Trabajo Práctico Nº2

Redes de Computadoras

Contenido

Condiciones para Aprobar

Entregable

Para la evaluación del presente trabajo, deben realizar los siguientes puntos:

- **Mail :** Enviar en formato digital con asunto “TP2: Sor1-1S-2019” e indicar en el cuerpo los integrantes del grupo. Adjuntar el código fuente de su implementación y un informe del trabajo realizado, dificultades encontradas y pseudocódigo de las soluciones propuestas.
- **Defensa/Demo :** El dia de la entrega debe mostrar a su docente la solución enviada funcionando en la computadora del laboratorio.

Puntaje / Calificación

Se califica con las notas:

- ★ I (insuficiente),
- ★ A- (aprobado menos, no puede tener dos A- en la cursada),
- ★ A (aprobado)
- ★ A+ (aprobado más, redondea para arriba la nota final en caso de promocionar)

Recuperatorio

En caso de no aprobar tiene un plazo de dos semanas para entregar el TP con las correcciones indicadas durante la demo (en recuperatorio no se pone A+). La demo del tp recuperatorio es de forma individual.

Introducción

Los objetivos de este trabajo son:

- Consolidar al alumno en el uso de la línea de comandos.
- Familiarizar al alumno con los elementos básicos de la administración de redes.
- Fomentar el trabajo en equipo

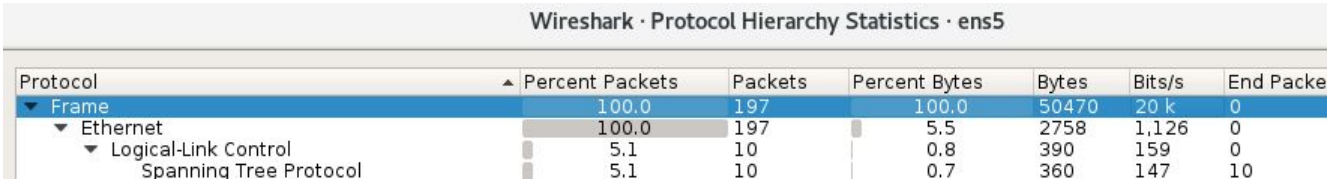
Capa Física

- Desde la línea de comandos investigar qué interfaces de red tiene disponibles. Que es una interfaz de red?
  Puede aplicar el comando: ip a ó ifconfig

- Instalar el analizador de tráfico wireshark
  sudo apt-get install wireshark

- Ejecutar wireshark y escuchar el tráfico en diferentes interfaces de red:
  - Wireless
  - Ethernet
  - Localhost

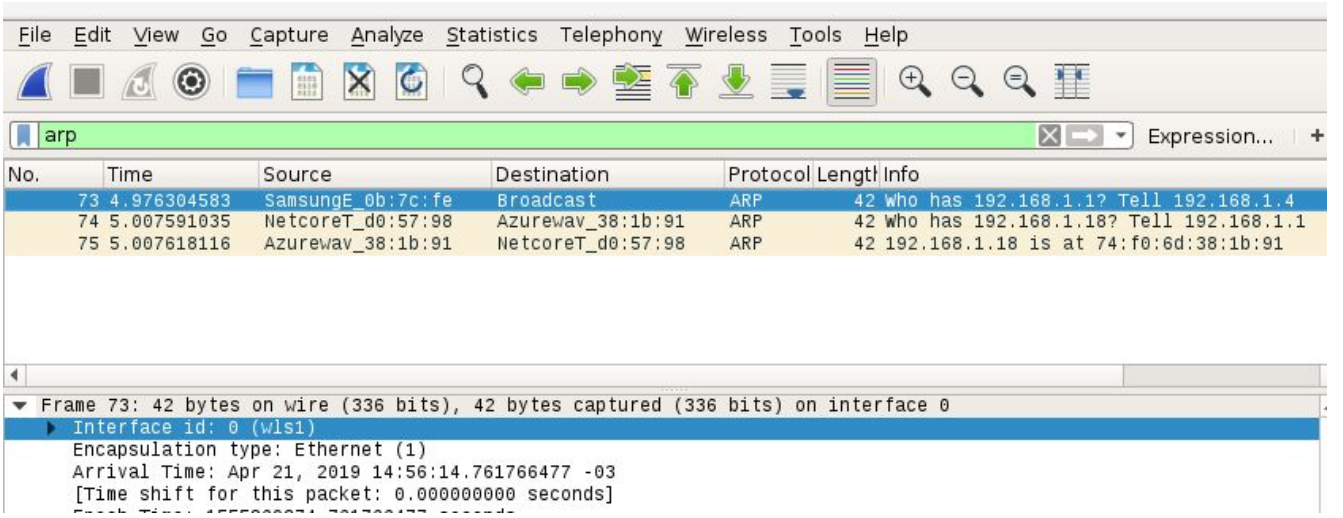
- Determinar el ancho de banda digital de su conexión a internet. Puede usar wireshark y la opción del menú statistics>protocol hierarchy. También puede graficar junto con la opción statistics>IO graph.



Capa de Enlace

- Investigar el protocolo ARP y su relación con las direcciones MAC.

- Utilizando wireshark realizar una captura de tal forma de identificar un envío ARP y su respuesta. Puede realizarlo sobre una interfaz con mucho tráfico y aplicando un filtro para el protocolo arp, por ejemplo, sobre la interfaz wireless wls1 tal como se muestra en la siguiente figura:



- Sobre su captura identificar los siguientes datos
    - MAC del origen
    - IP del destino
    - MAC del destino

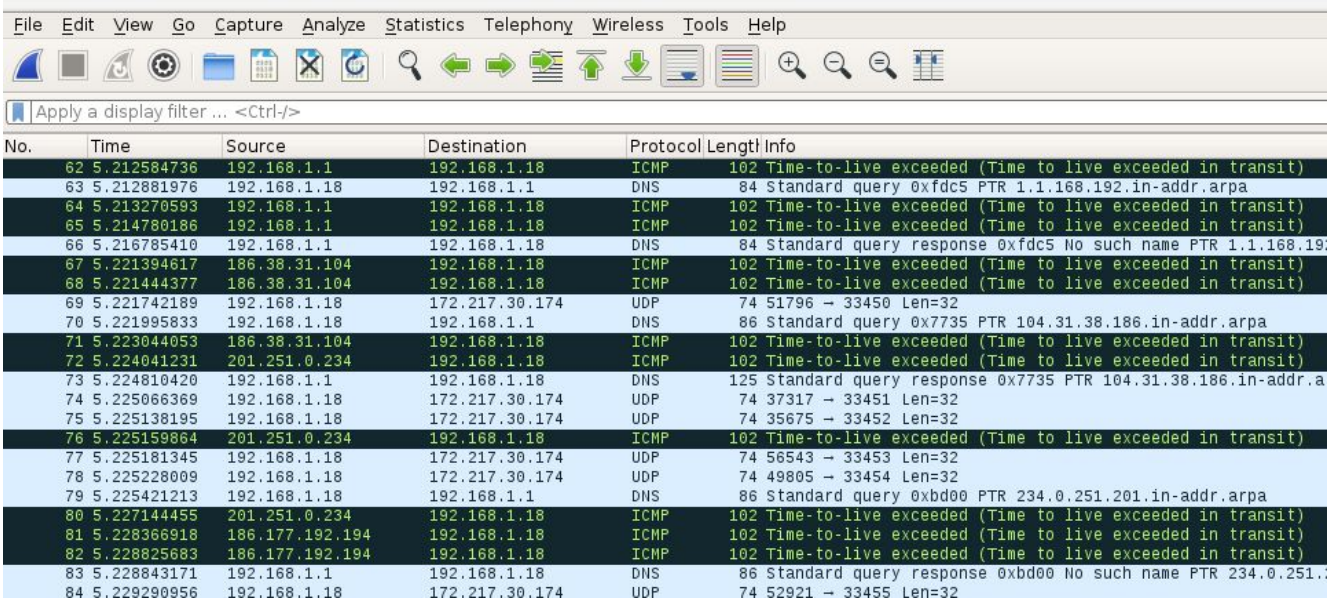
- Cuando se envía la pregunta “who has ip ….. ” porqué se realiza un broadcast? En ese momento qué datos son desconocidos para el que envía?

- Cuando llega la respuesta “ ip ….. is …...” qué nuevo dato se aprende?

Capa de Red

Traceroute

- Investigar cómo funciona traceroute y realizar un diagrama explicativo.
- Instalar traceroute: sudo apt-get install traceroute
- Ejecutar traceroute mientras realiza una captura con wireshark. Que función cumple el protocolo ICMP? Obtener por ejemplo:



- De los datos capturados anteriormente identificar:

- Qué direcciones ip aparecen en la columna Source o fuente, a qué hosts pertenecen estas direcciones ip?
- Porque los mensajes tienen el campo info igual a “Time-to-live exceeded”?
- Comparar estos datos con la salida standard de traceroute y marcar similitudes y diferencias, por ejemplo su salida puede ser:

```
~$ traceroute google.com
traceroute to google.com (172.217.30.174), 30 hops max, 60 byte packets
 1  gateway (192.168.1.1)  2.250 ms  2.855 ms  4.311 ms
 2  186-38-31-104.mrse.com.ar (186.38.31.104)  10.893 ms  10.912 ms  12.482 ms
 3  201.251.0.234 (201.251.0.234)  13.445 ms  14.534 ms  16.488 ms
 4  rtrpc1.interbourg.com.ar (186.177.192.194)  17.678 ms  18.106 ms  19.825 ms
 5  200.51.235.29 (200.51.235.29)  29.175 ms  30.814 ms  31.544 ms
 6  74.125.52.126 (74.125.52.126)  33.249 ms  23.985 ms  23.042 ms
 7  108.170.248.241 (108.170.248.241)  23.367 ms  24.609 ms 108.170.248.225 (108.170.248.225)
 8  216.239.62.245 (216.239.62.245)  23.022 ms  24.215 ms  25.311 ms
 9  eze03s36-in-f14.1e100.net (172.217.30.174)  22.012 ms  23.715 ms  24.028 ms
```

Capa de Transporte

Socket programming

En este ejercicio vamos a trabajar con un sistema cliente-servidor: puede encontrar los archivos en los links cliente.c y servidor.c. Realizar los siguientes puntos:

- Compilar el archivo server.c y ejecutarlo. En otra terminal compilar el archivo client.c y ejecutarlo. Se trata de un servidor “eco”. Qué comportamiento observa?

- Modificar los valores del socket para que cliente y servidor se ejecuten en diferentes computadoras del laboratorio.

- Realizar una captura del tráfico entre el cliente y el servidor y recopilar los siguientes datos:
  - Elegir una interfaz de red sobre la cual escuchar: sobre el cliente o sobre el servidor.
  - Identificar el momento del *three way handshake* y detallar la información que se intercambia. Como se utiliza esta información para implementar la confiabilidad de TCP?
  - Realizar un seguimiento del texto plano que se intercambió entre cliente y servidor. Puede usar la opción de wireshark Menu>Analize>Follow>TCPstream
  - Qué cambios piensa que se debe hacer en la comunicación para que la captura de wireshark no se pueda leer o entender como en el punto anterior?
  - Completar la siguiente tabla:

	Nro de IP	Nro de puerto
Socket cliente		
Socket servidor		

Capa de Aplicación

SSH y SCP

- Loguearse a otro host remoto con **ssh**. Investigar cual es la diferencia entre ssh y **telnet**. Realizar un login remoto con la posibilidad de lanzar aplicaciones gráficas. Una vez logueado, el usuario debería poder ejecutar programas de interfaz gráfica como gedit, emacs ó wireshark.

- Realizar una transferencia de archivos entre diferentes hosts. Usar **scp**. Cual es la relación entre ssh y scp?

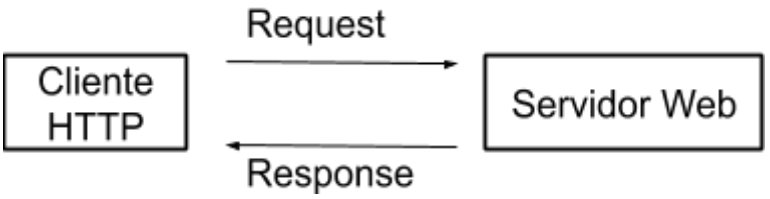
SSH y firewall con iptables

Averiguar su dirección IP.
Dar su ip, usuario y password a otro compañero para que se loguee a su computadora
Agregar una restricción de acceso con iptables
Verificar que su compañero ya no se puede conectar

- Agregar restricciones de acceso a su host, por ejemplo con iptables
    - Averiguar su dirección IP.
    - Dar su ip, usuario y password a otro compañero para que se loguee a su computadora
    - Agregar una restricción de acceso con iptables (sudo iptables -A INPUT -s 10.10.10.22 -j DROP)
    - Verificar que su compañero ya no se puede conectar
    - Remover la regla para volver su configuración a la normalidad (sudo iptables -D INPUT -s 10.10.10.22 -j DROP).
    - Verificar las reglas presentes en iptables (sudo iptables -L)

HTTP y Proxy

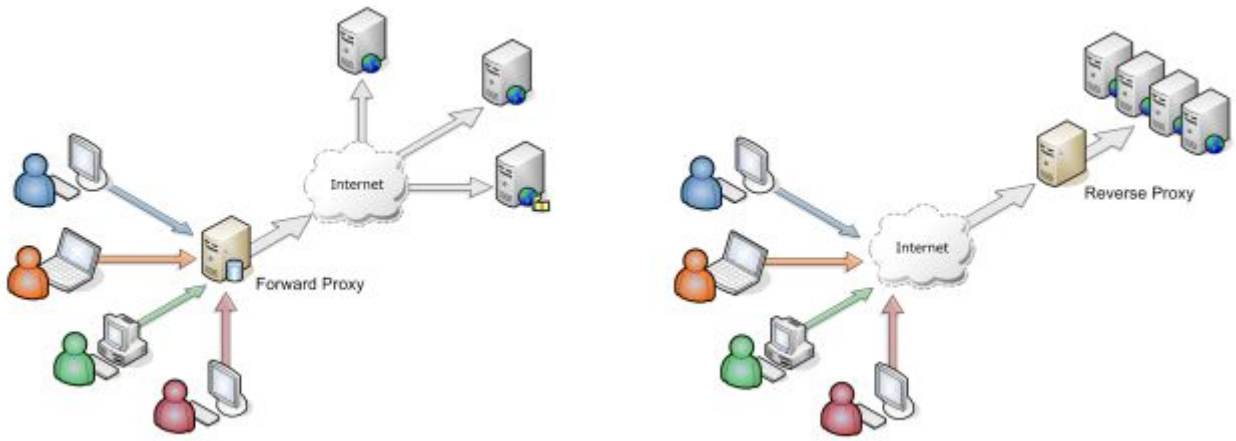
Para visualizar un página web, la comunicación entre dos hosts se realiza con intercambio de mensajes del protocolo HTTP. Estos mensajes pueden ser pedidos o requests (GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT) y también pueden ser respuestas o responses. Idealmente cada request tiene asociado un response:



Por ejemplo, desde la línea de comandos puede obtener el html de la página de google con
GET www.google.com

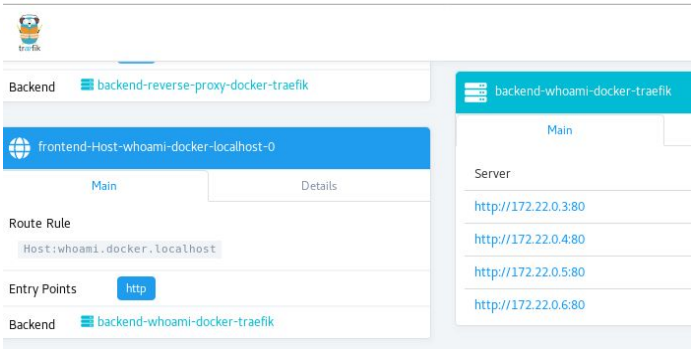
El gráfico anterior se puede completar agregando algunos intermediarios en la comunicación entre cliente y servidor a nivel del protocolo http. Por un lado y desde el lado del cliente podemos agregar

un servidor intermedio llamado servidor proxy cuya función es dar servicios de caché y proteger a los usuarios del mundo exterior (ver siguiente figura, lado izquierdo). Por otro lado podemos agregar en el lado del servidor un proxy reverso cuya función principal es proveer balanceo de carga entre diferentes servidores (ver siguiente figura, lado derecho).



En este ejercicio veremos una implementación de proxy reverso con la tecnología Docker que se ha ganado un lugar de importancia a la hora de montar servidores e infraestructura para aplicaciones web.

- Instalar Docker Community Edition (CE) (link)
- Instalar Docker Compose (link)
- Seguir las instrucciones del proxy reverso Traefik - Getting Started (link)
- Ingresar al panel de traefik, que debería quedar funcionando en http://localhost:8080/dashboard/



En la imagen anterior se observa que el servidor **whoami** está replicado 4 veces, eso permite realizar consultas de “who am I” desde la terminal y las mismas son redirigidas por traefik a servidores diferentes, por ejemplo:
curl -H Host:whoami.docker.localhost http://127.0.0.1; ejecutándose varias veces debe devolver la dirección ip de diferentes servidores.

- Reproducir la imagen anterior y verificar que efectivamente se accede a diferentes servidores.

Anexo: Instalación de Docker

Puede seguir las instrucciones de la página oficial:

- Instalar Docker Community Edition (CE) (link)
- Instalar Docker Compose (link)

y comparar con la siguiente instalación en un entorno Debian: link

Anexo: Instalación y edición de traceroute

Instalar: sudo apt-get install help2man
(es el programa que necesita Inetutils para generar su manual)

Bajar el código fuente (http://ftp.gnu.org/gnu/inetutils/inetutils-1.9.4.tar.gz) de la página oficial: http://ftp.gnu.org/gnu/inetutils/

Bajar y descomprimir inetutils-1.9.4.tar.gz, (tar xvzf inetutils-1.9.4.tar.gz)

se recomienda crear una carpeta “local” en su home, donde pueda descargar el código, por ejemplo /home/sor1/local/.

Ingresar a la carpeta donde esta el código, por ejemplo:
cd /home/sor1/local/inetutils-1.9.4/

Preparar Inetutils para compilar, ejecutar el script configure que verifica que todos los prerrequisitos están listos: ./configure

Compilar el código fuente, se realiza con un comando muy usado para compilar grandes paquetes con muchas dependencias: make

Acceder a la carpeta inetutils-1.9.4/src donde esta el código traceroute.c y el ejecutable previamente generado con el comando make. Ahora verificar que pueden usar el programa traceroute:
sudo ./traceroute www.ungs.edu.ar