CA169 Networks Assignment Two

Answer Sheets

STUDENT NAME:	Tomas Baltrunas
STUDENT NUMBER:	17350793
PROJECT NUMBER:	2
MODULE CODE:	CA169
DEGREE: {CA|EC|CPSSD|ECSA]	CA
LECTURER:	Brian Stone

Declaration

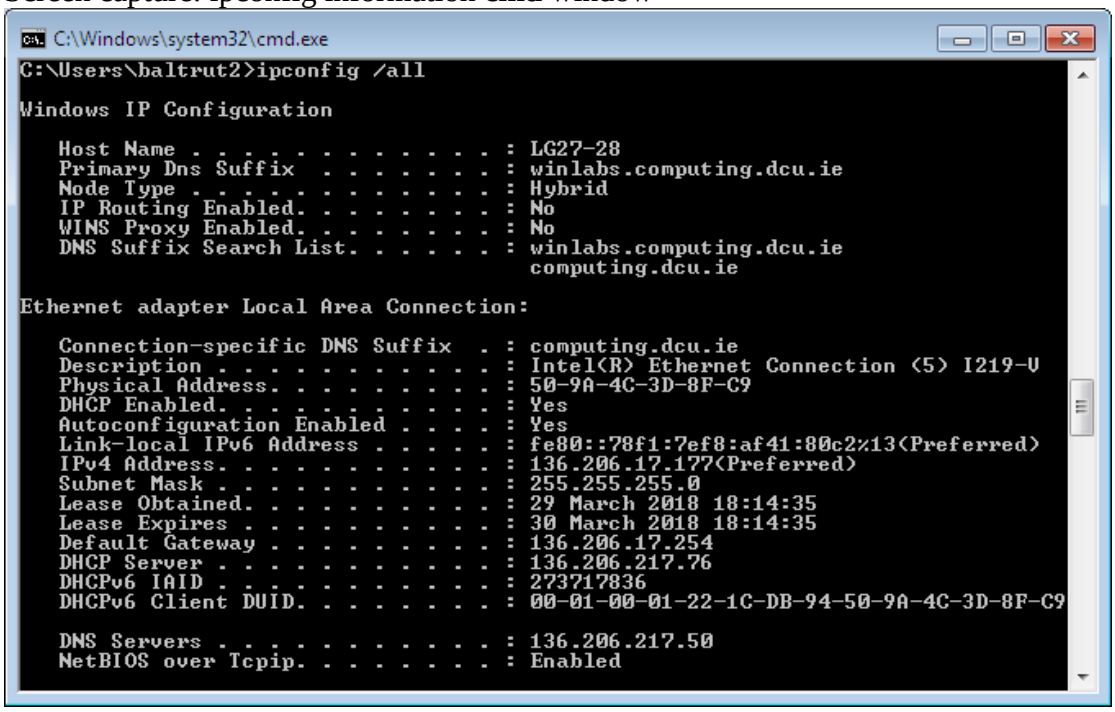
In submitting this project, I declare that the project material, which I now submit, is my own work. Any assistance received by way of borrowing from the work of others has been cited and acknowledged within the work. I make this declaration in the knowledge that a breach of the rules pertaining to project submission may carry serious consequences.

Part 1: DHCP traffic

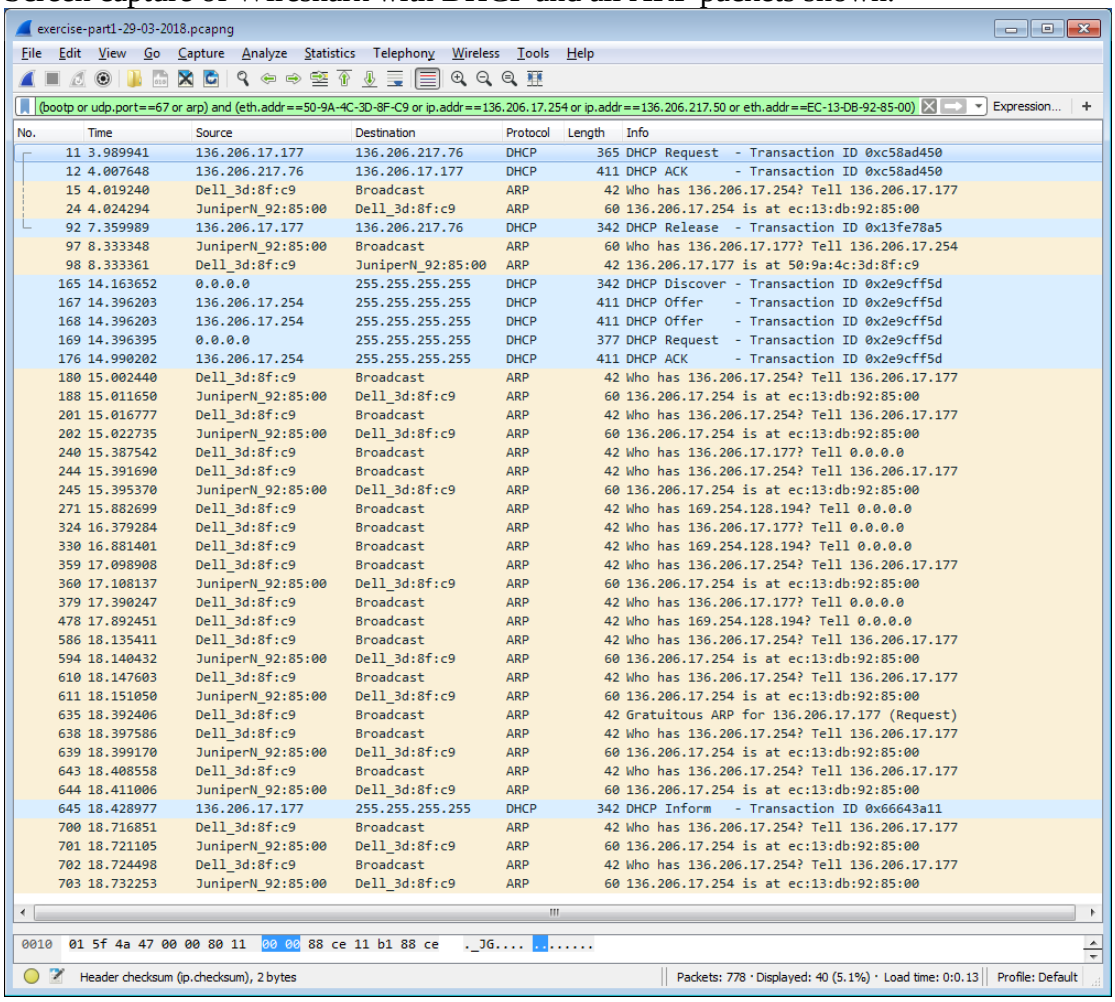
Your IP & MAC address for this experiment (use ipconfig)

136.206.17.177	50-9A-4C-3D-8F-C9

Screen capture: ipconfig information **cmd** window



Screen capture of Wireshark with DHCP and all ARP packets shown.



Packet numbers relevant to the DHCP interaction:

a. DHCP DISCOVER: **165**
b. DHCP OFFER: **167, 168**
c. DHCP Request: **169**
d. DHCP Acknowledgement: **176**
e. DHCP Release (if you release using `ipconfig /release`): **92**
f. All ARP packets used: **15, 24, 97, 98, 180, 188, 201, 202 … 702, 703**

Function of each packet

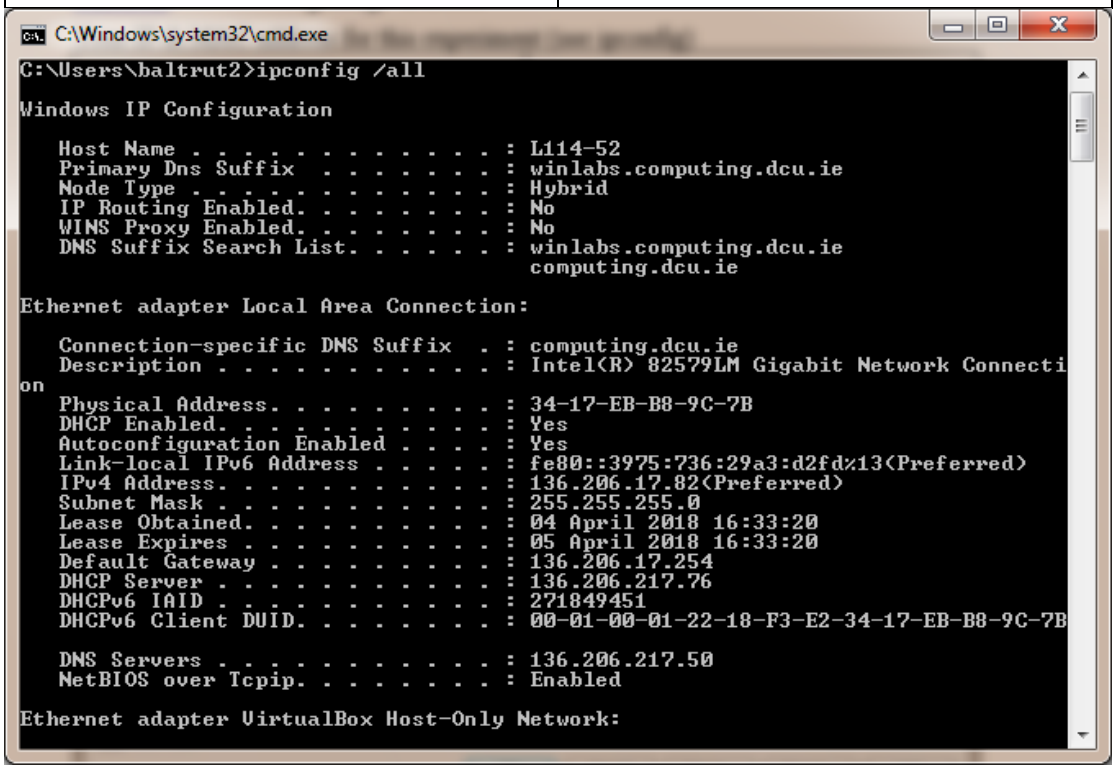
a. DHCP DISCOVER: **For the client to find a nearby DHCP server and request an IP address lease, by broadcasting to the entire network, with its IP address set to all 0's (no IP)**
b. DHCP OFFER: **To reserve and offer an IP lease and send additional configuration information to a client that sent a DISCOVER.**
c. DHCP Request: **For the client to accept and request the offered IP (also sent when renewing a lease)**
d. DHCP Acknowledgement: **For the server to confirm the client's new IP lease and to send additional configuration details (also a confirmation of lease requests)**

e. DHCP Release (if you release using `ipconfig /release`): **For the client to announce to the DHCP server that it gives up (releases) its IP lease**
f. ARP: **Lets computers find each other's MAC addresses from knowing their IP addresses. With this, the client can connect to the DHCP server. The server can check that an IP is not in use. And the client can test that it got the IP assigned correctly.**

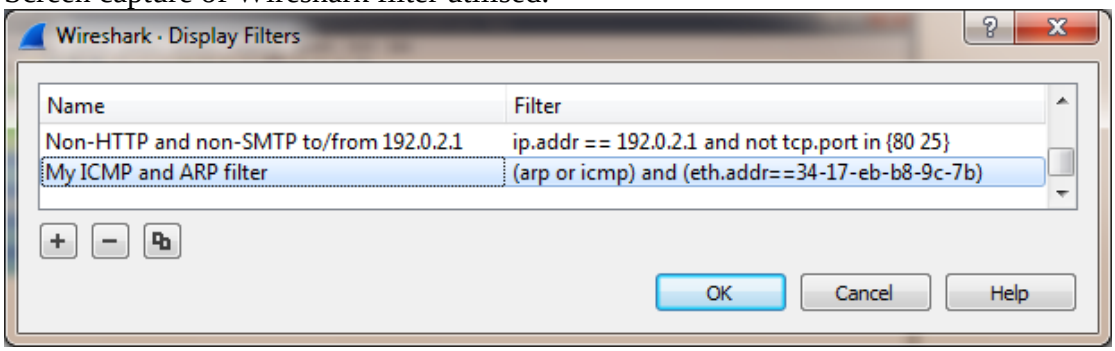
Part 2: `ping` traffic

Your IP & MAC address for this experiment (use ipconfig)

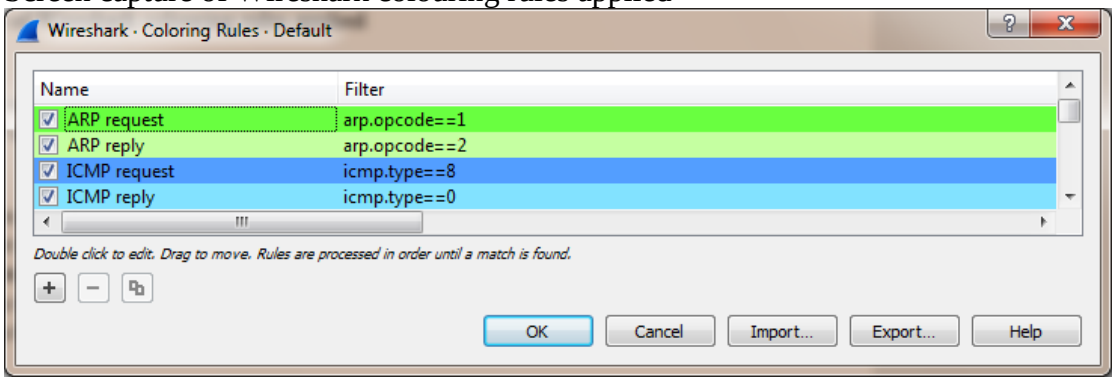
136.206.17.82	34-17-EB-B8-9C-7B



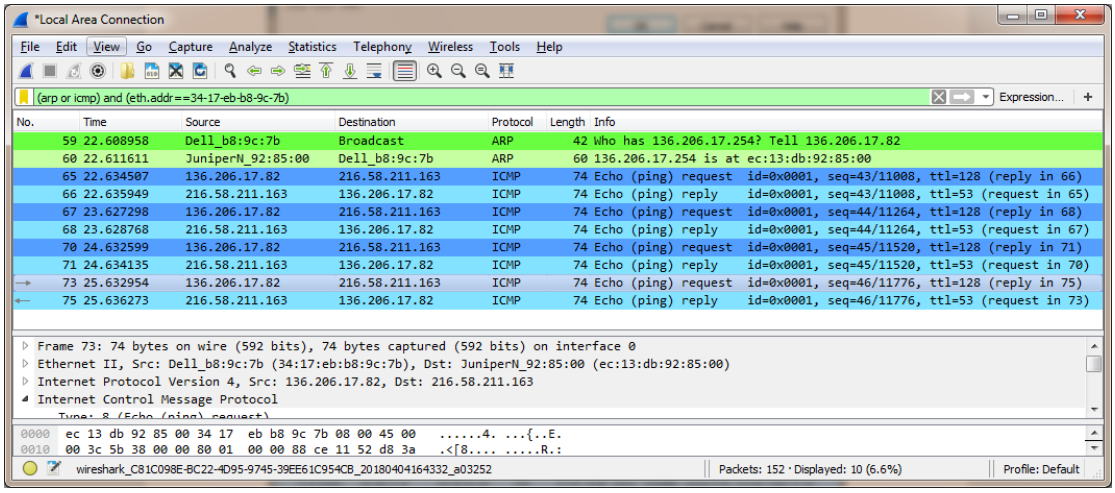
Screen capture of Wireshark filter utilised.



Screen capture of Wireshark colouring rules applied



Screen capture of Wireshark packet trace showing all relevant ping generated traffic, including ARP and ICMP traffic.



Packet numbers relevant to the experiment: **59, 60, 65, 66, 67, 68, 70, 71, 73, 75**
Explanation for each packet

- Function
- Explain why it is generated
- Explain the data contained in the packet

59: ARP request broadcast

- Asks for the MAC address of a computer, knowing its IP address.
- Here generated because my computer needs to find the MAC of (empty ARP cache) and send data to the gateway (to get the gateway to ask via proxy ARP for the MAC of the pinged computer on another network)
- Major data includes (besides the encapsulating Ethernet frame) the protocol type used (IPv4), corresponding protocol size (address length – 4 for IPv4), hardware size (hardware address length – 6 for Ethernet), opcode set to 1 (request packet), IP and MAC of sender/receiver (receiver's MAC set to all 0's to indicate a broadcast/unknown MAC)

60: ARP response

- The computer with the requested IP replies to the requesting computer with its MAC
- Generated in response to an ARP request
- Contents same as in ARP request packet, except that hardware type is explicitly stated (Ethernet), the opcode is set to 2 ('reply'), and now the sender includes its MAC.

65, 67, 70, 73: ICMP echo ping request.

- Functions as an internal checker of a computer's availability.
- Generated by the usage of the 'ping' command.
- Significant data (in the payload of the Ethernet frame and IP packet) includes the control message 'echo request' (from 'type' and 'code' fields), an error-detecting checksum, an echo request identifier and sequence number (request's number, incremented each time one is sent) (big endian and little endian formats)

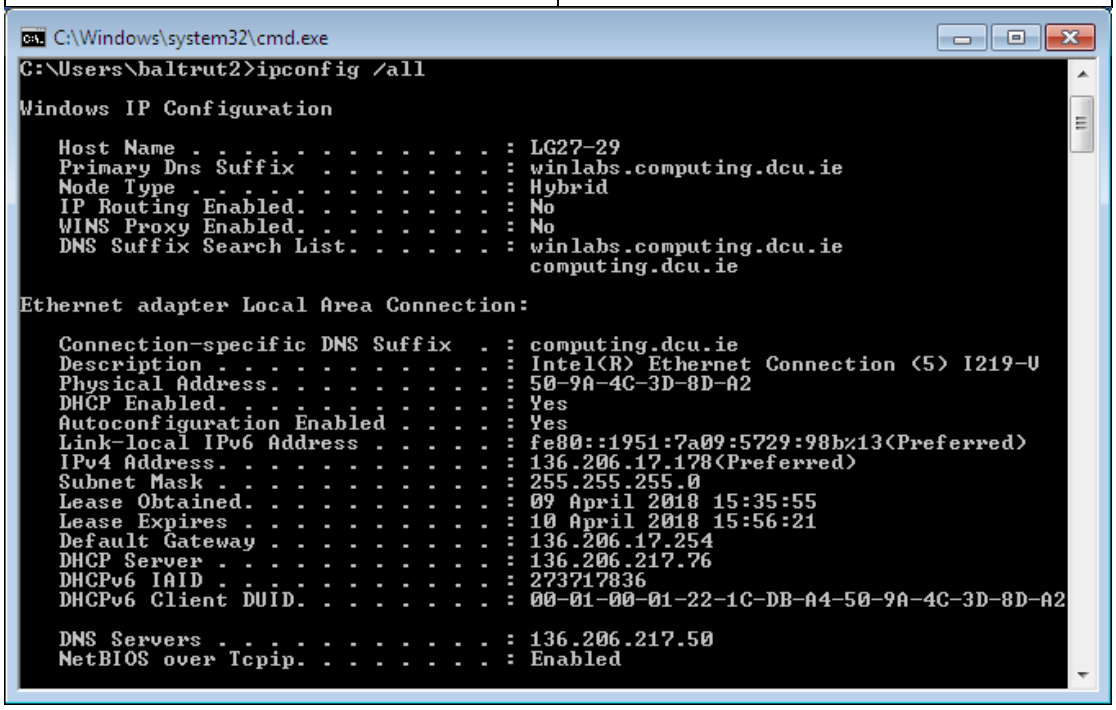
66, 68, 71, 75: ICMP echo ping reply.

- Responds to a ICMP request
- Generated by the target host
- Data inside particularly includes the control message ‘echo ping reply’, and a sequence number that corresponds to the request packet’s sequence number

Part 3:

Your IP & MAC address for this experiment (use ipconfig)

136.206.17.178	50-9A-4C-3D-8D-A2

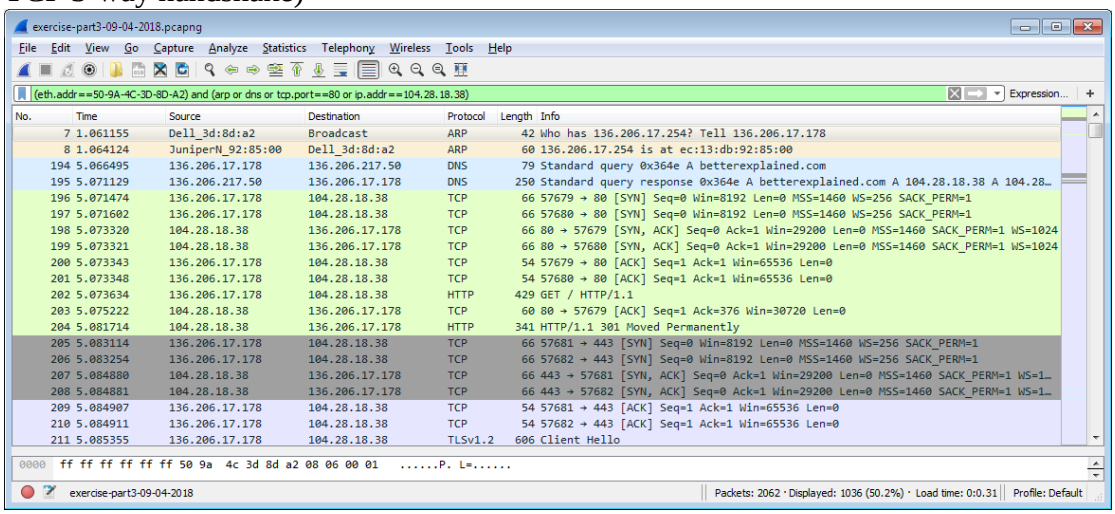


Filter to show only traffic concerning the test machine

Filter	**(eth.addr==50-9A-4C-3D-8D-A2) and (arp or dns or tcp.port==80 or ip.addr==104.28.18.38)**

Explain how you found the start of the interaction between your PC and the website.
Used parts of the filter above and the filter 'frame contains betterexplained' (name of the site I pinged)

Wireshark window showing the start of the interaction (should show ARP, DNS and TCP 3-way handshake)



Write down the numbers of the packets with the 3-way handshake.
196, 198, 200 (port 57679)

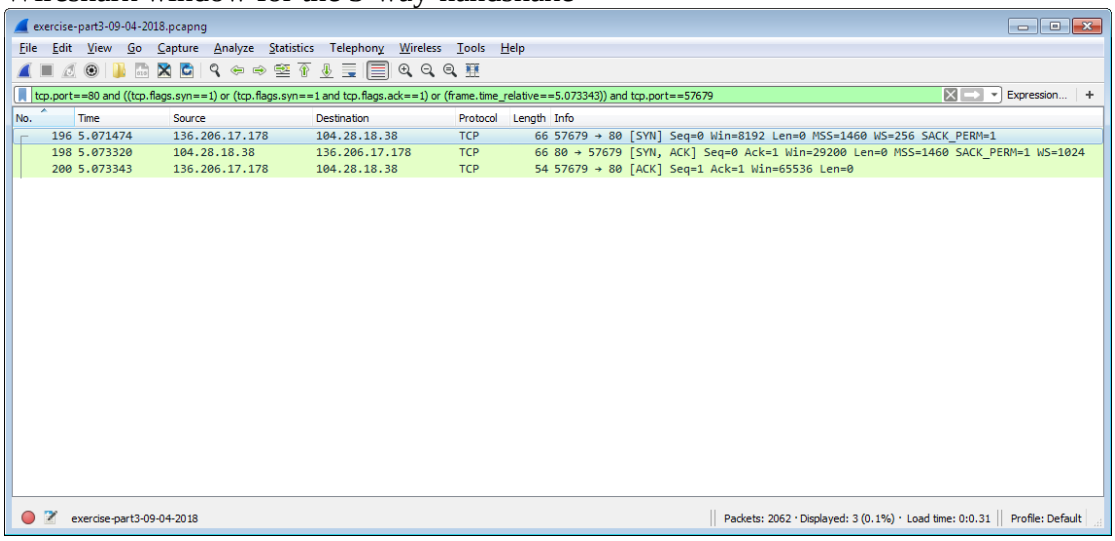
Explain what is happening with these 3 packets.
The client and server are trying to establish a TCP connection. The client sends a SYN(synchronise) packet to the server. The server acknowledges(ACK's) the packet and sends back a SYN to the client. The client acknowledges the server's SYN. Now the client and server can communicate.

Write down a filter to show only these three-way-handshake packets

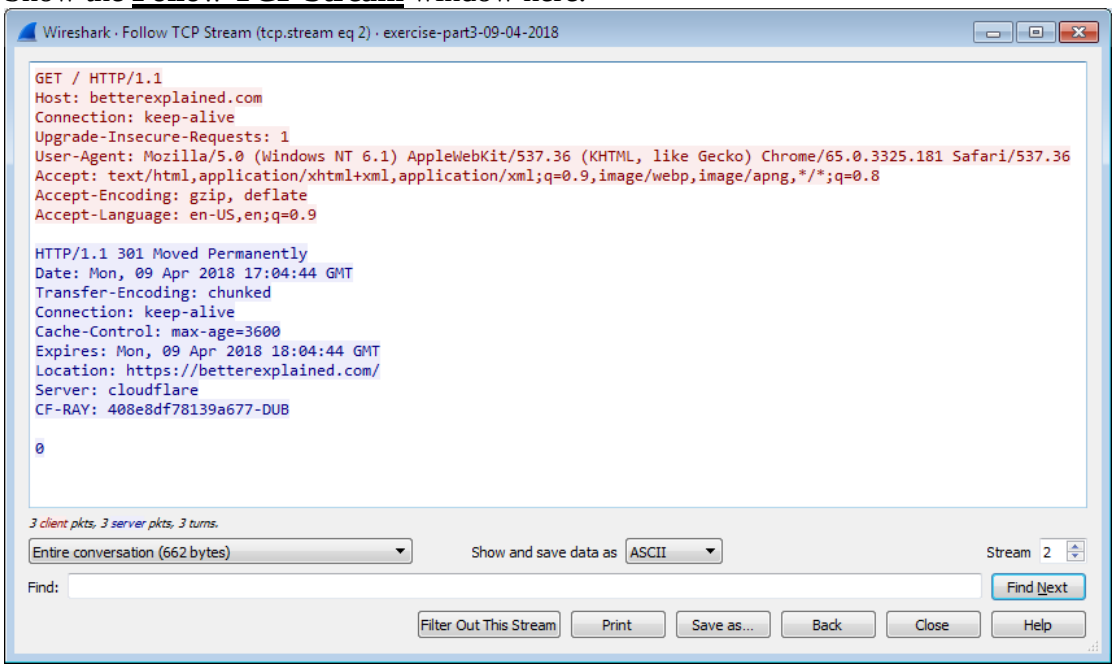
Filter	**tcp.port==80 and ((tcp.flags.syn==1) or (tcp.flags.syn==1 and tcp.flags.ack==1) or (frame.time_relative==5.073343)) and tcp.port==57679**

(I wasn't able to make the filter independent of the particular port used for the 3-way handshake and couldn't filter the last ACK packet, as many other TCP packets came with the ACK flag set)

Wireshark window for the 3-way-handshake



Show the **Follow TCP Stream** window here.



Your notes on…

a. The GET requests made
    i. **One request: '/ HTTP/1.1', which asks for all the files (going from root directory), using HTTP version 1.1. A few 'request headers' follow.**
b. The responses from the server
    i. **A response: The website has been 'moved permanently'. I think this is because the server is 'cloudflare' (DDoS protection service)**
c. The HTTP response codes used in the interaction and what they mean (look them up yourself on the Web)
    i. **The code: 301 or 'Moved Permanently'. Means that the URL has changed and needs to be updated. The site can still be accessed through URL forwarding (the 'Location' header provides a secure URL)**

Thank you for reading