Title: Parsing IP packets

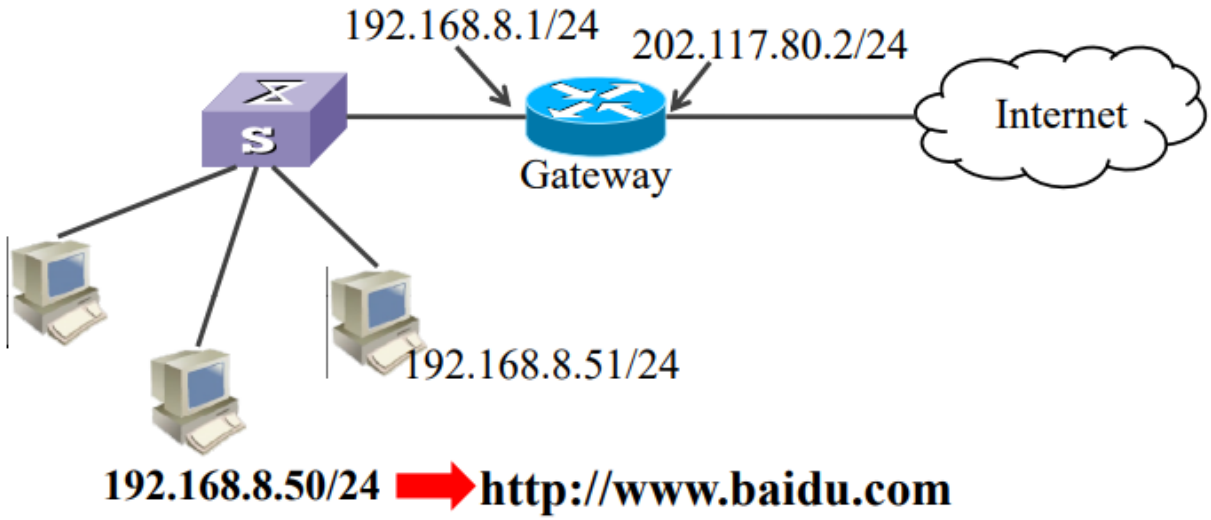
Name: Dikshya Kafle
Stu No: 2018380039
Class No: 10101801
Deadline: Dec 13th 2020

School of Computer Science and Engineering

Purpose:

By analyzing IP packets, we can understand the working principle of IP, ARP, TCP, HTTP and DNS protocols

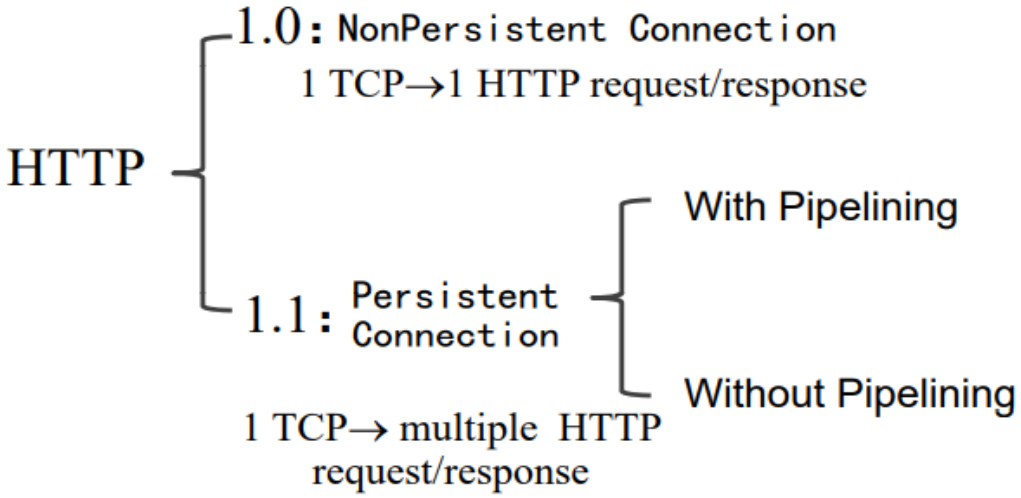
Network Topology For Experiment:

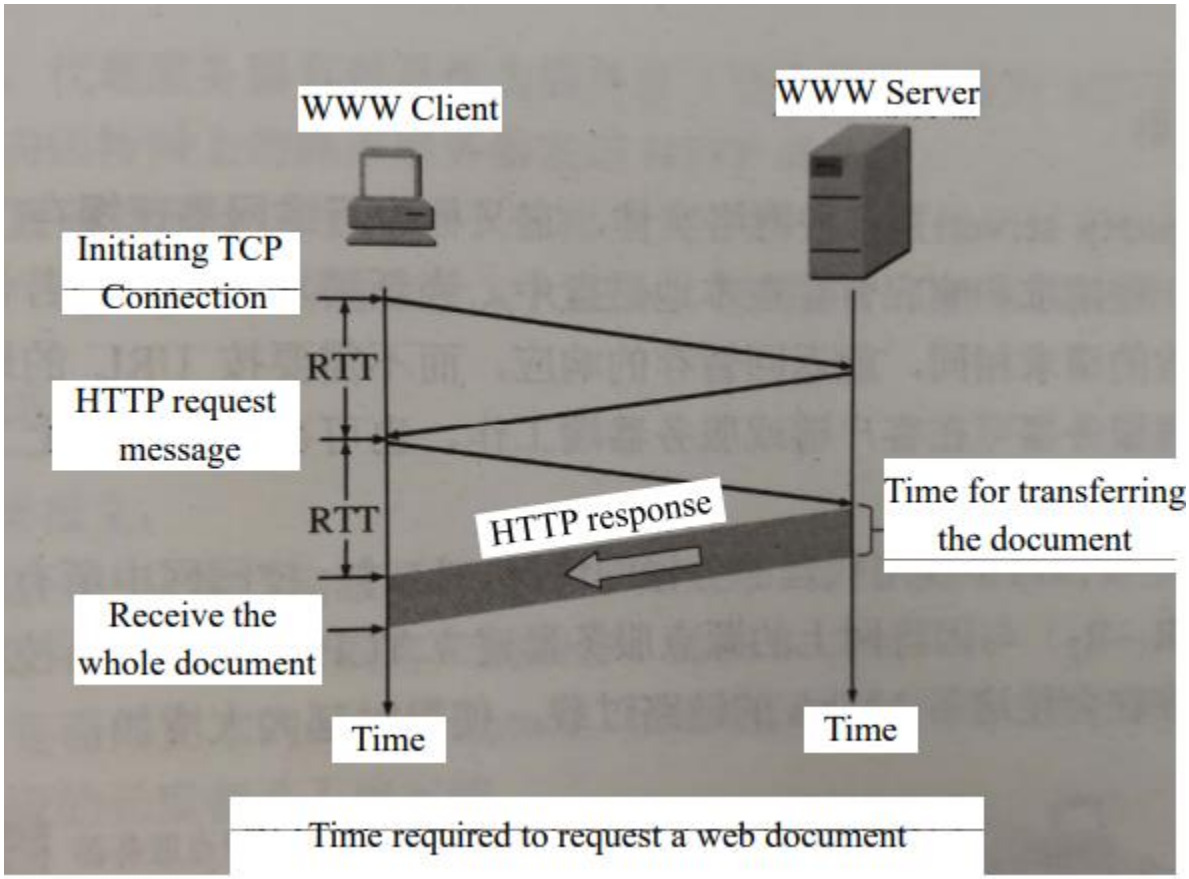


Process of accessing webpage:

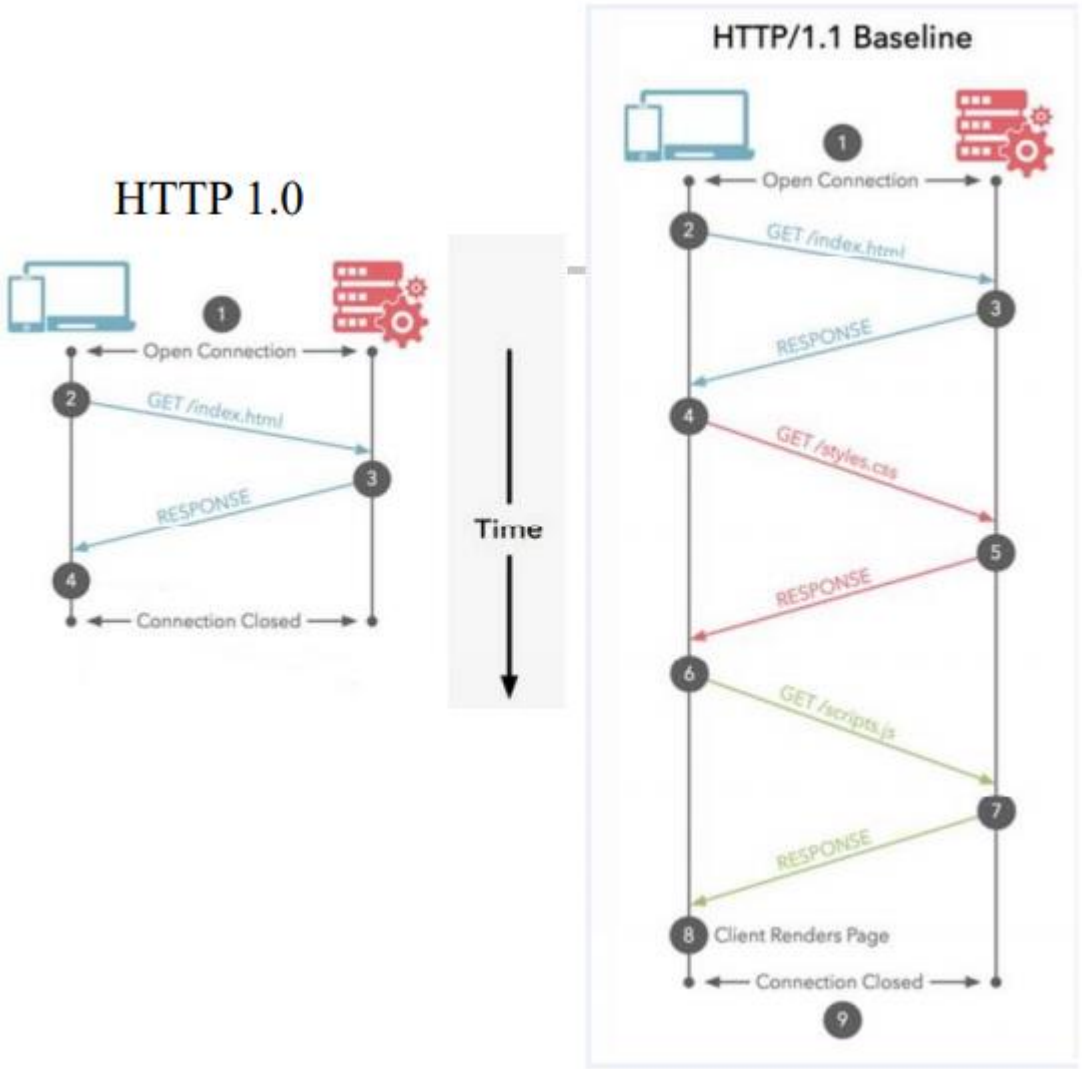
1. Access http://www.baidu.com ν
2. Step 1. DNS UDP IP ARP recursion/iteration ICMP
3. Step 2. Establish TCP connection by three-way handshake.
4. Step 3. Client sends HTTP request.
5. Step 4. Server receive and return HTTP response.
6. Step 5. Release TCP connection by four-way wave hand.

HTTP Working Mode:

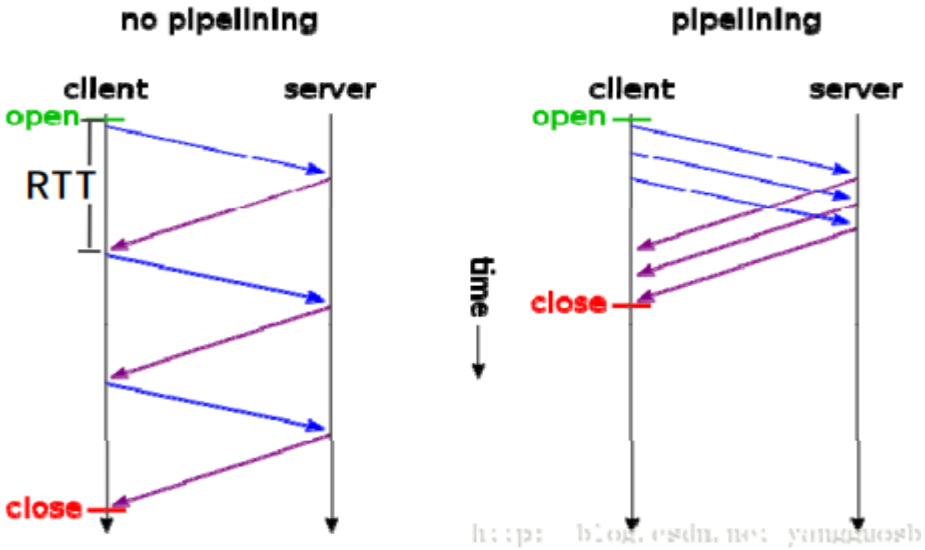




Pipeline Connection:



Pipeline Connection:



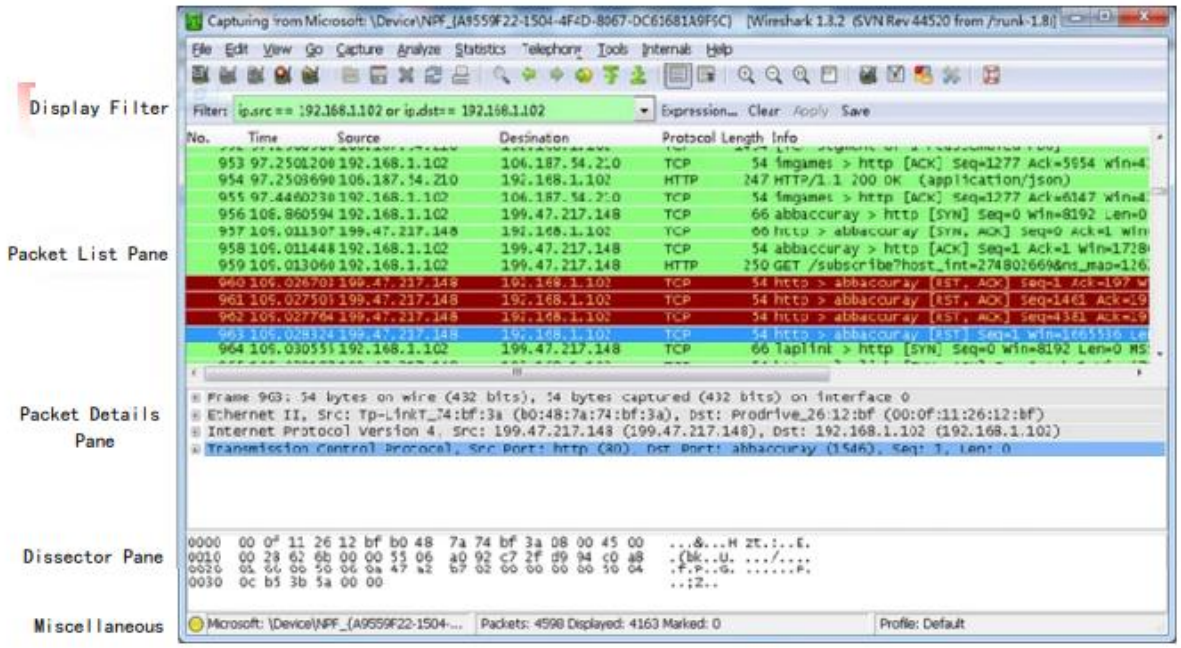
Data encapsulation:

DNS request	Frame header	IP Header	UDP Header	DNS Header	DATA	Frame Tail
HTTP request	Frame header	IP Header	TCP Header	HTTP Header	Entity	Frame Tail

ARP message format:

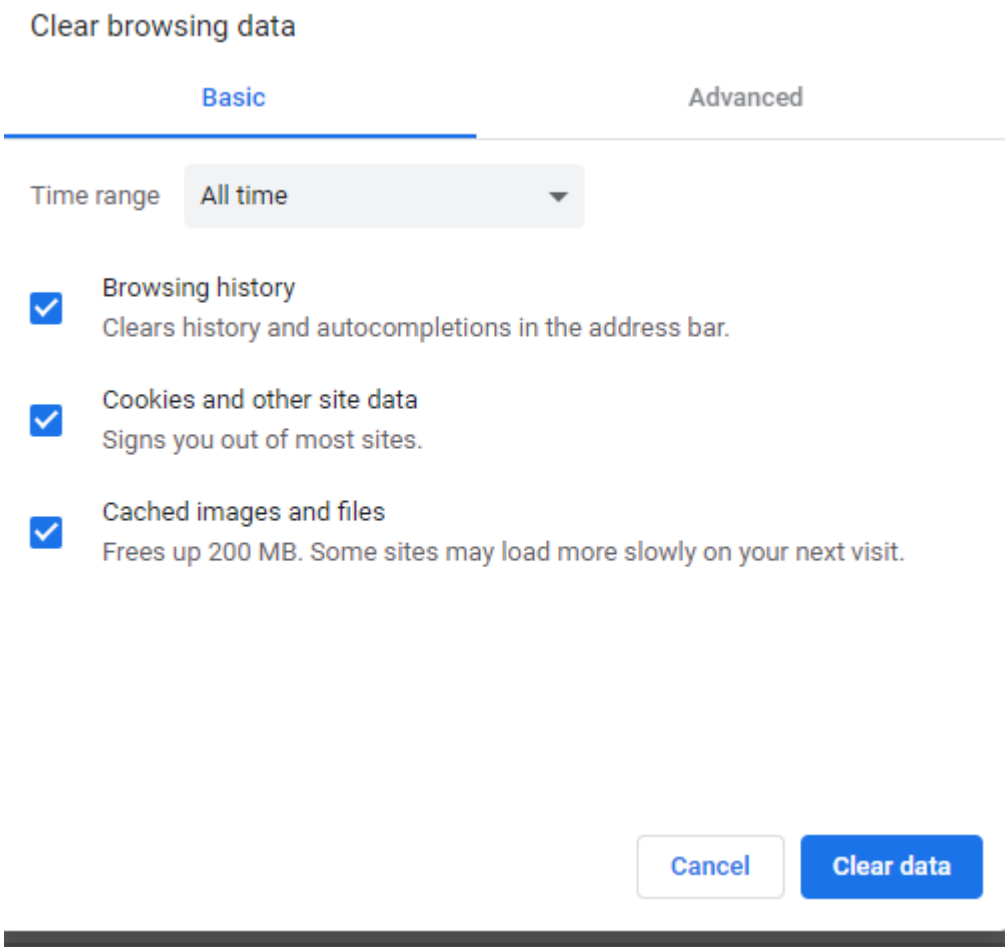
Byte		
2	Network type	"1" denote Ether
2	Protocol type	"0x0800" is IP
1	Length for PA	PA: Physical address
1	Length for IP	
2	Operation	1—ARP request 2—ARP response 3—RARP request 4—RARP response
6	Source PA	
4	Source IP	
6	Destination PA	Destination PA is empty in the request message
4	Destination IP	

9 2020/11/24

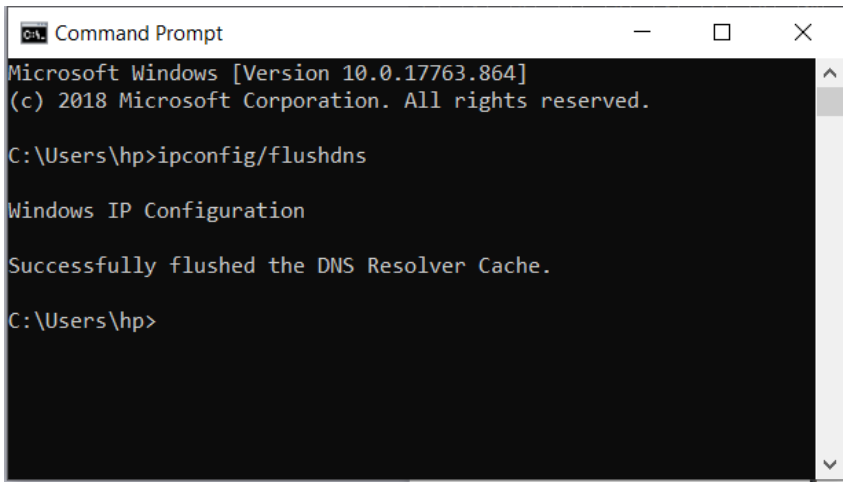


1- Preliminary:

(1) Clear browser cache Ensure that the Web is caught from network. Chrome: Options --> Under the Hood --> Clear browsing data。



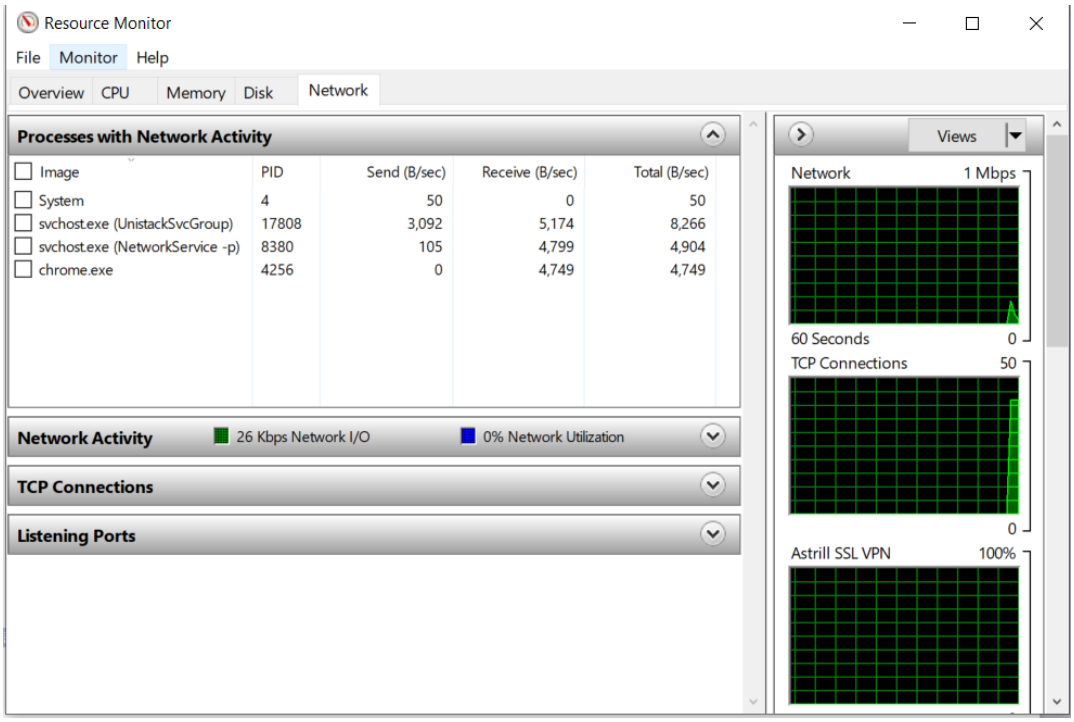
(2) Clear DNS cache Ensure that that the map of domain name and ip is got from network request. In Windows XP， input ipconfig /flushdns.



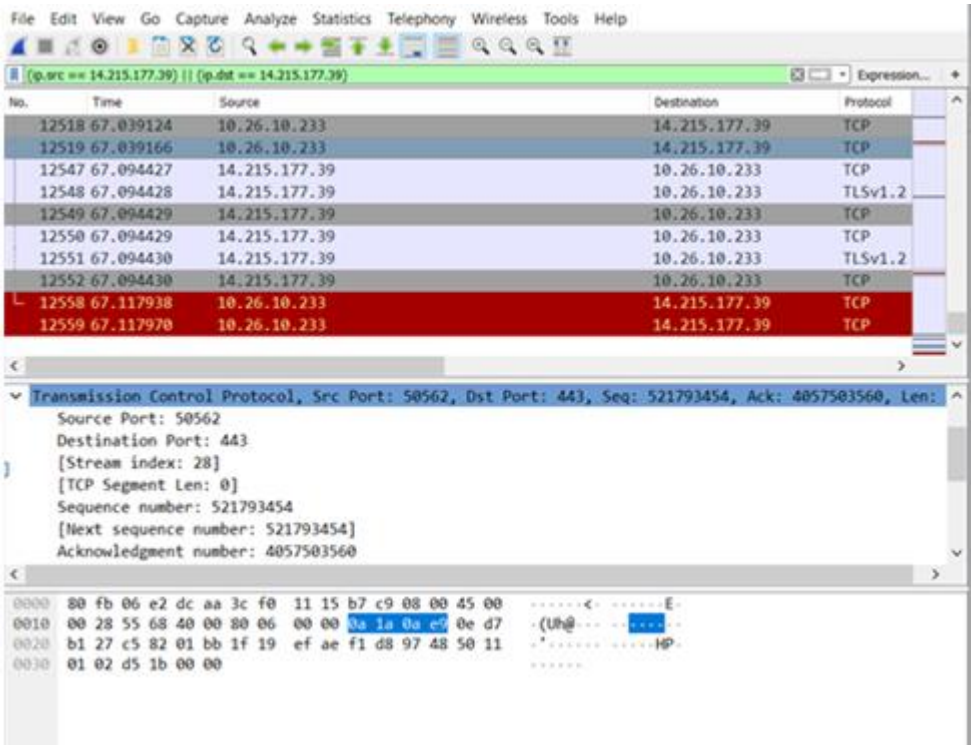
(3) Set filter rules In order to facilitate the analysis， set filter rules before catching the packets. In Filter ToolBar，Enter filter rule normal expression.

(ip.src == 14.215.177.39) || (ip.dst == 14.215.177.39)

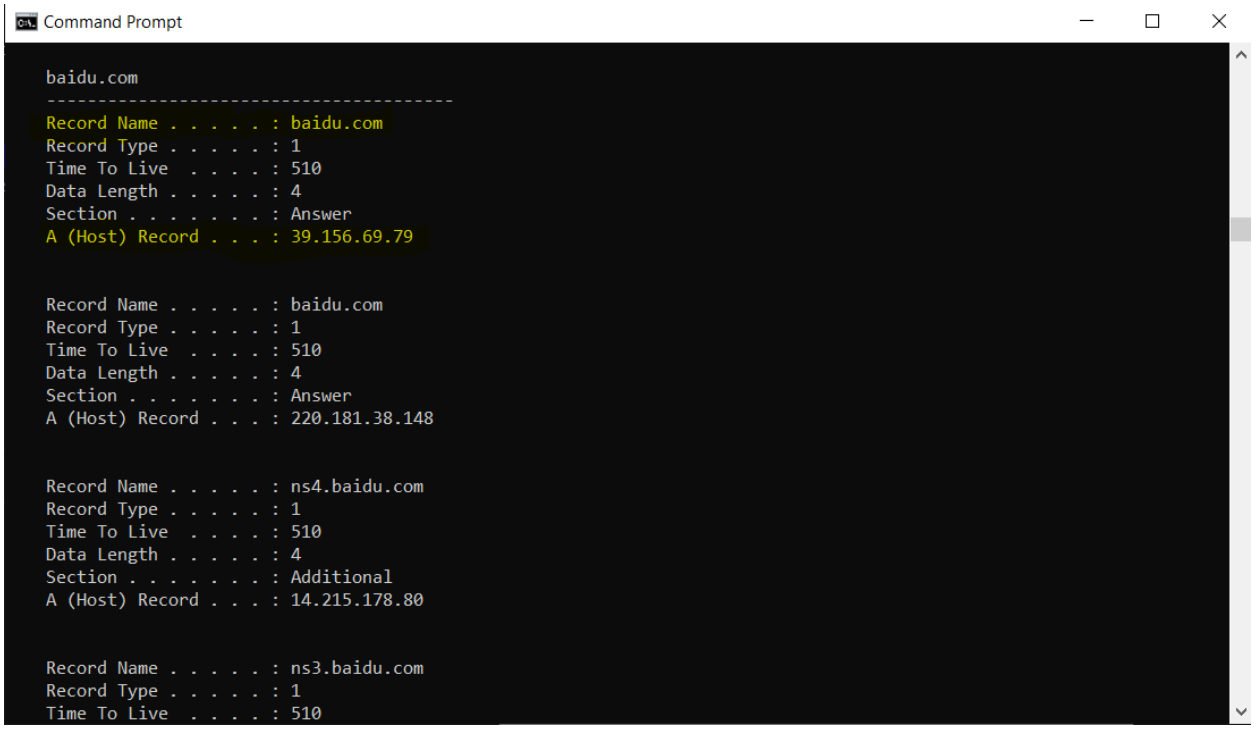
(4) Close network applications



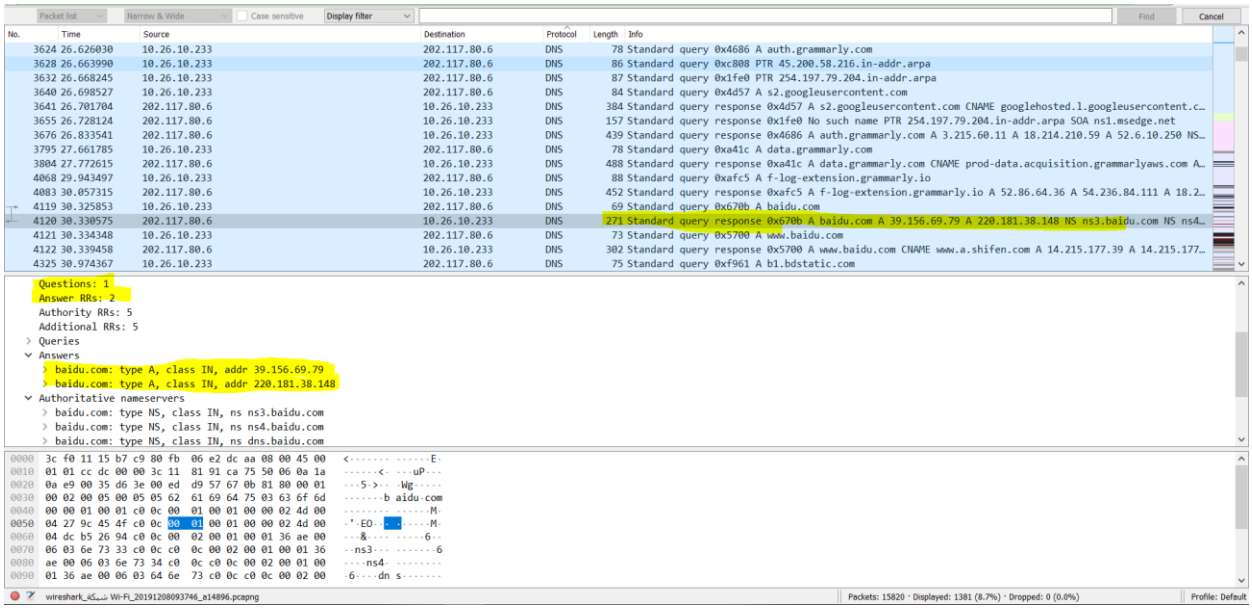
2- Start wireshark and Input url

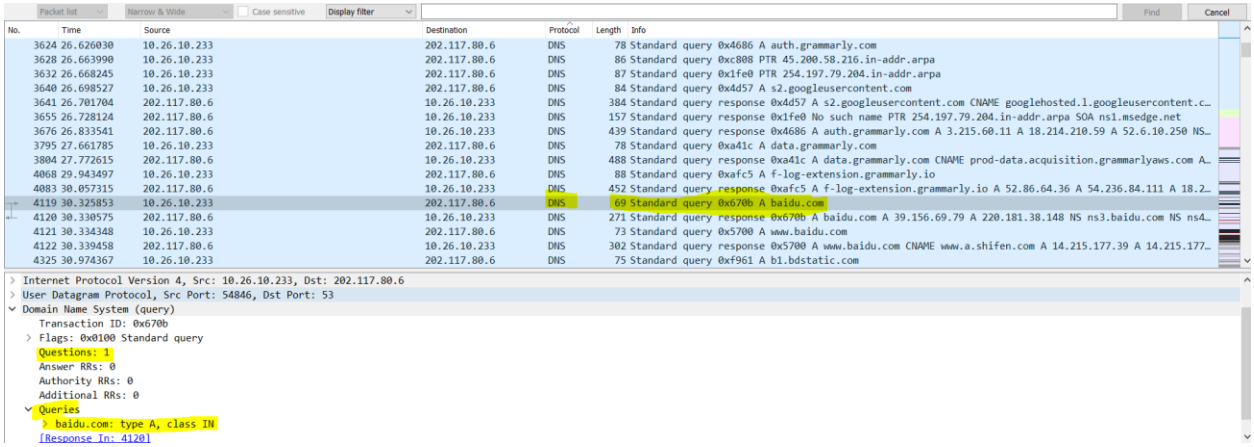


3- Ip address of baidu.com

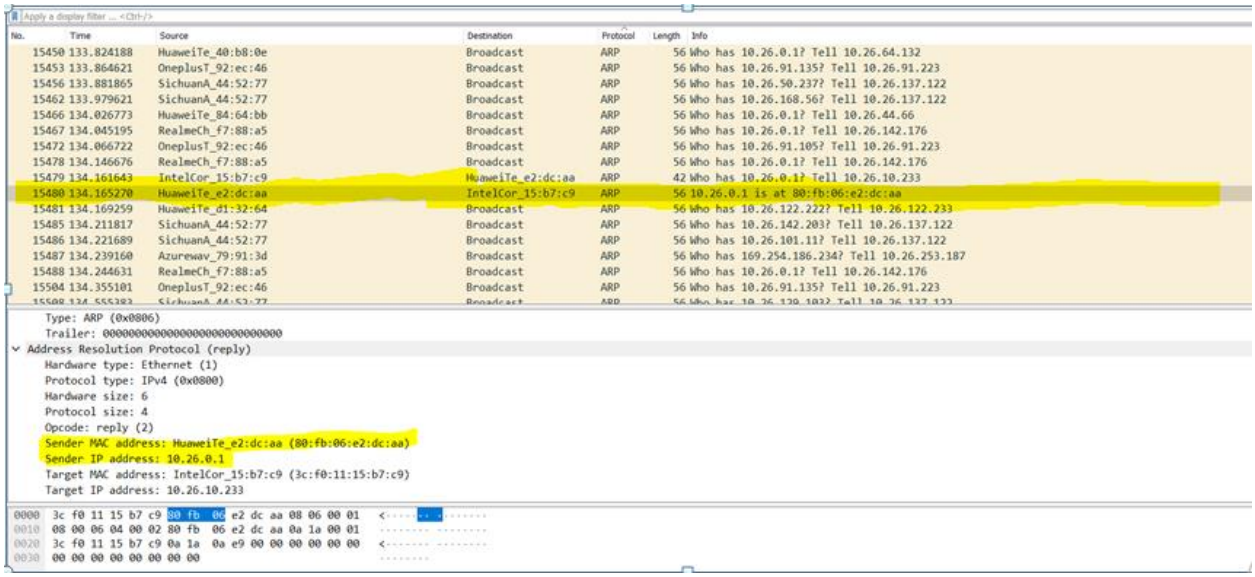


Or we can see DNS query and response in wireshark:

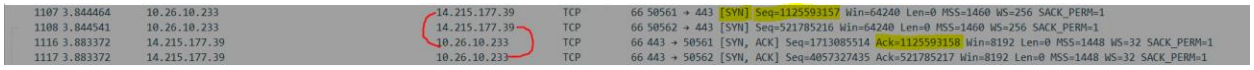


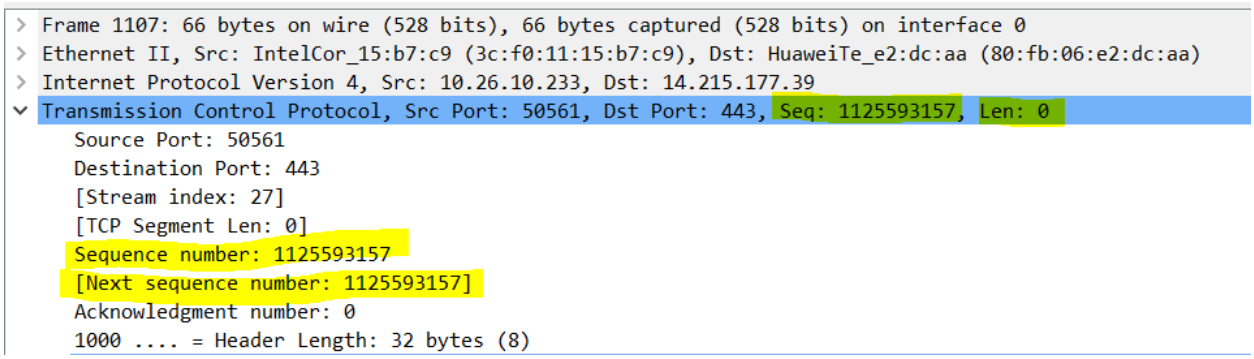


4- IP and MAC for gateway



5- ISN of data? Server ISN

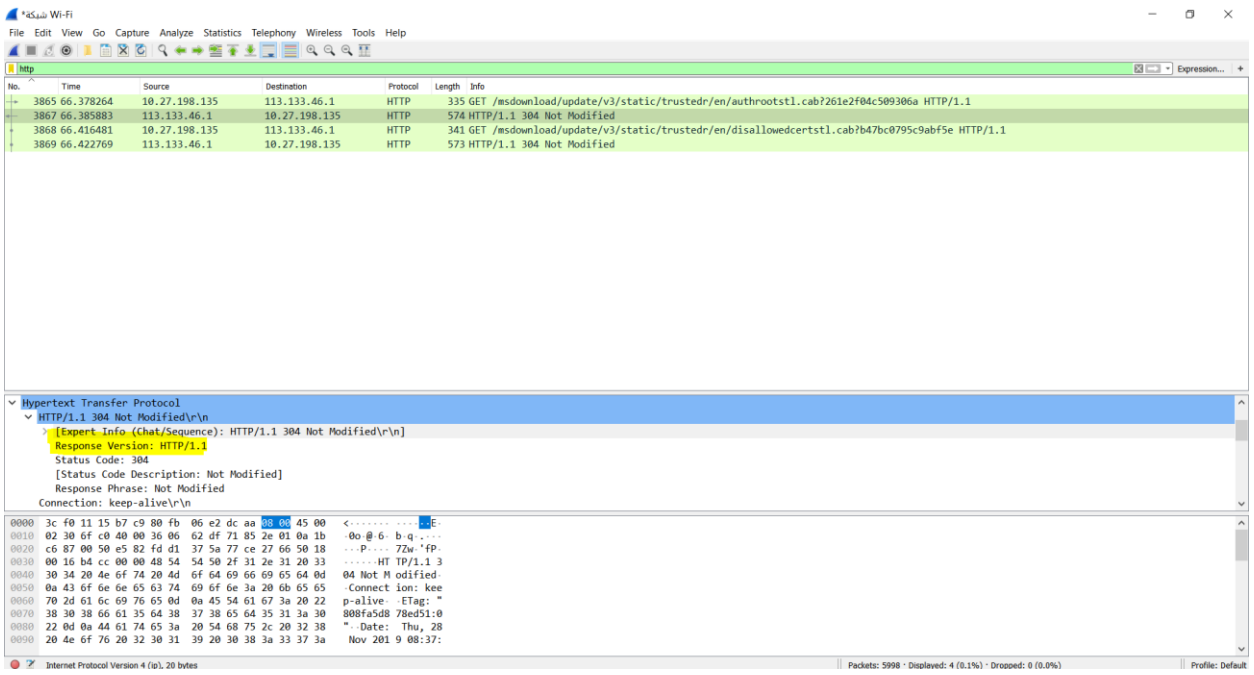




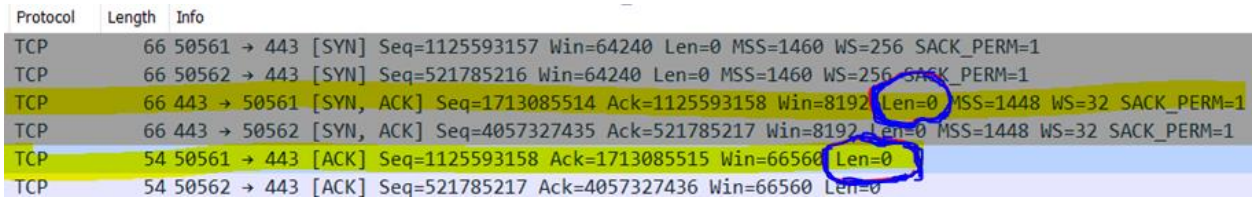
The client ISN is: 1125593157
The server ISN is: 1713085514

Note that the browser will set multiple TCP connection at the same time with the server to make a parallel retrieve of information as only one document can be asked per TCP connection, that;s why we find multiple SYN requests.

6- HTTP version， working mode

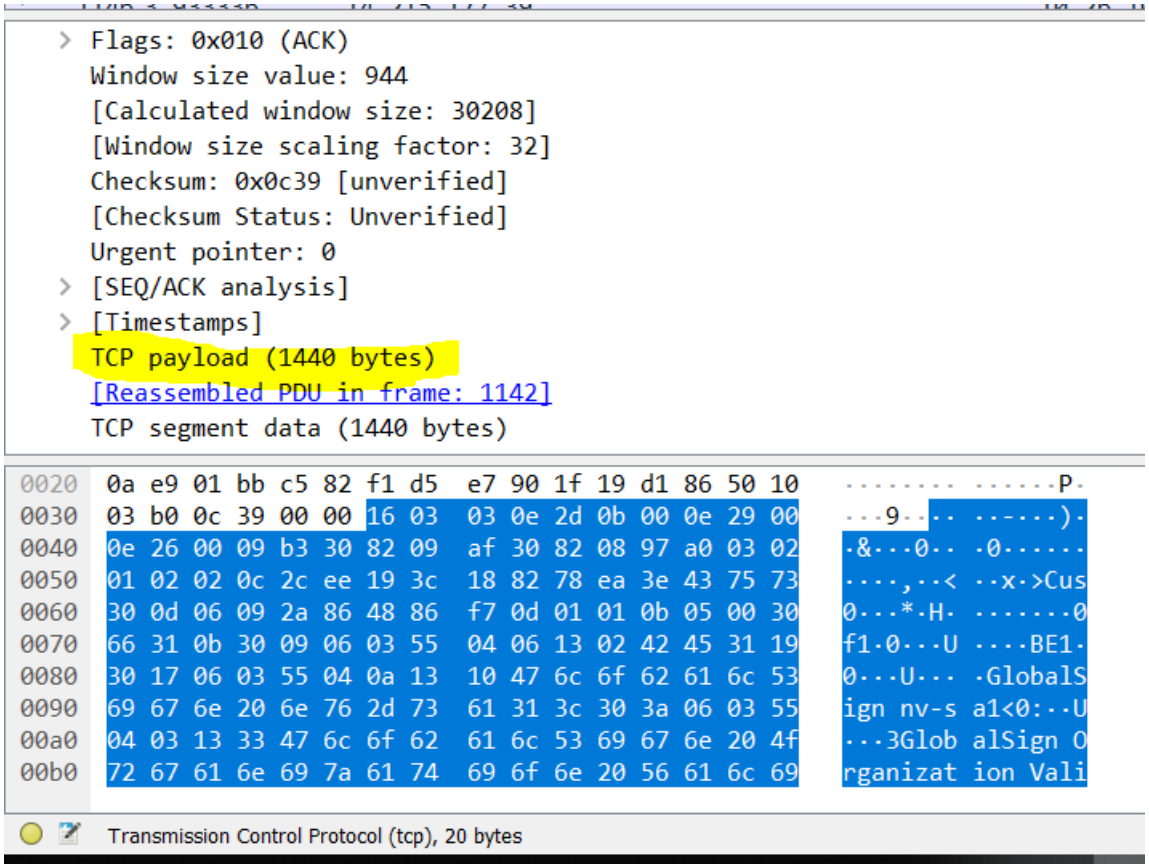


7- One TCP connection, Amount of data sent? Amount of data received



We can see in these two TCP packets, the length of data is clearly 0 as the client is only trying to establish the connection so it sends no data and the server only acknowledges the request and sends no data two. That why len =0.

However we can see another TCP packet where the payload length is 1440 bytes:



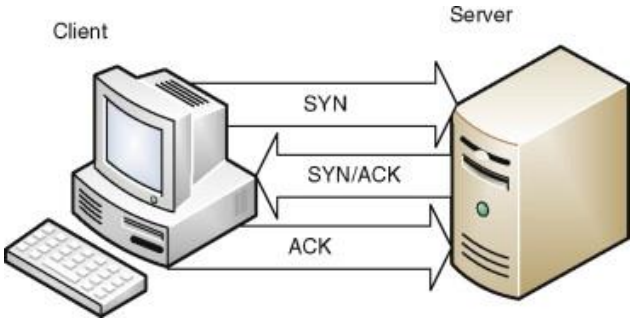
8- <u>The process of three-way handshake connection and four-way wavehand release.</u>

<u>Three-way handshake:</u>

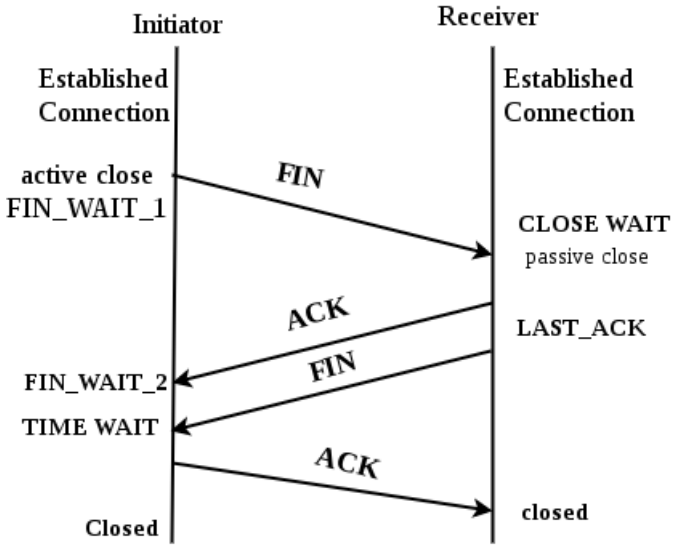
Step 1 (SYN) : In the first step, the client wants to create a server connection, so it sends a SYN (Synchronize Sequence Number) segment that tells the server that the client is likely to initiate contact and with which sequence number it begins segments with

Step 2 (SYN + ACK): The server responds with SYN-ACK signal bits set to a client request. Acknowledgement (ACK) refers to the reaction of the segment it obtained and SYN indicates with what sequence number the segments are likely to start with

Step 3 (ACK): The client acknowledges the server's response in the final part and both create a secure link with which they start the real data transfer.



Four-way wavehand release.



Step 1 (FIN From Client) – Suppose that the client application decides it wants to close the connection. (Note that the server could also choose to close the connection). This causes the client send a TCP segment with the **FIN** bit set to **1** to server and waits for a TCP segment from the server with an acknowledgment (ACK).

Step 2 (ACK From Server) – When Server received FIN bit segment from Sender (Client), Server Immediately send acknowledgement (ACK) segment to the Sender (Client).

Step 3 (Client waiting) – The client then waits for a TCP segment from the server with an acknowledgment. When it receives this segment, the client waits for another segment from the server with the FIN bit set to 1.

Step 4 (FIN from Server) – Server sends FIN bit segment to the Sender(Client) after some time when Server send the ACK segment.

Step 5 (ACK from Client) – When Client receive FIN bit segment from the Server, the client acknowledges the server's segment.

Summary:

We have finished running the software and produced a program that produces the values of the IP packet field and the data together. We will now explain the details of each byte that we use in the TCP header and UDP header packets. We have also examined IP packets and understood the working principle of the IP protocol.