Informe Tarea 2

Redes de Computadores

Integrantes:

Javier Figueroa Garrido

Diego Escobar Boehmwald

**Iniciar dos clientes y un servidor de Avioncito de Papel**

Se debe tener en cuenta que esta aplicación Chat fue realizada para un solo computador. Para empezar, se ejecuta la aplicación del Servidor (Servidor.java) y luego dos instancias de la aplicación Cliente (Redes2.java).

En las instancias de la aplicación Cliente se le solicitará que ingrese el número de puerto que usará el socket para conectarse con la aplicación Servidor. Tenga en cuenta que no puede usar el mismo número de puerto para dos o más aplicaciones Cliente, ya que un puerto sólo puede estar asociado a un socket. Asegúrese de que el puerto que va a utilizar este libre, el puerto 9999 está reservado para el servidor, así que no está disponible.

Como el número de puerto es único para cada cliente, este será su identificador a partir de ahora.

A continuación, abra en el navegador la siguiente dirección:

[http://localhost:**puerto**/](http://localhost:puerto/)

Donde puerto es el número de puerto escogido para una de sus aplicaciones clientes. En otra ventana o pestaña abra dicha dirección con el número de puerto correspondiente a la segunda aplicación cliente.

Una vez hecho esto, las aplicaciones comenzarán a trabajar. La aplicación Servidor estará “escuchando” a la espera de mensajes de consulta de las aplicaciones Cliente.

Las aplicaciones Cliente envían cada 5 segundos un mensaje de consulta al servidor, preguntando por nuevos mensajes, de la forma:

CONSULTA:**puerto**

Donde puerto es el número del puerto de la aplicación Cliente que consulta.

El servidor imprime en pantalla todos los mensajes que recibe, de la forma:

Recibido:**mensaje**

**Conectar ambos clientes (agregarse como contactos, intercambiar direcciones IP y puertos)**

El modelo utilizado se muestra en el siguiente diagrama:

(diagrama)

Dado que estamos trabajando en un solo computador, no hay intercambio de direcciones IP ya que corresponde a la misma.

(extender)

**Iniciar la captura de paquetes con Wireshark**

Las pruebas se hicieron en Windows, por lo tanto, no fue posible realizar un seguimiento de la interfaz loopback con Wireshark (ya que estamos trabajando en un solo computador, es necesario apuntar la dirección IP hacia sí misma). Por lo tanto, para monitorear la interface se utilizó un programa llamado RawCap (<http://www.netresec.com/?page=RawCap>).

**Enviar un mensaje (Cliente 1 envía mensaje a Cliente 2)**

Una vez ingrese a la aplicación HTML del Cliente ([http://localhost:**puerto**/](http://localhost:puerto/)), verá lo siguiente:

(screenshot)

Para enviar un mensaje, debe llenar el formulario ingresando el puerto del Cliente receptor, el mensaje y la dirección IP local (aunque dado que todos los Clientes se están ejecutando en el mismo computador, será “localhost” para todos).

En la parte superior se encuentra el Iframe del Chat, que se actualiza cada 5 segundos. El Chat muestra los mensajes que ha recibido y los que ha enviado (sólo se mostrarán los que ha enviado una vez el Cliente receptor haya recibido el mensaje).

**5. Enviar una respuesta (Cliente 2 envía mensaje a Cliente 1)**

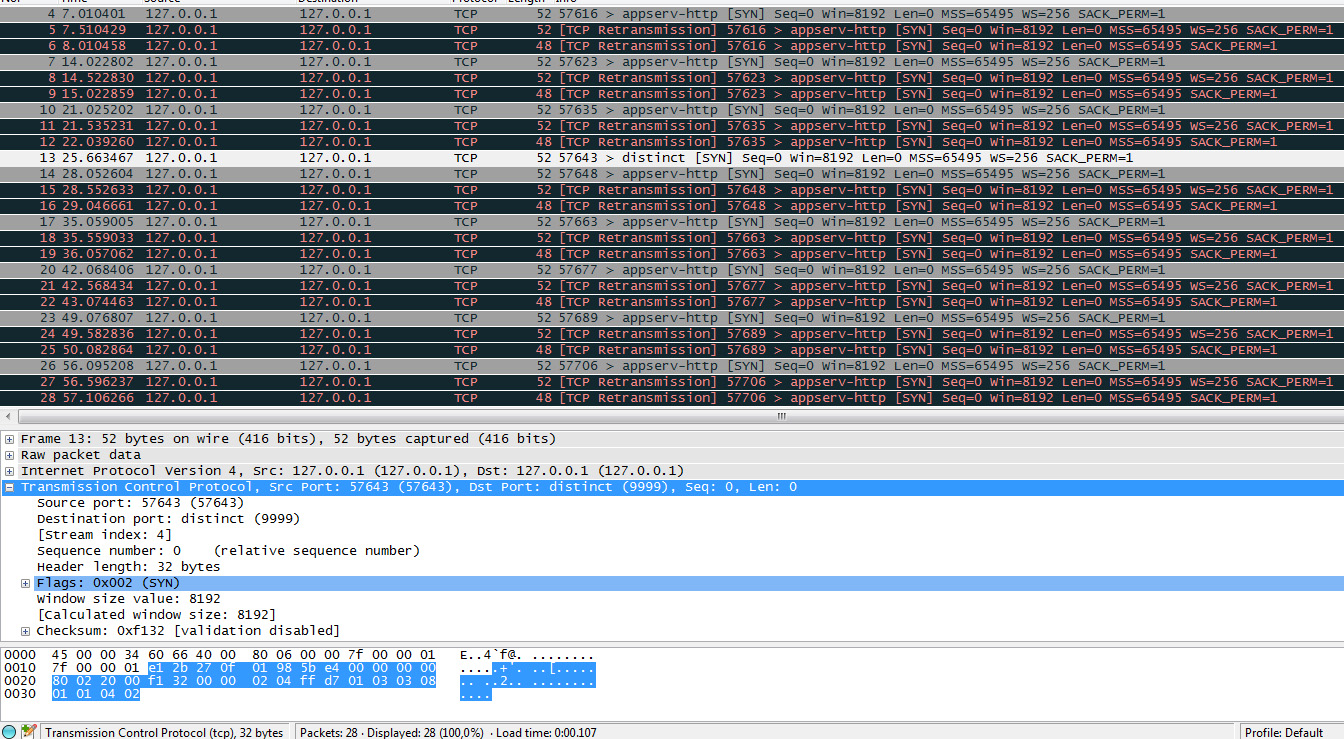
La respuesta se realiza de similar manera al envío del mensaje:

(screenshot formulario)

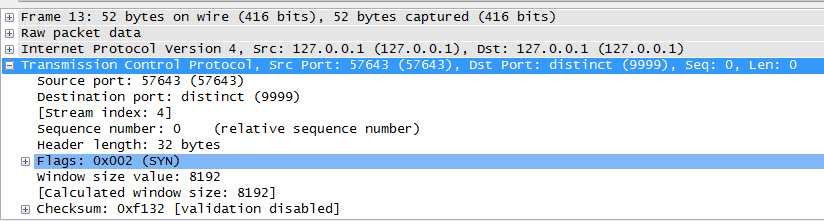
(screeenshot formulario post envio)

**6. Mostrar en Wireshark los mensajes que se enviaron, destacando direcciones IP y puertos de origen y destino. Comparar ACK y números de secuencia de acuerdo al protocolo escogido.**

RawCap genera un dump file que puede ser abierto por Wireshark. Una vez terminado el envío de mensaje y la respuesta, se revisan los paquetes enviados y recibidos:



Como era de esperarse, todas las IP correspondes a 127.0.0.1 (localhost).



Para un paquete seleccionado, es posible ver el puerto de origen y destino. Se puede ver que se corresponden con los esperados.