Name: ASHUTOSH SONI

HW-1

**COMPUTER NETWORKS**

Id: 2018ucp1505

Q1: Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

Ans: HTTP 1.1

Q2: What languages (if any) does your browser indicate that it can accept to the server?

Ans: US English (en-US)

Q4: What is the status code returned from the server to your browser?

Ans: 200 OK

Q5: When was the HTML file that you are retrieving last modified at the server?

Ans: We can filter messages by http.last\_modified and we see that the HTTP response. On-going inside I check in http://mnit.ac.in and it shows last modified at FRI 17-March 2017 14:04:53.

Q6: How many bytes of content are being returned to your browser?

Ans: On going to <http://mnit.ac.in/Images/new.gif> website the content return is 30080 bytes.

Q7: By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so name one

Ans: No. The raw data appears to match up exactly with what is shown in the packet-listing window.

Q8: Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?

Ans: NO.

Q9: Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

Ans: We can see after applying filter on html we can see HTTP/1.1 200 OK response beside that we can see the type of response it may be text/html of plain or CSS. From there we can tell the response of server.

Q10: Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?

Ans: If-Modified-Since: Wed, 12 Mar 2014 17:29:08 GMT\r\n.

Q12: How many HTTP GET request messages were sent by your browser?

Ans: One

Q13: How many data-containing TCP segments were needed to carry the single HTTP response?

Ans: I got 29 reassembled TCP Segments.

Q14: What is the status code and phrase associated with the response to the HTTP GET request?

Ans: HTTP/1.1 200 OK

Q15: Are there any HTTP status lines in the transmitted data associated with a TCP-induced “Continuation”?

Ans: No.

Q16: How many HTTP GET request messages were sent by your browser? To which Internet addresses were these GET requests sent?  
answer 4 HTTP GET request messages.

Ans: Three: 223.196.92.54 , 192.168.43.64 , 223.196.92.57

Q17: Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

Ans: Images are downloaded serially as TCP is connection oriented reliable protocol and in Wireshark, we can see this by seeing the timestamp of the images.

Q18: What is the server’s response (status code and phrase) in response to the initial HTTP GET message from your browser?

Ans: HTTP/1.1

Q19: When your browser’s sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

Ans: HTTP/1.1 200 OK

Q20: In the packet that contains the http GET message, what is the source mac address? Is this your computer's mac address?

Ans: The Source Mac Address is: 14:4f:8a:77: b7:6c.

It is my computer’s mac address.

Q21: What is the destination mac address of the above packet, is this the mac address of mnit.ac.in? If not, then which device has this mac address?

Ans: The Destination mac address is: dc:1a:c5:9b: 12:7f.

This is not the Ethernet address of mnit.ac.in. It is the mac address of my router or internet gateway address.

Q22: What is the hexadecimal frame type field in the ethernet header of this packet? What is the correspond upper layer protocol?

Ans: The hexadecimal frame type field in the ethernet header of this packet is 0x0800.

It indicates that the upper layer protocol is Internet Protocol version 4 (IPv4).

Q23: Do you notice that Wireshark can display the manufacturer of the sender (source) and receiver (destination) of this packet? How this can be done?

Ans: The mac address contains an OUI (Organizationally Unique Identifier) field. It is a 24 - bit number that uniquely identifies a vendor or manufacturer. They are purchased and assigned by the IEEE. The OUI is basically the first three octets of a MAC address.

Wireshark tries to convert the first 3 bytes of an ethernet address to an abbreviated manufacturer name by looking up OUI database.

Q24: How many bytes from the very start of the Ethernet frame does the ASCII 'G' in GET appear in the Ethernet frame? Explain how do you obtain this result.

Ans: After 528 bits or 66 bytes the G in get appears.

Or before ‘G’, we have 14 element header + 20 IP header + 32 tcp header = 66 bytes.