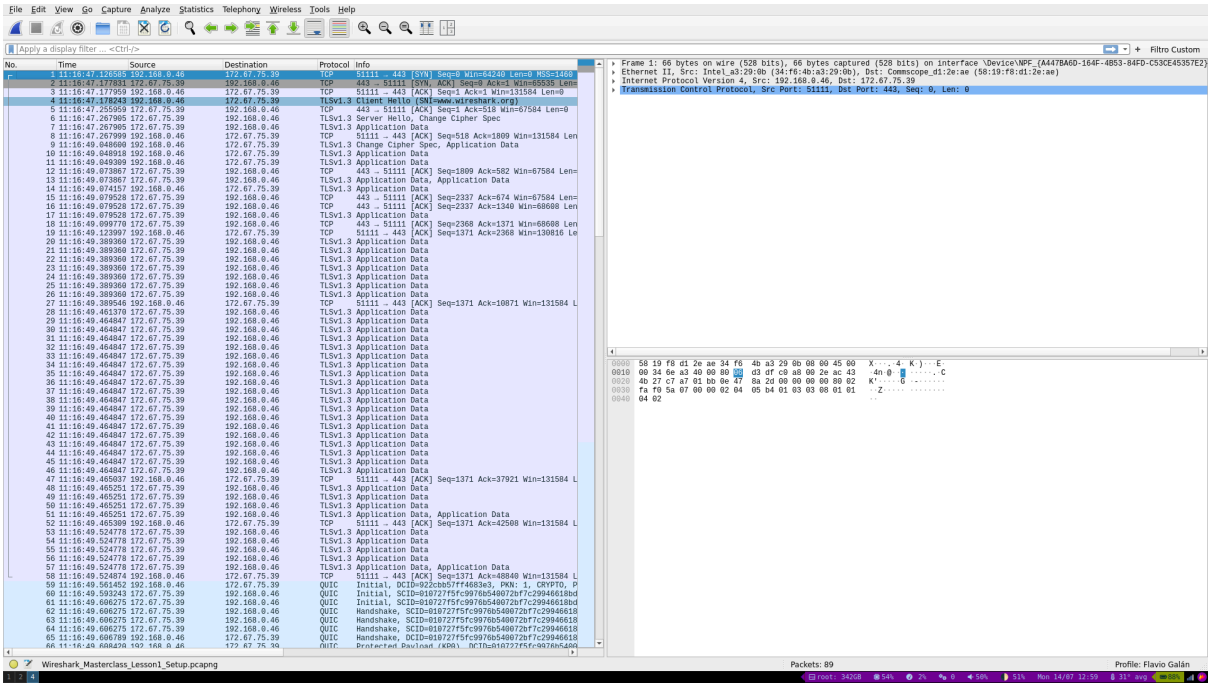
Lab 1.2 - Redes

Flavio Galán - 22386

Configuración Ambiente

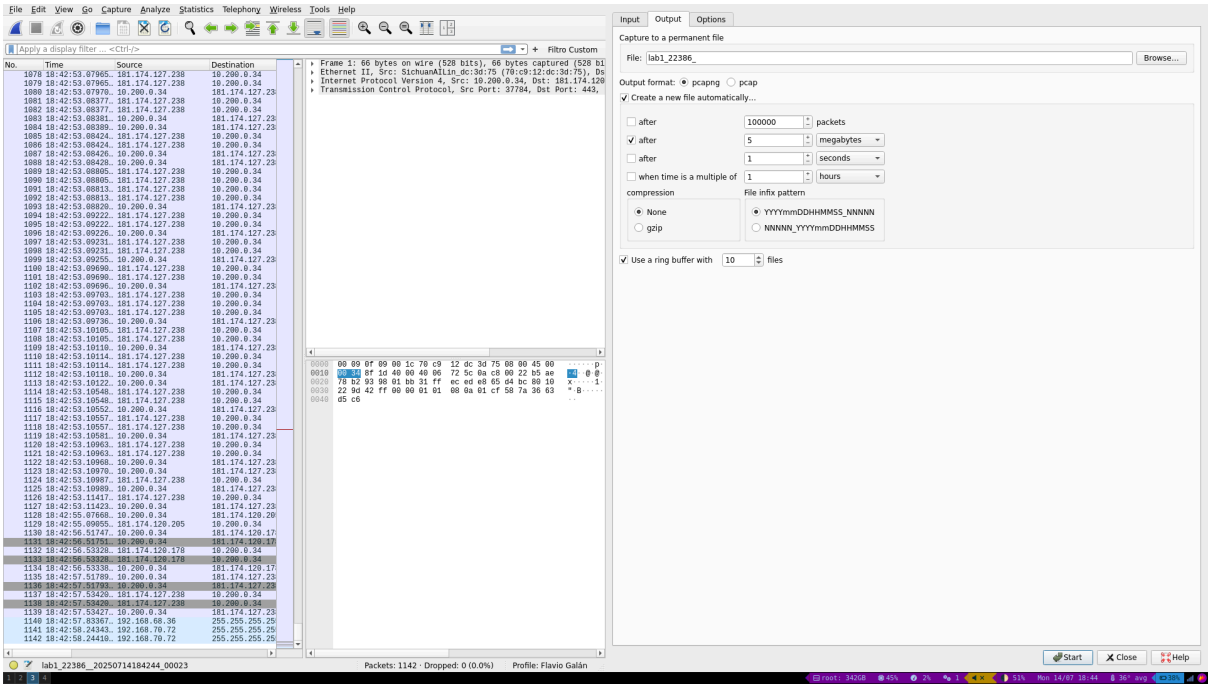


Configuración de Captura de Paquetes

```
[elrohirgt@elrohirgt:~]$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:1e:a8:c1:90  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vboxnet0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.56.1  netmask 255.255.255.0  broadcast 0.0.0.0
        ether 0a:00:27:00:00:00  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.200.0.34  netmask 255.255.240.0  broadcast 10.200.15.255
        inet6 fe80::1154:cf47:1c61:ca8c  prefixlen 64  scopeid 0x20<link>
        ether 70:c9:12:dc:3d:75  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


[elrohirgt@elrohirgt:~]$
```

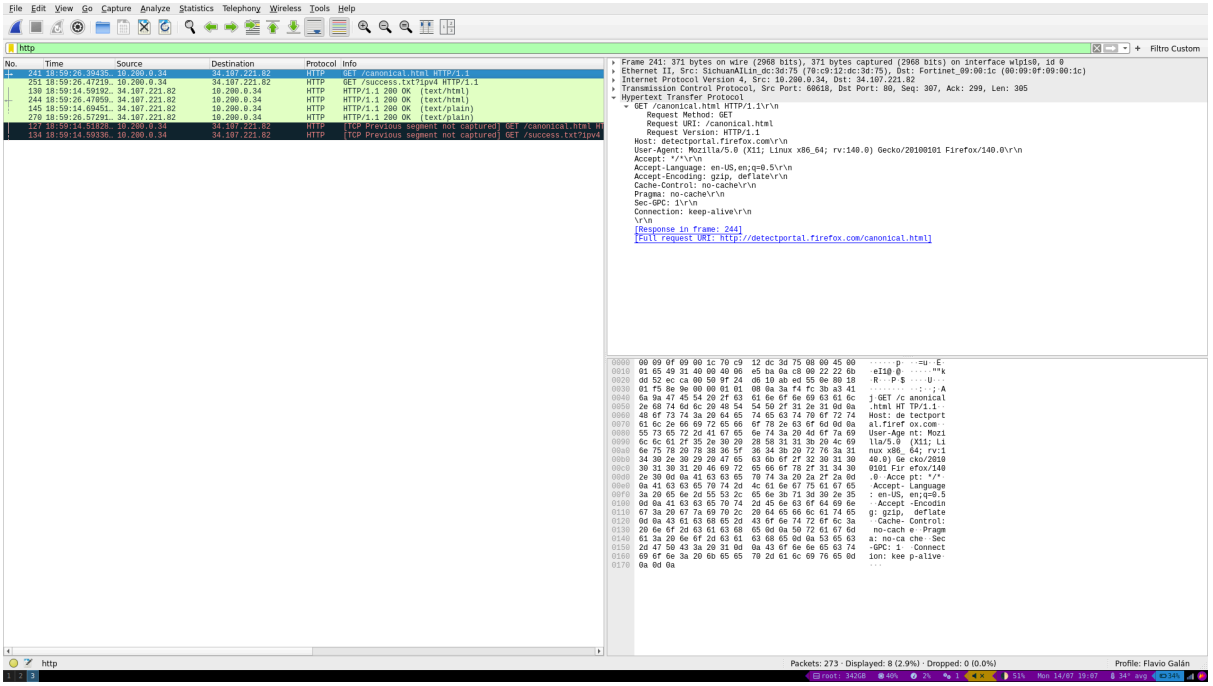
El comando que ejecuté es ifconfig ya que estoy en Linux.
Este comando te permite configurar las interfaces de red del sistema. Además tiene varias otras funciones de listado de información como el listar todas las interfaces como se ve en la captura anterior.

Captura de paquetes con wireshark



a. ¿Qué versión de HTTP está ejecutando su navegador?
1.1
b. ¿Qué versión de HTTP está ejecutando el servidor?
1.1
c. ¿Qué lenguajes (si aplica) indica el navegador que acepta a el servidor?
en-US
d. ¿Cuántos bytes de contenido fueron devueltos por el servidor?
90+90+8+8 = 196 bytes de contenido como tal, todo el resto fue overhead.
e. En el caso que haya un problema de rendimiento mientras se descarga la página ¿en que elementos de la red convendría “escuchar” los paquetes? ¿Es conveniente instalar Wireshark en el servidor? Justifique.

Si se tiene un problema de descarga de la página y solamente se tiene el problema en con esa página, entonces si es mejor instalar wireshark en el server. De lo contrario es más probable que el error sea del cliente, por lo que sería mejor instalar wireshark en la computadora que hace la request y ver si da o no errores ahí.



Discusión

Me pareció una actividad interesante, tuve algunos problemas al utilizar wireshark y obtener paquetes, principalmente con que no me registraba los paquetes HTTP de la página. Tuve que volver a grabar 3 veces invalidando la cache y ahí si me salieron los eventos de la captura de arriba.
Me sorprendió lo poco que ocupan los paquetes ya que grabé 50MB teniendo 3 videos de YT abiertos corriendo al mismo tiempo y aún así se tardó un su buen tiempo en llenarse.
Por último me sorprendió un poco que aunque ya vamos por HTTP 3 (que asco esa versión), se siga usando la versión 1.1. Que viva el HTTP 1.1.

Conclusiones

- Los paquetes aunque tienen un alto grado de overhead debido a que son varios protocolos los que se pueden ver en wireshark para cada paquete. Siguen siendo bien pequeños.
- Wireshark puede ser una herramienta útil al debuggear problemas de red de bajo nivel o en sistemas IoT.