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. Commer 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?

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.

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?

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?