

Práctico 1: Capa de aplicación y transporte

Objetivos:

- Comprender el funcionamiento de los protocolos de la capa de aplicación HTTP y FTP;
- Comprender el concepto de socket;
- Diferenciar entre los protocolos de capa 4: UDP y TCP;
- Entender performance de red mediante análisis de UDP y TCP (capacidad, delay, jitter);
- Aprender comandos básicos de la consola de linux;
- Aprender analizador de protocolos Wireshark/tcpdump.

Bibliografía y links de ayuda

- Computer Networking de Kurose y Ross, Capítulo 1 y 2;
- Computer Networking de Kurose y Ross, Capítulo 3;
- Douglas E. Comer Capítulo 10 y 11;
- https://www.wireshark.org/docs/wsug_html_chunked
- <https://iperf.fr/iperf-doc.php>
- <http://flask.pocoo.org/docs/0.12/>
- <http://arpitbhayani.me/techie/rest-the-hard-way-with-netcat.html>
- <http://crok-linkblog.homelinux.com/links-misc/how-to-use-iperf-properly-additions-to-the-tcp-throughput-post/>
- <https://www.sd-wan-experts.com/blog/iperf-bandwidth-testing/>
-

NOTA 1:

Los prácticos se deben ejecutar en Linux, idealmente Ubuntu 16.04 LTS. Aquellos que tengan Windows pueden instalar una máquina virtual para realizar los prácticos desde allí.

NOTA 2:

Para los ejercicios que involucren situaciones cliente/servidor (en especial EJ3 y EJ4) se sugiere, en caso de ser posible, no usar una red cableada entre los hosts (ej: usar wifi o internet, entre ellos).

NOTA 3:

Se requieren los siguientes paquetes para la ejecución del práctico

- `sudo apt-get install python-pip3`
- `sudo pip3 install flask`
- `sudo apt-get install iperf3`
- `sudo apt-get install netcat`
- `sudo apt-get install wireshark tcpdump`

Ejercicio 1: HTTP - Creación y utilización de una API

1.1.- Crear e iniciar un webserver básico con Python Flask en la máquina virtual. Para obtener su IP, busquen con el comando “ifconfig” o “ip a”.

La salida de ifconfig:

```
wlp2s0    Link encap:Ethernet  HWaddr fc:f8:ae:9a:c3:f6
          inet addr:192.168.0.4  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::1d9:6c7c:8232:3de4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13422 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6783 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14078283 (14.0 MB)  TX bytes:1422672 (1.4 MB)
```

La salida de ip:

```
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group de
fault qlen 1000
    link/ether fc:f8:ae:9a:c3:f6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.4/24 brd 192.168.0.255 scope global dynamic wlp2s0
        valid_lft 6557sec preferred_lft 6557sec
    inet6 fe80::1d9:6c7c:8232:3de4/64 scope link
        valid_lft forever preferred_lft forever
```

1.2.- Implementar un método GET simple, un método GET con parámetros y un método POST

1.3.- Establecer una conexión con el servidor web utilizando netcat desde el host. (Recordar que pueden obtener ayuda con “man netcat” o “nc -h” para entender mejor la herramienta.) Probar los diferentes métodos implementados con curl o alternativas. No utilizar una herramienta con interfaz gráfica.

1.4.- Analizar el tráfico con el wireshark ubicando peticiones y respuestas. Identificar y saber explicar los campos de la cabecera HTTP.

1.5.- Preguntas relacionadas:

¿Cuáles son otros métodos de HTTP?

¿Qué es JSON? ¿Cuáles son ventajas de devolver información en dicho formato?

¿Qué es REST API?

1.1.-

Se creo un webserver básico en Python, el cual recibe pedidos HTTP del tipo GET, GET con parámetros y POST. La IP del servidor se obtuvo con el comando **ifconfig** y el puerto utilizado fue el 5001.

```
wlp3s0    Link encap:Ethernet  direcciónHW 3c:a0:67:3f:0e:70
          Direc. inet:192.168.0.27  Difus.:192.168.0.255  Másc:255.255.255.0
          Dirección inet6: fe80::faf9:c42c:5c95:f7dc/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:94208 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:56591 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:127434764 (127.4 MB)  TX bytes:6483925 (6.4 MB)
```

1.2.-

Para implementar se definieron 3 funciones, cada una con una path definida para cada método:

GET: /

GET con parametros: /autenticacion

POST: /suma

El método GET simplemente devuelve un mensaje de bienvenida.

El método GET con parámetros pide un nombre de usuario como parámetro (user) y la respuesta es un echo del mismo nombre.

El método POST resuelve una suma de 2 números pasados como parámetro (num1 y num2)

1.3.-

Las peticiones del cliente al servidor web se realizaron mediante las siguientes herramientas:

Herramienta Curl:

```
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos/comdatos2018/Tp1/Punto1$ curl -X GET 192.168.0.22:5001
Bienvenidos al practico 1
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos/comdatos2018/Tp1/Punto1$ curl -X GET 192.168.0.22:5001/autenticacion?user=Andres
Andres
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos/comdatos2018/Tp1/Punto1$ curl -d "num1=2&num2=4" -X POST 192.168.0.22:5001/suma
El resultado de la suma es 6
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos/comdatos2018/Tp1/Punto1$
```

Herramienta netcat:

```
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos/comdatos2018/Tp1/Punto1$ nc 192.168.0.22 5001
GET / HTTP/1.1

HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 26
Server: Werkzeug/0.14.1 Python/3.5.2
Date: Sat, 08 Sep 2018 14:56:58 GMT

Bienvenidos al practico 1
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos/comdatos2018/Tp1/Punto1$ nc 192.168.0.22 5001
GET /autenticacion?user=Andres HTTP/1.1

HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 7
Server: Werkzeug/0.14.1 Python/3.5.2
Date: Sat, 08 Sep 2018 14:57:16 GMT

Andres
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos/comdatos2018/Tp1/Punto1$ nc 192.168.0.22 5001
POST /suma HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 14

num1=6&num2=3
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 29
Server: Werkzeug/0.14.1 Python/3.5.2
Date: Sat, 08 Sep 2018 14:58:07 GMT

El resultado de la suma es 9
```

Se puede observar la diferencia en la complejidad, siendo que Curl brinda una mayor facilidad al usuario para realizar las peticiones.

1.4.-

Luego se realizó el análisis del tráfico con la herramienta Wireshark.

A continuación se explican las cabeceras de **las peticiones HTTP**:

NOMBRE_MÉTODO /path HTTP/versión_http (ej. GET / HTTP/1.1)

Host: es la IP del servidor con el puerto

User-Agent: se refiere a la herramienta utilizada para realizar la petición (curl)

Accept: el tipo de contenido que el cliente puede procesar, expresado como un tipo MIME (MIME: forma estandarizada de indicar el formato de un documento). '/' indica que acepta cualquier tipo.

La respuesta HTTP tiene el siguiente formato:

HTTP/versión_http código_estado frase_respuesta (ej. HTTP/1.1 200 OK)

Content-Type: tipo de contenido presentado al client

Content-Length: largo en bytes del contenido

Server: software usado por el server

Date: fecha de la respuesta

Petición GET:

Wireshark capture showing a GET request. The filter is `http && ip.src == 192.168.0.22 || ip.src == 192.168.0.27 && ip.dst == 192.168.0.27 || ip.dst == 192.168.0.22`. The packet list shows two packets: a GET request (frame 316) and an HTTP 200 OK response (frame 320). The packet details for frame 316 show the request method, URI, version, host, user-agent, and accept headers.

No.	Time	Source	Destination	Protocol	Length	Info
316	95.126929367	192.168.0.22	192.168.0.27	HTTP	147	GET / HTTP/1.1
320	95.135722780	192.168.0.27	192.168.0.22	HTTP	229	HTTP/1.0 200 OK (text/html)

Frame 316: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits) on interface 0
Ethernet II, Src: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70)
Internet Protocol Version 4, Src: 192.168.0.22, Dst: 192.168.0.27
Transmission Control Protocol, Src Port: 56186, Dst Port: 5001, Seq: 1, Ack: 1, Len: 81
Hypertext Transfer Protocol
GET / HTTP/1.1
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
Host: 192.168.0.27:5001
User-Agent: curl/7.47.0
Accept: */*
Full request URI: http://192.168.0.27:5001/
HTTP request 1/1
Response in frame: 320

Respuesta GET:

Wireshark capture showing an HTTP 200 OK response. The filter is `http && ip.src == 192.168.0.22 || ip.src == 192.168.0.27 && ip.dst == 192.168.0.27 || ip.dst == 192.168.0.22`. The packet list shows two packets: a GET request (frame 316) and an HTTP 200 OK response (frame 320). The packet details for frame 320 show the response version, status code, status description, response phrase, content-type, content-length, server, date, and time since request.

No.	Time	Source	Destination	Protocol	Length	Info
316	95.126929367	192.168.0.22	192.168.0.27	HTTP	147	GET / HTTP/1.1
320	95.135722780	192.168.0.27	192.168.0.22	HTTP	229	HTTP/1.0 200 OK (text/html)

Frame 320: 229 bytes on wire (1832 bits), 229 bytes captured (1832 bits) on interface 0
Ethernet II, Src: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70), Dst: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23)
Internet Protocol Version 4, Src: 192.168.0.27, Dst: 192.168.0.22
Transmission Control Protocol, Src Port: 5001, Dst Port: 56186, Seq: 18, Ack: 82, Len: 163
[2 Reassembled TCP Segments (180 bytes): #318(17), #320(163)]
Hypertext Transfer Protocol
HTTP/1.0 200 OK
Response Version: HTTP/1.0
Status Code: 200
[Status Code Description: OK]
Response Phrase: OK
Content-Type: text/html; charset=utf-8
Content-Length: 26
Server: Werkzeug/0.14.1 Python/3.5.2
Date: Tue, 04 Sep 2018 13:03:47 GMT
Time since request: 0.008793413 seconds
Request in frame: 316
File Data: 26 bytes
Line-based text data: text/html (1 lines)
Bienvenidos al practico 1

Petición GET con parámetros: en este caso los parámetros se pasan en la URL.

Wireshark capture showing a GET request with parameters. The filter is `http && ip.src == 192.168.0.22 || ip.src == 192.168.0.27 && ip.dst == 192.168.0.27 || ip.dst == 192.168.0.22`. The packet list shows two packets: a GET request (frame 9) and an HTTP 200 OK response (frame 13). The packet details for frame 9 show the request method, URI, version, host, user-agent, and accept headers, including the query parameter `user=guille`.

No.	Time	Source	Destination	Protocol	Length	Info
9	14.593382170	192.168.0.22	192.168.0.27	HTTP	172	GET /autenticacion?user=guille HTTP/1.1
13	14.619032461	192.168.0.27	192.168.0.22	HTTP	209	HTTP/1.0 200 OK (text/html)

Frame 9: 172 bytes on wire (1376 bits), 172 bytes captured (1376 bits) on interface 0
Ethernet II, Src: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70)
Internet Protocol Version 4, Src: 192.168.0.22, Dst: 192.168.0.27
Transmission Control Protocol, Src Port: 56232, Dst Port: 5001, Seq: 1, Ack: 1, Len: 106
Hypertext Transfer Protocol
GET /autenticacion?user=guille HTTP/1.1
Request Method: GET
Request URI: /autenticacion?user=guille
Request URI Path: /autenticacion
Request URI Query: user=guille
Request URI Query Parameter: user=guille
Request Version: HTTP/1.1
Host: 192.168.0.27:5001
User-Agent: curl/7.47.0
Accept: */*
Full request URI: http://192.168.0.27:5001/autenticacion?user=guille
HTTP request 1/1
Response in frame: 13

Respuesta GET con parámetros:

The image shows a Wireshark packet capture. The top filter bar contains the expression: `http && ip.src == 192.168.0.22 || ip.src == 192.168.0.27 && ip.dst == 192.168.0.27 || ip.dst == 192.168.0.22`. The packet list shows four packets. Packet 13 is selected, showing details for the Hypertext Transfer Protocol. The status is 200 OK, and the content type is text/html. The request is shown in frame 0.

No.	Time	Source	Destination	Protocol	Length	Info
9	14.593302170	192.168.0.22	192.168.0.27	HTTP	172	GET /autenticacion?user=guille HTTP/1.1
13	14.619032461	192.168.0.27	192.168.0.22	HTTP	209	HTTP/1.0 200 OK (text/html)

Frame 13: 209 bytes on wire (1672 bits), 209 bytes captured (1672 bits) on interface 0
Ethernet II, Src: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70), Dst: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23)
Internet Protocol Version 4, Src: 192.168.0.27, Dst: 192.168.0.22
Transmission Control Protocol, Src Port: 5001, Dst Port: 56232, Seq: 18, Ack: 107, Len: 143
[2 Reassembled TCP Segments (160 bytes): #11(17), #13(143)]
Hypertext Transfer Protocol
HTTP/1.0 200 OK\r\n
[Expert Info (Chat/Sequence): HTTP/1.0 200 OK\r\n]
Response Version: HTTP/1.0
Status Code: 200
[Status Code Description: OK]
Response Phrase: OK
Content-Type: text/html; charset=utf-8\r\n
Content-Length: 7\r\n
Server: Werkzeug/0.14.1 Python/3.5.2\r\n
Date: Tue, 04 Sep 2018 13:19:29 GMT\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.025730291 seconds]
[Request in frame: 0]
File Data: 7 bytes
Line-based text data: text/html (1 lines)
guille\r\n

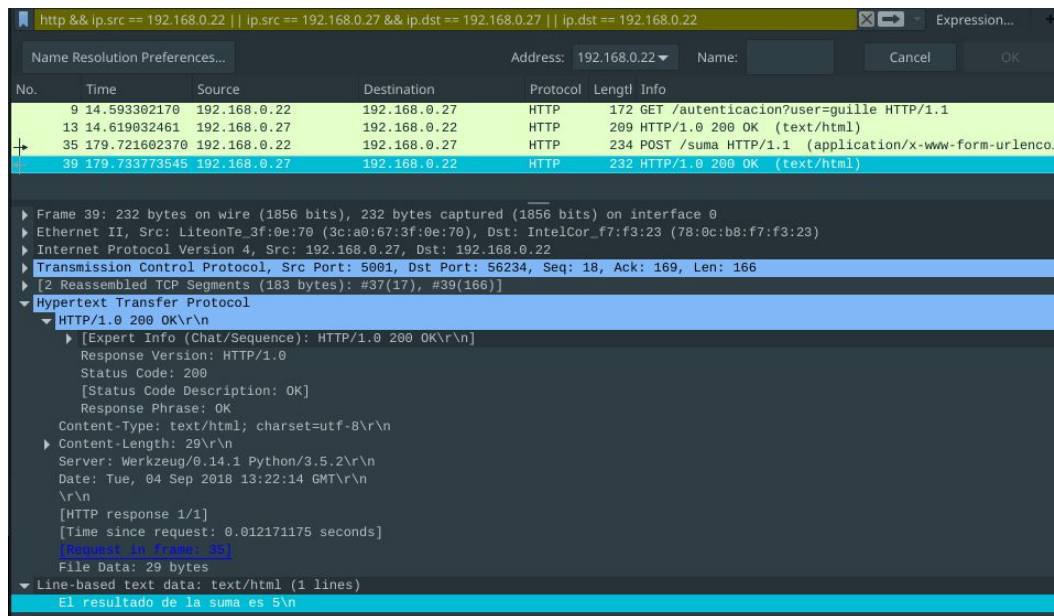
Petición POST: se pasan los parámetros en un formulario

The image shows a Wireshark packet capture. The top filter bar contains the expression: `http && ip.src == 192.168.0.22 || ip.src == 192.168.0.27 && ip.dst == 192.168.0.27 || ip.dst == 192.168.0.22`. The packet list shows four packets. Packet 35 is selected, showing details for the Hypertext Transfer Protocol. The status is 200 OK, and the content type is application/x-www-form-urlencoded. The request is shown in frame 0.

No.	Time	Source	Destination	Protocol	Length	Info
9	14.593302170	192.168.0.22	192.168.0.27	HTTP	172	GET /autenticacion?user=guille HTTP/1.1
13	14.619032461	192.168.0.27	192.168.0.22	HTTP	209	HTTP/1.0 200 OK (text/html)
35	179.721602370	192.168.0.22	192.168.0.27	HTTP	234	POST /suma HTTP/1.1 (application/x-www-form-urlencoded)
39	179.733773545	192.168.0.27	192.168.0.22	HTTP	232	HTTP/1.0 200 OK (text/html)

Frame 35: 234 bytes on wire (1872 bits), 234 bytes captured (1872 bits) on interface 0
Ethernet II, Src: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70)
Internet Protocol Version 4, Src: 192.168.0.22, Dst: 192.168.0.27
Transmission Control Protocol, Src Port: 56234, Dst Port: 5001, Seq: 1, Ack: 1, Len: 168
Hypertext Transfer Protocol
POST /suma HTTP/1.1\r\n
[Expert Info (Chat/Sequence): POST /suma HTTP/1.1\r\n]
Request Method: POST
Request URI: /suma
Request Version: HTTP/1.1
Host: 192.168.0.27:5001\r\n
User-Agent: curl/7.47.0\r\n
Accept: */*\r\n
Content-Length: 13\r\n
Content-Type: application/x-www-form-urlencoded\r\n
\r\n
[Full request URI: http://192.168.0.27:5001/suma]
[HTTP request 1/1]
[Response in frame: 39]
File Data: 13 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "num1" = "2"
Key: num1
Value: 2
Form item: "num2" = "3"
Key: num2
Value: 3

Respuesta POST:



Se puede observar que todas las respuestas fueron satisfactorias (código 200 OK). Las otras respuestas posibles pueden ser, entre otras:

404 Not Found, 400 Bad Request, 301 Moved Permanently, 505 HTTP Version Not Supported
500 Internal Server Error

1.5.-

Otros métodos HTTP son: PUT (insertar archivos en el server), DELETE (eliminar archivos), HEAD (similar a GET para depuración), CONNECT, OPTIONS, TRACE, PATCH.

JSON es un formato mas liviano para presentar los datos al cliente (navegador web) y se usa como alternativa a HTML y XML por ser muy ligero, fácil de usar e independiente de cualquier lenguaje de programación.

REST API es un estilo de arquitectura de software que permite utilizar los métodos HTTP ofreciendo una interfaz uniforme y estándar, además de permitir el uso de hipermedios (páginas web con texto, imagen, vídeo, etc.) para permitir al usuario navegar por enlaces HTML.

Ejercicio 2: Comunicación TCP y UDP con sockets

- 2.1.- Utilizando el módulo de sockets de python, implementar tanto un servidor UDP y TCP como un cliente UDP y TCP. El cliente debe poder mandar un mensaje y el servidor debe contestar con un eco (copia) del mismo mensaje.
- 2.2.- Analizar el flujo de mensajes con wireshark y explicar las diferencias entre UDP y TCP con detalle. Hacer énfasis en el handshake de TCP.
- 2.3.- Preguntas relacionadas:
 - ¿Cómo se define un socket?
 - ¿Qué diferencia hay, en tanto señalización, entre los protocolos UDP y TCP?
 - ¿En qué casos usaría TCP o UDP? Nombre algunos ejemplos.

2.1.-

A continuación se muestra el funcionamiento del cliente/servidor para TCP y UDP:

```
guillette@adelle ~/Facu/Comunicaciones_de_Datos/comdatos2018/Tp1/Punto2 $ python tcp_client.py
Ingrese mensaje a enviar: hola
Respuesta:
hola
Ingrese mensaje a enviar: exit
guillette@adelle ~/Facu/Comunicaciones_de_Datos/comdatos2018/Tp1/Punto2 $ python udp_client.py
Ingrese mensaje a enviar: hola
```

```
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos/comdatos2018/Tp1/Punto2$
python3 serverTCP.py
Connection address: ('192.168.0.22', 43434)
received data: hola
received data: exit
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos/comdatos2018/Tp1/Punto2$
python3 serverUDP.py
received message: hola
```

2.2.-

TCP

Inicio de conexión (Three-way Handshake): SYN, el cliente solicita el inicio de conexión

The image shows a Wireshark packet capture of a TCP SYN packet. The packet list at the top shows three packets: a SYN packet (No. 2), an ACK packet (No. 3), and another SYN packet (No. 1). The selected packet (No. 1) is a TCP SYN packet from 192.168.1.16 to 192.168.1.7. The packet details pane shows the following information:

- Source Port: 53166
- Destination Port: 5003
- [Stream index: 0]
- [TCP Segment Len: 0]
- Sequence number: 0 (relative sequence number)
- [Next sequence number: 0 (relative sequence number)]
- Acknowledgment number: 0
- 1010 = Header Length: 40 bytes (10)
- Flags: 0x002 (SYN)
- Window size value: 29200
- [Calculated window size: 29200]
- Checksum: 0x7923 [unverified]
- [Checksum Status: Unverified]
- Urgent pointer: 0
- Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
- [Timestamps]

SYN-ACK: el servidor acepta el inicio de conexión y solicita otro reconocimiento

tcp && ip.src == 192.168.1.16 ip.src == 192.168.1.7 && ip.dst == 192.168.1.7 ip.dst == 192.168.1.16						
No.	Time	Source	Destination	Protocol	Length	Info
3	0.009575452	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval...
2	0.009548467	192.168.1.7	192.168.1.16	TCP	74	5003 → 53166 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 ...
1	0.000000000	192.168.1.16	192.168.1.7	TCP	74	53166 → 5003 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SA...

▶ Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70), Dst: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23)
▶ Internet Protocol Version 4, Src: 192.168.1.7, Dst: 192.168.1.16
▼ Transmission Control Protocol, Src Port: 5003, Dst Port: 53166, Seq: 0, Ack: 1, Len: 0
Source Port: 5003
Destination Port: 53166
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
[Next sequence number: 0 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
1010 = Header Length: 40 bytes (10)
▶ Flags: 0x012 (SYN, ACK)
Window size value: 28960
[Calculated window size: 28960]
Checksum: 0xad5e [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
▶ [SEQ/ACK analysis]
▶ [Timestamps]

ACK: el cliente envía el reconocimiento y se establece la conexión

tcp && ip.src == 192.168.1.16 ip.src == 192.168.1.7 && ip.dst == 192.168.1.7 ip.dst == 192.168.1.16						
No.	Time	Source	Destination	Protocol	Length	Info
3	0.009575452	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval...
2	0.009548467	192.168.1.7	192.168.1.16	TCP	74	5003 → 53166 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 ...
1	0.000000000	192.168.1.16	192.168.1.7	TCP	74	53166 → 5003 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SA...

▶ Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶ Ethernet II, Src: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70)
▶ Internet Protocol Version 4, Src: 192.168.1.16, Dst: 192.168.1.7
▼ Transmission Control Protocol, Src Port: 53166, Dst Port: 5003, Seq: 1, Ack: 1, Len: 0
Source Port: 53166
Destination Port: 5003
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 1 (relative sequence number)
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
1000 = Header Length: 32 bytes (8)
▶ Flags: 0x010 (ACK)
Window size value: 229
[Calculated window size: 29312]
[Window size scaling factor: 128]
Checksum: 0x4c64 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
▶ [SEQ/ACK analysis]
▶ [Timestamps]

Transmisión del mensaje por TCP:

PSH y ACK:

tcp && ip.src == 192.168.1.16 ip.src == 192.168.1.7 && ip.dst == 192.168.1.7 ip.dst == 192.168.1.16						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.1.16	192.168.1.7	TCP	74	53166 → 5003 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SA...
2	0.009548467	192.168.1.7	192.168.1.16	TCP	74	5003 → 53166 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 ...
3	0.009575452	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval...
9920	711.016091199	192.168.1.16	192.168.1.7	TCP	73	53166 → 5003 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=7 ...
9930	711.061233984	192.168.1.7	192.168.1.16	TCP	66	5003 → 53166 [ACK] Seq=1 Ack=8 Win=29056 Len=0 TSval...
9947	711.332256579	192.168.1.7	192.168.1.16	TCP	73	5003 → 53166 [PSH, ACK] Seq=1 Ack=8 Win=29056 Len=7 ...
9948	711.332348152	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=8 Ack=8 Win=29312 Len=0 TSval...
9949	711.332743037	192.168.1.7	192.168.1.16	TCP	73	[TCP Spurious Retransmission] 5003 → 53166 [PSH, ACK]...
9950	711.332779370	192.168.1.16	192.168.1.7	TCP	78	[TCP Dup ACK 9948#1] 53166 → 5003 [ACK] Seq=8 Ack=8 ...

Frame 9920: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
Ethernet II, Src: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70)
Internet Protocol Version 4, Src: 192.168.1.16, Dst: 192.168.1.7
Transmission Control Protocol, Src Port: 53166, Dst Port: 5003, Seq: 1, Ack: 1, Len: 7
Source Port: 53166
Destination Port: 5003
[Stream index: 0]
[TCP Segment Len: 7]
Sequence number: 1 (relative sequence number)
[Next sequence number: 8 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
1000 ... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window size value: 229
[Calculated window size: 29312]
[Window size scaling factor: 128]
Checksum: 0xf3b6 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
TCP payload (7 bytes)
Data (7 bytes)
Data: 6d656e73616a65
[Length: 7]

ACK:

tcp && ip.src == 192.168.1.16 ip.src == 192.168.1.7 && ip.dst == 192.168.1.7 ip.dst == 192.168.1.16						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.1.16	192.168.1.7	TCP	74	53166 → 5003 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SA...
2	0.009548467	192.168.1.7	192.168.1.16	TCP	74	5003 → 53166 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 ...
3	0.009575452	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval...
9920	711.016091199	192.168.1.16	192.168.1.7	TCP	73	53166 → 5003 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=7 ...
9930	711.061233984	192.168.1.7	192.168.1.16	TCP	66	5003 → 53166 [ACK] Seq=1 Ack=8 Win=29056 Len=0 TSval...
9947	711.332256579	192.168.1.7	192.168.1.16	TCP	73	5003 → 53166 [PSH, ACK] Seq=1 Ack=8 Win=29056 Len=7 ...
9948	711.332348152	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=8 Ack=8 Win=29312 Len=0 TSval...
9949	711.332743037	192.168.1.7	192.168.1.16	TCP	73	[TCP Spurious Retransmission] 5003 → 53166 [PSH, ACK]...
9950	711.332779370	192.168.1.16	192.168.1.7	TCP	78	[TCP Dup ACK 9948#1] 53166 → 5003 [ACK] Seq=8 Ack=8 ...

Frame 9930: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70), Dst: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23)
Internet Protocol Version 4, Src: 192.168.1.7, Dst: 192.168.1.16
Transmission Control Protocol, Src Port: 5003, Dst Port: 53166, Seq: 1, Ack: 8, Len: 0
Source Port: 5003
Destination Port: 53166
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 1 (relative sequence number)
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 8 (relative ack number)
1000 ... = Header Length: 32 bytes (8)
Flags: 0x010 (ACK)
Window size value: 227
[Calculated window size: 29056]
[Window size scaling factor: 128]
Checksum: 0xbc74 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]

Transmision del echo:

PSH y ACK:

tcp && ip.src == 192.168.1.16 ip.src == 192.168.1.7 && ip.dst == 192.168.1.7 ip.dst == 192.168.1.16						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.1.16	192.168.1.7	TCP	74	53166 → 5003 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SA...
2	0.009548467	192.168.1.7	192.168.1.16	TCP	74	5003 → 53166 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 ...
3	0.009575452	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval...
9929	711.016091199	192.168.1.16	192.168.1.7	TCP	73	53166 → 5003 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=7 ...
9930	711.061233984	192.168.1.7	192.168.1.16	TCP	66	5003 → 53166 [ACK] Seq=1 Ack=8 Win=29056 Len=0 TSval...
9947	711.332256579	192.168.1.7	192.168.1.16	TCP	73	5003 → 53166 [PSH, ACK] Seq=1 Ack=8 Win=29056 Len=7 ...
9948	711.332348152	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=8 Ack=8 Win=29312 Len=0 TSval...
9949	711.332743037	192.168.1.7	192.168.1.16	TCP	73	[TCP Spurious Retransmission] 5003 → 53166 [PSH, ACK]...
9950	711.332779370	192.168.1.16	192.168.1.7	TCP	78	[TCP Dup ACK 9948#1] 53166 → 5003 [ACK] Seq=8 Ack=8 ...

Frame 9947: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
Ethernet II, Src: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70), Dst: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23)
Internet Protocol Version 4, Src: 192.168.1.7, Dst: 192.168.1.16
Transmission Control Protocol, Src Port: 5003, Dst Port: 53166, Seq: 1, Ack: 8, Len: 7
Source Port: 5003
Destination Port: 53166
[Stream index: 0]
[TCP Segment Len: 7]
Sequence number: 1 (relative sequence number)
[Next sequence number: 8 (relative sequence number)]
Acknowledgment number: 8 (relative ack number)
1000 = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window size value: 227
[Calculated window size: 29056]
[Window size scaling factor: 128]
Checksum: 0x1a22 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
TCP payload (7 bytes)
Data (7 bytes)
Data: 6d656e73616a65
[Length: 7]

ACK:

tcp && ip.src == 192.168.1.16 ip.src == 192.168.1.7 && ip.dst == 192.168.1.7 ip.dst == 192.168.1.16						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.1.16	192.168.1.7	TCP	74	53166 → 5003 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SA...
2	0.009548467	192.168.1.7	192.168.1.16	TCP	74	5003 → 53166 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 ...
3	0.009575452	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval...
9929	711.016091199	192.168.1.16	192.168.1.7	TCP	73	53166 → 5003 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=7 ...
9930	711.061233984	192.168.1.7	192.168.1.16	TCP	66	5003 → 53166 [ACK] Seq=1 Ack=8 Win=29056 Len=0 TSval...
9947	711.332256579	192.168.1.7	192.168.1.16	TCP	73	5003 → 53166 [PSH, ACK] Seq=1 Ack=8 Win=29056 Len=7 ...
9948	711.332348152	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=8 Ack=8 Win=29312 Len=0 TSval...
9949	711.332743037	192.168.1.7	192.168.1.16	TCP	73	[TCP Spurious Retransmission] 5003 → 53166 [PSH, ACK]...
9950	711.332779370	192.168.1.16	192.168.1.7	TCP	78	[TCP Dup ACK 9948#1] 53166 → 5003 [ACK] Seq=8 Ack=8 ...

Frame 9948: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70)
Internet Protocol Version 4, Src: 192.168.1.16, Dst: 192.168.1.7
Transmission Control Protocol, Src Port: 53166, Dst Port: 5003, Seq: 8, Ack: 8, Len: 0
Source Port: 53166
Destination Port: 5003
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 8 (relative sequence number)
[Next sequence number: 8 (relative sequence number)]
Acknowledgment number: 8 (relative ack number)
1000 = Header Length: 32 bytes (8)
Flags: 0x010 (ACK)
Window size value: 229
[Calculated window size: 29312]
[Window size scaling factor: 128]
Checksum: 0xbcb1c [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]

Cierre de conexión:

FIN:

No.	Time	Source	Destination	Protocol	Length	Info
2	0.009548467	192.168.1.7	192.168.1.16	TCP	74	5003 → 53166 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0
3	0.009575452	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=1 Ack=1 Win=29312 Len=0
9929	711.016091199	192.168.1.16	192.168.1.7	TCP	73	53166 → 5003 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=0
9930	711.061233984	192.168.1.7	192.168.1.16	TCP	66	5003 → 53166 [ACK] Seq=1 Ack=8 Win=29056 Len=0
9947	711.332256579	192.168.1.7	192.168.1.16	TCP	73	5003 → 53166 [PSH, ACK] Seq=1 Ack=8 Win=29056 Len=0
9948	711.332348152	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=8 Ack=8 Win=29312 Len=0
9949	711.332743037	192.168.1.7	192.168.1.16	TCP	73	[TCP Spurious Retransmission] 5003 → 53166 [PSH, A..
9950	711.332779370	192.168.1.16	192.168.1.7	TCP	78	[TCP Dup ACK 9948#1] 53166 → 5003 [ACK] Seq=8 Ack=..
27957	1739.6762178..	192.168.1.16	192.168.1.7	TCP	70	53166 → 5003 [PSH, ACK] Seq=8 Ack=8 Win=29312 Len=..
27958	1739.6917033..	192.168.1.7	192.168.1.16	TCP	70	5003 → 53166 [PSH, ACK] Seq=8 Ack=12 Win=29056 Len=..
27959	1739.6918876..	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=12 Ack=12 Win=29312 Len=0
27960	1739.6920177..	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [FIN, ACK] Seq=12 Ack=12 Win=29312 Le..
27961	1739.6992285..	192.168.1.7	192.168.1.16	TCP	66	5003 → 53166 [FIN, ACK] Seq=12 Ack=13 Win=29056 Le..
27962	1739.6992820..	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=13 Ack=13 Win=29312 Len=0

Frame 27960: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70)
Internet Protocol Version 4, Src: 192.168.1.16, Dst: 192.168.1.7

Transmission Control Protocol, Src Port: 53166, Dst Port: 5003, Seq: 12, Ack: 12, Len: 0

Source Port: 53166
Destination Port: 5003
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 12 (relative sequence number)
[Next sequence number: 12 (relative sequence number)]
Acknowledgment number: 12 (relative ack number)
1000 = Header Length: 32 bytes (8)

Flags: 0x011 (FIN, ACK)

000. = Reserved: Not set
...0. = Nonce: Not set
....0... = Congestion Window Reduced (CWR): Not set
....0... = ECN-Echo: Not set
....0... = Urgent: Not set
....1... = Acknowledgment: Set
....0... = Push: Not set
....0... = Reset: Not set
....0... = Syn: Not set
....1... = Fin: Set

[Expert Info (Chat/Sequence): Connection finish (FIN)]
[TCP Flags:A...F]
Window size value: 229
[Calculated window size: 29312]
[Window size scaling factor: 128]
Checksum: 0x1db3 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[Timestamps]

FIN ACK:

No.	Time	Source	Destination	Protocol	Length	Info
2	0.009548467	192.168.1.7	192.168.1.16	TCP	74	5003 → 53166 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0
3	0.009575452	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=1 Ack=1 Win=29312 Len=0
9929	711.016091199	192.168.1.16	192.168.1.7	TCP	73	53166 → 5003 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=0
9930	711.061233984	192.168.1.7	192.168.1.16	TCP	66	5003 → 53166 [ACK] Seq=1 Ack=8 Win=29056 Len=0
9947	711.332256579	192.168.1.7	192.168.1.16	TCP	73	5003 → 53166 [PSH, ACK] Seq=1 Ack=8 Win=29056 Len=0
9948	711.332348152	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=8 Ack=8 Win=29312 Len=0
9949	711.332743037	192.168.1.7	192.168.1.16	TCP	73	[TCP Spurious Retransmission] 5003 → 53166 [PSH, A..
9950	711.332779370	192.168.1.16	192.168.1.7	TCP	78	[TCP Dup ACK 9948#1] 53166 → 5003 [ACK] Seq=8 Ack=..
27957	1739.6762178..	192.168.1.16	192.168.1.7	TCP	70	53166 → 5003 [PSH, ACK] Seq=8 Ack=8 Win=29312 Len=..
27958	1739.6917033..	192.168.1.7	192.168.1.16	TCP	70	5003 → 53166 [PSH, ACK] Seq=8 Ack=12 Win=29056 Len=..
27959	1739.6918876..	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=12 Ack=12 Win=29312 Len=0
27960	1739.6920177..	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [FIN, ACK] Seq=12 Ack=12 Win=29312 Le..
27961	1739.6992285..	192.168.1.7	192.168.1.16	TCP	66	5003 → 53166 [FIN, ACK] Seq=12 Ack=13 Win=29056 Le..
27962	1739.6992820..	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=13 Ack=13 Win=29312 Len=0

Frame 27961: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70), Dst: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23)
Internet Protocol Version 4, Src: 192.168.1.7, Dst: 192.168.1.16

Transmission Control Protocol, Src Port: 5003, Dst Port: 53166, Seq: 12, Ack: 13, Len: 0

Source Port: 5003
Destination Port: 53166
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 12 (relative sequence number)
[Next sequence number: 12 (relative sequence number)]
Acknowledgment number: 13 (relative ack number)
1000 = Header Length: 32 bytes (8)

Flags: 0x011 (FIN, ACK)

000. = Reserved: Not set
...0. = Nonce: Not set
....0... = Congestion Window Reduced (CWR): Not set
....0... = ECN-Echo: Not set
....0... = Urgent: Not set
....1... = Acknowledgment: Set
....0... = Push: Not set
....0... = Reset: Not set
....0... = Syn: Not set
....1... = Fin: Set

[TCP Flags:A...F]
Window size value: 227
[Calculated window size: 29056]
[Window size scaling factor: 128]
Checksum: 0x1da4 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]

ACK:

tcp && ip.src == 192.168.1.16 ip.src == 192.168.1.7 && ip.dst == 192.168.1.7 ip.dst == 192.168.1.16						
No.	Time	Source	Destination	Protocol	Length	Info
2	0.009548467	192.168.1.7	192.168.1.16	TCP	74	5003 → 53166 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=...
3	0.009575452	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSV...
9929	711.016091199	192.168.1.16	192.168.1.7	TCP	73	53166 → 5003 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=...
9930	711.061233984	192.168.1.7	192.168.1.16	TCP	66	5003 → 53166 [ACK] Seq=1 Ack=8 Win=29056 Len=0 TSV...
9947	711.332256579	192.168.1.7	192.168.1.16	TCP	73	5003 → 53166 [PSH, ACK] Seq=1 Ack=8 Win=29056 Len=...
9948	711.332348152	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=8 Ack=8 Win=29312 Len=0 TSV...
9949	711.332743037	192.168.1.7	192.168.1.16	TCP	73	[TCP Spurious Retransmission] 5003 → 53166 [PSH, A...
9950	711.332779370	192.168.1.16	192.168.1.7	TCP	78	[TCP Dup ACK 9948#1] 53166 → 5003 [ACK] Seq=8 Ack=...
27957	1739.6762178...	192.168.1.16	192.168.1.7	TCP	70	53166 → 5003 [PSH, ACK] Seq=8 Ack=8 Win=29312 Len=...
27958	1739.6917033...	192.168.1.7	192.168.1.16	TCP	70	5003 → 53166 [PSH, ACK] Seq=8 Ack=12 Win=29056 Len=...
27959	1739.6918876...	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=12 Ack=12 Win=29312 Len=0 T...
27960	1739.6920177...	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [FIN, ACK] Seq=12 Ack=12 Win=29312 Le...
27961	1739.6992285...	192.168.1.7	192.168.1.16	TCP	66	5003 → 53166 [FIN, ACK] Seq=12 Ack=13 Win=29056 Le...
27962	1739.6992820...	192.168.1.16	192.168.1.7	TCP	66	53166 → 5003 [ACK] Seq=13 Ack=13 Win=29312 Len=0 T...
Frame 27962: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0						
Ethernet II, Src: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70)						
Internet Protocol Version 4, Src: 192.168.1.16, Dst: 192.168.1.7						
Transmission Control Protocol, Src Port: 53166, Dst Port: 5003, Seq: 13, Ack: 13, Len: 0						
Source Port: 53166						
Destination Port: 5003						
[Stream index: 0]						
[TCP Segment Len: 0]						
Sequence number: 13 (relative sequence number)						
[Next sequence number: 13 (relative sequence number)]						
Acknowledgment number: 13 (relative ack number)						
1000 = Header Length: 32 bytes (8)						
Flags: 0x010 (ACK)						
Window size value: 229						
[Calculated window size: 29312]						
[Window size scaling factor: 128]						
Checksum: 0x1da0 [unverified]						
[Checksum Status: Unverified]						
Urgent pointer: 0						
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps						
[SEQ/ACK analysis]						
[Timestamps]						

UDP

udp && ip.src == 192.168.1.16 ip.src == 192.168.1.7 && ip.dst == 192.168.1.7 ip.dst == 192.168.1.16						
No.	Time	Source	Destination	Protocol	Length	Info
19	55.944501270	192.168.1.16	192.168.1.7	UDP	49	41505 → 5006 Len=7
Frame 19: 49 bytes on wire (392 bits), 49 bytes captured (392 bits) on interface 0						
Ethernet II, Src: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70)						
Internet Protocol Version 4, Src: 192.168.1.16, Dst: 192.168.1.7						
User Datagram Protocol, Src Port: 41505, Dst Port: 5006						
Source Port: 41505						
Destination Port: 5006						
Length: 15						
Checksum: 0x2475 [unverified]						
[Checksum Status: Unverified]						
[Stream index: 4]						
Data (7 bytes)						
Data: 6d656e73616a65						
[Length: 7]						

Las diferencias entre UDP y TCP son:

- TCP es orientado a conexión, mientras que UDP es no-orientado a conexión.
- En UDP no hay una verificación de que el paquete haya llegado a destino
- En TCP se realizan todos los controles para asegurar el envío y recepción.
- El TCP siempre realiza reconocimientos (ACK) en cada operación.
- En caso de pérdida de paquetes, TCP lo retransmite, UDP no.
- En TCP, los paquetes llegan ordenados, en UDP no hay garantía de esto.
- UDP es más simple al no realizar ningún control.

2.3.-

Un socket es una abstracción para la comunicación que está definido por una dirección y un puerto, sirve para conectar la capa de aplicación con la capa de transporte.

La diferencia de señalización entre UDP y TCP es que en UDP la comunicación es unidireccional sin informar del envío del paquete, ni la llegada, ni la pérdida (no hay señalización); la cabecera UDP solo contiene la dirección, el puerto, el checksum y los datos. En TCP hay una fuerte señalización, teniendo en la cabecera de TCP muchos datos tales como: número de secuencia, número de reconocimiento, flags, ventana de datos, checksum, punteros, opciones.

TCP se usa para e-mail, acceso remoto, web, transferencia de archivos, etc.

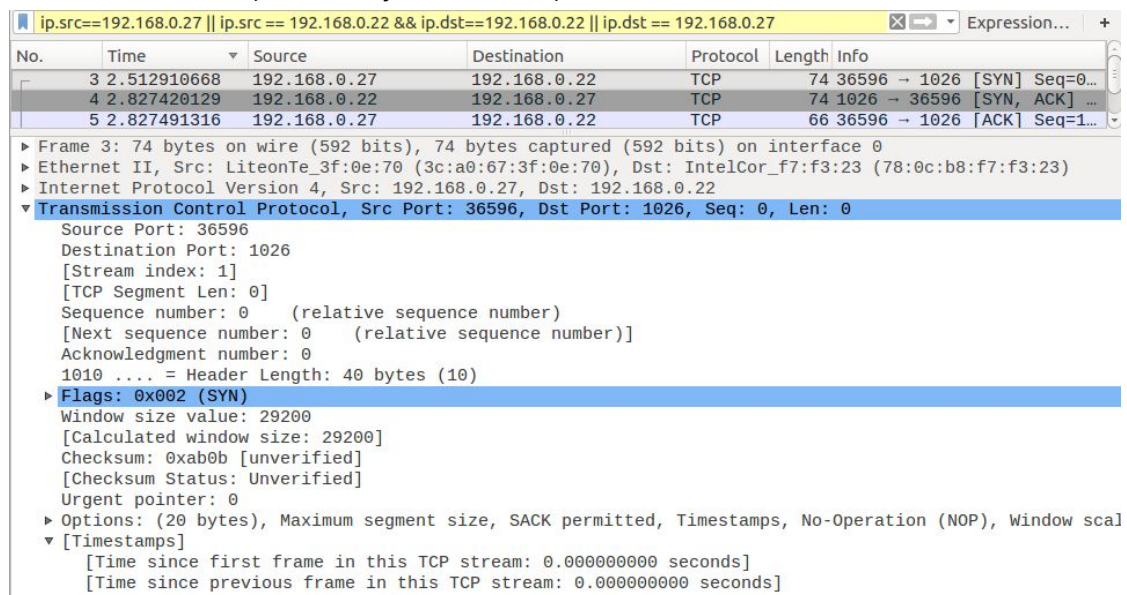
UDP se usa para aplicaciones donde no importa si algún paquete se pierde: streaming de audio y video, videojuegos, telefonía por IP, etc.

Ejercicio 3: Análisis de una transferencia TCP con FTP

- 3.1.- Instalar un servidor FTP en el host de uno de los integrantes del grupo y un cliente FTP en un host diferente. **NOTA:** debe haber conectividad IP entre server y cliente.
- 3.2.- En el server, crear un archivo con tamaño mínimo de 50 megabytes.
- 3.3.- Transferir el archivo del server al cliente, graficando los parámetros de transferencia con el wireshark.
- 3.4.- Explicar de manera cualitativa los gráficos 3.4.1) Time/Sequences Stevens; 3.4.2) Window Scaling y 3.4.3) RoundTripTime.
- 3.5.- Preguntas relacionadas:
¿Cómo funciona FTP?

3.3-

Inicio de conexión (Three-way Handshake):



The image shows a Wireshark packet capture of a TCP three-way handshake. The filter is set to `ip.src==192.168.0.27 || ip.src==192.168.0.22 && ip.dst==192.168.0.22 || ip.dst==192.168.0.27`. The packet list shows three packets:

No.	Time	Source	Destination	Protocol	Length	Info
3	2.512910668	192.168.0.27	192.168.0.22	TCP	74	36596 → 1026 [SYN] Seq=0...
4	2.827420129	192.168.0.22	192.168.0.27	TCP	74	1026 → 36596 [SYN, ACK] ...
5	2.827491316	192.168.0.27	192.168.0.22	TCP	66	36596 → 1026 [ACK] Seq=1...

The packet details for the first packet (Frame 3) are expanded, showing:

- Source Port: 36596
- Destination Port: 1026
- [Stream index: 1]
- [TCP Segment Len: 0]
- Sequence number: 0 (relative sequence number)
- [Next sequence number: 0 (relative sequence number)]
- Acknowledgment number: 0
- 1010 = Header Length: 40 bytes (10)
- Flags: 0x002 (SYN)
- Window size value: 29200
- [Calculated window size: 29200]
- Checksum: 0xab0b [unverified]
- [Checksum Status: Unverified]
- Urgent pointer: 0
- Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scaling
- [Timestamps]
- [Time since first frame in this TCP stream: 0.000000000 seconds]
- [Time since previous frame in this TCP stream: 0.000000000 seconds]

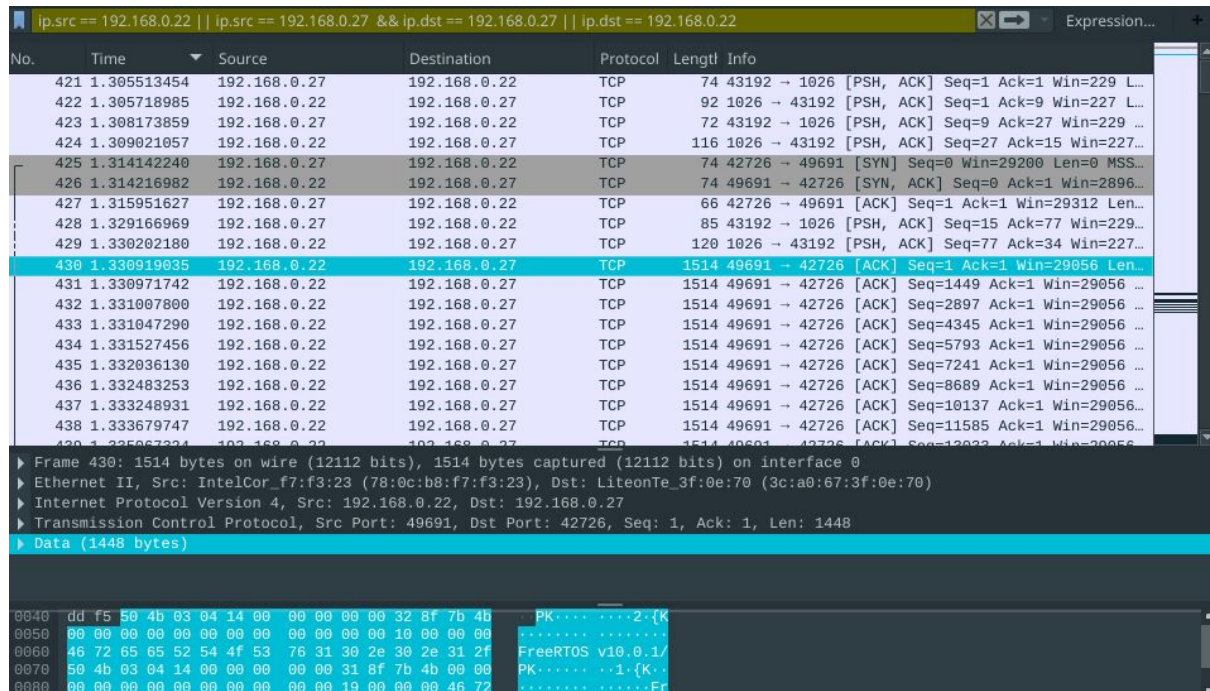
Autenticación de usuario en el servidor:

El servidor está configurado para permitir usuarios anónimos y un usuario con credenciales (user, 12345). El usuario se conecta al servidor indicando una dirección y un puerto y logueándose con esas credenciales.

```
[I 2018-09-07 19:09:36] >>> starting FTP server on 192.168.0.22:1026, pid=23046 <<<
[I 2018-09-07 19:09:36] concurrency model: async
[I 2018-09-07 19:09:36] masquerade (NAT) address: None
[I 2018-09-07 19:09:36] passive ports: None
[I 2018-09-07 19:09:49] 192.168.0.27:36442-[] FTP session opened (connect)
[I 2018-09-07 19:09:49] 192.168.0.27:36442-[user] USER 'user' logged in.
[I 2018-09-07 19:09:49] 192.168.0.27:36442-[user] CWD /home/guilletrejo/Downloads 250
[I 2018-09-07 19:14:21] 192.168.0.27:36442-[user] RETR /home/guilletrejo/Downloads/virtualbox-5.2_5.2.18-124319-Ubuntu
-xenial_amd64.deb completed=1 bytes=73493916 seconds=253.76
[I 2018-09-07 19:14:25] 192.168.0.27:36442-[user] FTP session closed (disconnect).
[I 2018-09-07 19:23:28] 192.168.0.27:36596-[] FTP session opened (connect)
[I 2018-09-07 19:23:28] 192.168.0.27:36596-[user] USER 'user' logged in.
[I 2018-09-07 19:23:28] 192.168.0.27:36596-[user] CWD /home/guilletrejo/Downloads 250
[I 2018-09-07 19:27:10] 192.168.0.27:36596-[user] RETR /home/guilletrejo/Downloads/virtualbox-5.2_5.2.18-124319-Ubuntu
-xenial_amd64.deb completed=1 bytes=73493916 seconds=206.69
[I 2018-09-07 19:27:11] 192.168.0.27:36596-[user] FTP session closed (disconnect).
```

Comienzo de la transferencia: (primer paquete de datos)

Se observa que el primer paquete de datos enviados tiene el número de secuencia 1 y número de reconocimiento 1. Además, en los bytes de datos se puede ver el nombre del archivo a descargar (en ASCII)



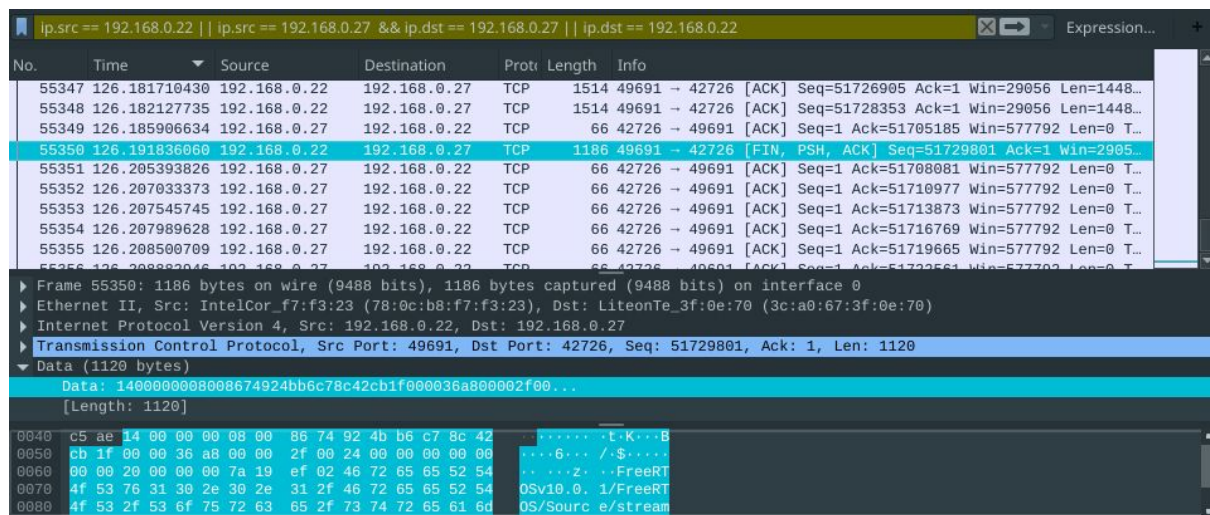
No.	Time	Source	Destination	Protocol	Length	Info
421	1.305513454	192.168.0.27	192.168.0.22	TCP	74	43192 → 1026 [PSH, ACK] Seq=1 Ack=1 Win=229 L...
422	1.305718985	192.168.0.22	192.168.0.27	TCP	92	1026 → 43192 [PSH, ACK] Seq=1 Ack=9 Win=227 L...
423	1.308173859	192.168.0.27	192.168.0.22	TCP	72	43192 → 1026 [PSH, ACK] Seq=9 Ack=27 Win=229 ...
424	1.309021057	192.168.0.22	192.168.0.27	TCP	116	1026 → 43192 [PSH, ACK] Seq=27 Ack=15 Win=227...
425	1.314142240	192.168.0.27	192.168.0.22	TCP	74	42726 → 49691 [SYN] Seq=0 Win=29200 Len=0 MSS...
426	1.314216982	192.168.0.22	192.168.0.27	TCP	74	49691 → 42726 [SYN, ACK] Seq=0 Ack=1 Win=2896...
427	1.315951627	192.168.0.27	192.168.0.22	TCP	66	42726 → 49691 [ACK] Seq=1 Ack=1 Win=29312 Len...
428	1.329166969	192.168.0.27	192.168.0.22	TCP	85	43192 → 1026 [PSH, ACK] Seq=15 Ack=77 Win=229...
429	1.330202180	192.168.0.22	192.168.0.27	TCP	120	1026 → 43192 [PSH, ACK] Seq=77 Ack=34 Win=227...
430	1.330919035	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=1 Ack=1 Win=29056 Len...
431	1.330971742	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=1449 Ack=1 Win=29056 ...
432	1.331007800	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=2897 Ack=1 Win=29056 ...
433	1.331047290	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=4345 Ack=1 Win=29056 ...
434	1.331527456	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=5793 Ack=1 Win=29056 ...
435	1.332036130	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=7241 Ack=1 Win=29056 ...
436	1.332483253	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=8689 Ack=1 Win=29056 ...
437	1.333248931	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=10137 Ack=1 Win=29056...
438	1.333679747	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=11585 Ack=1 Win=29056...
439	1.335077234	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=12833 Ack=1 Win=29056...

Frame 430: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
Ethernet II, Src: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70)
Internet Protocol Version 4, Src: 192.168.0.22, Dst: 192.168.0.27
Transmission Control Protocol, Src Port: 49691, Dst Port: 42726, Seq: 1, Ack: 1, Len: 1448
Data (1448 bytes)

0040 dd f5 50 4b 03 04 14 00 00 00 00 00 32 8f 7b 4b PK.....2..K
0050 00 00 00 00 00 00 00 00 00 00 00 00 10 00 00 00
0060 46 72 65 65 52 54 4f 53 76 31 30 2e 30 2e 31 2f FreeRTOS_v10.0.1/
0070 50 4b 03 04 14 00 00 00 00 00 31 8f 7b 4b 00 00 PK.....1..K
0080 00 00 00 00 00 00 00 00 00 00 19 00 00 00 46 72Fr

Fin de la transferencia: (último paquete de datos)

Se observa que el número de secuencia es similar al tamaño del archivo, y la flag de FIN indica la terminación de la transferencia.



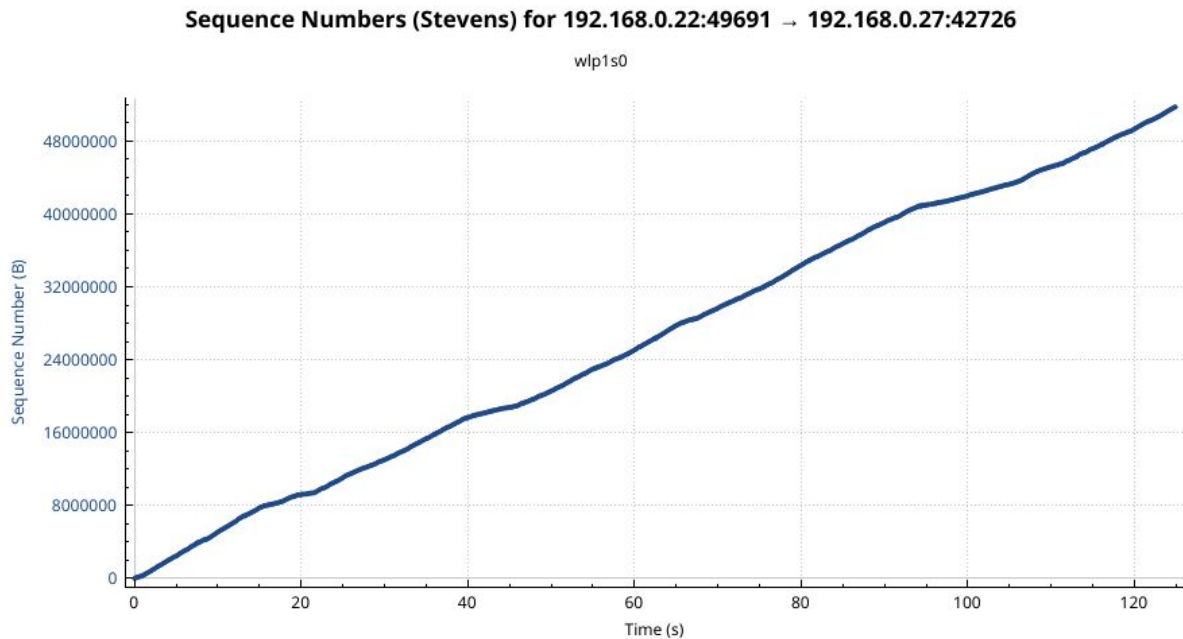
No.	Time	Source	Destination	Protocol	Length	Info
55347	126.181710430	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=51726905 Ack=1 Win=29056 Len=1448...
55348	126.182127735	192.168.0.22	192.168.0.27	TCP	1514	49691 → 42726 [ACK] Seq=51728353 Ack=1 Win=29056 Len=1448...
55349	126.185906634	192.168.0.27	192.168.0.22	TCP	66	42726 → 49691 [ACK] Seq=1 Ack=51705185 Win=577792 Len=0 T...
55350	126.191836060	192.168.0.22	192.168.0.27	TCP	1186	49691 → 42726 [FIN, PSH, ACK] Seq=51729801 Ack=1 Win=2905...
55351	126.205393826	192.168.0.27	192.168.0.22	TCP	66	42726 → 49691 [ACK] Seq=1 Ack=51708001 Win=577792 Len=0 T...
55352	126.207033373	192.168.0.27	192.168.0.22	TCP	66	42726 → 49691 [ACK] Seq=1 Ack=51710977 Win=577792 Len=0 T...
55353	126.207545745	192.168.0.27	192.168.0.22	TCP	66	42726 → 49691 [ACK] Seq=1 Ack=51713873 Win=577792 Len=0 T...
55354	126.207989628	192.168.0.27	192.168.0.22	TCP	66	42726 → 49691 [ACK] Seq=1 Ack=51716769 Win=577792 Len=0 T...
55355	126.208500709	192.168.0.27	192.168.0.22	TCP	66	42726 → 49691 [ACK] Seq=1 Ack=51719665 Win=577792 Len=0 T...
55356	126.208882015	192.168.0.27	192.168.0.22	TCP	66	42726 → 49691 [ACK] Seq=1 Ack=51722561 Win=577792 Len=0 T...

Frame 55350: 1186 bytes on wire (9488 bits), 1186 bytes captured (9488 bits) on interface 0
Ethernet II, Src: IntelCor_f7:f3:23 (78:0c:b8:f7:f3:23), Dst: LiteonTe_3f:0e:70 (3c:a0:67:3f:0e:70)
Internet Protocol Version 4, Src: 192.168.0.22, Dst: 192.168.0.27
Transmission Control Protocol, Src Port: 49691, Dst Port: 42726, Seq: 51729801, Ack: 1, Len: 1120
Data (1120 bytes)
Data: 1400000000008674924bb6c78c42cb1f000036a800002f00...
[Length: 1120]

0040 c5 ae 14 00 00 00 00 00 85 74 92 4b b6 c7 8c 42t.K...B
0050 cd 1f 00 00 36 a8 00 00 2f 00 24 00 00 00 00 00 ...6.../\$.
0060 00 00 20 00 00 00 7a 19 ef 02 46 72 65 65 52 54 ...z...FreeRT
0070 4f 53 76 31 30 2e 30 2e 31 2f 46 72 65 65 52 54 OSv10.0.1/FreeRT
0080 4f 53 2f 53 6f 75 72 63 65 2f 73 74 72 65 61 8d OS/Source/stream

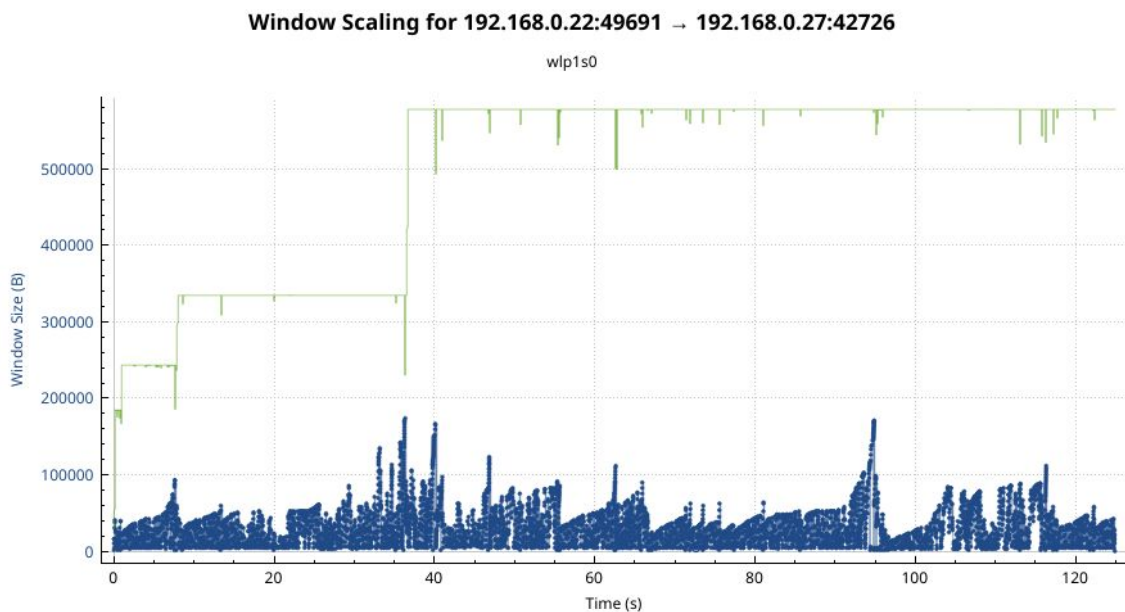
3.4-

1-



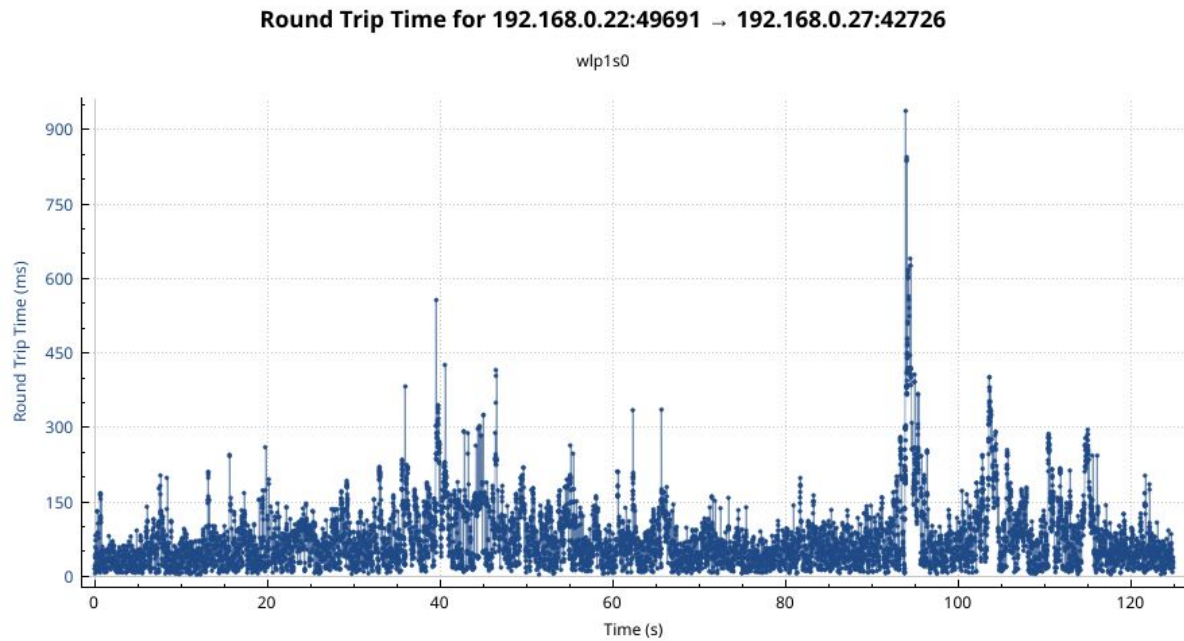
En este gráfico se observan los bytes enviados a lo largo del tiempo. La velocidad de transferencia se calcula aproximadamente con la pendiente de la recta, que es 500KB/s

2-



La línea verde representa el tamaño de la ventana de recepción, por la manera en que funciona TCP, la misma va aumentando porque el receptor, cada vez que recibe paquetes, hace el reconocimiento de todos ellos, por lo que se puede aumentar el tamaño de ventana. La línea azul representa los bytes de salida (enviados por el servidor). Se observa que están muy por debajo del tamaño de ventana, esto puede significar una baja velocidad de subida.

3-



El round-trip time es el tiempo de ida y vuelta de un paquete (tiempo que tarda en viajar del cliente al servidor y volver al cliente). Hay un tiempo máximo que tiene el receptor para enviar el reconocimiento de todos los paquetes que le han sido enviados en el tamaño actual de ventana, si no se cumple este tiempo, el tamaño de ventana se recibe. Alrededor del segundo 95 se observa un pico (algunos paquetes llegaron a demorar 900ms), lo que puede ser explicado con alguna congestión en la red.

Ejercicio 4: Análisis de performance usando iPerf

- 4.1.- Instalar iPerf3 en dos de las PCs de los integrantes del grupo. **NOTA:** debe haber conectividad IP entre server y cliente.
- 4.2.- en el servidor / cliente, ejecutar: `iperf3 -s / iperf3 -c %IP_Server -n 50M`. ¿Cuál es el ancho de banda de la red?
- 4.3.- Ejecutar en el cliente `iperf3 -c #IP_Server -u -b #M` (dónde #M es menor al ancho de banda averiguado en el punto 4.2)
- 4.4.- Ejecutar en el cliente `iperf3 -c #IP_Server -u -b #M` (dónde #M es mayor al ancho de banda averiguado en el punto 4.2)
- 4.5.- Explicar el comportamiento y los resultados.
- 4.6.- Preguntas relacionadas:
¿Qué relación hay entre los resultados vistos en los puntos 4.2 - 4.5 y la NOTA2?

4.2-

Inicio de transferencia: indicando un tamaño de paquete de 50MBytes

```
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos$ iperf3 -c 192.168.0.22 -n 50M
connecting to host 192.168.0.22, port 5201
 4] local 192.168.0.27 port 58420 connected to 192.168.0.22 port 5201
ID] Interval          Transfer    Bandwidth    Retr  Cwnd
 4]  0.00-1.00   sec    556 KBytes  4.55 Mbits/sec    0   33.9 KBytes
 4]  1.00-2.00   sec    595 KBytes  4.88 Mbits/sec    0   38.2 KBytes
 4]  2.00-3.00   sec    615 KBytes  5.04 Mbits/sec    0   46.7 KBytes
 4]  3.00-4.00   sec    742 KBytes  6.08 Mbits/sec    0   58.0 KBytes
 4]  4.00-5.00   sec    766 KBytes  6.28 Mbits/sec    0   73.5 KBytes
 4]  5.00-6.00   sec    744 KBytes  6.09 Mbits/sec    0   107 KBytes
```

Fin de la transmisión: se observa un ancho de banda alcanzado de 5.6Mb/s

Al ser el ancho de banda del servidor más grande que el del receptor, se “pierden” datos (se retransmiten 7), esto se debe a que se envían datos más rápidos de lo que se pueden recibir.

```
[ 5] 67.00-68.00   sec    713 KBytes  5.84 Mbits/sec
[ 5] 68.00-69.00   sec    532 KBytes  4.36 Mbits/sec
[ 5] 69.00-70.00   sec    471 KBytes  3.86 Mbits/sec
[ 5] 70.00-71.00   sec    737 KBytes  6.04 Mbits/sec
[ 5] 71.00-72.00   sec    837 KBytes  6.86 Mbits/sec
[ 5] 72.00-73.00   sec    897 KBytes  7.34 Mbits/sec
[ 5] 73.00-74.00   sec    757 KBytes  6.20 Mbits/sec
[ 5] 74.00-74.01   sec     8.48 KBytes  6.57 Mbits/sec
-----
[ ID] Interval          Transfer    Bandwidth    Retr
[ 5]  0.00-74.01   sec   50.0 MBytes  5.67 Mbits/sec    7
[ 5]  0.00-74.01   sec   49.7 MBytes  5.63 Mbits/sec
-----
Server listening on 5201
sender
receiver
```

4.3-

Inicio transferencia UDP con ancho de banda menor al detectado anteriormente:

Servidor: se observa la pérdida de algunos paquetes (no se retransmiten)

```
Accepted connection from 192.168.0.27, port 58552
[ 5] local 192.168.0.22 port 5201 connected to 192.168.0.27 port 47379
[ ID] Interval          Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 5] 0.00-1.00 sec      536 KBytes   4.38 Mbits/sec  10.908 ms    1/68 (1.5%)
[ 5] 1.00-2.00 sec      600 KBytes   4.93 Mbits/sec  14.845 ms    0/75 (0%)
[ 5] 2.00-3.00 sec      616 KBytes   5.05 Mbits/sec  16.399 ms    0/77 (0%)
[ 5] 3.00-4.00 sec      616 KBytes   5.05 Mbits/sec  10.021 ms    1/78 (1.3%)
[ 5] 4.00-5.00 sec      552 KBytes   4.52 Mbits/sec  11.998 ms    7/76 (9.2%)
[ 5] 5.00-6.00 sec      568 KBytes   4.65 Mbits/sec  18.773 ms    0/71 (0%)
[ 5] 6.00-7.00 sec      624 KBytes   5.11 Mbits/sec  19.871 ms    0/78 (0%)
[ 5] 7.00-8.00 sec      640 KBytes   5.24 Mbits/sec  13.556 ms    0/80 (0%)
[ 5] 8.00-9.00 sec      576 KBytes   4.72 Mbits/sec  17.460 ms    2/74 (2.7%)
[ 5] 9.00-10.00 sec     616 KBytes   5.04 Mbits/sec  12.205 ms    0/77 (0%)
[ 5] 10.00-10.05 sec    16.0 KBytes   2.82 Mbits/sec  11.233 ms    0/2 (0%)
-----
[ ID] Interval          Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 5] 0.00-10.05 sec     5.91 MBytes   4.93 Mbits/sec  11.233 ms   11/756 (1.5%)
-----
```

Cliente: indica UDP y limita el ancho de banda a 5Mb/s

```
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos$ iperf3 -c 192.168.0.22 -u -b 5M
Connecting to host 192.168.0.22, port 5201
[ 4] local 192.168.0.27 port 47379 connected to 192.168.0.22 port 5201
[ ID] Interval          Transfer      Bandwidth      Total Datagrams
[ 4] 0.00-1.00 sec      552 KBytes   4.52 Mbits/sec    69
[ 4] 1.00-2.00 sec      616 KBytes   5.05 Mbits/sec    77
[ 4] 2.00-3.00 sec      608 KBytes   4.98 Mbits/sec    76
[ 4] 3.00-4.00 sec      608 KBytes   4.98 Mbits/sec    76
[ 4] 4.00-5.00 sec      608 KBytes   4.98 Mbits/sec    76
[ 4] 5.00-6.00 sec      616 KBytes   5.04 Mbits/sec    77
[ 4] 6.00-7.00 sec      608 KBytes   4.98 Mbits/sec    76
[ 4] 7.00-8.00 sec      608 KBytes   4.98 Mbits/sec    76
[ 4] 8.00-9.00 sec      616 KBytes   5.05 Mbits/sec    77
[ 4] 9.00-10.00 sec     608 KBytes   4.98 Mbits/sec    76
-----
[ ID] Interval          Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 4] 0.00-10.00 sec     5.91 MBytes   4.95 Mbits/sec  11.233 ms   11/756 (1.5%)
[ 4] Sent 756 datagrams
```

4.4-

Inicio transferencia UDP con ancho de banda mayor al detectado anteriormente:

Servidor: se observa una pérdida considerable de paquetes debido a la exageración en el ancho de banda solicitado.

```
Accepted connection from 192.168.0.27, port 58712
[ 5] local 192.168.0.22 port 5201 connected to 192.168.0.27 port 34696
[ ID] Interval          Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 5] 0.00-1.00 sec      672 KBytes   5.50 Mbits/sec   7.571 ms    19/103 (18%)
[ 5] 1.00-2.00 sec      464 KBytes   3.80 Mbits/sec  12.933 ms   68/126 (54%)
[ 5] 2.00-3.00 sec      272 KBytes   2.23 Mbits/sec  13.667 ms   98/132 (74%)
[ 5] 3.00-4.00 sec      168 KBytes   1.38 Mbits/sec  10.270 ms  98/119 (82%)
[ 5] 4.00-5.00 sec      104 KBytes   852 Kbits/sec   14.365 ms  139/152 (91%)
[ 5] 5.00-6.00 sec      56.0 KBytes  459 Kbits/sec   13.940 ms  158/165 (96%)
[ 5] 6.00-7.00 sec      136 KBytes   1.11 Mbits/sec   9.630 ms  175/192 (91%)
[ 5] 7.00-8.00 sec      16.0 KBytes  131 Kbits/sec   9.353 ms  161/163 (99%)
[ 5] 8.00-9.00 sec      128 KBytes   1.05 Mbits/sec  13.896 ms  146/162 (90%)
[ 5] 9.00-10.00 sec     176 KBytes   1.44 Mbits/sec  13.052 ms  115/137 (84%)
[ 5] 10.00-10.07 sec     8.00 KBytes   956 Kbits/sec  12.442 ms   14/15 (93%)
-----
[ ID] Interval          Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 5] 0.00-10.07 sec    11.5 MBytes   9.60 Mbits/sec  12.442 ms  1191/1466 (81%)
-----
```

Ciente: indica UDP y especifica un ancho de banda de 10Mb/s

```
andres@andres:/media/andres/F147-BDF2/Facultad/ComDatos$ iperf3 -c 192.168.0.22 -u -b 10M
Connecting to host 192.168.0.22, port 5201
[ 4] local 192.168.0.27 port 34696 connected to 192.168.0.22 port 5201
[ ID] Interval          Transfer      Bandwidth    Total Datagrams
[ 4] 0.00-1.00 sec      944 KBytes   7.73 Mbits/sec  118
[ 4] 1.00-2.00 sec     1.02 MBytes  8.59 Mbits/sec  131
[ 4] 2.00-3.00 sec     1.00 MBytes  8.39 Mbits/sec  128
[ 4] 3.00-4.00 sec     1.08 MBytes  9.04 Mbits/sec  138
[ 4] 4.00-5.00 sec     1.22 MBytes  10.2 Mbits/sec  156
[ 4] 5.00-6.00 sec     1.31 MBytes  11.0 Mbits/sec  168
[ 4] 6.00-7.00 sec     1.33 MBytes  11.1 Mbits/sec  170
[ 4] 7.00-8.00 sec     1.38 MBytes  11.6 Mbits/sec  177
[ 4] 8.00-9.00 sec     1.11 MBytes  9.30 Mbits/sec  142
[ 4] 9.00-10.00 sec    1.15 MBytes  9.63 Mbits/sec  147
- - - - -
[ ID] Interval          Transfer      Bandwidth    Jitter      Lost/Total Datagrams
[ 4] 0.00-10.00 sec    11.5 MBytes  9.67 Mbits/sec  12.442 ms  1191/1466 (81%)
[ 4] Sent 1466 datagrams
```

4.5.-

El programa se utilizó para medir el rendimiento de la red y observar algunos parámetros característicos. Se establece una conexión TCP (o UDP si se especifica) y se envían datos para probar el rendimiento de la conexión y el ancho de banda. Cada cierto intervalo de tiempo especificado (1 segundo por defecto) se indican estos parámetros y el total de datos transmitidos.

4.6.-

Al usar una conexión Wi-Fi, el ancho de banda es menor al que se hubiera obtenido usando, por ej., Ethernet, y también se observa una variación mayor en la red (menos estabilidad por las interferencias electromagnéticas). Las variaciones en el ancho de banda también serían menores con una conexión cableada.