Herramientas para el análisis de comunicaciones TCP/UDP

Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación

Diciembre 2023

Contenidos

netstat

- La herramienta `netstat` permite obtener información sobre varios aspectos del estado de la red en un sistema Unix/Linux.
- Entre otros usos, permite ver el listado de comunicaciones activas en una máquina: detalles de las conexiones TCP y comunicaciones UDP que hay establecidas en ese momento.
- Sintaxis:

```
netstat -tna
```

```
netstat -una
```

  - la opción `-t` muestra información de las conexiones TCP
  - la opción `-u` muestra información de las comunicaciones UDP
  - la opción `-n` muestra direcciones IP (si se omite, se trata de mostrar nombres de máquinas por DNS en su lugar)
  - la opción `-a` muestra información de todas las comunicaciones, incluyendo aquellas en las que está máquina ha lanzado un servidor que está esperando recibir mensajes de clientes

netstat

- `netstat` mostrará la siguiente información para las comunicaciones activas:

```
Active Internet connections (servers and established)
+------+--------+--------+---------------+-----------------+-------+
|Proto | Recv-Q | Send-Q | Local Address | Foreign Address | State |
+------+--------+--------+---------------+-----------------+-------+
```

  - La columna `Proto` indica el protocolo utilizado (UDP o TCP)
  - La columna `Local Address` muestra la dirección IP local de la máquina donde se esperan recibir datos y el número de puerto.
  - En la columna `Foreign Address` muestra la dirección IP y puerto de las máquinas remota con la que se ha establecido una comunicación.
  - Las columnas `Recv-Q` (`receiving queue`) y `Send-Q` (`sending queue`) muestran la cantidad de bytes que hay almacenados en los buffers locales reservados para la recepción de datos y emisión de datos de este servidor.
  - La columna `State` indicará el estado de la comunicación.

netstat: comunicaciones UDP (I)

- Para visualizar las comunicaciones UDP activas:

```
pc1:~# netstat -una
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 0.0.0.0:7777            0.0.0.0:*
```

  - El resultado de ejecutar este comando muestra un servidor UDP esperando recibir conexiones de clientes en el puerto 7777.
  - La columna `Local Address` muestra la dirección 0.0.0.0 que indica que se esperan recibir comunicaciones UDP en cualquiera de las direcciones IP configuradas actualmente en la máquina local.
  - En la columna `Foreign Address` se mostrarán las direcciones IP y puertos de las máquinas clientes remotos que se conecten con este servidor. Actualmente no hay ninguna.
  - Las columnas `Recv-Q` y `Send-Q` muestran que no hay datos almacenados en los buffers.

netstat: comunicaciones UDP (II)

- Para visualizar las comunicaciones UDP activas:

```
pc1:~# netstat -una
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 11.0.0.1:7777           11.0.0.2:32768          ESTABLISHED
```

  - El resultado de ejecutar este comando muestra una comunicación UDP entre la dirección IP local 11.0.0.1 y puerto 7777 y la dirección IP remota 11.0.0.2 y puerto 32768.
  - Las columnas `Recv-Q` y `Send-Q` muestran que no hay datos almacenados en los buffers.

netstat: comunicaciones TCP (I)

- Para visualizar las comunicaciones TCP activas:

```
pc1:~# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:7777            0.0.0.0:*               LISTEN
```

  - El resultado de ejecutar este comando muestra un servidor TCP esperando recibir conexiones de clientes en el puerto 7777.
  - La columna `Local Address` muestra la dirección 0.0.0.0 que indica que se esperan recibir comunicaciones UDP en cualquiera de las direcciones IP configuradas actualmente en la máquina local.
  - En la columna `Foreign Address` se mostrarán las direcciones IP y puertos de las máquinas clientes remotos que se conecten con este servidor. Actualmente no hay ninguna.
  - Las columnas `Recv-Q` y `Send-Q` muestran que no hay datos almacenados en los buffers.

netstat: comunicaciones TCP (II)

- Para visualizar las comunicaciones TCP activas:

```
pc1:~# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 11.0.0.1:7777           11.0.0.2:33715          ESTABLISHED
```

  - El resultado de ejecutar este comando muestra una comunicación TCP entre la dirección IP local 11.0.0.1 y puerto 7777 y la dirección IP remota 11.0.0.2 y puerto 33715.
  - Las columnas `Recv-Q` y `Send-Q` muestran que no hay datos almacenados en los buffers.

Contenidos

Herramienta nc

- Usaremos `nc` para generar tráfico TCP y UDP según el modelo de comunicaciones cliente/servidor.
- Se arrancarán 2 aplicaciones: una funcionando con el rol cliente y la otra con el rol servidor.
- **Siempre es necesario lanzar primero la aplicación que funciona como servidor**. El servidor quedará a la espera de recibir tráfico procedente de la aplicación cliente, que se deberá lanzar después.
  - Una aplicación lanzada con `nc` como **cliente** lee de la entrada estándar (por omisión el teclado) los caracteres introducidos y al pulsar la tecla `INTRO` la línea de texto es enviada usando TCP o UDP a la aplicación servidor.
  - Al recibir la línea de texto, la aplicación **servidor** lanzada con `nc` mostrará en la pantalla los datos recibidos de la aplicación cliente lanzada con `nc`.

Contenidos

Aplicación servidor UDP

- Para arrancar `nc` como **servidor** utilizando el protocolo UDP ejecutaremos la siguiente orden:

```
nc -u -l -p <Pto-Loc>
```

  - `-u`: UDP
  - `-l`: *listen* = modo servidor
  - `-p <Pto-Loc>`: número de puerto local UDP en el que la aplicación servidor esperará recibir los datagramas UDP de una aplicación cliente.

- Por ejemplo, si queremos arrancar una aplicación servidor UDP en el puerto 7777 de la máquina `pc1` utilizaremos la siguiente orden:

```
pc1:~# nc -u -l -p 7777
```

Aplicación cliente UDP

- Para arrancar nc como **cliente** utilizando el protocolo UDP ejecutaremos la siguiente orden:

```
nc -u -p <Pto-Loc> <IP-dest> <Pto-dest>
```

  Donde:

  - -u: UDP
  - -p <Pto-Loc>: número de puerto local UDP en el que la aplicación cliente esperará recibir los datagramas UDP que vengan del servidor.
  - <IP-dest>: dirección IP de la máquina donde se está ejecutando la aplicación servidor UDP.
  - <Pto-dest> es el número de puerto UDP en el que escucha la aplicación servidor UDP.

- Por ejemplo, si queremos arrancar una aplicación cliente UDP en pc2 que espere recibir datagramas UDP en el puerto 6666 y que envíe datagramas UDP a la dirección IP 200.0.0.1 y puerto 7777 (donde se encuentra esperando recibir datagramas UDP la aplicación servidor) utilizaremos la siguiente orden:

```
pc2:~# nc -u -p 6666 200.0.0.1 7777
```

Envío de datos UDP

- Una vez lanzadas las aplicaciones servidor UDP y cliente UDP, el cliente puede enviarle líneas de texto al servidor.
- Después de que el cliente haya enviado al menos una línea de texto al servidor, todo lo que escribamos a través de la entrada estándar de un extremo será enviado al otro extremo como datagramas UDP: si escribimos en el terminal de la aplicación cliente, esto será enviado a la aplicación servidor, y viceversa.
- Pasado un cierto tiempo desde el último mensaje del cliente, el servidor "olvida" al cliente (recuerda que en UDP no hay conexiones) y es necesario volver a enviar un mensaje desde el cliente para que el servidor pueda volver a enviarle mensajes.
- Para interrumpir la ejecución de estas aplicaciones se debe utilizar `Ctrl+C`.

Contenidos

Aplicación servidor TCP

- Para arrancar `nc` como **servidor** utilizando el protocolo TCP ejecutaremos la siguiente orden:

```
nc -l -p <Pto-Loc>
```

  Donde:

  - `-l`: *listen* = modo servidor
  - `-p <Pto-Loc>`: es el número de puerto local TCP en el que la aplicación servidor esperará recibir mensajes TCP de una aplicación cliente.

- Por ejemplo, si queremos arrancar una aplicación servidor TCP en el puerto 7777 de la máquina `pc1` utilizaremos la siguiente orden:

```
pc1:~# nc -l -p 7777
```

Aplicación cliente TCP

- Para arrancar `nc` como **cliente** utilizando el protocolo TCP ejecutaremos la siguiente orden:

```
nc -p <Pto-Loc> <IP-dest> <Pto-dest>
```

  Donde:

  - `-p <Pto-Loc>`: número de puerto local TCP en el que la aplicación cliente esperará recibir los mensajes de la aplicación servidor TCP.
  - `<IP-dest>`: dirección IP de la máquina donde se está ejecutando la aplicación servidor TCP.
  - `<Pto-dest>`: número de puerto TCP en el que escucha la aplicación servidor TCP.

- Por ejemplo, si queremos arrancar una aplicación cliente TCP en pc2 que utilice el puerto origen 6666 para establecer una conexión TCP con un servidor TCP que escuche en el puerto destino 7777 de la máquina 200.0.0.1, utilizaremos la siguiente orden:

```
pc2:~# nc -p 6666 200.0.0.1 7777
```

Envío de datos TCP

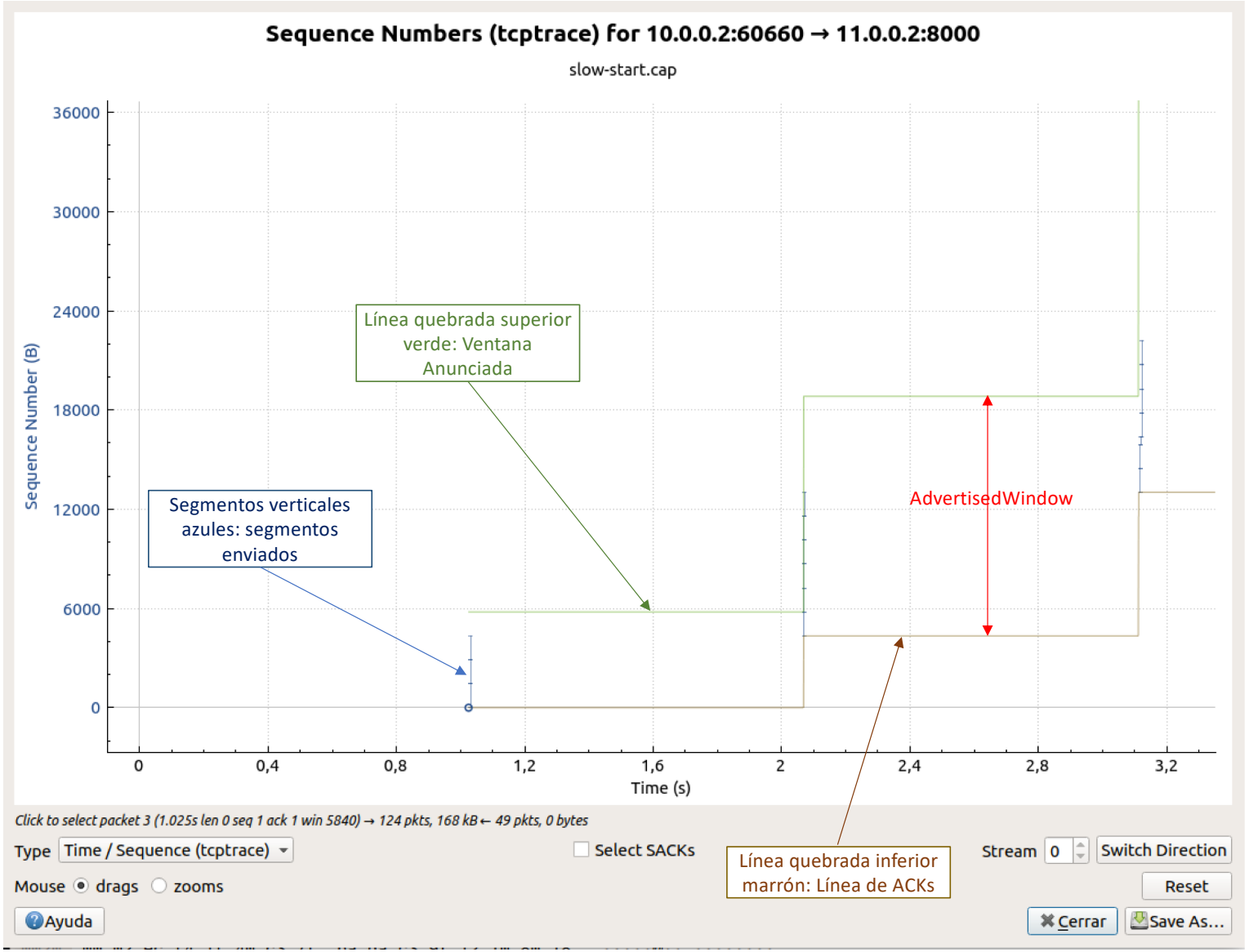
- Una vez iniciada la aplicación servidor TCP, ésta se queda esperando recibir mensajes de una aplicación cliente TCP.
- Una vez iniciada la aplicación cliente TCP, ésta intercambiará unos mensajes de control (apertura de conexión) con la aplicación servidor, por lo que es imprescindible que dicha aplicación servidor haya sido lanzada antes.
- Si la comunicación entre ambas aplicaciones es posible, a partir de este momento todo lo que escribamos a través de la entrada estándar de una aplicación será enviada a la otra: si escribimos en el terminal de la aplicación cliente, esto será enviado a la aplicación servidor, y viceversa.
- Para interrumpir la ejecución de estas aplicaciones se debe utilizar `Ctrl+C`.

Contenidos

Gráfica de *tcptrace* dentro de Wireshark

- En Wireshark, además de mirar el contenido de los paquetes de una conexión TCP, puede verse en una gráfica la **evolución del envío de datos y recepción de acks respecto al tiempo**.
- Wireshark permite mostrar varios tipos de gráficas de una conexión TCP: Nosotros utilizaremos la gráfica de *tcptrace*.
- Como una conexión TCP permite el envío de datos en ambos sentidos, se pueden visualizar 2 gráficas de *tcptrace* diferentes: las correspondientes a cada sentido de la comunicación.
- Para ver en Wireshark la gráfica de *tcptrace* de uno de los sentidos de una conexión TCP es necesario:
  - Cargar el fichero de una captura que contenga los paquetes de una conexión TCP.
  - Seleccionar un segmento de la conexión del sentido de la comunicación que queremos analizar (si el segmento seleccionado va del proceso A al proceso B, la gráfica que se mostrará será la correspondiente al envío de datos de A a B).
  - Seleccionar en el menú de Wireshark: **Statistics→TCP Stream Graph→Time-Sequence Graph (tcptrace)**

wireshark: Apariencia



wireshark: Controles

- **Rueda del ratón**: zoom in/out
- **Arrastrar con el botón izquierdo**: desplazar el gráfico (útil si se ha hecho "zoom in")
- **ESPACIO**: activa/desactiva una cruz para ayudar a ver sobre los ejes la posición del ratón.
- **Click izquierdo sobre un segmento**: seleccionar el paquete concreto en la lista de paquetes de Wireshark.
- **Botón 'Switch Direction'**: pasa a mostrar el otro sentido de la conexión.

Seguimiento de conexiones con tcpdump en el terminal