

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



Mạng máy tính

Bài thực hành 2

GVHD: Nguyễn Lê Duy Lai
SV: Nguyễn Minh Khánh - 2211523

TP. HỒ CHÍ MINH, THÁNG 10/2024

4a

Question 1

What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the "details of the selected packet header window" (refer to Figure 2 in the "Getting Started with Wireshark" Lab if you're uncertain about the Wireshark windows).

Answer: IP address: 192.168.137.66, TCP port: 52435

Question 2

What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

Answer: IP address: 128.119.245.12, TCP port: 80

Question 3

What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

Answer: I do above



Question 4

What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

Answer: Sequence number: 0. By a TCP flag: (SYN)

Question 5

What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

Answer: According to the above figure, the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN is 0. The value of the acknowledgement field in the SYNACK segment is 1. The value of the Acknowledgement field in the SYNACK segment is determined by the server gaia.cs.umass.edu. The server adds 1 to the initial sequence number of the SYN segment from the client computer. For this case, the initial sequence number of the SYN segment from the client computer is 0, thus the value of the Acknowledgement field in the SYNACK segment is 1. A segment will be identified as a SYNACK segment if both SYN flag and Acknowledgement in the segment are set to 1.

Question 6

What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.

Answer: Sequence number: 1

No.	Time	Source	Destination	Protocol	Length	Info
88	1.066410	192.168.137.66	216.239.36.178	TCP	54	52436 → 443 [ACK] Seq=1 Win=262400 Len=0
89	1.068458	192.168.137.66	216.239.36.178	TLSv1.3	340	Client Hello (SHA=www.google-analytics.com)
90	1.080541	192.168.137.66	216.239.36.178	TCP	54	443 → 52437 [ACK] Seq=1 Win=262400 Len=0
91	1.081541	192.168.137.66	216.239.36.178	TCP	54	52437 → 443 [ACK] Seq=1 Win=262400 Len=0
92	1.082437	192.168.137.66	216.239.36.178	TLSv1.3	571	Client Hello (SHA=www.google-analytics.com)
93	1.080178	216.239.36.178	192.168.137.66	TCP	54	443 → 52436 [ACK] Seq=1 Win=262400 Len=0
94	1.078112	216.239.36.178	192.168.137.66	TCP	54	443 → 52437 [ACK] Seq=1 Win=262400 Len=0
95	1.101069	192.168.137.66	216.239.36.178	TLSv1.3	1514	Server Hello, Change Cipher Spec
96	1.101069	192.168.137.66	216.239.36.178	TCP	1514	443 → 52437 [ACK] Seq=1 Win=262400 Len=0
97	1.101069	192.168.137.66	216.239.36.178	TLSv1.3	1096	Application Data
98	1.101069	192.168.137.66	216.239.36.178	TCP	54	52437 → 443 [ACK] Seq=1 Win=262400 Len=0
99	1.111145	192.168.137.66	216.239.36.178	TLSv1.3	134	Change Cipher Spec, Application Data
100	1.111269	192.168.137.66	216.239.36.178	TLSv1.3	276	Application Data
101	1.112769	216.239.36.178	192.168.137.66	TLSv1.3	1466	Server Hello, Change Cipher Spec
102	1.112769	216.239.36.178	192.168.137.66	TCP	1466	443 → 52436 [ACK] Seq=1 Win=262400 Len=0
103	1.112769	216.239.36.178	192.168.137.66	TLSv1.3	539	Application Data
104	1.112769	216.239.36.178	192.168.137.66	TCP	54	52436 → 443 [ACK] Seq=1 Win=262400 Len=0
105	1.116324	192.168.137.66	216.239.36.178	TLSv1.3	134	Change Cipher Spec, Application Data
106	1.124736	192.168.137.66	216.239.36.178	TLSv1.3	141	Application Data

Frame 101: 1466 bytes on wire (11728 bits), 1466 bytes captured (11728 bits) on interface v...
Ethernet II, Src: P2000 (08:00:27:00:00:00), Dst: CloudNetwork_44-9a:7f (08:00:27:00:00:00)
Internet Protocol Version 4, Src: 216.239.36.178, Dst: 192.168.137.66
Transmission Control Protocol, Src Port: 443, Dst Port: 52436, Seq: 1413, Ack: 287, Len: 14
Source Port: 443
Destination Port: 52436
[Stream index: 2]
[Stream Packet Number: 7]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP segment len: 1413]
[Sequence Number: 1413] (relative sequence number)
[Next Sequence Number: 2825] (relative sequence number)
[Acknowledgment Number: 287] (relative ack number)
[Acknowledgment number (raw): 3557135968]

Question 7

Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent,

and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the EstimatedRTT value (see Section 3.5.3, page 242 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 242 for all subsequent segments.

Answer: According to above figures, the segments 1-6 are No. 182, 183, 184, 185, 186, 187. The ACK of segments 1-6 are No. 221, 222, 242, 247, 248 and 249. Segment 1 sequence number is 1. Segment 2 sequence number is 643. Segment 3 sequence number is 2095. Segment 4 sequence number is 3547. Segment 5 sequence number is 4999. Segment 6 sequence number is 6451.

	A	B	C	D
	Sent time		ACK received time	RTT
Seg1		2.694624	2.979896	0.285272
Seg2		2.694742	2.979896	0.285154
Seg3		2.694742	3.250986	0.556244
Seg4		2.694742	3.268827	0.574085
Seg5		2.694742	3.268827	0.574085
Seg6		2.694742	3.268827	0.574085

Question 8

What is the length of each of the first six TCP segments?

Answer: The length of the first TCP segment is 696 bytes. The length of each of the following five TCP segments is 1506 bytes.

Time	Source	Destination	Sequence	Length	Flags
132.1.221662	192.168.137.66	216.239.36.178	TCP	54	52436 → 443 [ACK] Seq=1773 Ack=693 Min=26120 Len=0
133.1.224640	192.168.137.66	216.239.36.178	TLSv1.3	311	Application Data
134.1.225726	192.168.137.66	216.239.36.178	TLSv1.3	1303	Application Data
135.1.253383	192.168.137.66	184.16.83.69	TCP	54	52437 → 443 [ACK] Seq=1820 Ack=5813 Min=263168 Len=0
136.1.256322	216.239.36.178	192.168.137.66	TCP	54	443 → 52436 [ACK] Seq=6993 Ack=3279 Min=267008 Len=0
138.1.268879	216.239.36.178	192.168.137.66	TLSv1.3	853	Application Data
140.1.314593	192.168.137.66	216.239.36.178	TCP	54	52436 → 443 [ACK] Seq=3279 Ack=6892 Min=262400 Len=0
182.2.694624	192.168.137.66	128.119.245.12	TCP	1506	52435 → 80 [PSH, ACK] Seq=1 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
183.2.694742	192.168.137.66	128.119.245.12	TCP	1506	52435 → 80 [ACK] Seq=443 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
184.2.694742	192.168.137.66	128.119.245.12	TCP	1506	52435 → 80 [ACK] Seq=2095 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
185.2.694742	192.168.137.66	128.119.245.12	TCP	1506	52435 → 80 [ACK] Seq=3547 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
186.2.694742	192.168.137.66	128.119.245.12	TCP	1506	52435 → 80 [ACK] Seq=4999 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
187.2.694742	192.168.137.66	128.119.245.12	TCP	1506	52435 → 80 [ACK] Seq=6451 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
188.2.694742	192.168.137.66	128.119.245.12	TCP	1506	52435 → 80 [ACK] Seq=7983 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
189.2.694742	192.168.137.66	128.119.245.12	TCP	1506	52435 → 80 [ACK] Seq=9355 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
190.2.694742	192.168.137.66	128.119.245.12	TCP	1506	52435 → 80 [ACK] Seq=10887 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
191.2.694742	192.168.137.66	128.119.245.12	TCP	1506	52435 → 80 [ACK] Seq=12259 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
221.2.979896	128.119.245.12	192.168.137.66	TCP	54	80 → 52435 [ACK] Seq=1 Ack=643 Min=30592 Len=0
222.2.979896	128.119.245.12	192.168.137.66	TCP	54	80 → 52435 [ACK] Seq=1 Ack=13711 Min=56784 Len=0
223.2.980877	192.168.137.66	128.119.245.12	TCP	1506	52435 → 80 [PSH, ACK] Seq=13711 Ack=1 Min=302656 Len=1452 [TCP PDU reassembled in 465]

Question 9

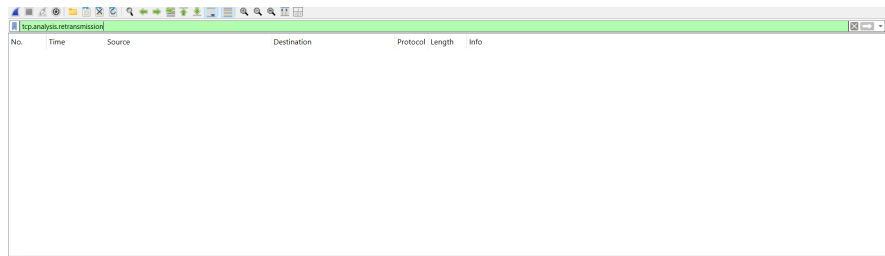
What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

Answer: The minimum amount of available buffer space advertised at the receiver for the entire trace is 29200. According to the trace, the sender is never throttled due to lacking receiver buffer space.

Question 10

This web page contains images. Before retrieving each image, does your host issue new DNS queries?

Answer: There are no retransmitted segments in the trace file.



Question 11

How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 247 in the text)?

Answer: The difference between the acknowledged sequence numbers of two consecutive ACKs indicates the data received by the server between these two ACKs. The receiver is ACKing every other segment. For example, the segment No. 13 acknowledged data with 1452 bytes.

285	3.522299	128.119.245.12	192.168.137.66	TCP	54	80 → 52435 [ACK] Seq=1 Ack=45655 Win=126576 Len=0
286	3.522407	192.168.137.66	128.119.245.12	TCP	1586	52435 → 80 [ACK] Seq=90607 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
287	3.522407	192.168.137.66	128.119.245.12	TCP	1586	52435 → 80 [ACK] Seq=92119 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
288	3.522407	192.168.137.66	128.119.245.12	TCP	1586	52435 → 80 [ACK] Seq=93571 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
289	3.522407	192.168.137.66	128.119.245.12	TCP	1586	52435 → 80 [ACK] Seq=95023 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
290	3.522407	192.168.137.66	128.119.245.12	TCP	1586	52435 → 80 [ACK] Seq=96475 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
291	3.522407	192.168.137.66	128.119.245.12	TCP	1586	52435 → 80 [PSH, ACK] Seq=97927 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
292	3.522407	192.168.137.66	128.119.245.12	TCP	1586	52435 → 80 [ACK] Seq=99379 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
293	3.522407	192.168.137.66	128.119.245.12	TCP	1586	52435 → 80 [ACK] Seq=100831 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
294	3.551945	128.119.245.12	192.168.137.66	TCP	54	80 → 52435 [ACK] Seq=1 Ack=98011 Win=123208 Len=0
295	3.551945	128.119.245.12	192.168.137.66	TCP	54	80 → 52435 [ACK] Seq=1 Ack=98011 Win=123208 Len=0
296	3.551945	128.119.245.12	192.168.137.66	TCP	54	80 → 52435 [ACK] Seq=1 Ack=98011 Win=123208 Len=0
297	3.551945	128.119.245.12	192.168.137.66	TCP	54	80 → 52435 [ACK] Seq=1 Ack=98011 Win=123208 Len=0
298	3.551945	128.119.245.12	192.168.137.66	TCP	54	80 → 52435 [ACK] Seq=1 Ack=98011 Win=123208 Len=0
299	3.551945	128.119.245.12	192.168.137.66	TCP	54	80 → 52435 [ACK] Seq=1 Ack=98011 Win=123208 Len=0
300	3.551945	128.119.245.12	192.168.137.66	TCP	54	80 → 52435 [ACK] Seq=1 Ack=98011 Win=123208 Len=0
301	3.551945	128.119.245.12	192.168.137.66	TCP	54	80 → 52435 [ACK] Seq=1 Ack=98011 Win=123208 Len=0
302	3.552067	192.168.137.66	128.119.245.12	TCP	1586	52435 → 80 [ACK] Seq=102283 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]
303	3.552067	192.168.137.66	128.119.245.12	TCP	1586	52435 → 80 [ACK] Seq=103735 Ack=1 Min=262656 Len=1452 [TCP PDU reassembled in 465]

Question 12

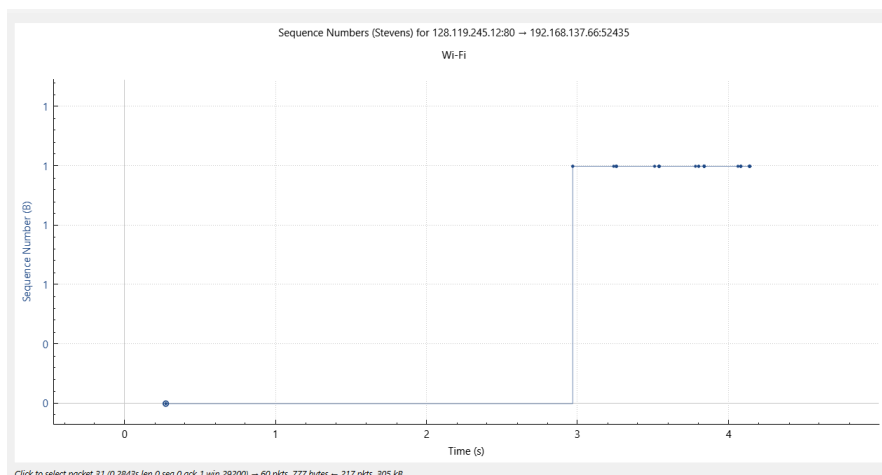
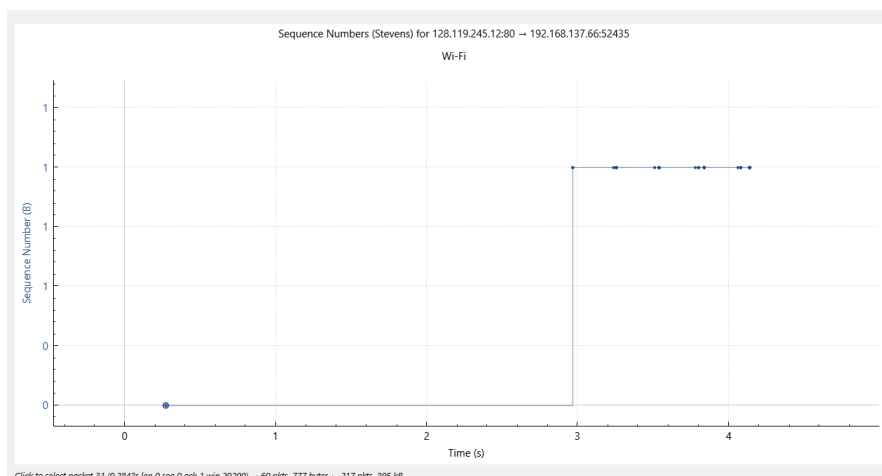
What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

Answer: The alice.txt on the hard drive is bytes, and the download time is (First TCP segment) - (last ACK) = second. Therefore, the throughput for the TCP connection is computed as bytes/second.

Question 13

Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP's slow start phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

Answer: The slow start of the TCP seems to begin at about 0.2 seconds and then ends at about 2.9 seconds. Congestion avoidance takes over at about 2.7 seconds because it cut down the amount being sent.



Question 14

This web page contains images. Before retrieving each image, does your host issue new DNS queries?

Answer: There are no retransmitted segments in the trace file.

Question 5a

Question 1

What is the IP address of your computer?

Answer: IP address: 192.168.137.66

Question 2

Within the IP packet header, what is the value in the upper layer protocol field?

Answer: The value of the upper layer protocol field is ICMP.



Question 3

Question 4

Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

Answer: The fragment offset is set to 0, therefore, the packet has not been fragmented.

Question 5

Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?

Answer: The header checksum and the Identification changes from each datagram to the next.

Question 6

Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?

Answer: Version(IPv4), Length of header, Source IP(sending from same place), Destination IP(contacting same site), Upper layer protocol(always using ICMP,) Fields that must stay constant: Same as above. The fields that must change are: The header checksum (header changes), Identification(to verify packets).

Question 7

Describe the pattern you see in the values in the Identification field of the IP datagram.

Answer: The pattern in the identification field is that the field increases by one in each strand of echo requests. (see video for proof)

Question 8

What is the value in the Identification field and the TTL field?

Question 9

Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?

Answer: The Identification field changes from all of the replies because this field has to have a unique value. If they(2 or more replies) have the same value then the replies must be fragments of a bigger packet. The TTL field does not change because the time to live to the first hop router is always the same.

Question 10

Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 2000. Has that message been fragmented across more than one IP datagram?

Answer: Yes, that message has been fragmented across more than one IP datagram.



Question 11

Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?

Answer:

Question 12

Print out the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are there more fragments? How can you tell?

Answer: [Your answer here]

Question 13

What fields change in the IP header between the first and second fragment?

Answer: The fields that change are Length, Flags Set, Fragment offset, header checksum

Question 14

How many fragments were created from the original datagram?

Answer:

Question 15

What fields change in the IP header among the fragments?

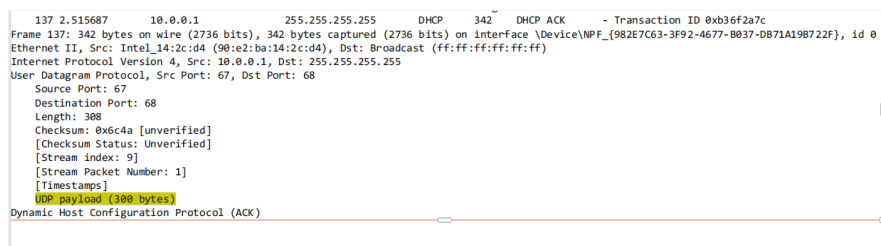
Answer:

Question 5b

Question 1

Are DHCP messages sent over UDP or TCP?

Answer: UDP

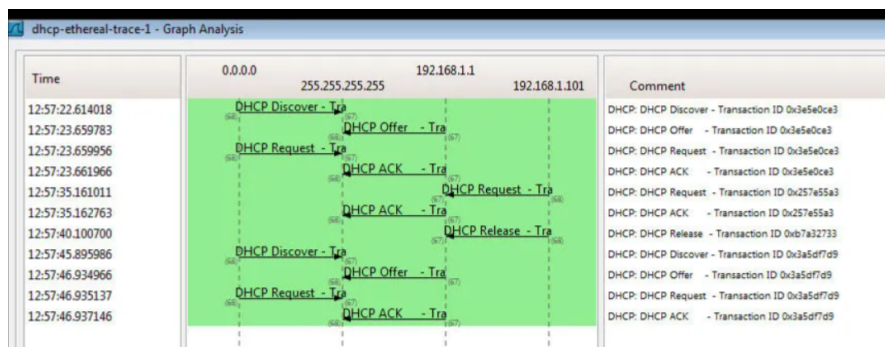


Question 2

Draw a timing diagram illustrating the sequence of the first four-packet Discover/Offer/Request/ACK DHCP exchange between the client and server. For each packet, indicate the source

and destination port numbers. Are the port numbers the same as in the example given in this lab assignment?

Answer: The port numbers are the same as in the example given in this lab assignment. The



Discover packet has a source port of 68 and destination port of 67. The Offer packet has a source port of 67 and a destination port of 68. The Request packet has a source port of 68 and a destination of 67. The ACK packet has a source port of 67 and a destination of 68. All of this corresponds to the example given in the lab.

Question 3

Consider now the HTTP GET sent from the client to the Google server (whose IP address is 64.233.169.104) at time 7.109267. What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET?

Answer: 38:d5:7a:44:9a:7f



Question 4

What values in the DHCP discover message differentiate this message from the DHCP request message?

Answer: Message type, Discover: Boot Request, Request: Boot Reply

Question 5

What is the value of the Transaction-ID in each of the first four (Discover/Offer/Request/ACK) DHCP messages? What are the values of the Transaction-ID in the second set (Request/ACK) of DHCP messages? What is the purpose of the Transaction-ID field?

Answer: The first set Transaction ID: Discover: 0xba9836e8 / Offer: 0xba9836e8 / Request: 0xba9836e8 / ACK: 0xba9836e8. The second set Transaction ID: Request: 0x7d44c87b / ACK: 0x7d44c87b. The Transaction-ID helps track and match requests and responses within a DHCP transaction session.

Question 6

A host uses DHCP to obtain an IP address, among other things. But a host's IP address is not confirmed until the end of the four-message exchange! If the IP address is not set until the end of the four-message exchange, then what values are used in the IP datagrams in the four-message exchange? For each of the four DHCP messages (Discover/Offer/Request/ACK), indicate the source and destination IP addresses that are carried in the encapsulating IP datagram.

Answer:

- Discover: Source IP = 0.0.0.0, Destination IP = 255.255.255.255
- Offer: Source IP = 10.0.0.1, Destination IP = 10.0.125.98
- Request: Source IP = 0.0.0.0, Destination IP = 255.255.255.255
- ACK: Source IP = 10.0.0.1, Destination IP = 10.0.125.98

Question 7

What is the IP address of your DHCP server?

Answer: DHCP address 10.0.0.1

Question 8

What IP address is the DHCP server offering to your host in the DHCP Offer message? Indicate which DHCP message contains the offered DHCP address.

Answer: The DHCP server offers an IP address in the DHCP Offer message. Your (Client) IP Address indicates

Question 9

In the example screenshot in this assignment, there is no relay agent between the host and the DHCP server. What values in the trace indicate the absence of a relay agent? Is there a relay agent in your experiment? If so what is the IP address of the agent?

Answer: The IP address being 0.0.0.0 indicates the absence of a relay agent. There is no relay agent in my experiment.

```
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 0
Transaction ID: 0xba9836e8
Seconds elapsed: 0
> Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0
Your (client) IP address: 10.0.125.98
Next server IP address: 10.0.0.1
Relay agent IP address: 0.0.0.0
Client MAC address: CloudNetwork_44:9a:7f (38:d5:7a:44:9a:7f)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
```

Question 10

Explain the purpose of the router and subnet mask lines in the DHCP offer message.

Answer: The IP address for the router identifies the default internet gateway. The subnet mask defines the subnet that is available.

Question 11

In the DHCP trace file noted in footnote 2, the DHCP server offers a specific IP address to the client (see also question 8 above). In the client's response to the first server OFFER message, does the client accept this IP address? Where in the client's RESPONSE is the client's requested address?

Answer: Yes. Option: (t=50,l=4) Requested IP Address = 192.168.1.101

Question 12

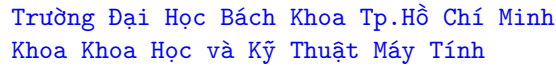
Explain the purpose of the router and subnet mask lines in the DHCP offer message.

Answer: The lease time is the amount of time the user is allowed to connect to the router. Option: (t=51,l=4) IP Address Lease Time = 4 hours, 10 minutes.

```
Relay agent IP address: 0.0.0.0
Client MAC address: CloudNetwork_44:9a:7f (38:d5:7a:44:9a:7f)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
> Option: (53) DHCP Message Type (Offer)
> Option: (1) Subnet Mask (255.255.0.0)
> Option: (3) Router
> Option: (6) Domain Name Server
> Option: (51) IP Address Lease Time
  Length: 4
  IP Address Lease Time: 4 hours, 10 minutes (15000)
> Option: (54) DHCP Server Identifier (10.0.0.1)
> Option: (255) End
```

Question 13

What is the purpose of the DHCP release message? Does the DHCP server issue an acknowledgment of receipt of the client's DHCP request? What would happen if the client's DHCP release



Answer: The DHCP release message tells the DHCP server that you want to cancel the IP address offered. The DHCP server will not issue an acknowledgment of receipt of the client's DHCP request. If the release message is lost, then the DHCP server retains the IP address until the lease time expires.

Answer: Yes, there were ARP packets sent and received to map the MAC address with the IP address.

Answer: 192.168.1.100

Question 3

Consider the HTTP GET sent from the client to the Google server at time 7.109267. What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET?

Answer: Source IP address: 192.168.1.100, Destination IP address: 64.233.169.104, TCP Source Port: 4335, TCP Destination Port: 80

```
No.    Time           Source            Destination      Protocol Length Info
 56    7.109267      192.168.1.100    64.233.169.104  HTTP        689    GET / HTTP/1.1
Frame 56: 689 bytes on wire (5512 bits), 689 bytes captured (5512 bits) on interface 0
Ethernet II, Src: HonHaiPrecis_0d:ca:8f (00:22:68:0d:ca:8f), Dst: CiscoLinksys_45:1f:1b (00:22:6b:45:1f:1b)
Internet Protocol Version 4, Src: 192.168.1.100, Dst: 64.233.169.104
 0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 675
 Identification: 0xa2ac (41644)
 0100 .... = Flags: 0x2, Don't fragment
 ...0 0000 0000 0000 = Fragment Offset: 0
 Time to Live: 128
 Protocol: TCP (6)
 Header Checksum: 0xa94a [validation disabled]
 [Header checksum status: Unverified]
 Source Address: 192.168.1.100
 Destination Address: 64.233.169.104
 [Stream index: 5]
Transmission Control Protocol, Src Port: 4335, Dst Port: 80, Seq: 1, Ack: 1, Len: 635
Hypertext Transfer Protocol
```

Question 4

At what time is the corresponding 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message?

Answer: Time: 7.158797, Source IP address: 64.233.169.104, Destination IP address: 192.168.1.100, TCP Source Port: 80, TCP Destination Port: 4335

```
60    7.158797      64.233.169.104    192.168.1.100  HTTP        814    HTTP/1.1 200 OK (text/html)
Frame 60: 814 bytes on wire (6512 bits), 814 bytes captured (6512 bits) on interface 0
Ethernet II, Src: CiscoLinksys_45:1f:1b (00:22:6b:45:1f:1b), Dst: HonHaiPrecis_0d:ca:8f (00:22:68:0d:ca:8f)
Internet Protocol Version 4, Src: 64.233.169.104, Dst: 192.168.1.100
 0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x20 (DSCP: CS1, ECN: Not-ECT)
 Total Length: 800
 Identification: 0xf61e (63006)
 0000 .... = Flags: 0x0
 ...0 0000 0000 0000 = Fragment Offset: 0
 Time to Live: 50
 Protocol: TCP (6)
 Header Checksum: 0xe33b [validation disabled]
 [Header checksum status: Unverified]
 Source Address: 64.233.169.104
 Destination Address: 192.168.1.100
 [Stream index: 5]
Transmission Control Protocol, Src Port: 80, Dst Port: 4335, Seq: 2861, Ack: 636, Len: 760
[3 Reassembled TCP Segments (3620 bytes): #58(1430), #59(1430), #60(760)]
Hypertext Transfer Protocol
Line-based text data: text/html (12 lines)
```

Question 5

At what time is the client-to-server TCP SYN segment sent that sets up the connection used by the GET sent at time 7.109267? What are the source and destination IP addresses and source and destination ports for the TCP SYN segment? What are the source and destination IP addresses and source and destination ports of the ACK sent in response to the SYN? At what time is this ACK received at the client?

Answer: SYN: Time: 7.075657, Source IP address: 192.168.1.100, Destination IP address: 64.233.169.104, TCP Source Port: 4335, TCP Destination Port: 80.

```

53 7.075657 192.168.1.100 64.233.169.104 TCP 66 4335 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4
SACK_PERM
Frame 53: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
Ethernet II, Src: HonHaiPrecis_0d:ca:8f (00:22:68:0d:ca:8f), Dst: CiscoLinksys_45:1f:1b (00:22:6b:45:1f:1b)
Internet Protocol Version 4, Src: 192.168.1.100, Dst: 64.233.169.104
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 52
Identification: 0xa2aa (41642)
010. .... = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: TCP (6)
Header Checksum: 0xabbb [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.1.100
Destination Address: 64.233.169.104
[Stream index: 5]
Transmission Control Protocol, Src Port: 4335, Dst Port: 80, Seq: 0, Len: 0

```

ACK: Time: 7.108986, Source IP address: 64.233.169.104, Destination IP address: 192.168.1.100, TCP Source Port: 80, TCP Destination Port: 4335

```

54 7.108986 64.233.169.104 192.168.1.100 TCP 66 80 → 4335 [ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
SACK_PERM WS=64
Frame 54: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
Ethernet II, Src: CiscoLinksys_45:1f:1b (00:22:6b:45:1f:1b), Dst: HonHaiPrecis_0d:ca:8f (00:22:68:0d:ca:8f)
Internet Protocol Version 4, Src: 64.233.169.104, Dst: 192.168.1.100
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x20 (DSCP: CS1, ECN: Not-ECT)
Total Length: 52
Identification: 0xf61a (63002)
000. .... = Flags: 0x0
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 50
Protocol: TCP (6)
Header Checksum: 0xe62b [validation disabled]
[Header checksum status: Unverified]
Source Address: 64.233.169.104
Destination Address: 192.168.1.100
[Stream index: 5]
Transmission Control Protocol, Src Port: 80, Dst Port: 4335, Seq: 0, Ack: 1, Len: 0

```

Question 6

In the NAT_ISP_side trace file, find the HTTP GET message that was sent from the client to the Google server at time 7.109267. At what time does this message appear in the NAT_ISP_side trace file? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET? Which of these fields are the same, and which are different than your answer to question 3 above?

Answer:



Question 7

Are any fields in the HTTP GET message changed? Which of the following fields in the IP datagram carrying the HTTP GET are changed: Version, Header Length, Flags, Checksum? If any of these fields have changed, give a reason (in one sentence) stating why this field needed to change.

Answer:

Question 8

In the NAT_ISP_side trace file, at what time is the first 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message? Which of these fields are the same, and which are different than your answer to question 4 above?

Answer:

Question 9

In the NAT_ISP_side trace file, at what time were the client-to-server TCP SYN segment and the server-to-client TCP ACK segment corresponding to the segments in question 5 above captured? What are the source and destination IP addresses and source and destination ports for these two segments? Which of these fields are the same, and which are different than your answer to question 5 above?

Answer:

Question 10

Using your answers to 1-8 above, fill in the NAT translation table entries for the HTTP connection considered in questions 1-8 above.

Answer: