Wireshark Lab: TCP

En este laboratorio, se investigará el comportamiento de TCP en detalle. Se hará así mediante el análisis de una traza de los segmentos TCP enviados y recibidos en la transferencia de un archivo de 150KB (conteniendo el texto de Alicia en el País de las Maravillas de Lewis Carrol) desde una computadora a un servidor remoto. Se estudiará el uso de los números de secuencia y de acuse de recibo de TCP (ACK) para proveer transferencia de datos confiable, se verá el algoritmo de control de congestión de TCP – arranque lento (slow Start) y el que evita la congestión (congestion avoidance) – en acción; y se verá el mecanismo de control de flujo TCP anunciado-por-el-receptor. También brevemente se considerará el setup ó configuración de la conexión de TCP y se investigará el rendimiento (throughput y RTT) de la conexión TCP entre la computadora y el servidor.

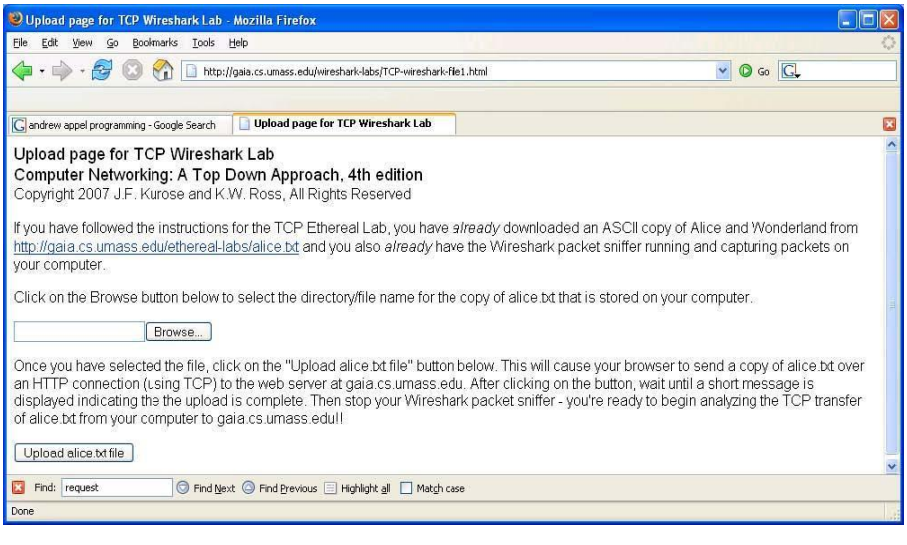
Antes de hacer el laboratorio, sería ideal revisar las secciones 3.5 y 3.7 en el libro[1].

1. Capturando paquetes TCP de un envío masivo a un servidor remoto

Antes de empezar la exploración de TCP necesitamos usar Wireshark para obtener una serie de trazas TCP. Específicamente vamos a obtener una captura de la transferencia de un fichero desde nuestra computadora hasta un servidor remoto. Para ello, nos vamos a meter en una web que nos va permitir introducir el nombre del fichero que quiero subir (que contiene texto en ASCII de Alicia en el País de las Maravillas), y luego transferir el archivo a un servidor Web usando el método HTTP POST (ver sección 2.2.3 en el texto). Se va a usar el método POST en vez del método GET ya que se quiere transferir una gran cantidad de datos desde una computadora a otra computadora. Por supuesto, se estará ejecutando Wireshark durante este tiempo para obtener la captura de los segmentos TCP enviados y recibidos desde la computadora.

Haga lo siguiente:

1. **Pre-condición:** Iniciar el navegador web. Ir a https://gaia.cs.umass.edu/wireshark-labs/alice.txt y recuperar una copia ASCII de Alicia en el País de las Maravillas
2. A continuación ir a http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html
3. Debería ver una pantalla como la siguiente:

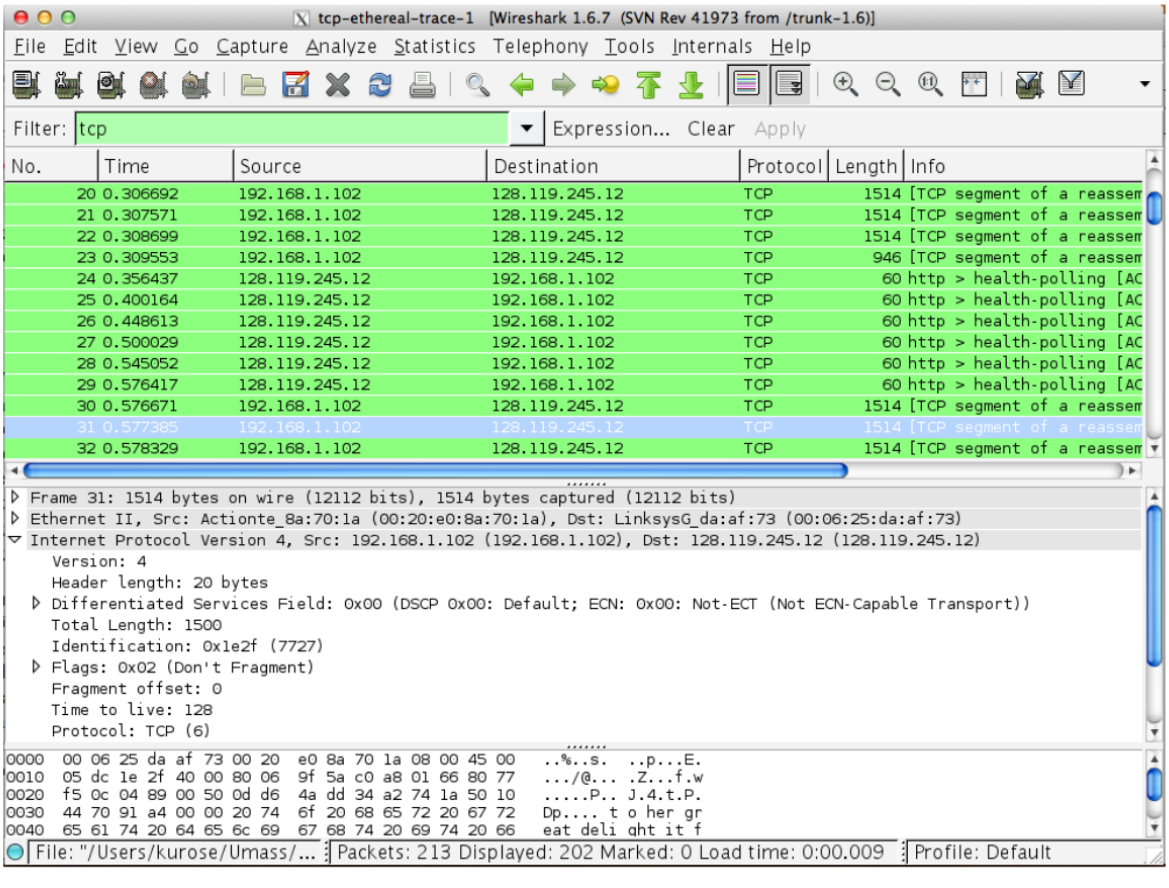


4. Usar el botón Browse o Examinar en este formulario para introducir el nombre del archivo (el nombre de ruta completo) en tu computadora que contiene Alicia en el País de las Maravillas (o hacerlo manualmente), para que quede listo para el siguiente paso. Todavía no presionéis el botón "Upload alice.txt file".

5. Ahora iniciar Wireshark y comenzar la captura de paquetes (Capture -> Options)
    a. Si fuese necesario, seleccionar la interfaz que realmente está siendo utilizada para enviar y recibir paquetes.
    b. Si fuese necesario, desactivar (*Capture packets in promiscuous mode*) y luego presionar Start en la pantalla Wireshark Packet Capture Options (no se necesitará seleccionar ninguna opción aquí).

6. Regresar al navegador, presionar el botón "Upload alice.text file" para subir el archivo al servidor gaia.cs.umass.edu. Una vez que el archivo ha sido subido, se mostrará un pequeño mensaje de felicitación en la ventana del navegador.

7. Detener la captura de paquetes de Wireshark (Capture -> Stop). La ventana de Wireshark debería parecer similar a la ventana mostrada abajo.



Recuerda que todos estos pasos son facultativos. Puedes usar las capturas ofrecidas en ALUD.

2. Un primer vistazo a las tramas capturadas

Antes de analizar el comportamiento de la conexión TCP en detalle, vamos a realizar una vista rápida a la trama descargada o bien de ALUD o bien capturada.

- Primero, filtrar los paquetes en la ventana Wireshark introduciendo "tcp" dentro de la ventana de especificación de “filtro” de visualización en la parte superior de la ventana Wireshark.

Lo que se debería ver son series de mensajes TCP y HTTP entre la computadora y gaia.cs.umass.edu. Se debería ver el **handshake** inicial de 3-vías conteniendo un mensaje SYN. Se debería ver un mensaje HTTP POST y una serie de mensajes de "Continuación HTTP" que son enviados desde la computadora a gaia.cs.umass.edu. Recordar de la discusión en el lab Wireshark HTTP inicial, que no hay como tal un "mensaje de Continuación HTTP" – esto es el modo que Wireshark tiene para indicar que hay múltiples segmentos TCP que se utilizan para enviar un único mensaje HTTP. Se debería también ver segmentos TCP ACK que son devueltos desde gaia.cs.umass.edu a la computadora.

Responder a las siguientes cuestiones.

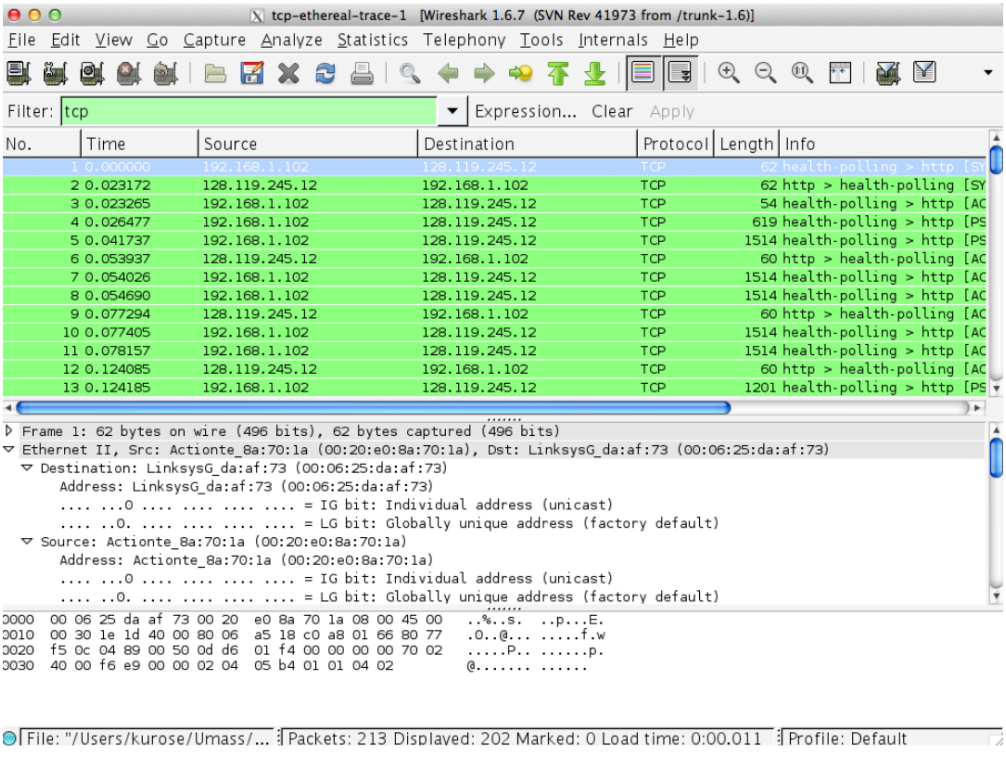
1. ¿Cuál es la dirección IP y el número de puerto TCP usados por la computadora cliente (fuente) que está transfiriendo el archivo a gaia.cs.umass.edu? Para responder esta pregunta, probablemente es más fácil seleccionar el mensaje HTTP y explorar los detalles del paquete TCP usado para transportar este mensaje HTTP, usando la ventana "detalles del encabezado del paquete seleccionado"/details of the selected packet header window”.

2. ¿Cuál es la dirección IP de gaia.cs.umass.edu? ¿Sobre qué número de puerto se está enviando y recibiendo segmentos TCP para esta conexión?

Puesto que este Lab es sobre TCP y no HTTP, vamos a cambiar la ventana "Lista de paquetes capturados" de Wireshark a fin que muestre la información acerca de los segmentos TCP conteniendo los mensajes HTTP, en lugar de los mensajes HTTP. Para que Wireshark haga esto, seleccione Analyze -> Enabled Protocols. Una vez ahí, deshabilita la casilla HTTP y selecciona OK. Se debería ahora ver una ventana Wireshark como la siguiente:

1. Cliente:
-IP:192.168.1.182
-Port: 1161

2. Servidor:
-IP:128.119.245.12
-Port: 80



Esto es lo que se está buscando – una serie de segmentos TCP enviados entre la computadora y gaia.cs.umass.edu.
Consejo práctico: si necesario y si fuera más conveniente aplicar filtros, tal como el siguiente:

- tcp and ((ip.src==ip_cliente and ip.dst==ip_servidor) or (ip.src==ip_servidor and ip.dst==ip_cliente))
- (ip.src==ip_cliente and ip.dst==ip_servidor) or (ip.src==ip_servidor and ip.dst==ip_cliente)

3. Fundamentos de TCP

Responder las siguientes preguntas para los segmentos TCP:

3. ¿Cuál es el número de secuencia relativo y absoluto del segmento TCP SYN que es usado para iniciar la conexión TCP entre la computadora cliente y gaia.cs.umass.edu? ¿Qué es lo que identifica al segmento como un segmento SYN? Sequence Number: 0 (relative sequence number)
   Sequence Number (raw): 232129012
4. ¿Cuál es el número de secuencia relativo y absoluto del segmento SYN+ACK enviado por gaia.cs.umass.edu a la computadora cliente en réplica al SYN? ¿Cuál es el valor relativo y absoluto del campo ACK en el segmento SYN+ACK? ¿Cómo determina gaia.cs.umass.edu ese valor? ¿Qué es lo que identifica al segmento como un segmento SYN+ACK? Seq relativo: 0 absoluto: 883061785

5. ¿Cuál es el número de secuencia relativo y absoluto del segmento TCP conteniendo el comando HTTP POST? Obsérvese que a fin de encontrar el comando POST, se necesitará seleccionar un paquete TCP capturado, luego en el Panel "Detalles del Paquete Seleccionado" hacer un clic en "TCP segment data (### bytes)", y finalmente en el Panel "Bytes del Paquete Seleccionado" observar el campo de contenido del paquete (al final de la ventana Wireshark), buscando un segmento con la palabra "POST" dentro de su campo de DATOS; también indicar el Nº del segmento que contiene esta palabra.
   Seq relativo: 1 absoluto: 232193013

6. Considerar el segmento TCP conteniendo el HTTP POST como el primer segmento en la conexión TCP ya establecida. ¿Cuáles son los números de secuencia absolutos de los primeros 6 segmentos en la conexión TCP (incluyendo el segmento conteniendo el HTTP POST)? ¿A qué hora fue cada segmento enviado? ¿Cuándo se recibió el ACK para cada segmento enviado?
   Dada la diferencia entre el momento en que cada segmento TCP fue enviado y cuando su acuse de recibo fue recibido ¿cuál es el valor de RTT para cada uno de los 6 segmentos? ¿Cuál es el valor de RTTestimado después del recibo de cada ACK? (Asumir que el valor de RTTestimado es igual al RTTmedido para el primer segmento en la conexión ya establecida, y luego se calcula usando la ecuación de RTTestimado para todos los segmentos subsecuentes: página 242 del libro de Kurose[1])

$$RTT_{estimado} = (1-\alpha) * RTT_{old} + \alpha * RTT_{medido\,(muestreado)}$$

7. ¿Cuál es la longitud de los datos en cada uno de los 6 primeros segmentos TCP, después que la conexión ha sido establecida?

8. ¿Cuál es la mínima cantidad del espacio del buffer disponible que es anunciada por el lado receptor para la traza entera? ¿La carencia de espacio en el buffer receptor regula en algún momento el flujo del emisor? ¿Qué se debería observar en la captura (en la traza entera) a fin de responder esta segunda parte?

9. ¿Hay algunos segmentos retransmitidos en la captura? ¿se debería observar en la captura (en la traza) a fin de responder esta pregunta?

10. ¿Cuántos datos "acusa de recibo" el receptor a través de los 6 primeros ACKs? ¿Puedes identificar casos donde el receptor está ACKeando cada segmento recibido (ver Tabla 3.2 en el capítulo Nº 05 de Kurose)?

Nro. Segmento	# Ack recibido en el Ex desde el servidor	Cantidad de datos ACKeados.

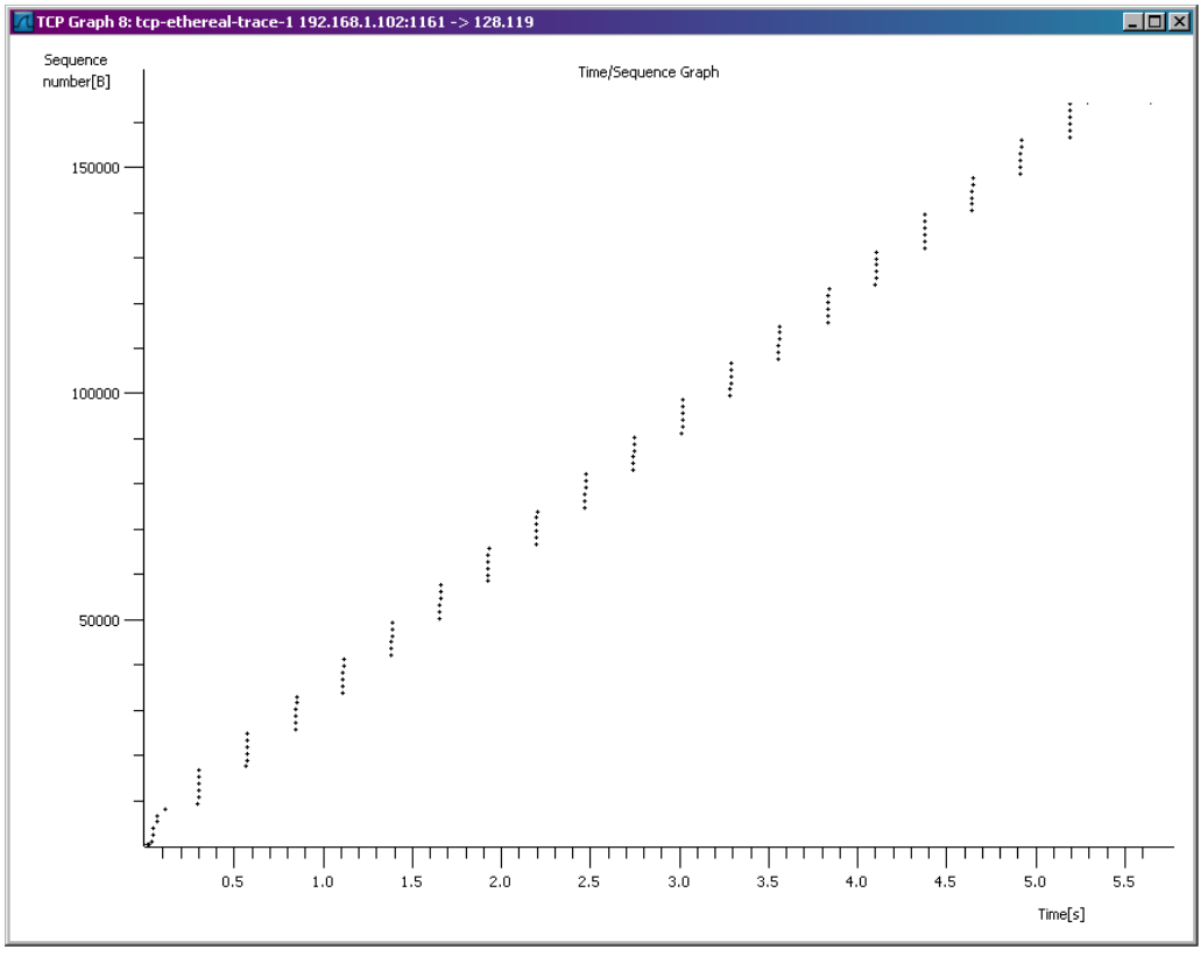
11. ¿Cuál es el throughput (los bytes transferidos por unidad de tiempo) para la conexión TCP? Explicar cómo se calculó este valor.

[1] Wireshark tiene una característica interesante que permite plotear el RTT para cada uno de los segmentos TCP enviados. Seleccionar un segmento TCP en la ventana de "Lista de paquetes capturados" que se está enviando desde el cliente al servidor gaia.cs.umass.edu. Luego seleccionar: Statistics->TCP Stream Graph->Round Trip Time Graph.

4. Control de la congestión TCP en acción

Ahora examinar la cantidad de datos enviados por unidad de tiempo desde el cliente al servidor. En vez de calcular (a mano) esto a partir de los datos en bruto en la ventana de Wireshark, se usará una de la utilidades de gráficos TCP de Wireshark – Time-Sequence-Graph (Stevens) – para "plotear" los datos.

- Seleccionar un segmento TCP en la ventana "Lista de paquetes capturados" de Wireshark. Luego seleccionar el menú: Statistics -> TCP Stream Graph -> Time- Sequence-Graph (Stevens). Se debería ver un plot que parece similar al siguiente plot:



Aquí, cada punto representa un segmento TCP enviado, ploteando el número de secuencia del segmento vs. el tiempo en el cual fue enviado. Obsérvese que un conjunto de puntos apilados uno encima del otro representa una serie de paquetes que fueron enviados en inmediata sucesión por el emisor (back-to-back).
Responder las siguientes preguntas para los segmentos TCP de la traza capturada (ALUD):

12. Usar la herramienta de ploteo Time-Sequence-Graph (Stevens) para visualizar el número de secuencia vs. el plot de tiempo de los segmentos siendo enviados desde el cliente al servidor gaia.cs.umass.edu. ¿Puede identificar donde la fase de arranque lento (slowstart) de TCP comienza y termina, y donde la evitación de congestión (congestion avoidance) toma el relevo? Comentar sobre los modos en los cuales los datos medidos difieren del comportamiento idealizado de TCP que se estudió en el texto.