

# Network Final Project

## Part 1:

- All comments on the relevant functions are recorded in the file itself.

When executing `recv_udp.c (recv)`, a datagram socket is been created and listening to the port for incoming requests to come.

When executing `send_udp.c (send)`, a datagram socket is been created and sends a datagram to the receive socket (`recv`).

In the following picture we can see how the message is sent and received in the UDP protocol:

```
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./recv
RECV_UDP:
Local socket is:: ip= 0.0.0.0, port= 13107
recv_udp: :
Packet from:: ip= 127.0.0.1, port= 60646
Got data ::<41568>
[]

ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./send 127.0.0.1
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$
```

We can also see the sending of the message in Wireshark:

4	0.115858961	127.0.0.53	127.0.0.1	DNS	210 Standard query response 0x1d2a AAAA mobile.events.data.microsoft.com C
5	31.176113770	127.0.0.1	127.0.0.1	UDP	66 59116 → 13107 Len=24
6	125.587254767	127.0.0.1	127.0.0.53	DNS	100 Standard query 0xeec1 AAAA connectivity-check.ubuntu.com OPT
7	125.65553078	127.0.0.53	127.0.0.1	DNS	100 Standard query response 0xeec1 AAAA connectivity-check.ubuntu.com OPT

▶ Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface lo, id 0  
 ▶ Ethernet II, Src: 00:00:00\_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
 ▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1  
 ▶ User Datagram Protocol, Src Port: 59116, Dst Port: 13107  
 ▶ Data (24 bytes)

```
0000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 00  .....E.
0010  00 34 f1 29 40 00 40 11 4b 8d 7f 00 00 01 7f 00  .4.)@.K.....
0020  00 01 e6 ec 33 33 00 20 fe 33 3c 24 96 28 fe 7f  ....33. 3<$.(..
0030  00 00 00 00 a2 60 00 00 00 00 3e c2 b6 ee aa 7f  .....>.....
0040  00 00
```

**When we run sink, gateway, source (of course an IP must be given to the gateway and source) we will get:**

```
OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS

send: 23      0.919667 > 0.5 ,the data gram has been sent!
              sleep 1 second...
send: 24      Received data gram from source
              The random number is: 0.779248
              0.779248 > 0.5 ,the data gram has been sent!
              sleep 1 second...
send: 25      Received data gram from source
              The random number is: 0.612073
              0.612073 > 0.5 ,the data gram has been sent!
              sleep 1 second...
send: 26      Received data gram from source
              The random number is: 0.227663
              0.227663 < 0.5 ,the data gram has been blocked!
              sleep 1 second...
send: 27      Received data gram from source
              The random number is: 0.271271
              0.271271 < 0.5 ,the data gram has been blocked!
              sleep 1 second...
send: 28      Received data gram from source
              The random number is: 0.497684
              0.497684 < 0.5 ,the data gram has been blocked!
              sleep 1 second...
send: 29      Received data gram from source
              The random number is: 0.131784
              0.131784 < 0.5 ,the data gram has been blocked!
              sleep 1 second...
send: 30      Received data gram from source
              The random number is: 0.131784
              0.131784 < 0.5 ,the data gram has been blocked!
              sleep 1 second...
send: 31      Received data gram from source
              The random number is: 0.131784
              0.131784 < 0.5 ,the data gram has been blocked!
              sleep 1 second...
send: 32      Received data gram from source
              The random number is: 0.131784
              0.131784 < 0.5 ,the data gram has been blocked!
              sleep 1 second...
send: 33      Received data gram from source
              The random number is: 0.131784
              0.131784 < 0.5 ,the data gram has been blocked!
              sleep 1 second...
send: 34      Received data gram from source
              The random number is: 0.131784
              0.131784 < 0.5 ,the data gram has been blocked!
              sleep 1 second...
send: 35      Received data gram from source
              The random number is: 0.131784
              0.131784 < 0.5 ,the data gram has been blocked!
              sleep 1 second...

Got Data: 19
Recv data:
Packet from: ip= 127.0.0.1, port= 45005
Got Data: 21

Recv data:
Packet from: ip= 127.0.0.1, port= 45005
Got Data: 23

Recv data:
Packet from: ip= 127.0.0.1, port= 45005
Got Data: 28

Recv data:
Packet from: ip= 127.0.0.1, port= 45005
Got Data: 29

Recv data:
Packet from: ip= 127.0.0.1, port= 45005
Got Data: 30

Recv data:
Packet from: ip= 127.0.0.1, port= 45005
Got Data: 31
```

The source sends a datagram to gateway every single second.

At the gateway, a number between 0 and 1 is drawn, and in case the result is greater than 0.5, the datagram is sent to sink, otherwise the datagram is blocked.

**You can also see the sending of the datagrams in Wireshark:**

[illegible]

In the wireshark recording you can see that among every few packets that arrive at port 13107 there are some that arrive at port 13108, and this is exactly what we wanted to happen - that some of the packets would arrive at port  $p+1$  and some at port  $p$  according to the result of the random function.

# TCP programming:

## Part a:

When executing net\_server.c and then net\_client.c at the first time, we get this result:

```
OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./client
Client is alive and establishing socket connection.
Error establishing communications: Connection refused
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$

ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./server
```

The server is waiting and listening, but the client can't create connection because it can't allocate address to himself.

This can be explained because of 2 reasons:

1. net\_server.c (9999) and net\_client.c (1337) are not working with the same port.
2. The IP address of net\_client.c is not defined.

for this we need to execute nslookup.c with hostname(localhost) of my computer:

```
OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./lookup localhost
Address for localhost is 127.0.0.1
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$
```

Of course we will change the IP address before compilation:

```
#include <netinet/in.h>
#define SIM_LENGTH 10
#define IP_ADDRESS "127.0.0.1"
#define PORT 1337
```

Now we run net\_server.c and net\_client.c again and get this:

```
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./server
Server has written 1 to socket.
Server has written 2 to socket.
Server has written 3 to socket.
Server has written 4 to socket.
Server has written 5 to socket.
Server has written 6 to socket.
Server has written 7 to socket.
Server has written 8 to socket.
Server has written 9 to socket.
Server has written 10 to socket.
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$

ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./client
Client is alive and establishing socket connection.
Client has received 1 from socket.
Client has received 2 from socket.
Client has received 3 from socket.
Client has received 4 from socket.
Client has received 5 from socket.
Client has received 6 from socket.
Client has received 7 from socket.
Client has received 8 from socket.
Client has received 9 from socket.
Client has received 10 from socket.
Exiting now.
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$
```

We will be able to see the sending in wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	38302 → 9999 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=17...
2	0.000016393	127.0.0.1	127.0.0.1	TCP	74	9999 → 38302 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=...
3	0.000030046	127.0.0.1	127.0.0.1	TCP	66	38302 → 9999 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1773162737 TSecr=1...
4	0.000102812	127.0.0.1	127.0.0.1	TCP	70	9999 → 38302 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=4 TSval=1773162738 TS...
5	0.000119909	127.0.0.1	127.0.0.1	TCP	66	38302 → 9999 [ACK] Seq=1 Ack=5 Win=65536 Len=0 TSval=1773162738 TSecr=1...
6	0.000283548	127.0.0.1	127.0.0.1	TCP	70	9999 → 38302 [PSH, ACK] Seq=5 Ack=1 Win=65536 Len=4 TSval=1773162738 TS...
7	0.000289102	127.0.0.1	127.0.0.1	TCP	66	38302 → 9999 [ACK] Seq=1 Ack=9 Win=65536 Len=0 TSval=1773162738 TSecr=1...
8	0.000311675	127.0.0.1	127.0.0.1	TCP	70	9999 → 38302 [PSH, ACK] Seq=9 Ack=1 Win=65536 Len=4 TSval=1773162738 TS...
9	0.000319765	127.0.0.1	127.0.0.1	TCP	66	38302 → 9999 [ACK] Seq=1 Ack=13 Win=65536 Len=0 TSval=1773162738 TSecr=...
10	0.000327496	127.0.0.1	127.0.0.1	TCP	78	9999 → 38302 [PSH, ACK] Seq=13 Ack=1 Win=65536 Len=12 TSval=1773162738 TS...
11	0.000333078	127.0.0.1	127.0.0.1	TCP	66	38302 → 9999 [ACK] Seq=1 Ack=25 Win=65536 Len=0 TSval=1773162738 TSecr=...
12	0.000336606	127.0.0.1	127.0.0.1	TCP	74	9999 → 38302 [PSH, ACK] Seq=25 Ack=1 Win=65536 Len=8 TSval=1773162738 TS...
13	0.000340643	127.0.0.1	127.0.0.1	TCP	66	38302 → 9999 [ACK] Seq=1 Ack=33 Win=65536 Len=0 TSval=1773162738 TSecr=...
14	0.000344014	127.0.0.1	127.0.0.1	TCP	74	9999 → 38302 [PSH, ACK] Seq=33 Ack=1 Win=65536 Len=8 TSval=1773162738 TS...
15	0.000347315	127.0.0.1	127.0.0.1	TCP	66	38302 → 9999 [FIN, ACK] Seq=41 Ack=1 Win=65536 Len=0 TSval=1773162738 TS...
16	0.000348148	127.0.0.1	127.0.0.1	TCP	66	38302 → 9999 [ACK] Seq=1 Ack=41 Win=65536 Len=0 TSval=1773162738 TSecr=...
17	0.000384527	127.0.0.1	127.0.0.1	TCP	66	38302 → 9999 [FIN, ACK] Seq=1 Ack=42 Win=65536 Len=0 TSval=1773162738 TS...
18	0.000389634	127.0.0.1	127.0.0.1	TCP	66	9999 → 38302 [ACK] Seq=42 Ack=2 Win=65536 Len=0 TSval=1773162738 TSecr=...
19	36.783499005	127.0.0.1	127.0.0.53	DNS	100	Standard query 0xef43 AAAA connectivity-check.ubuntu.com OPT
20	36.913229484	127.0.0.53	127.0.0.1	DNS	100	Standard query response 0xef43 AAAA connectivity-check.ubuntu.com OPT
21	36.913461609	127.0.0.1	127.0.0.53	DNS	105	Standard query 0x702c AAAA connectivity-check.ubuntu.com.Home OPT
22	37.101256218	127.0.0.53	127.0.0.1	DNS	105	Standard query response 0x702c No such name AAAA connectivity-check.ubu...

You can see the handshake of the protocol (SYN, SYN ACK, ACK) at the top of the screen.

After going through the net\_server.c and net\_client.c programs and writing notes on them, I ran them according to the IP of my computer that I discovered using the nslookup program in the terminal and I saw that one program sends information and the other program receives the same information as requested. (In the case of these programs, we sent the number 1-10 according to the order and after all the information is received, the programs close themselves).

Now let's try run net\_client when net\_server is not running:

```
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./client
Client is alive and establishing socket connection.
Error establishing communications: Connection refused
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$
```

And the Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	38304 → 9999 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=17...
2	0.000010672	127.0.0.1	127.0.0.1	TCP	54	9999 → 38304 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
3	129.947887202	127.0.0.1	127.0.0.53	DNS	99	Standard query 0xc75c A contile.services.mozilla.com OPT
4	129.948200465	127.0.0.1	127.0.0.53	DNS	99	Standard query 0xa421 AAAA contile.services.mozilla.com OPT
5	129.949058995	127.0.0.1	127.0.0.53	DNS	99	Standard query 0x61be A contile.services.mozilla.com OPT

Here you can see an error (in the red line the Ack return with reset in TCPDUMP I will show what is mean..), so we tried to run a client without a port on the other side waiting for connection and therefore we could not run the program.

We can use TCPDUMP to check our assumption, and we can see that the ack is return has the flag - R that mean:

```
ariel@ariel-VirtualBox:~/Desktop$ sudo tcpdump -i any
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
23:48:17.800268 IP localhost.38318 > localhost.9999: Flags [S], seq 1154102444, win 65495, options [mss 65495,sackOK,TS val 1775409760 ecr 0,nop,wscale 7], length 0
23:48:17.800275 IP localhost.9999 > localhost.38318: Flags [R.], seq 0, ack 1154102445, win 0, length 0
```

Flags		
TCP Flag	tcpdump Flag	Meaning
RESET	R	Reset, indication of <u>immediate abort of conn.</u>

After the previous run, I looked at the program nslookup.c which gets an address and returns the corresponding IP address, wrote notes on it and ran it several times.

With the help of this program, I understood how the program net\_client.c can be upgraded so that instead of having to manually enter the address for the program itself, anyone who uses the program can, together with running the program - enter their address, and the system call getaddrinfo() will locate the appropriate IP address by itself.

Here are some examples:

```
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./lookup walla.com
Address for walla.com is 65.9.112.9
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./lookup google.com
Address for google.com is 142.250.201.14
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./lookup ynet.com
Address for ynet.com is 123.60.211.213
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./lookup walla.co.il
Address for walla.co.il is 13.226.2.15
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$
```

Net\_client.c

According to the last section we takes the hostname as a command-line argument.

```
C:\net_client.c
40 struct sockaddr_in* saddr;
41
42 if (argc != 2) {
43     perror("Usage: hostnamelookup <hostname>\n");
44     exit(1);
45 }
46
47 hostname = argv[1]; //Taking the argument that comes after the running command.
48
49 //The sys call getaddrinfo() getting the IP address of the name we insert it..
50 if (0 != getaddrinfo(hostname, NULL, NULL, &res)) {
51     fprintf(stderr, "Error in resolving hostname %s\n", hostname);
52     exit(1);
53 }
54
55 saddr = (struct sockaddr_in*)res->ai_addr;
56 hostaddr = inet_ntoa(saddr->sin_addr);
```

OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./client localhost
Client is alive and establishing socket connection.
Address for localhost is 127.0.0.1
Client has received 1 from socket.
Client has received 2 from socket.
Client has received 3 from socket.
Client has received 4 from socket.
Client has received 5 from socket.
Client has received 6 from socket.
Client has received 7 from socket.
Client has received 8 from socket.
Client has received 9 from socket.
Client has received 10 from socket.
Exiting now.
```

```
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./server
Server has written 1 to socket.
Server has written 2 to socket.
Server has written 3 to socket.
Server has written 4 to socket.
Server has written 5 to socket.
Server has written 6 to socket.
Server has written 7 to socket.
Server has written 8 to socket.
Server has written 9 to socket.
Server has written 10 to socket.
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$
```

We use the code from nslookup.c to resolve this hostname to a result of type struct sockaddr, then use the result to connect to this address rather than the hard-coded IP\_ADDRESS.



## Part b:

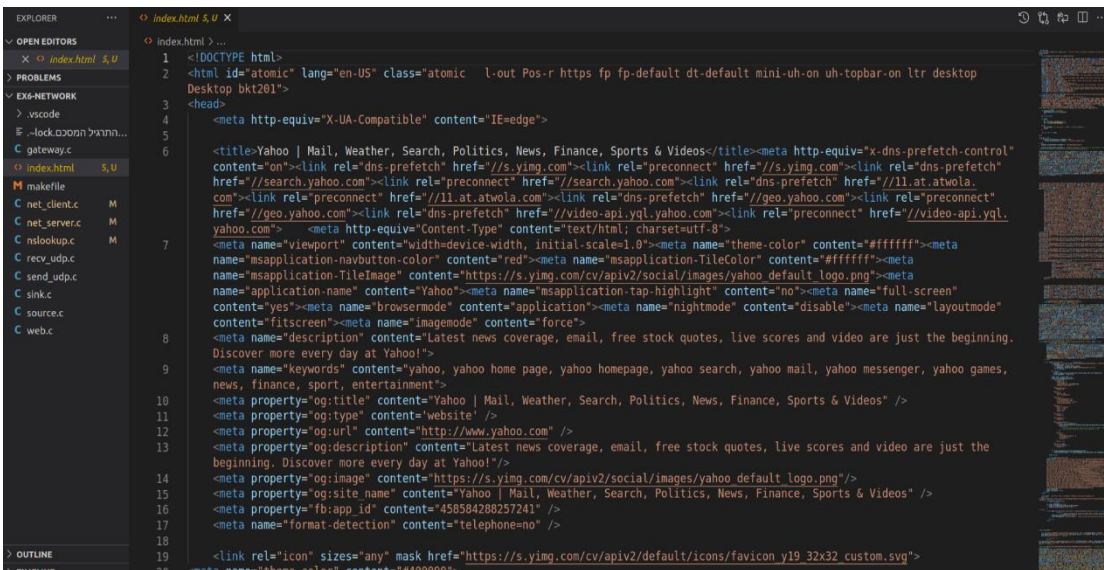
First lets try to use wget command, we will use it in the command line In order to receive information (html file) from the website yahoo.com:

```
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ wget http://www.yahoo.com
--2022-08-03 13:23:29-- http://www.yahoo.com/
Resolving www.yahoo.com (www.yahoo.com)... 87.248.100.216, 87.248.100.215, 2a00:1288:110:c305::1:8000, ...
Connecting to www.yahoo.com (www.yahoo.com)|87.248.100.216|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.yahoo.com/ [following]
--2022-08-03 13:23:29-- https://www.yahoo.com/
Connecting to www.yahoo.com (www.yahoo.com)|87.248.100.216|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html           [  <=> ] 726.14K  716KB/s  in 1.0s

2022-08-03 13:23:31 (716 KB/s) - 'index.html' saved [743570]

ariel@ariel-VirtualBox:~/Desktop/Ex6-network$
```



wget() is a free utility for non-interactive download of files from the Web. It supports HTTP, HTTPS, and FTP protocols, as well as retrieval through HTTP proxies.

In this part we were asked to write a program that simulates the wget() program.

wget() is a program that receives a URL and writes the content from the URL into a new file.

We will build our program so that it receives as an argument in the terminal a URL address and prints the content it received from the website to STDOUT.

The program works so that after we have located the host at the address entered in the terminal, a socket will be opened through which we will transfer the information. (the "message").

Since we are dealing with the web here, we will have to send an HTTP request to receive the information from the address.

After we have sent the request to the host and received confirmation, we will start receiving information using the read function that reads the information transmitted through the socket.

## Below is an example of the run on the address <http://www.yahoo.com>

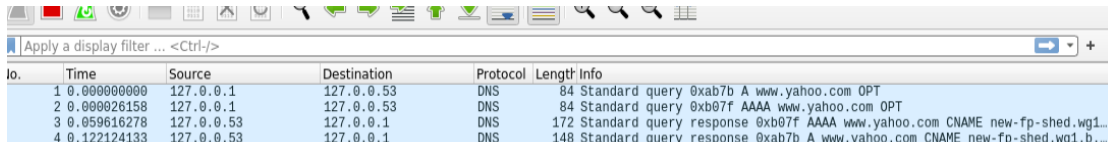
```
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./web http://www.yahoo.com
87.248.100.216
Client is alive and establishing socket connection.
message has sent!

HTTP/1.0 301 Moved Permanently
Date: Thu, 04 Aug 2022 09:12:43 GMT
Server: ATS
Cache-Control: no-store, no-cache
Content-Type: text/html
Content-Language: en
Connection: keep-alive
Content-Security-Policy: frame-ancestors 'self' https://*.builtbygirls.com https://*.rivals.com https://*.engadget.com http
s://*.intheknow.com https://*.autoblog.com https://*.techcrunch.com https://*.yahoo.com https://*.aol.com https://*.huffing
tonpost.com https://*.oath.com https://*.search.yahoo.com https://*.search.aol.com https://*.search.huffpost.com https://*.
onsearch.com https://*.verizonmedia.com https://*.publishing.oath.com https://*.autoblog.com; sandbox allow-forms allow-sa
me-origin allow-scripts allow-popups allow-popups-to-escape-sandbox allow-presentation; report-uri https://csp.yahoo.com/be
acon/csp?src=ats&site=frontpage&region=US&lang=en-US&device=&yrid=1977b59hen3cd&partner=;
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; report="https://csp.yahoo.com/beacon/csp?src=fp-hpkk-www"
Location: https://www.yahoo.com/
Content-Length: 8

redirectExiting now.
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$
```

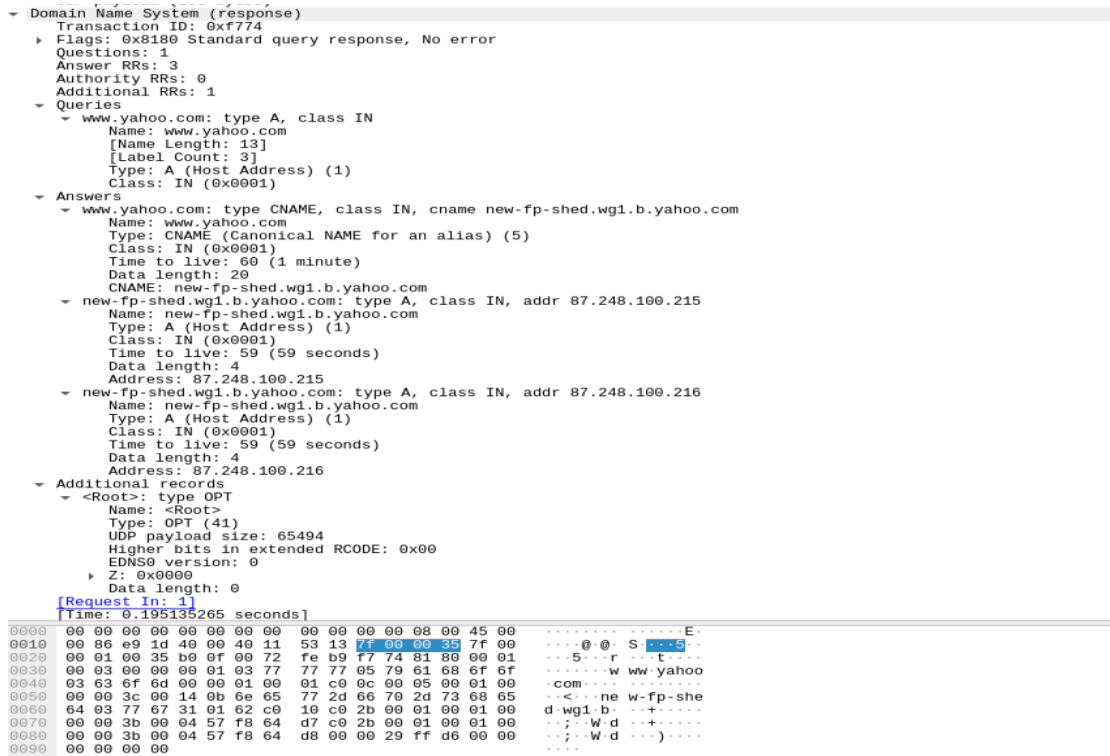
we can see the header that comes back from the site surrounded in yellow.

## We can see the wireshark record:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.53	DNS	84	Standard query 0xab7b A www.yahoo.com OPT
2	0.000026158	127.0.0.1	127.0.0.53	DNS	84	Standard query 0xb07f AAAA www.yahoo.com OPT
3	0.059616278	127.0.0.53	127.0.0.1	DNS	172	Standard query response 0xb07f AAAA www.yahoo.com CNAME new-fp-shed.wg1...
4	0.122124133	127.0.0.53	127.0.0.1	DNS	148	Standard query response 0xab7b A www.yahoo.com CNAME new-fp-shed.wg1.b...

## Documentation of the request in the Wireshark recording:



```
Domain Name System (response)
  Transaction ID: 0xf774
  Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 3
  Authority RRs: 0
  Additional RRs: 1
  Queries
    www.yahoo.com: type A, class IN
      Name: www.yahoo.com
      [Name Length: 13]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
  Answers
    www.yahoo.com: type CNAME, class IN, cname new-fp-shed.wg1.b.yahoo.com
      Name: www.yahoo.com
      Type: CNAME (Canonical NAME for an alias) (5)
      Class: IN (0x0001)
      Time to live: 60 (1 minute)
      Data length: 20
      CNAME: new-fp-shed.wg1.b.yahoo.com
    new-fp-shed.wg1.b.yahoo.com: type A, class IN, addr 87.248.100.215
      Name: new-fp-shed.wg1.b.yahoo.com
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 59 (59 seconds)
      Data length: 4
      Address: 87.248.100.215
    new-fp-shed.wg1.b.yahoo.com: type A, class IN, addr 87.248.100.216
      Name: new-fp-shed.wg1.b.yahoo.com
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 59 (59 seconds)
      Data length: 4
      Address: 87.248.100.216
  Additional records
    <Root>: type OPT
      Name: <Root>
      Type: OPT (41)
      UDP payload size: 65494
      Higher bits in extended RCODE: 0x00
      Z: 0x0000
      Data length: 0
  [Request In: 1]
  [Time: 0.195135265 seconds]
```

Time	Source	Destination	Protocol	Length	Info
0.000000000	127.0.0.1	127.0.0.53	DNS	84	Standard query 0xab7b A www.yahoo.com OPT
0.000026158	127.0.0.1	127.0.0.53	DNS	84	Standard query 0xb07f AAAA www.yahoo.com OPT
0.059616278	127.0.0.53	127.0.0.1	DNS	172	Standard query response 0xb07f AAAA www.yahoo.com CNAME new-fp-shed.wg1...
0.122124133	127.0.0.53	127.0.0.1	DNS	148	Standard query response 0xab7b A www.yahoo.com CNAME new-fp-shed.wg1.b...

## In the same way the request can be forwarded with PORT:

```
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./web http://www.yahoo.com 80
87.248.100.215
Client is alive and establishing socket connection.
message has sent!

HTTP/1.0 301 Moved Permanently
Date: Thu, 04 Aug 2022 09:46:30 GMT
Server: ATS
Cache-Control: no-store, no-cache
Content-Type: text/html
Content-Language: en
Connection: keep-alive
Content-Security-Policy: frame-ancestors 'self' https://*.builtbygirls.com https://*.rivals.com https://*.engadget.com https://*.intheknow.com https://*.autoblog.com https://*.techcrunch.com https://*.yahoo.com https://*.aol.com https://*.huffingtonpost.com https://*.oath.com https://*.search.yahoo.com https://*.search.aol.com https://*.search.huffpost.com https://*.onesearch.com https://*.verizonmedia.com https://*.publishing.oath.com https://*.autoblog.com; sandbox allow-forms allow-same-origin allow-scripts allow-popups allow-popups-to-escape-sandbox allow-presentation; report-uri https://csp.yahoo.com/beacon/csp?src=ats&site=frontpage&region=US&lang=en-US&device=&yrid=4viuas9hen5bm&partner=;
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; report="https://csp.yahoo.com/beacon/csp?src=fp-hpkip-www"
Location: https://www.yahoo.com/
Content-Length: 8

redirectExiting now.

ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./web http://www.yahoo.com:80
87.248.100.216
Client is alive and establishing socket connection.
message has sent!

HTTP/1.0 301 Moved Permanently
Date: Thu, 04 Aug 2022 09:48:22 GMT
Server: ATS
Cache-Control: no-store, no-cache
Content-Type: text/html
Content-Language: en
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
Location: https://www.yahoo.com/
Content-Length: 8

redirectExiting now.
```

## And with path:

```
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./web http://www.yahoo.com/80/path
87.248.100.216
Client is alive and establishing socket connection.
message has sent!

HTTP/1.0 301 Moved Permanently
Date: Thu, 04 Aug 2022 09:50:45 GMT
Server: ATS
Cache-Control: no-store, no-cache
Content-Type: text/html
Content-Language: en
Connection: keep-alive
Content-Security-Policy: frame-ancestors 'self' https://*.builtbygirls.com https://*.rivals.com https://*.engadget.com https://*.intheknow.com https://*.autoblog.com https://*.techcrunch.com https://*.yahoo.com https://*.aol.com https://*.huffingtonpost.com https://*.oath.com https://*.search.yahoo.com https://*.search.aol.com https://*.search.huffpost.com https://*.onesearch.com https://*.verizonmedia.com https://*.publishing.oath.com https://*.autoblog.com; sandbox allow-forms allow-same-origin allow-scripts allow-popups allow-popups-to-escape-sandbox allow-presentation; report-uri https://csp.yahoo.com/beacon/csp?src=ats&site=frontpage&region=US&lang=en-US&device=&yrid=05oq2mhhen5jl&partner=;
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; report="https://csp.yahoo.com/beacon/csp?src=fp-hpkip-www"
Location: https://www.yahoo.com/80/path
Content-Length: 8

redirectExiting now.
```

When executing web client with `http://www.yahoo.com/does-not-exist` the result is:

```
ariel@ariel-VirtualBox:~/Desktop/Ex6-network$ ./web http://www.yahoo.com/does-not-exist
87.248.100.215
Client is alive and establishing socket connection.
message has sent!

HTTP/1.0 301 Moved Permanently
Date: Thu, 04 Aug 2022 09:56:11 GMT
Server: ATS
Cache-Control: no-store, no-cache
Content-Type: text/html
Content-Language: en
Connection: keep-alive
Content-Security-Policy: frame-ancestors 'self' https://*.builtbygirls.com https://*.rivals.com https://*.engadget.com https://*.intheknow.com https://*.autoblog.com https://*.techcrunch.com https://*.yahoo.com https://*.aol.com https://*.huffingtonpost.com https://*.oath.com https://*.search.yahoo.com https://*.search.aol.com https://*.search.huffpost.com https://*.onesearch.com https://*.verizonmedia.com https://*.publishing.oath.com https://*.autoblog.com; sandbox allow-forms allow-same-origin allow-scripts allow-popups allow-popups-to-escape-sandbox allow-presentation; report-uri https://csp.yahoo.com/beacon/csp?src=ats&site=frontpage&region=US&lang=en-US&device=&yrid=208mucihen5tr&partner=;
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; report="https://csp.yahoo.com/beacon/csp?src=fp-hpkip-www"
Location: https://www.yahoo.com/does-not-exist
Content-Length: 8

redirectExiting now.
```

We received the same header as in the previous lectures.