

Práctico Nº 8

Año 2025

Objetivo: uso WireShark y TCP

1) Manteniendo en funcionamiento el Wireshark, realice una captura de tráfico con la ejecución de los siguientes comandos. Desde la consola de windows o de Linux

1- Ping www.google.com

2- Tracert -d www.google.com

Muestre por pantalla las salidas que se obtiene al ejecutar el comando: ping www.google.com

De la ejecución anterior, realice una captura e indique qué protocolos puede identificar.

Idem con: Tracert -d www.yahoo.com


Idem con Tracert -d www.unl.edu.ar

Ejecutar ping www.google.com desde la consola

ping www.google.com

¿Qué hace esto?


Envía paquetes ICMP type Echo Request al servidor de Google y espera respuestas tipo Echo Reply.

 Captura esperada en Wireshark:

Protocolo: ICMP

Tramas: Request (eco) y Reply (respuesta)

Dirección destino: la IP de Google (ej. 142.250.xxx.xxx)

 Salida por consola:

ping www.google.com

Haciendo ping a www.google.com [142.251.129.36] con 32 bytes de datos:

Respuesta desde 142.251.129.36: bytes=32 tiempo=311ms TTL=116

Respuesta desde 142.251.129.36: bytes=32 tiempo=53ms TTL=116

Respuesta desde 142.251.129.36: bytes=32 tiempo=22ms TTL=116

Respuesta desde 142.251.129.36: bytes=32 tiempo=29ms TTL=116

Estadísticas de ping para 142.251.129.36:

Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos), Tiempos aproximados de ida y vuelta en milisegundos: Mínimo = 22ms, Máximo = 311ms, Media = 103ms

Ejecutar tracert -d www.google.com

¿Qué hace esto?


Práctico Nº 8

Año 2025

Objetivo: uso WireShark y TCP

Envía paquetes ICMP con distintos TTL (Time-To-Live) para identificar los routers intermedios hasta llegar al destino.


El -d evita la resolución DNS y muestra IPs directamente.

 Captura esperada en Wireshark:

Protocolo: ICMP

IPs de varios routers (saltos)

ICMP Type 11 (Time Exceeded) y luego Echo Reply

 Salida esperada en consola:

tracert -d www.google.com

Traza a la dirección www.google.com [142.251.129.36] sobre un máximo de 30 saltos:


```
1  10 ms  6 ms  8 ms 192.168.0.1
2  *      *      *  Tiempo de espera agotado para esta solicitud.
3  *      *      *  Tiempo de espera agotado para esta solicitud.
4  18 ms  15 ms  13 ms 181.96.62.110
5  *      *      *  Tiempo de espera agotado para esta solicitud.
6  *      *      *  Tiempo de espera agotado para esta solicitud.
7  *      17 ms  *    181.89.51.39
8  *      24 ms  26 ms 72.14.194.198
9  21 ms  20 ms  23 ms 108.170.255.29
10 25 ms  25 ms  21 ms 142.251.239.155
11 30 ms  24 ms  24 ms 142.251.129.36
```

Traza completa

 ¿Qué protocolos vas a ver en Wireshark?

Comando	Protocolos identificados en Wireshark
ping www.google.com	ICMP, IP, ARP
tracert -d www.google.com	ICMP, IP
tracert -d www.yahoo.com	ICMP, IP
tracert -d www.unl.edu.ar	ICMP, IP

También podés ver DNS si no usás -d, ya que intenta resolver los nombres.

 Resumen/Conclusiones

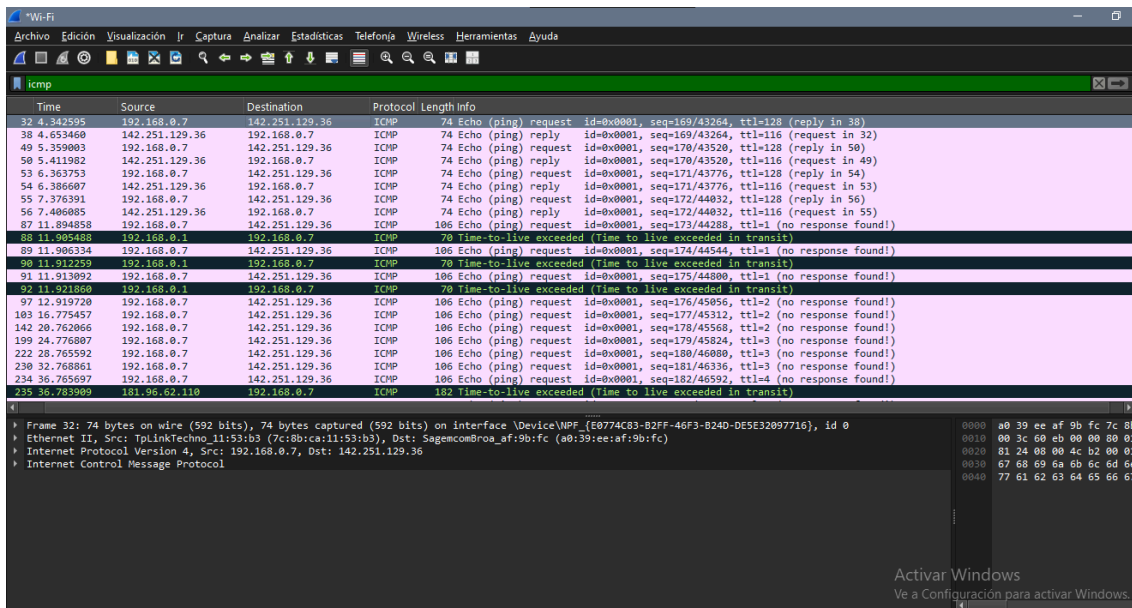
El comando ping utiliza ICMP Echo Request y Reply.

El comando tracert usa ICMP, pero genera también respuestas de tipo Time Exceeded.

La opción -d en tracert evita tráfico DNS.

Wireshark permite ver en detalle cada paquete enviado, su protocolo, TTL, respuesta y tamaño.

Con estas herramientas podés analizar la conectividad y el camino hacia un destino en red.



2) Realice un flushdns antes de realizar este ejercicio. Manteniendo en funcionamiento el Wireshark, realice una captura de tráfico ingresando a la dirección web que no haya ingresado en los últimos días, una vez que ingrese al sitio corte la captura.

- identifique uno de los paquetes de la conexión y establezca el filtro de conversión “TCP”
- identifique los 3 paquetes del “three handshake”. Analice sus datos, sobre todo las banderas de cada uno.

1) Limpiar caché DNS

Antes de comenzar, ejecutá el siguiente comando en consola de Windows:

ipconfig /flushdns

¿Por qué?

Esto elimina las direcciones DNS guardadas localmente, lo que obliga al navegador a hacer una nueva consulta DNS. Así Wireshark puede capturar el proceso completo de conexión, incluyendo la resolución del dominio y el handshake.

2) Abrir Wireshark y comenzar captura

Abrí Wireshark.

Seleccioná la interfaz de red activa (por ejemplo, Wi-Fi).

Iniciá la captura antes de abrir el navegador.

No pongas filtros todavía, así ves todo el proceso.

3) Ingresar a un sitio nuevo

Práctico Nº 8

Año 2025

Objetivo: uso WireShark y TCP

Abrió un navegador (Chrome, Firefox, etc.) y entró a un sitio que no hayas visitado recientemente. Por ejemplo:

<https://www.udec.cl>

Esto genera una nueva conexión TCP y tráfico DNS/TLS que Wireshark va a capturar.

Cuando la página cargue, detené la captura.

4) Aplicar filtro en Wireshark

Para ver sólo el tráfico TCP, aplicá este filtro:

tcp

¿Por qué?


Porque el protocolo TCP es el que establece las conexiones mediante el three-way handshake.

5) Buscar los 3 paquetes del three-way handshake

Buscá tres paquetes consecutivos entre el cliente (tu IP) y el servidor web (la IP pública del sitio).

Los tres paquetes son:

Nº	Flag	Descripción	Campo en Wireshark
1	SYN	El cliente solicita conexión	Flags: 0x002 (SYN)
2	SYN, ACK	El servidor acepta y responde	Flags: 0x012 (SYN, ACK)
3	ACK	El cliente confirma	Flags: 0x010 (ACK)

 Ayuda visual en Wireshark:

En la columna "Info" vas a ver algo como:

SYN → Client → Server

SYN, ACK → Server → Client

ACK → Client → Server

 Ejemplo:

No.	Source	Destination	Protocol	Info
12	192.168.1.6	45.60.32.5	TCP	[SYN]
13	45.60.32.5	192.168.1.6	TCP	[SYN, ACK]
14	192.168.1.6	45.60.32.5	TCP	[ACK]

6) Análisis de cada paquete (abre el panel inferior)

En cada uno, expandí el bloque:

Transmission Control Protocol

Y revisá estos campos:

Flags: Aquí ves SYN, ACK activados.

UNL-FICH Dpto. de Informática
Docentes: Gabriel Filippa - Franco Cian - Marcelo T. Gentile – Joaquín Nepotti
Redes y Comunicaciones de Datos II

Práctico N° 8

Año 2025

Objetivo: uso Wireshark y TCP

Seq (sequence number) y Ack (acknowledgment number):

En el SYN → Seq = x, sin Ack

En el SYN, ACK → Seq = y, Ack = x+1

En el ACK → Ack = y+1

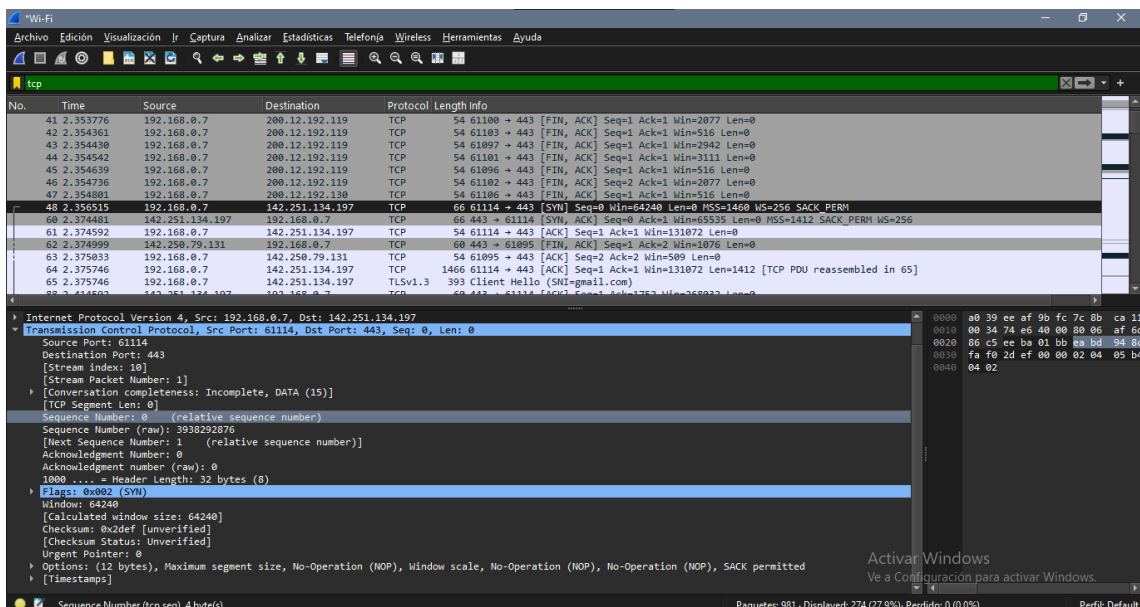
☒ ¿Qué aprendiste de este ejercicio?

El protocolo TCP está orientado a conexión, y siempre comienza con un handshake de 3 pasos.

Estos tres paquetes aseguran que ambos extremos están preparados para transmitir datos.

Podés observar detalles de la negociación, como los números de secuencia, puertos, y ventanas de recepción.

Este proceso es fundamental en la navegación web y todas las apps que usan TCP (correo, FTP, etc.).



3) utilice la topología en PT adjunta al ejercicio y capture los paquetes de la conexión de 3 vías utilizando el modo simulación ingresando del PC1 a Server a la página

<http://10.8.0.2/index.html>

- a) captura de la primera conexión
- b) captura del retorno de conexión más ack
- c) captura del ack final
- d) identifique números de secuencia y puertos de cada TPDU

PASO A PASO: Captura de conexión de 3 vías (Three-Way Handshake)

 Paso previo: configuración básica (si aún no está hecha)

PC1: IP: 10.8.0.1

Práctico Nº 8

Año 2025

Objetivo: uso WireShark y TCP


Gateway: dejar vacío (no se necesita si está en la misma red)

DNS: vacío

Server:IP: 10.8.0.2

Activar servicio HTTP (Pestaña Services > HTTP > On)

Cargar archivo index.html si es necesario.

 Modo Simulación


Cambiá Packet Tracer a Modo Simulación (abajo a la derecha).

En "Edit Filters", marcá solo el protocolo TCP (y HTTP si querés ver más allá).

Desde PC1, abrí el navegador y poné: http://10.8.0.2/index.html

Usá el botón "Capture/Forward" paso a paso para avanzar en la simulación.

Observá los 3 paquetes TCP iniciales (SYN, SYN-ACK, ACK).

 Respuestas del ejercicio

a) Captura de la primera conexión

 Paquete 1 – Solicitud de conexión desde PC1 al servidor

Tipo: TCP

Flags: SYN

Origen: 10.8.0.1, puerto aleatorio (ej: 1025)

Destino: 10.8.0.2, puerto 80 (HTTP)

Número de secuencia (Seq): 0

ACK: No hay aún (—)

b) Captura del retorno de conexión + ACK

 Paquete 2 – Respuesta del servidor al cliente

Tipo: TCP

Flags: SYN, ACK

Origen: 10.8.0.2, puerto 80

Destino: 10.8.0.1, puerto 1025

Número de secuencia (Seq): 0

Número de reconocimiento (ACK): 1 (respondiendo al SYN del cliente)

c) Captura del ACK final

 Paquete 3 – Confirmación del cliente

Tipo: TCP

Flags: ACK

Origen: 10.8.0.1, puerto 1025

Destino: 10.8.0.2, puerto 80

Número de secuencia (Seq): 1

ACK: 1 (confirmando el SYN del servidor)

 Hasta aquí termina el handshake TCP.

d) Identificación de TPDU: puertos y números de secuencia

Paquete	Origen IP:Puerto		Destino IP:Puerto		Flags	Seq	Ack
1	10.8.0.1:1025	10.8.0.2:80	SYN	0	—		
2	10.8.0.2:80	10.8.0.1:1025	SYN, ACK	0	1		


UNL-FICH Dpto. de Informática
Docentes: Gabriel Filippa - Franco Cian - Marcelo T. Gentile – Joaquín Nepotti
Redes y Comunicaciones de Datos II

Práctico Nº 8

Año 2025

Objetivo: uso WireShark y TCP

3 10.8.0.1:1025 10.8.0.2:80 ACK 1 1

 Notas:

Los números de secuencia comienzan en 0 por convención en Packet Tracer.

Los números de puerto del cliente son aleatorios (ephemeral ports), mientras que el servidor usa el puerto estándar 80.

Video ejemplo de control de flujo en TCP

<https://www.youtube.com/watch?v=7mcVikm5csQ>

Video de retransmisiones en TCP

<https://www.youtube.com/watch?v=gjOloj6Ct7E>