

## Tarea 1

### Ejercicio 1

#### I) Exercise One

Open “Wireshark”, then use the “File” menu and the “Open” command to open the file “WS-Ex-01.pcap”. You should see 26 packets listed.

This set of packets describes a ‘conversation’ between a user’s client and a central server. This entire conversation happens automatically, after a user types something and hits enter. Look at the packets to answer the following questions in relation to this conversation.

In answering the following questions, use brief descriptions. For example, “In frame X, the client requests a web page, and in frame Y, the server delivers the content of the page.”

- What is the IP address of the client that initiates the conversation?
- Use the first two packets to identify the server that is going to be contacted. List the common name, and three IP addresses that can be used for the server.
- What is happening in frames 3, 4, and 5?
- What is happening in frames 6 and 7?
- Ignore frame eight. However, for your information, frame eight is used to manage flow control.
- What is happening in frames nine and ten? How are these two frames related?
- What happens in packet 11?
- After the initial set of packets is received, the client sends out a new request in packet 12. This occurs automatically without any action by the user. Why does this occur? See the first “hint” to the left.
- What is occurring in packets 13 through 22?
- Explain what happens in packets 23 through 26. See the second “hint” to the left.
- In one sentence describe what the user was doing (Reading email? Accessing a web page? FTP? Other?).

| No. | Time     | Source         | Destination    | Protocol | Length | Info   |
|-----|----------|----------------|----------------|----------|--------|--|
| 1   | 0.000000 | 131.247.95.216 | 131.247.92.200 | DNS      | 74     | Standard query 0xefc3 A www.google.com   |
| 2   | 0.000405 | 131.247.92.200 | 131.247.95.216 | DNS      | 142    | Standard query response 0xefc3 A www.google.com CNAME www.l.google.com A 64.233.161.99 A 64.233.161.104 A 64.233.161.147 |
| 3   | 0.001025 | 131.247.95.216 | 64.233.161.99  | TCP      | 60     | 60 → 1143 [SYN] Seq=0 Win=65536 Len=0 MSS=1460 SACK_PERM=1   |
| 4   | 0.002728 | 64.233.161.99  | 131.247.95.216 | TCP      | 60     | 60 → 1143 [SYN, ACK] Seq=0 Ack=1 Win=65536 Len=0 MSS=1460  |
| 5   | 0.002735 | 131.247.95.216 | 64.233.161.99  | TCP      | 54     | 54 → 1143 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0   |
| 6   | 0.002797 | 131.247.95.216 | 64.233.161.99  | HTTP     | 546    | GET / HTTP/1.1   |
| 7   | 0.053913 | 64.233.161.99  | 131.247.95.216 | TCP      | 60     | 60 → 1143 [ACK] Seq=1 Ack=493 Win=7680 Len=0   |
| 8   | 0.054815 | 64.233.161.99  | 131.247.95.216 | TCP      | 60     | [TCP Window Update] 80 → 1143 [ACK] Seq=1 Ack=493 Win=6432 Len=0   |
| 9   | 0.073552 | 64.233.161.99  | 131.247.95.216 | TCP      | 1484   | 80 → 1143 [ACK] Seq=1 Ack=493 Win=6432 Len=1430 [TCP PDU reassembled in 10]  |
| 10  | 0.073623 | 64.233.161.99  | 131.247.95.216 | HTTP     | 275    | HTTP/1.1 200 OK (text/html)  |
| 11  | 0.073677 | 131.247.95.216 | 64.233.161.99  | TCP      | 54     | 54 → 1143 → 80 [ACK] Seq=493 Ack=1652 Win=17520 Len=0  |
| 12  | 0.129180 | 131.247.95.216 | 64.233.161.99  | HTTP     | 522    | GET /img/ma/images/logo.gif HTTP/1.1   |
| 13  | 0.166185 | 64.233.161.99  | 131.247.95.216 | TCP      | 1484   | 80 → 1143 [ACK] Seq=1652 Ack=961 Win=7584 Len=1430 [TCP PDU reassembled in 22]   |
| 14  | 0.166293 | 64.233.161.99  | 131.247.95.216 | TCP      | 1484   | 80 → 1143 [ACK] Seq=3082 Ack=961 Win=7584 Len=1430 [TCP PDU reassembled in 22]   |
| 15  | 0.166353 | 131.247.95.216 | 64.233.161.99  | TCP      | 54     | 54 → 1143 → 80 [ACK] Seq=961 Ack=512 Win=17520 Len=0   |
| 16  | 0.166397 | 64.233.161.99  | 131.247.95.216 | TCP      | 1484   | 80 → 1143 [ACK] Seq=4512 Ack=961 Win=7584 Len=1430 [TCP PDU reassembled in 22]   |
| 17  | 0.193399 | 64.233.161.99  | 131.247.95.216 | TCP      | 1484   | 80 → 1143 [ACK] Seq=5942 Ack=961 Win=7584 Len=1430 [TCP PDU reassembled in 22]   |
| 18  | 0.193515 | 64.233.161.99  | 131.247.95.216 | TCP      | 1484   | 80 → 1143 [ACK] Seq=7372 Ack=961 Win=7584 Len=1430 [TCP PDU reassembled in 22]   |
| 19  | 0.193548 | 131.247.95.216 | 64.233.161.99  | TCP      | 54     | 54 → 1143 → 80 [ACK] Seq=961 Ack=7372 Win=17520 Len=0  |
| 20  | 0.193598 | 64.233.161.99  | 131.247.95.216 | TCP      | 1484   | 80 → 1143 [ACK] Seq=8802 Ack=961 Win=7584 Len=1430 [TCP PDU reassembled in 22]   |
| 21  | 0.193626 | 131.247.95.216 | 64.233.161.99  | TCP      | 54     | 54 → 1143 → 80 [ACK] Seq=961 Ack=18032 Win=17520 Len=0   |
| 22  | 0.220899 | 64.233.161.99  | 131.247.95.216 | HTTP     | 238    | HTTP/1.1 200 OK (GIF89a)   |
| 23  | 0.260899 | 131.247.95.216 | 64.233.161.99  | HTTP     | 477    | GET /favicon.ico HTTP/1.1  |
| 24  | 0.294356 | 64.233.161.99  | 131.247.95.216 | TCP      | 1484   | 80 → 1143 [ACK] Seq=10456 Ack=1384 Win=8576 Len=1430 [TCP PDU reassembled in 25]   |
| 25  | 0.294500 | 64.233.161.99  | 131.247.95.216 | HTTP     | 239    | HTTP/1.1 200 OK (image/x-icon)   |
| 26  | 0.294600 | 131.247.95.216 | 64.233.161.99  | TCP      | 54     | 54 → 1143 → 80 [ACK] Seq=1384 Ack=12031 Win=17520 Len=0  |

- La dirección IP del cliente que inicia la conversación es 131.247.95.216
- El server que quiere ser contactado es [www.google.com](http://www.google.com) y sus 3 IPs son 64.233.161.99, 64.233.161.104 y 64.233.161.104
- Lo que pasa entre el frame 3 y 5 es que se inicia un proceso de three way handshake:

- a. En el frame 3, se intenta realizar una conexión con un servidor realizando una solicitud con el número de secuencia 0.
  - b. En el frame 4, el servidor responde a la solicitud con su propio SYN y un ACK confirmando que recibió la solicitud.
  - c. En el frame 5, el cliente aumenta su número de secuencia y envía otro ACK para poder terminar de establecer la conexión con el servidor, permitiendo de este modo el intercambio de información.
- D. Una vez establecida la conexión TCP, entre el frame 6 y 7 pasa:
  - a. En el frame 6, se realiza una solicitud HTTP de la página principal del servidor
  - b. En el frame 7, el servidor envía un ACK indicando que recibió la solicitud.
- E. Se ignora el frame 8, según indicaciones. Sin embargo, se menciona que se encarga del control de flujo.
- F. En el frame 9 y 10 pasa:
  - a. En el frame 9, inicia la respuesta por parte del servidor al cliente.
  - b. En el frame 10, se envía la información solicitada por el cliente, terminando la respuesta del servidor. Ambos se relacionan porque juntos forman la respuesta completa del servidor. Ambos contienen información importante que una vez que se une contiene la información solicitada.
- G. En el frame 11, el cliente envía un ACK al servidor para indicarle que recibió los datos enviados.
- H. Esto sucede porque la solicitud del frame 12 es un elemento necesario para cargar por completo la página, por lo que después de que el usuario solicite la página, se pueden realizar solicitudes automáticas para cargar otros elementos necesarios para poder mostrar al usuario correctamente su solicitud inicial, que en este caso es la página. Esto también está relacionado con el comportamiento de la caché DNS, la cual en este caso realiza una consulta.
- I. Entre el frame 13 y 22 sucede:
  - a. En los frames 13,14,16,17,18 y 19 el servidor envía fragmentos de la información solicitada por el cliente, en este caso una imagen.
  - b. Entre los frames 15, 19 y 21 el cliente envía un ACK al servidor para indicar que recibió la información.
  - c. En el frame 23, el servidor indica la respuesta HTTP 200 OK que incluye los datos completos de la imagen y el tipo de dato.
- J. Entre el frame 23 y 26 sucede:
  - a. En el frame 23, el cliente solicita al servidor un archivo llamado favicon.ico, que como se menciona en la pista, es un gráfico pequeño que se utiliza como icono para identificar una página web.
- K. El usuario se conectó a una página web para solicitar archivos.

## Ejercicio 2

## II) Exercise Two

Open “Wireshark”, then use the “File” menu and the “Open” command to open the file “WS-Ex-02.pcap”. You should see 176 packets listed.

- In the first few packets, the client machine is looking up the common name (cname) of a web site to find its IP address. What is the cname of this web site? Give two IP addresses for this web site.
- How many packets/frames does it take to receive the web page (the answer to the first http get request only)?
- Does this web site use gzip to compress its data for sending? Does it write cookies? In order to answer these questions, look under the payload for the reassembled packet that represents the web page. This will be the last packet from question b above. Look to see if it has “Content-Encoding” set to gzip, and to see if it has a “Set-Cookie” to write a cookie.
- What is happening in packets 26 and 27? Does every component of a web page have to come from the same server? See the Hint to the left.
- In packet 37 we see another DNS query, this time for us.i1.yimg.com. Why does the client need to ask for this IP address? Didn’t we just get this address in packet 26? (This is a trick question; carefully compare the two common names in packet 26 and 37.)
- In packet 42 we see a HTTP “Get” statement, and in packet 48 a new HTTP “Get” statement. Why didn’t the system need another DNS request before the second get statement? Click on packet 42 and look in the middle window. Expand the line titled “Hypertext Transfer Protocol” and read the “Host:” line. Compare that line to the “Host:” line for packet 48.
- Examine packet 139. It is one segment of a PDU that is reassembled with several other segments in packet 160. Look at packets 141, 142, and 143. Are these three packets also part of packet 160? What happens if a set of packets that are supposed to be reassembled do not arrive in a continuous stream or do not arrive in the proper order?
- Return to examine frames 141 and 142. Both of these are graphics (GIF files) from the same source IP address. How does the client know which graphic to match up to each get statement? Hint: Click on each and look in the middle window for the heading line that starts with “Transmission Control Protocol”. What difference do you see in the heading lines for the two files? Return to the original “Get” statements. Can you see the same difference in the “Get” statements?

| No. | Time     | Source          | Destination     | Protocol | Length | Info  |
|-----|----------|-----------------|-----------------|----------|--------|---|
| 1   | 0.000000 | 131.247.95.216  | 131.247.92.200  | DNS      | 73     | Standard query 0x159b A www.yahoo.com   |
| 2   | 0.001630 | 131.247.92.200  | 131.247.95.216  | DNS      | 542    | Standard query response 0x159b A www.yahoo.com CNAME www.yahoo.akadns.net A 216.109.117.106 A 216.109.117.109 A 216.109.117.110 A 216.109.117.204 A 216.109.117.206 A 216.109.118.70 A 216.109.118.71 |
| 3   | 0.001754 | 131.247.95.216  | 216.109.117.106 | TCP      | 62     | 1223 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1  |
| 4   | 0.028756 | 216.109.117.106 | 131.247.95.216  | TCP      | 60     | 80 → 1221 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460   |
| 5   | 0.028873 | 131.247.95.216  | 216.109.117.106 | TCP      | 54     | 1221 → 80 [ACK] Seq=1 Win=17520 Len=0   |
| 6   | 0.032252 | 131.247.95.216  | 216.109.117.106 | HTTP     | 402    | GET / HTTP/1.1  |
| 7   | 0.002807 | 216.109.117.106 | 131.247.95.216  | TCP      | 1514   | 80 → 1221 [ACK] Seq=1 Ack=439 Win=65535 Len=1460 [TCP PDU reassembled in 22]  |
| 8   | 0.002945 | 216.109.117.106 | 131.247.95.216  | TCP      | 1514   | 80 → 1221 [ACK] Seq=1461 Ack=439 Win=65535 Len=1460 [TCP PDU reassembled in 22]   |
| 9   | 0.002998 | 131.247.95.216  | 216.109.117.106 | TCP      | 54     | 1221 → 80 [ACK] Seq=439 Ack=2921 Win=17520 Len=0  |
| 10  | 0.003805 | 216.109.117.106 | 131.247.95.216  | TCP      | 1514   | 80 → 1221 [ACK] Seq=2921 Ack=439 Win=65535 Len=1460 [TCP PDU reassembled in 22]   |
| 11  | 0.108801 | 216.109.117.106 | 131.247.95.216  | TCP      | 1514   | 80 → 1221 [ACK] Seq=4381 Ack=439 Win=65535 Len=1460 [TCP PDU reassembled in 22]   |
| 12  | 0.108805 | 216.109.117.106 | 131.247.95.216  | TCP      | 1514   | 80 → 1221 [ACK] Seq=5841 Ack=439 Win=65535 Len=1460 [TCP PDU reassembled in 22]   |
| 13  | 0.108808 | 131.247.95.216  | 216.109.117.106 | TCP      | 54     | 1221 → 80 [ACK] Seq=439 Ack=5841 Win=17520 Len=0  |
| 14  | 0.108906 | 216.109.117.106 | 131.247.95.216  | TCP      | 1514   | 80 → 1221 [ACK] Seq=7903 Ack=439 Win=65535 Len=1460 [TCP PDU reassembled in 22]   |
| 15  | 0.108914 | 131.247.95.216  | 216.109.117.106 | TCP      | 54     | 1221 → 80 [ACK] Seq=439 Ack=8761 Win=17520 Len=0  |
| 16  | 0.114687 | 216.109.117.106 | 131.247.95.216  | TCP      | 1514   | 80 → 1221 [ACK] Seq=8763 Ack=439 Win=65535 Len=1460 [TCP PDU reassembled in 22]   |
| 17  | 0.134777 | 216.109.117.106 | 131.247.95.216  | TCP      | 1514   | 80 → 1221 [ACK] Seq=10221 Ack=439 Win=65535 Len=1460 [TCP PDU reassembled in 22]  |
| 18  | 0.134834 | 131.247.95.216  | 216.109.117.106 | TCP      | 54     | 1221 → 80 [ACK] Seq=439 Ack=10881 Win=17520 Len=0   |
| 19  | 0.135010 | 216.109.117.106 | 131.247.95.216  | TCP      | 1514   | 80 → 1221 [ACK] Seq=11681 Ack=439 Win=65535 Len=1460 [TCP PDU reassembled in 22]  |
| 20  | 0.135036 | 216.109.117.106 | 131.247.95.216  | TCP      | 1514   | 80 → 1221 [ACK] Seq=13141 Ack=439 Win=65535 Len=1460 [TCP PDU reassembled in 22]  |
| 21  | 0.135069 | 131.247.95.216  | 216.109.117.106 | TCP      | 54     | 1221 → 80 [ACK] Seq=439 Ack=14081 Win=17520 Len=0   |
| 22  | 0.135124 | 216.109.117.106 | 131.247.95.216  | HTTP     | 1115   | HTTP/1.1 200 OK (text/html)   |
| 23  | 0.135175 | 131.247.95.216  | 216.109.117.106 | TCP      | 54     | 1221 → 80 [ACK] Seq=439 Ack=15663 Win=16459 Len=0   |
| 24  | 0.135590 | 131.247.95.216  | 216.109.117.106 | TCP      | 54     | 1221 → 80 [FIN, ACK] Seq=439 Ack=15663 Win=16459 Len=0  |
| 25  | 0.160526 | 216.109.117.106 | 131.247.95.216  | TCP      | 60     | 80 → 1221 [ACK] Seq=15663 Ack=440 Win=65535 Len=0   |
| 26  | 0.232293 | 131.247.95.216  | 131.247.92.200  | DNS      | 75     | Standard query 0x0400 A us.i1.yimg.com  |
| 27  | 0.234735 | 131.247.92.200  | 131.247.95.216  | DNS      | 198    | Standard query response 0x0400 A us.i1.yimg.com CNAME a321.yimg.com georedirector.akadns.net CNAME a321.x.a.yimg.com A 64.21.46.151 A 64.21.46.134 A 64.21.46.150                                     |
| 28  | 0.237844 | 131.247.95.216  | 64.21.46.151    | TCP      | 62     | 1223 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1  |
| 29  | 0.287086 | 64.21.46.151    | 131.247.95.216  | TCP      | 60     | 80 → 1223 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM=1   |
| 30  | 0.287939 | 131.247.95.216  | 64.21.46.151    | TCP      | 54     | 1223 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0   |
| 31  | 0.318189 | 131.247.95.216  | 64.21.46.151    | HTTP     | 425    | GET /us.yimg.com/i/mu/ut3.1.3.js HTTP/1.1   |
| 32  | 0.340927 | 64.21.46.151    | 131.247.95.216  | TCP      | 60     | 80 → 1223 [ACK] Seq=1 Ack=372 Win=6432 Len=0  |
| 33  | 0.350907 | 64.21.46.151    | 131.247.95.216  | TCP      | 1514   | 80 → 1223 [ACK] Seq=1 Ack=372 Win=6432 Len=1460 [TCP PDU reassembled in 35]   |
| 34  | 0.351853 | 64.21.46.151    | 131.247.95.216  | TCP      | 1514   | 80 → 1223 [ACK] Seq=1461 Ack=372 Win=6432 Len=1460 [TCP PDU reassembled in 35]  |
| 35  | 0.351862 | 64.21.46.151    | 131.247.95.216  | HTTP     | 107    | HTTP/1.1 200 OK (application/javascript)  |
| 36  | 0.351120 | 131.247.95.216  | 64.21.46.151    | TCP      | 54     | 1223 → 80 [ACK] Seq=372 Ack=2974 Win=17520 Len=0  |
| 37  | 0.402117 | 131.247.95.216  | 131.247.92.200  | DNS      | 74     | Standard query 0x0f09 A us.i1.yimg.com  |
| 38  | 0.404549 | 131.247.92.200  | 131.247.95.216  | DNS      | 181    | Standard query response 0x0f09 A us.i1.yimg.com CNAME a943.yimg.com georedirector.akadns.net CNAME a943.x.a.yimg.com A 64.21.46.137 A 64.21.46.144  |
| 39  | 0.406236 | 131.247.95.216  | 64.21.46.137    | TCP      | 62     | 1225 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1  |
| 40  | 0.517684 | 64.21.46.137    | 131.247.95.216  | TCP      | 62     | 80 → 1225 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM=1   |
| 41  | 0.517588 | 131.247.95.216  | 64.21.46.137    | TCP      | 54     | 1225 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0   |
| 42  | 0.518881 | 131.247.95.216  | 64.21.46.137    | HTTP     | 423    | GET /us.yimg.com/i/mu/dt1.1.1.js HTTP/1.1   |
| 43  | 0.519264 | 64.21.46.137    | 131.247.95.216  | TCP      | 60     | 80 → 1225 [ACK] Seq=1 Ack=378 Win=6432 Len=0  |
| 44  | 0.553408 | 64.21.46.137    | 131.247.95.216  | TCP      | 1514   | 80 → 1225 [ACK] Seq=1 Ack=378 Win=6432 Len=1460 [TCP PDU reassembled in 47]   |
| 45  | 0.553530 | 64.21.46.137    | 131.247.95.216  | TCP      | 1514   | 80 → 1225 [ACK] Seq=1461 Ack=378 Win=6432 Len=1460 [TCP PDU reassembled in 47]  |
| 46  | 0.553593 | 131.247.95.216  | 64.21.46.137    | TCP      | 54     | 1225 → 80 [ACK] Seq=378 Ack=2921 Win=17520 Len=0  |
| 47  | 0.553629 | 64.21.46.137    | 131.247.95.216  | HTTP     | 1287   | HTTP/1.1 200 OK (application/javascript)  |
| 48  | 0.504952 | 131.247.95.216  | 64.21.46.137    | HTTP     | 440    | GET /us.yimg.com/i/mu/dt2.1.1.gif HTTP/1.1  |
| 49  | 0.597939 | 131.247.95.216  | 64.21.46.137    | TCP      | 62     | 1226 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1  |
| 50  | 0.656088 | 64.21.46.137    | 131.247.95.216  | TCP      | 1514   | 80 → 1226 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM=1   |
| 51  | 0.626482 | 64.21.46.137    | 131.247.95.216  | TCP      | 705    | HTTP/1.1 200 OK (GIF89a)  |
| 52  | 0.626534 | 131.247.95.216  | 64.21.46.137    | TCP      | 54     | 1226 → 80 [ACK] Seq=756 Ack=6325 Win=17520 Len=0  |
| 53  | 0.626641 | 64.21.46.137    | 131.247.95.216  | TCP      | 62     | 80 → 1226 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM=1   |
| 54  | 0.626746 | 131.247.95.216  | 64.21.46.137    | TCP      | 54     | 1226 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0   |
| 55  | 0.735985 | 131.247.95.216  | 64.21.46.137    | HTTP     | 440    | GET /us.yimg.com/i/mu/dt1/125.gif HTTP/1.1  |
| 56  | 0.736061 | 131.247.95.216  | 64.21.46.137    | HTTP     | 442    | GET /us.yimg.com/i/mu/dt1/13441.gif HTTP/1.1  |

| No. | Time     | Source         | Destination    | Protocol | Length | Info  |
|-----|----------|----------------|----------------|----------|--------|---|
| 55  | 0.755985 | 131.247.95.216 | 64.21.46.137   | HTTP     | 440    | GET /us.yimg.com/i/w/bt/125.gif HTTP/1.1  |
| 56  | 0.755611 | 131.247.95.216 | 64.21.46.137   | HTTP     | 442    | GET /us.yimg.com/i/w/bt/13441.gif HTTP/1.1  |
| 57  | 0.741545 | 131.247.95.216 | 131.247.92.200 | DNS      | 74     | Standard query 0x0e99 A us.s1.yimg.com  |
| 58  | 0.741998 | 131.247.92.200 | 131.247.95.216 | HTTP     | 187    | Standard query response 0x0e99 A us.s1.yimg.com CNAME a566.yimg.com.georedirector.akadns.net CNAME a321.a.yimg.com A 64.21.46.134 A 64.21.46.150 A 64.21.46.151 |
| 59  | 0.743562 | 131.247.95.216 | 64.21.46.134   | TCP      | 62     | 1228 → 80 [SYN] Seq=1541384 Len=0 MSS=1460 SACK_PERM  |
| 60  | 0.767242 | 64.21.46.137   | 131.247.95.216 | TCP      | 60     | 80 → 1226 [ACK] Seq=1 Acks=387 Win=6432 Len=0   |
| 61  | 0.767669 | 64.21.46.137   | 131.247.95.216 | HTTP     | 1218   | HTTP/1.0 200 OK (GIF89a)  |
| 62  | 0.767792 | 64.21.46.137   | 131.247.95.216 | HTTP     | 1313   | HTTP/1.0 200 OK (GIF89a)  |
| 63  | 0.773998 | 64.21.46.134   | 131.247.95.216 | TCP      | 62     | 80 → 1228 [SYN] Seq=1 Acks=1 Win=5840 Len=0 MSS=1460 SACK_PERM  |
| 64  | 0.774114 | 131.247.95.216 | 64.21.46.134   | TCP      | 54     | 1228 → 80 [ACK] Seq=1 Acks=1 Win=17320 Len=0  |
| 65  | 0.777929 | 131.247.95.216 | 64.21.46.134   | HTTP     | 444    | GET /us.yimg.com/a/1-/java/promotions/js/ad_oe_1.1.js HTTP/1.1  |
| 66  | 0.778217 | 131.247.95.216 | 64.21.46.137   | HTTP     | 439    | GET /us.yimg.com/i/w/bt/121.gif HTTP/1.1  |
| 67  | 0.778281 | 131.247.95.216 | 64.21.46.137   | HTTP     | 442    | GET /us.yimg.com/i/w/bt/123511.gif HTTP/1.1   |
| 68  | 0.808999 | 64.21.46.134   | 131.247.95.216 | TCP      | 60     | 80 → 1228 [ACK] Seq=1 Acks=391 Win=6432 Len=0   |
| 69  | 0.809459 | 64.21.46.137   | 131.247.95.216 | HTTP     | 1836   | HTTP/1.0 200 OK (application/x-javascript)  |
| 70  | 0.809948 | 64.21.46.137   | 131.247.95.216 | HTTP     | 1396   | HTTP/1.0 200 OK (GIF89a)  |
| 71  | 0.810034 | 64.21.46.137   | 131.247.95.216 | HTTP     | 441    | GET /us.yimg.com/i/w/bt/123511.gif HTTP/1.1   |
| 72  | 0.810761 | 131.247.95.216 | 64.21.46.137   | HTTP     | 439    | GET /us.yimg.com/i/w/bt/121.gif HTTP/1.1  |
| 73  | 0.810832 | 131.247.95.216 | 64.21.46.137   | HTTP     | 1109   | HTTP/1.0 200 OK (GIF89a)  |
| 74  | 0.842345 | 64.21.46.137   | 131.247.95.216 | TCP      | 1300   | HTTP/1.0 200 OK (GIF89a)  |
| 75  | 0.842450 | 64.21.46.137   | 131.247.95.216 | HTTP     | 436    | GET /us.yimg.com/i/w/new.gif HTTP/1.1   |
| 76  | 0.843376 | 131.247.95.216 | 64.21.46.137   | HTTP     | 442    | GET /us.yimg.com/i/w/vr_mall_t.gif HTTP/1.1   |
| 77  | 0.843446 | 131.247.95.216 | 64.21.46.137   | HTTP     | 405    | HTTP/1.0 200 OK (GIF89a)  |
| 78  | 0.875131 | 64.21.46.137   | 131.247.95.216 | HTTP     | 977    | HTTP/1.0 200 OK (GIF89a)  |
| 79  | 0.875209 | 64.21.46.137   | 131.247.95.216 | HTTP     | 453    | GET /us.yimg.com/i/wt/123511.gif HTTP/1.1   |
| 80  | 0.875156 | 131.247.95.216 | 64.21.46.137   | HTTP     | 457    | GET /us.yimg.com/i/wt/123511.gif HTTP/1.1   |
| 81  | 0.875226 | 131.247.95.216 | 64.21.46.137   | HTTP     | 457    | GET /us.yimg.com/i/wt/123511.gif HTTP/1.1   |
| 82  | 0.907734 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1225 [ACK] Seq=1 Acks=2 Acks=2781 Win=12864 Len=1460 [TCP PDU reassembled in 98]   |
| 83  | 0.907855 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1225 [ACK] Seq=1 Acks=2 Acks=2781 Win=12864 Len=1460 [TCP PDU reassembled in 98]   |
| 84  | 0.907954 | 131.247.95.216 | 64.21.46.137   | TCP      | 54     | 1225 → 80 [ACK] Seq=2781 Acks=1342 Win=17320 Len=0  |
| 85  | 0.907953 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1225 [ACK] Seq=1342 Acks=2781 Win=12864 Len=1460 [TCP PDU reassembled in 98]   |
| 86  | 0.908009 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1225 [ACK] Seq=14802 Acks=2781 Win=12864 Len=1460 [TCP PDU reassembled in 98]  |
| 87  | 0.908104 | 131.247.95.216 | 64.21.46.137   | TCP      | 54     | 1225 → 80 [ACK] Seq=2781 Acks=16262 Win=17320 Len=0   |
| 88  | 0.908223 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1225 [ACK] Seq=1813 Acks=1847 Win=18720 Len=1460 [TCP PDU reassembled in 89]   |
| 89  | 0.908290 | 64.21.46.137   | 131.247.95.216 | HTTP     | 841    | HTTP/1.0 200 OK (GIF89a)  |
| 90  | 0.908318 | 131.247.95.216 | 64.21.46.137   | TCP      | 54     | 1225 → 80 [ACK] Seq=1847 Acks=6800 Win=17320 Len=0  |
| 91  | 0.909189 | 131.247.95.216 | 64.21.46.137   | HTTP     | 459    | GET /us.yimg.com/i/w/news/2006/05/08/050808blaine.jpg HTTP/1.1  |
| 92  | 0.938923 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1225 [ACK] Seq=16262 Acks=2781 Win=12864 Len=1460 [TCP PDU reassembled in 98]  |
| 93  | 0.939019 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1225 [ACK] Seq=17722 Acks=2781 Win=12864 Len=1460 [TCP PDU reassembled in 98]  |
| 94  | 0.939078 | 131.247.95.216 | 64.21.46.137   | TCP      | 54     | 1225 → 80 [ACK] Seq=2781 Acks=19182 Win=17320 Len=0   |
| 95  | 0.939122 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1225 [ACK] Seq=1342 Acks=2781 Win=12864 Len=1460 [TCP PDU reassembled in 98]   |
| 96  | 0.939261 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1225 [ACK] Seq=20642 Acks=2781 Win=12864 Len=1460 [TCP PDU reassembled in 98]  |
| 97  | 0.939293 | 131.247.95.216 | 64.21.46.137   | TCP      | 54     | 1225 → 80 [ACK] Seq=2781 Acks=22182 Win=17320 Len=0   |
| 98  | 0.939321 | 64.21.46.137   | 131.247.95.216 | TCP      | 807    | HTTP/1.0 200 OK (JPEG JFIF image)   |
| 99  | 0.939794 | 131.247.95.216 | 64.21.46.137   | HTTP     | 440    | GET /us.yimg.com/i/w/beta/news/video.gif HTTP/1.1   |
| 100 | 0.940081 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1226 [ACK] Seq=8008 Acks=2352 Win=11792 Len=1460 [TCP PDU reassembled in 103]  |
| 101 | 0.940719 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1226 [ACK] Seq=7308 Acks=2352 Win=11792 Len=1460 [TCP PDU reassembled in 103]  |
| 102 | 0.940761 | 131.247.95.216 | 64.21.46.137   | TCP      | 54     | 1226 → 80 [ACK] Seq=2352 Acks=8908 Win=17320 Len=0  |
| 103 | 0.940794 | 64.21.46.137   | 131.247.95.216 | TCP      | 1127   | HTTP/1.0 200 OK (JPEG JFIF image)   |
| 104 | 0.941205 | 131.247.95.216 | 64.21.46.137   | HTTP     | 455    | GET /us.yimg.com/i/buzz/2006/05/05/050505small.jpg HTTP/1.1   |
| 105 | 0.947094 | 131.247.95.216 | 64.21.46.134   | TCP      | 54     | 1226 → 80 [ACK] Seq=91 Acks=1268 Win=16261 Len=0  |
| 106 | 0.947459 | 131.247.95.216 | 64.21.46.134   | HTTP     | 467    | GET /us.yimg.com/a/1-/Flash/promotions/nbc/080508/7011.gif HTTP/1.1   |
| 107 | 0.971439 | 64.21.46.137   | 131.247.95.216 | HTTP     | 617    | HTTP/1.0 200 OK (GIF89a)  |
| 108 | 0.972142 | 131.247.95.216 | 64.21.46.137   | HTTP     | 442    | GET /us.yimg.com/i/w/pe_sm11y.gif HTTP/1.1  |
| 109 | 0.972565 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1226 [ACK] Seq=10853 Acks=2753 Win=12864 Len=1460 [TCP PDU reassembled in 110]   |
| 110 | 0.972665 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1226 [ACK] Seq=10853 Acks=2753 Win=12864 Len=1460 [TCP PDU reassembled in 110]   |
| 111 | 0.972765 | 64.21.46.137   | 131.247.95.216 | TCP      | 54     | 1226 → 80 [ACK] Seq=2753 Acks=12768 Win=17320 Len=0   |
| 112 | 0.972780 | 131.247.95.216 | 64.21.46.137   | HTTP     | 443    | GET /us.yimg.com/i/w/pe_errone.png HTTP/1.1   |
| 113 | 0.972950 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=1260 Acks=804 Win=7504 Len=1460 [TCP PDU reassembled in 116]  |
| 114 | 0.978691 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=2720 Acks=804 Win=7504 Len=1460 [TCP PDU reassembled in 116]  |
| 115 | 0.978765 | 131.247.95.216 | 64.21.46.134   | TCP      | 54     | 1228 → 80 [ACK] Seq=804 Acks=14108 Win=17320 Len=0  |
| 116 | 0.978781 | 64.21.46.134   | 131.247.95.216 | TCP      | 630    | HTTP/1.0 200 OK (GIF89a)  |
| 117 | 0.978861 | 131.247.95.216 | 64.21.46.134   | TCP      | 428    | GET /us.yimg.com/i/w/vr_1.1.js HTTP/1.1   |
| 118 | 0.980556 | 64.21.46.137   | 131.247.95.216 | TCP      | 1514   | 80 → 1225 [ACK] Seq=23562 Acks=3483 Win=15008 Len=1460 [TCP PDU reassembled in 119]   |
| 119 | 0.980593 | 64.21.46.137   | 131.247.95.216 | HTTP     | 408    | HTTP/1.0 200 OK (GIF89a)  |
| 120 | 0.980656 | 131.247.95.216 | 64.21.46.137   | TCP      | 54     | 1225 → 80 [ACK] Seq=3483 Acks=25376 Win=17320 Len=0   |
| 121 | 0.980426 | 131.247.95.216 | 64.21.46.137   | HTTP     | 430    | GET /us.yimg.com/i/w/pld3n.gif HTTP/1.1   |
| 122 | 0.980438 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1226 [ACK] Seq=2780 Acks=3142 Win=13936 Len=1460 [TCP PDU reassembled in 123]  |
| 123 | 0.980434 | 64.21.46.137   | 131.247.95.216 | TCP      | 1167   | 0 → 200 OK (PNG)  |
| 124 | 0.980568 | 131.247.95.216 | 64.21.46.137   | TCP      | 54     | 1226 → 80 [ACK] Seq=3142 Acks=15490 Win=17320 Len=0   |
| 125 | 0.980550 | 131.247.95.216 | 64.21.46.137   | HTTP     | 454    | GET /us.yimg.com/i/w/ntgr/search/r1_wd2.gif HTTP/1.1  |
| 126 | 0.929157 | 64.21.46.131   | 131.247.95.216 | TCP      | 1514   | 80 → 1223 [ACK] Seq=2974 Acks=742 Win=7504 Len=1460 [TCP PDU reassembled in 129]  |
| 127 | 0.929262 | 64.21.46.131   | 131.247.95.216 | TCP      | 1514   | 80 → 1223 [ACK] Seq=434 Acks=742 Win=7504 Len=1460 [TCP PDU reassembled in 129]   |
| 128 | 0.929337 | 131.247.95.216 | 64.21.46.131   | TCP      | 54     | 1223 → 80 [ACK] Seq=742 Acks=8884 Win=17320 Len=0   |
| 129 | 0.929353 | 64.21.46.131   | 131.247.95.216 | HTTP     | 603    | HTTP/1.1 200 OK (application/x-javascript)  |
| 130 | 0.932353 | 131.247.95.216 | 64.21.46.131   | HTTP     | 505    | GET /us.yimg.com/i/Flash/promotions/state/far/060508/100ffcClickId=javascript:sfAction() HTTP/1.1   |
| 131 | 0.935910 | 64.21.46.137   | 131.247.95.216 | HTTP     | 455    | HTTP/1.0 200 OK (GIF89a)  |
| 132 | 0.936422 | 64.21.46.137   | 131.247.95.216 | HTTP     | 329    | HTTP/1.0 200 OK (GIF89a)  |
| 133 | 0.936937 | 131.247.95.216 | 64.21.46.137   | HTTP     | 442    | GET /us.yimg.com/i/w/rtfc_bckt.gif HTTP/1.1   |
| 134 | 0.937085 | 131.247.95.216 | 64.21.46.137   | HTTP     | 440    | GET /us.yimg.com/i/w/answers.gif HTTP/1.1   |
| 135 | 0.936365 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=6756 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]   |
| 136 | 0.936778 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=6216 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]   |
| 137 | 0.938343 | 131.247.95.216 | 64.21.46.134   | TCP      | 54     | 1228 → 80 [ACK] Seq=1335 Acks=7676 Win=17320 Len=0  |
| 138 | 0.938389 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=7676 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]   |
| 139 | 0.940423 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=9136 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]   |
| 140 | 0.940497 | 131.247.95.216 | 64.21.46.134   | TCP      | 54     | 1228 → 80 [ACK] Seq=1335 Acks=10866 Win=17320 Len=0   |
| 141 | 0.940563 | 64.21.46.137   | 131.247.95.216 | HTTP     | 1089   | HTTP/1.0 200 OK (GIF89a)  |
| 142 | 0.940688 | 64.21.46.137   | 131.247.95.216 | HTTP     | 917    | HTTP/1.0 200 OK (GIF89a)  |
| 143 | 0.940450 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=10856 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]  |
| 144 | 0.940761 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=12056 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]  |
| 145 | 0.940434 | 131.247.95.216 | 64.21.46.134   | TCP      | 54     | 1228 → 80 [ACK] Seq=1335 Acks=13516 Win=17320 Len=0   |
| 146 | 0.940470 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=13516 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]  |
| 147 | 0.940497 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=14976 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]  |
| 148 | 0.959020 | 131.247.95.216 | 64.21.46.134   | TCP      | 54     | 1228 → 80 [ACK] Seq=13516 Acks=16436 Win=17320 Len=0  |
| 149 | 0.959110 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=16436 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]  |
| 150 | 0.959212 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=17896 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]  |
| 151 | 0.959278 | 131.247.95.216 | 64.21.46.134   | TCP      | 54     | 1228 → 80 [ACK] Seq=1335 Acks=13566 Win=17320 Len=0   |
| 152 | 0.959272 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=13566 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]  |
| 153 | 0.959275 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=20816 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]  |
| 154 | 0.959460 | 64.21.46.134   | 131.247.95.216 | TCP      | 54     | 1228 → 80 [ACK] Seq=1335 Acks=22276 Win=17320 Len=0   |
| 155 | 0.959802 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=22276 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]  |
| 156 | 0.959817 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=23776 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]  |
| 157 | 0.959803 | 131.247.95.216 | 64.21.46.134   | TCP      | 54     | 1228 → 80 [ACK] Seq=1335 Acks=25196 Win=17320 Len=0   |
| 158 | 0.960154 | 64.21.46.134   | 131.247.95.216 | TCP      | 1514   | 80 → 1228 [ACK] Seq=25196 Acks=1335 Win=8576 Len=1460 [TCP PDU reassembled in 160]  |
| 159 | 0.960154 | 64.21.46.134   | 131.247.95.216 | TCP      | 54     | 1228 → 80 [ACK] Seq=1335 Acks=25196 Win=17320 Len=0   |
| 160 | 0.960154 | 64.21.46.134   | 131.247.95.216 | TCP      | 54     | 1228 → 80 [ACK] Seq=1335 Acks=25196 Win=17320 Len=0   |
| 161 | 0.960154 | 64.21.46.134   | 131.247.95.216 | TCP      | 54     | 1228 → 80 [ACK] Seq=1335 Acks=25196 Win=17320 Len=0   |
| 162 | 0.960154 | 64.21.46.134   | 131.247.95.216 | TCP      | 54     | 1228 → 80 [ACK] Seq=1335 Acks=25196 Win=17320 Len=0   |
| 163 | 0.960154 |                |                |          |        |   |

- B. Para recibir la página se toman 20, tomando en cuenta el paquete que realiza la solicitud y el paquete de confirmación de la respuesta completa y cierre de la comunicación que van desde el frame 6 que es el inicio de la solicitud, el frame 22 que es el ensamblado de la respuesta y el frame 25 que es el frame de cierre de la comunicación. Cómo tal la respuesta acaba en el frame 22.
- C. Sí, el servidor utiliza gzip para enviar contenido comprimido y escribe cookies en su respuesta, como se puede observar en la siguiente imagen:

```
Vary: User-Agent\r\n
Set-Cookie: FPB=ol1uquj7e125v8pe; expires=Thu, 01 Jun 2006 19:00:00 GMT; path=/; domain=www.yahoo.com\r\n
Set-Cookie: D=_ylh=X3oDMTFmdXFnazJs8F9TAzI3MTYxNDkEc6lkAzExNDcxMTC5NTQEdGVzdA%wBHRtcGwDaw5kZXgtY3Nz; path=/; domain=.yahoo.com\r\n
Connection: close\r\n
Transfer-Encoding: chunked\r\n
Content-Type: text/html\r\n
Content-Encoding: gzip\r\n
\r\n
```

- D. En los frames 26 y 27 sucede:
- En el frame 26, el cliente realiza una solicitud DNS para obtener la dirección de la página [js2.yimg.com](http://js2.yimg.com)
  - En el frame 27, el servidor DNS envía la respuesta, indicando que la dirección de la página está asociada a otros dominios como [a321.yimg.com.georedirector.akadns.net](http://a321.yimg.com.georedirector.akadns.net) y a [a321.x.a.yimg.com](http://a321.x.a.yimg.com). Además, le devuelve una serie de direcciones IPs asociadas a la página, como, 64.21.46.151, 64.21.46.134 y 64.21.46.150
- E. El cliente solicita la dirección IP puesto que, aunque ambas páginas pertenecen al mismo dominio, no pertenecen al mismo subdominio, lo que puede indicar que pertenecen a funciones o servidores diferentes.
- F. El sistema no necesita otra solicitud DNS porque como se observa en la línea de Host tanto del frame 42 como 47, ese nombre ya se había resuelto y es el mismo en las solicitudes de ambos frames.
- G. Solo los paquetes 139 y 143 son parte del paquete 160. Los paquetes 141 y 142 son respuestas independientes del paquete 160. Si los paquetes no llegan en orden, el protocolo TCP se encarga de ordenarlos, almacenando los en un buffer para ordenarlos por su número de secuencia.
- H. Para poder identificar a qué solicitud get asigna cada gráfico, el cliente utiliza el número de puerto de destino. En el frame 141 el puerto es 1226 y en el 143 es 1225

### Ejercicio 3



### III) Exercise Three

Open “Wireshark”, then use the “File” menu and the “Open” command to open the file “WS-Ex-03.pcap”. You should see 22 packets listed.

These packets represent two different requests for web pages. Packets 1-7 involve the request for the web page [www.yahoo.com](http://www.yahoo.com). Packets 8-22 involve the request for the web page [my.usf.edu](http://my.usf.edu).

- a) Compare the destination port in the TCP packet in frame 3 with the destination port in the TCP packet in frame 12. What difference do you see? What does this tell you about the difference in the two requests?

The following table compares the two requests for web pages. For example, row i) shows that frames 1-2 and frames 8-9 represent the DNS lookups for each of the web requests.

| Row  | www.yahoo.com frames | my.usf.com frames | Brief Explanation of Activity                                 |
|------|----------------------|-------------------|---|
| i)   | 1-2                  | 8-9               | DNS Request to find IP address for common name & DNS Response |
| ii)  | 3-5                  | 10-12             | Three-way handshake   |
| iii) | --                   | 13-20             |   |
| iv)  | 6                    | 21                | “Get” request for web page                                    |
| v)   | 7                    | 22                | First packet from web server with web page content.           |

- b) Explain what is happening in row “iii” above. Why are there no frames listed for yahoo in row “iii”?
- c) Look at the “Info” column on frame 6. It says: “GET / HTTP/1.1. What is the corresponding Info field for the [my.usf.com](http://my.usf.edu) web request (frame 21)? Why doesn’t it read the same as in frame 6?

| No. | Time        | Source         | Destination    | Protocol | Length | Info  |
|-----|-------------|----------------|----------------|----------|--------|---|
| 1   | 0.000000    | 192.168.1.3    | 192.168.1.1    | DNS      | 73     | Standard query 0xe493 A www.yahoo.com   |
| 2   | 0.011187    | 192.168.1.1    | 192.168.1.3    | DNS      | 514    | Standard query response 0xe493 A www.yahoo.com CNAME www.yahoo.akadns.net A 68.142.226.44 A 68.142.226.48 A 68.142.226.50 A 68.142.226.54 A 68.142.226.35 A 68.142.226.36 A 68.142.226.37 A 68.142.226.38 |
| 3   | 0.004097    | 192.168.1.3    | 68.142.226.44  | TCP      | 62     | 3904 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM  |
| 4   | 0.135169    | 68.142.226.44  | 192.168.1.3    | TCP      | 60     | 80 → 3904 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460   |
| 5   | 0.135287    | 192.168.1.3    | 68.142.226.44  | TCP      | 54     | 3904 → 80 [ACK] Seq=1 Ack=1 Win=0 Len=0   |
| 6   | 0.186585    | 192.168.1.3    | 68.142.226.44  | HTTP     | 1089   | GET / HTTP/1.1  |
| 7   | 0.208554    | 68.142.226.44  | 192.168.1.3    | TCP      | 1514   | 80 → 3904 [ACK] Seq=1 Ack=1036 Win=5535 Len=1408  |
| 8   | 0.5545849   | 192.168.1.3    | 192.168.1.1    | DNS      | 70     | Standard query 0x5baa A my.usf.edu  |
| 9   | 0.556867    | 192.168.1.1    | 192.168.1.3    | DNS      | 223    | Standard query response 0x5baa A my.usf.edu CNAME cluster6.acomp.usf.edu A 131.247.100.94 NS lead.acomp.usf.edu NS ritchie.acomp.usf.edu NS gold.acomp.usf.edu A 131.247.100.25 A 131.247.100.24          |
| 10  | 0.5599818   | 192.168.1.3    | 131.247.100.94 | TCP      | 62     | 3924 → 443 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM   |
| 11  | 0.5607788   | 131.247.100.94 | 192.168.1.3    | TCP      | 62     | 443 → 3924 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 SACK_PERM  |
| 12  | 0.5608785   | 192.168.1.3    | 131.247.100.94 | TCP      | 54     | 3924 → 443 [ACK] Seq=1 Ack=1 Win=5535 Len=0   |
| 13  | 0.5608813   | 192.168.1.3    | 131.247.100.94 | SSLV2    | 159    | Client Hello  |
| 14  | 0.56097828  | 131.247.100.94 | 192.168.1.3    | TCP      | 60     | 443 → 3924 [ACK] Seq=1 Ack=106 Win=49335 Len=0  |
| 15  | 0.56176946  | 131.247.100.94 | 192.168.1.3    | TLSv1    | 1514   | Server Hello, Certificate   |
| 16  | 0.56177817  | 131.247.100.94 | 192.168.1.3    | TLSv1    | 77     | Server Key Exchange, Server Hello Done  |
| 17  | 0.56177889  | 192.168.1.3    | 131.247.100.94 | TCP      | 54     | 3924 → 443 [ACK] Seq=106 Ack=1484 Win=5535 Len=0  |
| 18  | 0.56178455  | 192.168.1.3    | 131.247.100.94 | TLSv1    | 252    | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message  |
| 19  | 0.56232977  | 131.247.100.94 | 192.168.1.3    | TCP      | 60     | 443 → 3924 [ACK] Seq=1484 Ack=384 Win=49337 Len=0   |
| 20  | 0.56239114  | 131.247.100.94 | 192.168.1.3    | TLSv1    | 113    | Change Cipher Spec, Encrypted Handshake Message   |
| 21  | 0.56299838  | 192.168.1.3    | 131.247.100.94 | TLSv1    | 491    | Application Data  |
| 22  | 0.563151301 | 131.247.100.94 | 192.168.1.3    | TLSv1    | 491    | Application Data  |

- A. Se puede observar que el frame 3 y el 12 tienen puertos de destino diferentes. El frame 3 tiene el puerto de destino 80, lo que sugiere que se utiliza HTTP y el frame 12 tiene como puerto de destino 443, indicando que utiliza HTTPS. Esto da a entender que el frame 12 utiliza un protocolo cifrado y el frame 3 no.
- B. Al analizar la tabla se observa que entre los frames 13 y 20, se realiza el proceso para establecer una conexión segura para acceder al sitio [my.usf.com](http://my.usf.com). El hecho de que no aparezca nada en la fila III de la columna de [yahoo.com](http://yahoo.com) indica que no hubieron paquetes o frames dedicados a establecer una conexión segura con esta página.
- C. La razón por la cual la solicitud realizada por el frame 6 y el 21 se ven distintas es porque utilizan protocolos de comunicación diferentes. El frame 6 utiliza HTTP, mientras que el frame 21 utiliza TLSv1, el cuál es un protocolo seguro y se encarga de cifrar las solicitudes, por lo que estas no son visibles como sí lo son con HTTP.

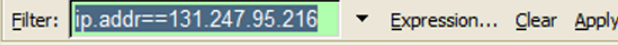
## Ejercicio 4

#### IV) Exercise Four

In this exercise, you are going to capture live traffic from your computer. Open up Wireshark and use the “Capture” menu to save live traffic. The Wireshark “QuickStart” guide distributed with these exercises contains more instructions on using Wireshark.

Start capturing data, visit a live web site ([os.ecci.ucr.ac.cr/ci0121](http://os.ecci.ucr.ac.cr/ci0121)) using your standard Internet browser, and stop capturing data.

If you have a large amount of network traffic, the relevant data may be hidden among a lot of broadcast messages. To focus on just the key frames, you can set a display filter like this.



For the IP number enter the IP number of your client machine. Type it as shown (ip.addr==your.ip.address) in the graphic above. Then click on “Apply”.

Using an approach similar to the approach in Exercise One, describe the set of frames that you captured.

- For this description think of this as a conversation – every discussion starts with a question and follows with an answer.
- For example, two of the frames will contain the DNS request for an IP address for the web site, and the DNS answer with the IP number.
- Remember that some answers may take several frames if they need to be reassembled from segmented packets.

|     |   |        |                |                 |        |      |  |
|-----|---|--------|----------------|-----------------|--------|------|--|
| 921 | 3 | 745504 | 192.168.100.3  | 239.255.255.258 | SSDP   | 485  | NOTIFY * HTTP/1.1  |
| 922 | 3 | 745853 | 192.168.100.3  | 239.255.255.258 | SSDP   | 581  | NOTIFY * HTTP/1.1  |
| 923 | 3 | 745853 | 192.168.100.3  | 239.255.255.258 | SSDP   | 485  | NOTIFY * HTTP/1.1  |
| 924 | 3 | 772205 | 23.223.184.112 | 192.168.100.62  | QUIC   | 68   | Protected Payload (PFB)  |
| 925 | 3 | 813752 | 192.168.100.62 | 200.91.75.5     | DNS    | 77   | Standard query 0x81c1 A os.ecci.ucr.ac.cr                                    |
| 926 | 3 | 824863 | 192.168.100.62 | 200.91.75.5     | DNS    | 77   | Standard query 0x81c1 A os.ecci.ucr.ac.cr                                    |
| 927 | 3 | 825183 | 192.168.100.62 | 200.91.75.5     | DNS    | 77   | Standard query 0x81c1 HTTPS os.ecci.ucr.ac.cr                                |
| 928 | 3 | 825156 | 200.91.75.5    | 192.168.100.62  | DNS    | 93   | Standard query response 0x81c1 A os.ecci.ucr.ac.cr A 163.178.104.62          |
| 929 | 3 | 829354 | 200.91.75.5    | 192.168.100.62  | DNS    | 93   | Standard query response 0x81c1 A os.ecci.ucr.ac.cr A 163.178.104.62          |
| 930 | 3 | 831287 | 192.168.100.62 | 52.168.112.67   | TLV1.3 | 4316 | Application Data   |
| 931 | 3 | 831833 | 192.168.100.62 | 52.168.112.67   | TLV1.3 | 5542 | Application Data   |
| 932 | 3 | 832281 | 200.91.75.5    | 192.168.100.62  | DNS    | 125  | Standard query response 0x81c1 HTTPS os.ecci.ucr.ac.cr SOA os.ecci.ucr.ac.cr |
| 933 | 3 | 832636 | 192.168.100.62 | 163.178.104.62  | TCP    | 68   | 51555 -> 443 [WIN] Seq=0 Win=65535 Len=0 MSG=1440 ID=256 SACK_PERM           |
| 934 | 3 | 832790 | 192.168.100.62 | 163.178.104.62  | TCP    | 68   | 51556 -> 443 [SYN] Seq=0 Win=65535 Len=0 MSG=1440 ID=256 SACK_PERM           |
| 935 | 3 | 835775 | 163.178.104.62 | 192.168.100.62  | TCP    | 68   | 443 -> 51555 [SYN, ACK] Seq=0 Ack=1 Win=2048 Len=0 MSG=1412 SACK_PERM ID=128 |
| 936 | 3 | 835817 | 163.178.104.62 | 192.168.100.62  | TCP    | 68   | 443 -> 51556 [SYN, ACK] Seq=0 Ack=1 Win=2048 Len=0 MSG=1412 SACK_PERM ID=128 |
| 937 | 3 | 835838 | 192.168.100.62 | 163.178.104.62  | TCP    | 54   | 51555 -> 443 [ACK] Seq=1 Ack=1 Win=65535 Len=0                               |
| 938 | 3 | 835863 | 192.168.100.62 | 163.178.104.62  | TCP    | 54   | 51556 -> 443 [ACK] Seq=1 Ack=1 Win=65535 Len=0                               |
| 939 | 3 | 836065 | 192.168.100.62 | 163.178.104.62  | TLV1.2 | 1877 | Client Hello [OS:os.ecci.ucr.ac.cr]  |
| 940 | 3 | 836268 | 192.168.100.62 | 163.178.104.62  | TLV1.2 | 1845 | Client Hello [OS:os.ecci.ucr.ac.cr]  |
| 941 | 3 | 840677 | 163.178.104.62 | 192.168.100.62  | TCP    | 68   | 443 -> 51555 [ACK] Seq=1 Ack=1413 Win=21218 Len=0                            |
| 942 | 3 | 840804 | 163.178.104.62 | 192.168.100.62  | TCP    | 68   | 443 -> 51555 [ACK] Seq=1 Ack=1824 Win=34944 Len=0                            |
| 943 | 3 | 840925 | 163.178.104.62 | 192.168.100.62  | TCP    | 68   | 443 -> 51556 [ACK] Seq=1 Ack=1413 Win=21218 Len=0                            |
| 944 | 3 | 840925 | 163.178.104.62 | 192.168.100.62  | TCP    | 68   | 443 -> 51556 [ACK] Seq=1 Ack=1792 Win=34944 Len=0                            |
| 945 | 3 | 844736 | 163.178.104.62 | 192.168.100.62  | TLV1.2 | 1466 | Server Hello   |
| 946 | 3 | 844736 | 163.178.104.62 | 192.168.100.62  | TLV1.2 | 1466 | Certificate  |
| 947 | 3 | 844736 | 163.178.104.62 | 192.168.100.62  | TLV1.2 | 224  | Server Key Exchange, Server Hello Done                                       |
| 948 | 3 | 844799 | 163.178.104.62 | 192.168.100.62  | TLV1.2 | 1466 | Server Hello   |
| 949 | 3 | 844799 | 163.178.104.62 | 192.168.100.62  | TLV1.2 | 1466 | Certificate  |
| 950 | 3 | 844799 | 163.178.104.62 | 192.168.100.62  | TLV1.2 | 224  | Server Key Exchange, Server Hello Done                                       |
| 951 | 3 | 844817 | 192.168.100.62 | 163.178.104.62  | TCP    | 54   | 51555 -> 443 [ACK] Seq=1824 Ack=2095 Win=65535 Len=0                         |
| 952 | 3 | 844851 | 192.168.100.62 | 163.178.104.62  | TCP    | 54   | 51556 -> 443 [ACK] Seq=1792 Ack=2095 Win=65535 Len=0                         |
| 953 | 3 | 844857 | 192.168.100.3  | 239.255.255.258 | SSDP   | 421  | NOTIFY * HTTP/1.1  |
| 954 | 3 | 844867 | 192.168.100.62 | 163.178.104.62  | TLV1.2 | 188  | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message         |
| 955 | 3 | 846500 | 192.168.100.3  | 239.255.255.258 | SSDP   | 428  | NOTIFY * HTTP/1.1  |
| 956 | 3 | 846582 | 192.168.100.3  | 239.255.255.258 | SSDP   | 473  | NOTIFY * HTTP/1.1  |
| 957 | 3 | 846796 | 192.168.100.3  | 192.168.100.255 | SSDP   | 428  | NOTIFY * HTTP/1.1  |
| 958 | 3 | 846796 | 192.168.100.3  | 192.168.100.255 | SSDP   | 438  | NOTIFY * HTTP/1.1  |
| 959 | 3 | 846750 | 192.168.100.62 | 163.178.104.62  | TLV1.2 | 188  | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message         |
| 960 | 3 | 847029 | 192.168.100.3  | 192.168.100.255 | SSDP   | 473  | NOTIFY * HTTP/1.1  |
| 961 | 3 | 847043 | 192.168.100.3  | 239.255.255.258 | SSDP   | 407  | NOTIFY * HTTP/1.1  |
| 962 | 3 | 847043 | 192.168.100.3  | 239.255.255.258 | SSDP   | 489  | NOTIFY * HTTP/1.1  |
| 963 | 3 | 847262 | 192.168.100.3  | 239.255.255.258 | SSDP   | 485  | NOTIFY * HTTP/1.1  |
| 964 | 3 | 847269 | 192.168.100.3  | 239.255.255.258 | SSDP   | 561  | NOTIFY * HTTP/1.1  |
| 965 | 3 | 847398 | 192.168.100.3  | 239.255.255.258 | SSDP   | 495  | NOTIFY * HTTP/1.1  |
| 966 | 3 | 850805 | 163.178.104.62 | 192.168.100.62  | TLV1.2 | 328  | New Session Ticket, Change Cipher Spec, Encrypted Handshake Message          |
| 967 | 3 | 851282 | 163.178.104.62 | 192.168.100.62  | TLV1.2 | 328  | New Session Ticket, Change Cipher Spec, Encrypted Handshake Message          |

En la imagen anterior se observan los paquetes o frames obtenidos al intentar a ingresar a la página [os.ecci.ucr.ac.cr/ci0121](http://os.ecci.ucr.ac.cr/ci0121)

Entre los frames 925 y 929 ocurre la búsqueda del nombre DNS de la página. Cómo el cliente quiere entrar a [os.ecci.ucr.ac.cr](http://os.ecci.ucr.ac.cr), consulta su dirección IP y el servidor DNS responde indicando que esta es 163.178.104.62

Entre los frames 933 y 938 se establece una conexión TCP con el servidor. Para esta conexión se realiza un three-way handshake. PARA esto el cliente envía paquetes al puerto 443 del servidor, el servidor responde con un SYN y un ACK y finalmente el cliente responde con un ACK.

Por último, entre el frame 939 y 959 se empieza el protocolo TLSv1.2, para cifrar la comunicación con el servidor. Para esto, el cliente envía un Hello al servidor, este responde con un Hello de vuelta, envía su certificado y se realiza un intercambio de claves para poder cifrar la comunicación.