Redes y Comunicaciones 2017
Práctica 5

Autor: Fermín Minetto (http://www.github.com/ferminmine)

1) ¿Cuál es la función de la capa de transporte?

La función de la capa de transporte es proporcionar una comunicación lógica entre procesos de aplicación que se ejecutan en hosts diferentes. Por comunicación lógica se quiere decir que desde la perspectiva de la aplicación, es como si los dos hosts que ejecutan los procesos estuvieran conectados directamente cuando en realidad los hosts pueden encontrarse en puntos opuestos del planeta conectados mediante numerosos routers y un amplio rango de tipos de enlace.

En el lado emisor, la capa de transporte convierte los mensajes querecibe procedentes de un proceso de apliación emisor en paquetes de la capa de transporte conocidos como segmentos de la capa de transporte. Esto se hacedividiendo los mensajes de la aplicación en fragmentos más pequeños y añadiendo una cabecera de la capa de transporte a cada fragmento con el fin de crear el segmento de la capa de transporte.

2) Describa la estructura del segmento TCP y UDP

Estructura del segmento UDP: *no. Puerto Origen, no. Puerto Destino, Longitud, Suma de Comprobacion, Datos de la Aplicacion (Mensaje)*

- ➔ ***Datos de la aplicación:*** Los datos de la aplicicación se almacenan en este campo. Por ejemplo, podría contener un mensaje de consulta o respuesta DNS.
- ➔ **Puerto de Origen y Destino:** Permiten al host de destino pasar los datos de la aplicación al proceso apropiado que está ejecutandose en el sistema terminal del destino (es decir, realizar la función de demultiplexación). Tareas de multiplexación y demultiplexación.
- ➔ **Suma de Comprobación**: El host receptor utiliza la suma de comprobación para detectar si se han introducido errores en el segmento.
- ➔ **Longitud:** Especifica la longitud del segmento UDP en bytes, incluyendo la cabecera.

Estructura del segmento TCP: *no Puerto Origen, no Puerto Destino, Suma de Comprobación, Número de Secuencia, Número de Reconocimiento, Ventana de Recepción, Longitud de Cabecera, Opciones (Opcional), Indicador*

- ➔ **Puerto de Origen y Destino:** Permiten para tareas de multiplexación y demultiplexación.
- ➔ **Suma de Comprobación**: El host receptor utiliza la suma de comprobación para detectar si se han introducido errores en el segmento.
- ➔ **Número de secuencia y número de reconocimiento:** Utilizados para implementar un servicio de transferencia de datos fiables. Los saltos en los números de secuencia permiten identificar si se han perdido paquetes.
- ➔ **Ventana de recepción:** Se emplea para mdocar el número de bytes que un receptor está dispuesto a aceptar.
- ➔ **Longitud de cabecera:** La cabecera TCP puede tener una longitud variable a causa del campo opciones TCP (normalmente este campo está vacío por lo que la longitud de una cabecera TCP típica es de 20 bytes).

- ➔ **Opciones:** Es opcional y de longitud variable. Se utiliza cuando un emisor y un receptor negocian el tamaó máximo de segmento (MSS) o como un factor de escala de la ventana en las redes de alta velocidad.
- ➔ **Indicador:** Tiene 6 bits. El *bit ACK* se utiliza para indicar que el valor transportado en el campo de reconocimiento es válido. Los bits *RST, SYN y FIN* se utilizan para el establecimiento y cierre de conexiones. *PSH* indica que el receptor deberá pasar los datos a la capa superior de forma inmediata. Por último, el *bit URG* se utiliza para indicar que hay datos en e ste segmento que la entidad de la capa superior del lado emisor ha marcado como "urgentes".

3) ¿Cuál es el objetivo del uso de puertos en el modelo TCP/IP?

Sirve para identificar los servicios o aplicaciones para el que esta dirigido un paquete que llega a la computadora.

4) Compare TCP con UDP

- ➔ **Confiabilidad:** UDP no es un sistema de transferencia de datos confiables ya que no garantiza que los datos enviados por un proceso lleguen intactos. Algunas aplicaciones utilizan UDP ya que son tolerantes a pérdidas de cantidades pequeñas de paquetes. En cambio, TCP si puede garantizar que los datos enviados lleguen intactos ya que es un servicio de transferencia de datos fiable.
- ➔ **Multiplexación:** Ambos UDP y TCP realizan tareas de multiplexación ya que es uno de los servicios que deben implementar los protocolos de la capa de transporte. La multiplexación es la única característica que tienen en común TCP y UDP.
- ➔ **Orientado a la conexión:** TCP lleva a cabo un proceso de establecimieto de la conexión en tres fases antes de iniciar la transferencia de datos. UDP inicia la transmisión de manera inmediata, sin formalidades preliminares. UDP no añade ningún retardo adicional a causa del establecimiento de una conexión, es por eso que DNS opera sobre UDP y no TCP.
- ➔ **Controles de congestión:** TCP si brinda mecanismosde control de congestión, que evitan que cualquier conexión TCP inunde con una cantidad de tráfico excesiva los enlaces y routers existentes entre los hosts que están comunicándose. UDP no proporciona mecanismos de control de congestión, puesto que es más simple y más apropiado a las aplicaciones en tiempo real (como comunicaciones o multimedia) las cuales no responden demasiado bien a estos mecanismos. El tráfico UDP, por el contrario de TCP, no está regulado.
- ➔ **Utilización de puertos (Lo mismo que la multiplexación)**
- ➔ **¿Cuál es el campo del datagrama IP y los valores que se utilizan en este para diferenciar que se transporta TCP o UDP? (Ayuda:)**

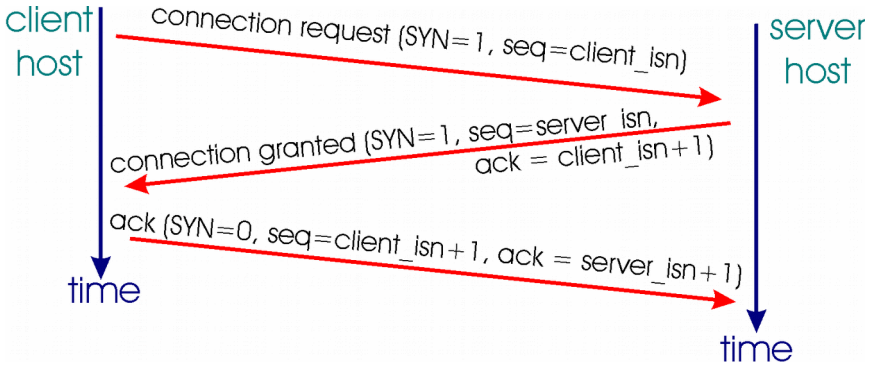
5) La PDU de la capa de transporte es el segmento. Sin embargo, en algunos contextos suele utilizarse el término datagrama. Indique cuando.

En los RFC los paquetes que manejan tanto TCP como UDP son los segmentos. Sin embargo no es poco común encontrar que en algunos textos mencionan como segmentos sólo a los paquetes TCP y datagrama a los paquetes UDP.

6) Describa el saludo de tres vías de TCP

El establecimiento y finalización de una conexión TCP tiene una gran importancia, puesto que el establecimiento de una conexión TCP puede aumentar significativamente el retardo percibido (por ejemplo, cuando se navega por la Web). Para establecer una conexión TCP, el proceso de palicación cliente informa en primer lugar al cliente TCP que desea establecer una conexión con un proceso servidor. A continuación, el protocolo TCP en el cliente establece una conexión TCP con el protocolo TCP en el servidor de la siguiente manera:

1. TCP del lado del cliente envía un segmento TCP especial al TCP del lado del servidor. Este segmento no contiene datos de la capa de aplicación pero uno de los bits indicadores en la cabecera del segmento, el bit *SYN se pone a 1.* Además, el cliente selecciona de forma aleatoria un número de secuencia inicial (client_isn) y lo coloca en el campo número de secuencia del segmento TCP inicial SYN.
2. Una vez que el datagrama IP que contiene el segmento SYN TCP llega al host servior (suponiendo que llega), el servidor extrae dicho segmento SYN del datagrama, asigna los buffers y variables TCP a la conexión y envía un segmento de conexión concedida al cliente TCP. Este segmento *de conexión concedida se conoce como segmento SYNACK* y contiene tres ragmentos imprtantes de la cabecera del segmento. El primero, el bit SYN se pone a 1. El seegundo, el campo reconocimiento (ACK) se hace igual a cliente_isn+1. Por último el servidor elige su propio numero de secuencia inicial (server_isn) que lo coloca en el indicador de secuencia.
3. Al recibir el segmento SYNACK, el cliente también asigna buffers y variables a la conexión. El host cliente envía entonces al servidor un último segmento que confirma el segmento de conexión concedida del servidor (almacena el valor server_isn+1 en el campo de reconocimiento de la cabecera). El bit SYN se pone a cero, ya que la conexión está establecida.



7) Utilizar comando ss...

- ➔ **Listar comunicaciones TCP establecidas:** *ss -net*
- ➔ **Listar comunicaciones UDP establecidas:** *ss -neu*
- ➔ **Listar servicios TCP que estén esperando comunicaciones:** ss -nlt
- ➔ **Listar servicios UDP que estén esperando comunicaciones:** ss -nlu
- ➔ **Con los nombres de procesos asociados a los puertos:** *-netp,neup,nltp, nlup*
- ➔ **Utilizando netsat:** *netstat -nap, -punta*

Nota: recomendable utilizar sudo para poder ver algunos procesos del sistema que estén escuchando.

8) Utilizar hping3 y mandar segmentos TCP con SYN activado

hping3 -S 127.0.0.1 -p 22 (-S mayúscula activa el flag de SYN)

hping3 -S 127.0.0.1 -p 40

Cuando se envía un segmento TCP para establecer la conexión a un puerto que no está escuchando (40) entonces se recibe una respuesta con el flag RA, que quiere decir *conexión rechazada (answer refused?)*. En cambio, cuando se envía a un puerto que si está escuchando, se recibe un flag SA que significa Syn ACK, que sería el segundo paso del saludo de tres vías para iniciar una conexión TCP.

9) Utilizar hping3 y mandar segmentos UDP

hping3 -2 127.0.0.1 -p 68
hping3 -2 127.0.0.1 -p 40

Cuando se envían los datos hacia un puerto que está escuchando entonces no se va a recibir respuesta ya que con el protocolo UDP simplemente se envían y se reciben datos y no hay un protocolo de saludo para establecer una conexión. Cuando se envía un segmento UDP a un puerto que no está escuchando o no es alcanzable, se va a recibir un mensaje desde la capa de red avisando que el puerto es inalcanzable.

Es posible establecer una conexión TCP y recibir datos sobre el mismo número de puerto, puesto que el SO toma un puerto UDP o un puerto TCP como puertos distintos de acuerdo al protocolo. Es decir, que el SO vería el puerto 63UDP como algo distinto a un puerto 63TCP.

10) Multicast. ¿Sobre cual protocolo de capa de transporte funciona? ¿Se podría adaptar para que funcione sobre otro protocolo de capa de transporte? ¿Por qué?

Se denomina multicast a una comunicación grupal donde la información es dirigida hacia un grupo de computadoras de manera simultánea. Puede ser multicast de aplicación o multicast de red, siendo la última la que permita efectivamente enviar de manera simultánea la información. El protocolo de la capa de transporte utilizado para multicast es *el protocolo UDP* y suele ser utilizado por aplicaciones de estilo transmisión de televisión, o para charlas grupales.
Algunas empresas han desarrollado un sistema de transferencia fiable en la capa de aplicación que usa UDP como protocolo de transporte, para poder utilizar un multicast más fiable.
No se podría implementar con TCP, puesto que multicast sería el envío de un mensaje a muchos hosts, mientras que con TCP se imitaría pero no sería la misma función ya que deberían enviarse un mensaje por cada conexión abierto.

11) Topología CORE. Levantar servicio que escuche un puerto y escuchar desde otra máquina usando ncat.

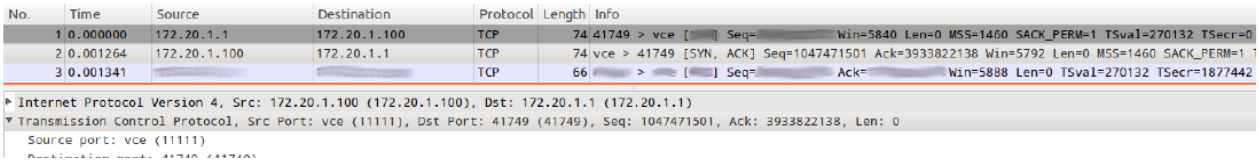
Levantar servicio que escuche un puerto: *ncat -k -l 8001* (-k para mantener viva, l para escuchar y 8001 es el número de puerto donde se escuchará)
Establecer una conexión: *ncat 10.0.0.21 8001* (la dirección IP de la máquina con la que se establecerá la conexión y el número de puerto al que se va a conectar)

La conexión levantada estará en estado listen hasta que otra máquina se conecte. Cuando otra máquina se conecte, el puerto pasará a la categoría de established, y se listará con que IP se estableció la conexión. Una vez establecida la conexión ambos hosts pueden

intercambiar segmentos. Cada una de las conexiones abiertas se va a identificar por el número de puerto.
Si un servidor establece varias conexiones sobre el mismo puerto entonces lo que distinguirá a las conexiones que estén establecidas en el mismo puerto será el campo *IP de origen*.

Si la conexión se cierra primero desde el servidor, entonces se queda en *estado FIN-WAIT* y hasta que no se cierra desde el cliente no se deja de listar; desde el cliente se puede ver el estado *CLOSE-WAIT*.
Si el cliente cierra la conexión, desde el lado del cliente se puede ver el *estado TIME-WAIT* por unos segundos hasta que desaparece de la lista.

12) Completar los campos de la imagen



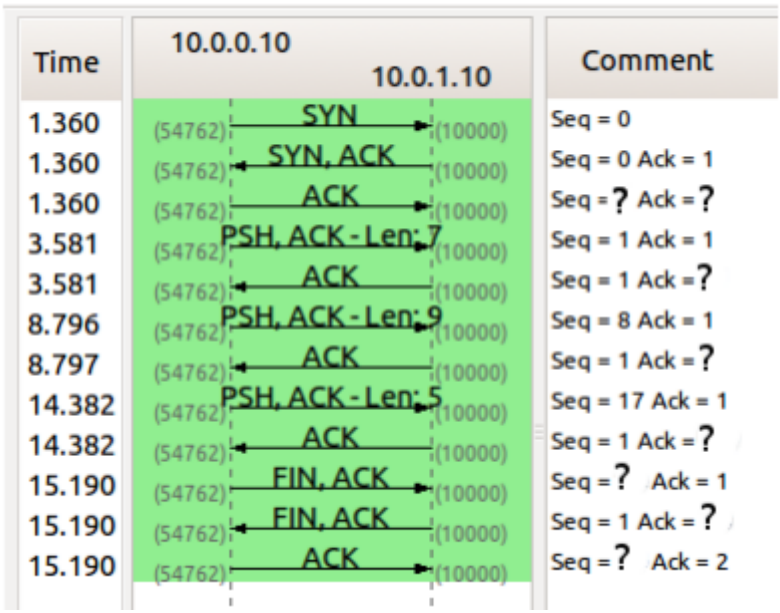
La forma de resolver la imagen es sencilla. El *SYN ACK* es el segundo segmento TCP que es parte del saludo de las tres vías utilizado por TCP para establecer una conexión. El campo de secuencia en el segmento corresponde a un numero aleatorio, y el campo de ack corresponde a un número aleatorio elegido por el emisor+1.
Conociendo esta información, sabemos que en el primer segmento SYN, el emisor envía por el seq su número elegido original, es decir, sin sumarle el +1, por lo tanto el número del ACK en el SYNACK debería restársele uno para obtener el valor original.
En el tercer segmento (ACK), el emisor debería enviar en el campo de secuencia su número elegido+1 y en el ACK el número enviado por el servidor+1.

[SYN] seq: 3933822137
[SYN, ACK] seq: 1047471501 ; **ack:** 3933822138
[ACK] seq: 3933822138 ; **ack:** 1047471502

La forma de resolver la imagen es sencilla. El *SYN ACK* es el segundo segmento TCP que es parte del saludo de las tres vías utilizado por TCP para establecer una conexión. El campo de secuencia en el segmento corresponde a un numero aleatorio, y el campo de ack corresponde a un número aleatorio elegido por el emisor+1.
Conociendo esta información, sabemos que en el primer segmento SYN, el emisor envía por el seq su número elegido original, es decir, sin sumarle el +1, por lo tanto el número del ACK en el SYNACK debería restársele uno para obtener el valor original.
En el tercer segmento (ACK), el emisor debería enviar en el campo de secuencia su número elegido+1 y en el ACK el número enviado por el servidor+1.

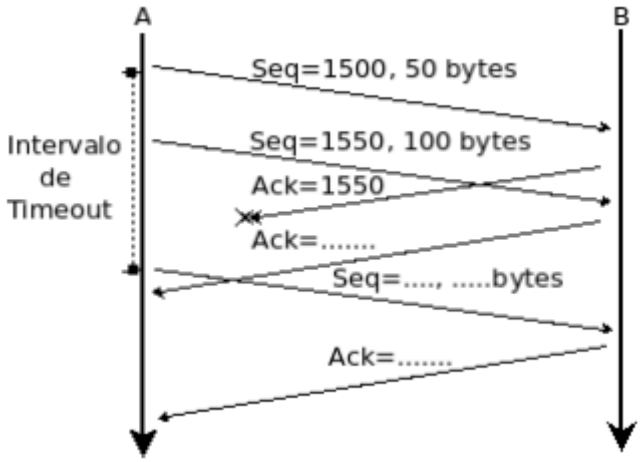
[SYN] seq: 3933822137
[SYN, ACK] seq: 1047471501 ; **ack:** 3933822138
[ACK] seq: 3933822138 ; **ack:** 1047471502

13) Completar los valores marcados



Syn → | seq 0
← **Syn Ack** | seq 0; ack 1
Ack → | seq 1; ack 1
PSH, ACK -len: 7 → | seq 1; ack 1
←**Ack** | seq 1; ack 8
PSH, Ack – Len:9 → | seq=8; ack 1
← **Ack** | seq 1; ack 17
PSH, Ack – Len: 5 → | Seq 17; ack 1
←**Ack** | seq 1; ack 22
FIN, ACK → | seq 22; ack 1
← **FIN, ACK** | seq 1; ack 23
Ack → | seq 23; ack 2

14) Completar los datos faltantes



TCP utiliza el algoritmo Selective Repeat.

1. Ack = 1650
2. Seq= 1500, 50 bytes
3. Ack = 1550