and the HTTP OK Packet. The packets included in the entity body are listed in the HTTP OK message.

When a conditional GET request is made and the server has a new version of the object than the local cache the requested object is also returned inside the entity body as for a basic GET request.

For example, for a basic GET: In frame 53 there is a response to the conditional GET in frame 49. The HTTP response tells us that the object was returned in a series of TCP packets (including the response)

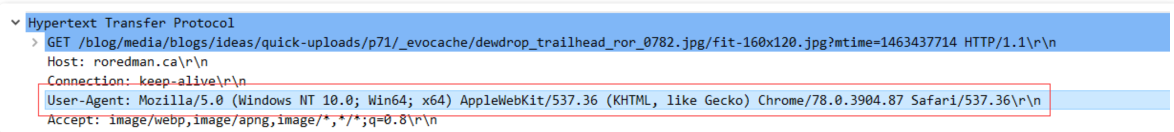
```
[3 Reassembled TCP Segments (4042 bytes): #51(1460), #52(1460), #53(1122)]
```

For a conditional GET request is made (as in Frame 18) and the requested object has not been modified since the last GET request, then the server returns a message indicating that the object has not been modified. This can be seen in frame 20. In this case the object is not present in the (empty) entity body. Instead a message `"HTTP/1.1 304 Not Modified\r\n"` is present in the status line of the HTTP response.

```
[6 Reassembled TCP segments (8072 bytes): #9(1460), #10(1460), #11(1460),
#12(1460), #13(1460), #14(1402)]
```

c) [3 points] What browser is making the request? What version of HTTP is that browser running? What version of HTTP is the queried web server running? Give reasons for your answers based on selected packets from the supplied packet captures.

From frame 114 in HTTP2020conversational.pcapng (or any HTTP request frame in lists in part a) each of the questions above can be answered

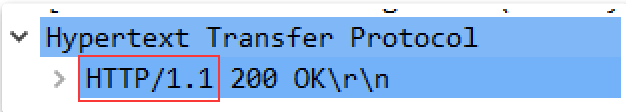

```
Hypertext Transfer Protocol
  GET /blog/media/blogs/ideas/quick-uploads/p71/_evocache/dewdrop_trailhead_ror_0782.jpg/fit-160x120.jpg?mtime=1463437714 HTTP/1.1\r\n
  Host: roredman.ca\r\n
  Connection: keep-alive\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.87 Safari/537.36\r\n
  Accept: image/webp,image/apng,image/*,*/*;q=0.8\r\n
```

The browser being used is Chrome or Safari (only one need be mentioned in a student solution, this information line is available in any of the response frames listed in a)

Chrome is running version HTTP1.1.

In the HTTP GET request (the first line in the Wireshark packet analysis) inside the HTTP GET message the last item in the line is HTTP/1.1 which indicates that the browser is using HTTP 1.1.

The webserver is running HTTP1.1 as is seen in the first statement in HTTP OK message in frame 47



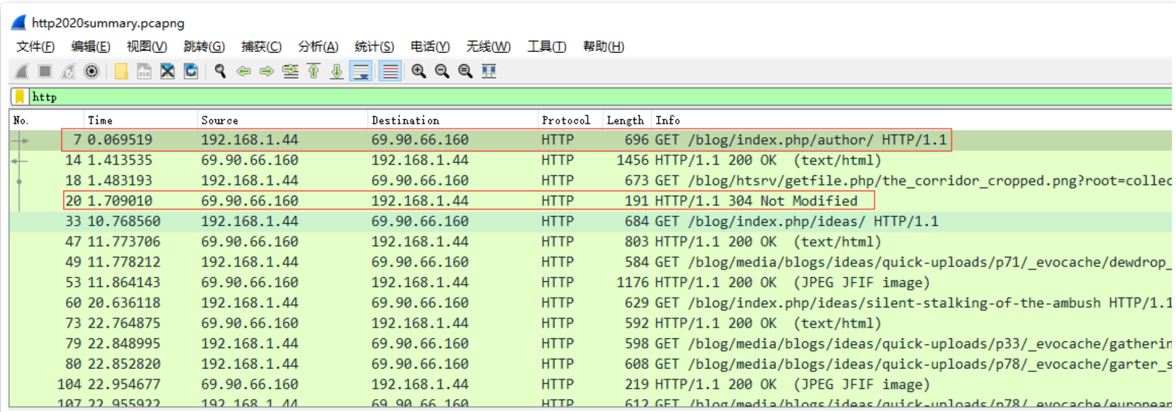
d) [2 points] What was the IP address of the computer running the browser? What was the IP address of the web server it queried? Use evidence from the file HTTP2020summary.pcapng.

我的答案:

The IP address of the computer running the browser: 192.168.1.44

The IP address of the web server it queried: 69.90.66.160

As the screenshot below shows, request No.7 is a HTTP GET request send from 192.168.1.44 to 69.90.66.160, so 192.168.1.44 is the client, and 69.90.66.160 is a server. No.20 is a GET response message send from 69.90.66.160 to 192.168.1.44, so 69.90.66.160 should be the queried server and 192.168.1.44 should be the client.



No.	Time	Source	Destination	Protocol	Length	Info
7	0.069519	192.168.1.44	69.90.66.160	HTTP	696	GET /blog/index.php/author/ HTTP/1.1
14	1.413535	69.90.66.160	192.168.1.44	HTTP	1456	HTTP/1.1 200 OK (text/html)
18	1.483193	192.168.1.44	69.90.66.160	HTTP	673	GET /blog/htsrv/getfile.php/the_corridor_cropped.png?root=collec
20	1.709010	69.90.66.160	192.168.1.44	HTTP	191	HTTP/1.1 304 Not Modified
33	10.768560	192.168.1.44	69.90.66.160	HTTP	684	GET /blog/index.php/ideas/ HTTP/1.1
47	11.773706	69.90.66.160	192.168.1.44	HTTP	803	HTTP/1.1 200 OK (text/html)
49	11.778212	192.168.1.44	69.90.66.160	HTTP	584	GET /blog/media/blogs/ideas/quick-uploads/p71/_evocache/dewdrop_
53	11.864143	69.90.66.160	192.168.1.44	HTTP	1176	HTTP/1.1 200 OK (JPEG JFIF image)
60	20.636118	192.168.1.44	69.90.66.160	HTTP	629	GET /blog/index.php/ideas/silent-stalking-of-the-ambush HTTP/1.1
73	22.764875	69.90.66.160	192.168.1.44	HTTP	592	HTTP/1.1 200 OK (text/html)
79	22.848995	192.168.1.44	69.90.66.160	HTTP	598	GET /blog/media/blogs/ideas/quick-uploads/p33/_evocache/gatherin
80	22.852820	192.168.1.44	69.90.66.160	HTTP	608	GET /blog/media/blogs/ideas/quick-uploads/p78/_evocache/garter_s
104	22.954677	69.90.66.160	192.168.1.44	HTTP	219	HTTP/1.1 200 OK (JPEG JFIF image)

老师的答案:

Consider frame 49 which contains an HTTP GET message. The description of the captured packet for that line is

```
49 11.778212      192.168.1.44          69.90.66.160              49416 HTTP
                  GET /blog/media/blogs/ideas/quick-uploads/p71/_evocache…
```

Since this packet is a query,

- The source address is the address of the computer running the browser
- Address of computer running browser is 192.162.1.44.
- The destination address is the address of the HTTP server
- Address of the HTTP server is 69.90.66.160

e) [2 point] For one of the HTTP GET responses shown in file HTTP2020conversation.pcapng how many packets were used to carry the HTTP GET response from the server to the client? How did you determine your answer?

As an example consider the HTTP GET in frame 60 and the object returned by the server in response to that GET. The information regarding this exchange is in frames 60, 61 and 65-73. The HTTP GET request is in frame 60, the HTTP REPLY is in frame 73. Frame 61 and 65-73 are returning the object to the client. These packets contain data being sent from the server to the client. We can count these packets or look inside the HTTP OK frame (73) and find the line

```
[9 Reassembled TCP Segments (12070 bytes): #65(1460), #66(1460), #67(1460),
, #68(1460), #69(1460), #70(1312), #71(1460), #72(1460), #73(538)]
```

So there are 9 frames used to carry data from the server to the client in response to the HTTP query.

Again consider the file HTTP2020summary.pcapng and the file HTTP20202conversation.pcapng. These files contains a query made for a web page including multiple objects. Based on the contents of these files answer the following questions:

f) [3 points] Were persistent or non-persistent connections used to download the webpage information from the server? Explain why you think so?

Persistent connections were used to download more than half of the webpage information from the server. Several connections using different TCP ports (49414-49421) are present in the sample packets. Half of the connections are used to transmit more than one GET message and GET reply (and are therefore definitely persistent). For example, the connection on port 49414 requests multiple objects in HTTP GET requests in packets 7 and 18. It is reasonable to assume that all (or most) connections are persistent connections.

g) [2 points] Can you see any evidence that pipelining was used to download the webpage information from the server? Explain why you think pipelining was used, or why you think pipelining was not used.

<u>Prof's Answer</u>

Pipelining does not seem to have been used to download the webpage information from the server. When each connection is separately considered each HTTP reply and request pair is separate, one request reply pair completes before the next pair starts. For example for the connection using port 49416 the GET in frame 33 and reply 47 in frame 47 is followed by a GET in frame 49 by an OK in frame 53.

<u>My Answer</u>

Pipelining was used to download the webpage information form the server. As shown in the screenshot below, No. 107, No. 113, and No. 114 are HTTP GET requests, and the response of them are No. 150, No. 164, and No. 194. The three requests all come to the server before the server response to request No. 107. Hence, pipelining was used to download the webpage information from the server.

107	22.955922	192.168.1.44	69.90.66.160	HTTP	612	GET /blog/media/blogs/ideas/quick-uploads/p78/_evocache/european_skipper_ror_1200.jpg/fit-400x320.jpg?mtime=1467062443 HTTP…
113	22.974354	192.168.1.44	69.90.66.160	HTTP	607	GET /blog/media/blogs/ideas/quick-uploads/p78/_evocache/red_admiral_ror_1257.jpg/fit-400x320.jpg?mtime=1467062444 HTTP/1.1
114	22.974557	192.168.1.44	69.90.66.160	HTTP	598	GET /blog/media/users/leslie/profile_pictures/_evocache/p7210161.jpg/crop-top-64x64.jpg?mtime=1439568253 HTTP/1.1
138	23.020807	69.90.66.160	192.168.1.44	HTTP	214	HTTP/1.1 200 OK (JPEG JFIF image)
150	23.057949	69.90.66.160	192.168.1.44	HTTP	973	HTTP/1.1 200 OK (JPEG JFIF image)
164	23.060910	69.90.66.160	192.168.1.44	HTTP	1450	HTTP/1.1 200 OK (JPEG JFIF image)
194	23.123033	69.90.66.160	192.168.1.44	HTTP	756	HTTP/1.1 200 OK (JPEG JFIF image)