Práctica 3 - Midiendo retardos en Ethernet

1. Introducción

El tiempo que tarde la información que quiera enviar una aplicación a otra atravesando una red de conmutación de paquetes depende del tiempo que le cueste a cada paquete atravesar la red y de cómo se secuencien entre ellos los paquetes enviados. En esta práctica nos vamos a centrar en el tiempo que tarde un paquete en atravesar un camino por una red.

El tiempo que tarde un paquete va a depender del tiempo que se tarde en transmitir por cada enlace, del tiempo que tarde en propagarse la señal por cada medio y del tiempo que pase el paquete en cada uno de los equipos de conmutación (mientras es procesado o mientras espera a que se le asigne el medio de salida y pueda comenzar su transmisión). En general, en redes de área local como la que vamos a emplear, los tiempos de propagación van a ser muy pequeños y difíciles de medir (por debajo del microsegundo). Los tiempos de transmisión van a depender del tamaño de los paquetes y de la tasa de transmisión.

Dado que vamos a emplear los PCs del laboratorio para medir, debemos entender que el propio proceso de medida va a introducir errores. Por ejemplo, si se está enviando una trama Ethernet de 65 bytes por un enlace a 100Mb/s el tiempo de transmisión de ésta será de 5.12µs. Necesitaríamos que la marca de tiempo que indica cuándo se recibe un paquete fuera puesta por la tarjeta de red, ya que si esperamos a que el paquete sea copiado a memoria del sistema operativo es probable que cualquier otra tarea que esté llevando a cabo la CPU retrase la atención a la tarjeta de red e introduzca un error en esa marca de tiempo, posiblemente de órdenes de magnitud superiores. Si queremos medir tiempos en diferentes máquinas nos encontraremos con que los relojes de éstas no están sincronizados con la precisión necesaria para medidas tan finas. Por no complicar el proceso de medida intentaremos llevar a cabo la práctica con mecanismos simples, entendiendo siempre que algunas medidas pueden tener error en los tiempos por estos y otros fenómenos.

El objetivo de esta práctica es familiarizarse con las dependencias de los diferentes retardos con parámetros como los tamaños de los paquetes, las tasas de transmisión o el número de saltos.

Emplearemos uno solo de los PC (PC A, B o C). Enviaremos tramas Ethernet por uno de sus interfaces para que sigan un camino por una LAN hasta llegar a otro de los interfaces de ese mismo PC. Queremos medir cuándo enviamos el paquete y cuándo llega al destino, y la mejor forma es hacer ambas medidas en la misma máquina ya que eso elimina problemas de sincronización de relojes. Si enviáramos desde un PC y el destino final donde medimos la llegada fuera otro PC, un pequeño desfase en los relojes de las dos máquinas haría muy complicado medir diferencias de tiempos como los que se van a dar en escenario de LAN, como mucho de unos pocos milisegundos.

2. Tiempos en un enlace

Escoja uno de los PCs (A, B o C). Interconecte su interfaz `eth0` con el interfaz `eth1` (Figura 1) empleando un cable cruzado (los interfaces de esos PCs no implementan

Auto-MDI/MDI-X). Active ambos interfaces. Lance el programa `wireshark` y póngalo a capturar de ambos interfaces simultáneamente (en la ventana donde selecciona interfaz puede seleccionar más de uno a la vez).

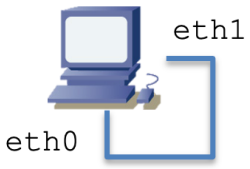

Figura 1 – Dos interfaces del mismo PC interconectados

Emplee `ethSend` para enviar paquetes por el interfaz `eth0`. Haga un envío sin fin de paquetes del tamaño mínimo con separación de 5s entre ellos. Debería aparecer en `wireshark` cada paquete dos veces, una de ellas al verlo por el interfaz `eth0` cuando se envía y otra al recibirlo en el interfaz `eth1`.

En `wireshark` puede mostrar la diferencia de tiempo entre un paquete y el anterior. Busque la opción en el menú `View > Time Display Format`.

Deberían verse tiempos pequeños entre el paquete enviado y el recibido, seguidos por el tiempo de 5s hasta el siguiente envío. En ocasiones podrían aparecer diferencias de tiempo negativas por la forma en que `wireshark` haya leído de los dos interfaces. Nos quedaremos con el valor absoluto de estas diferencias de tiempo.

La herramienta `tshark` es similar a `wireshark` pero su uso es enteramente en línea de comandos. Pruebe en otro terminal mientras `ethSend` está enviando, por ejemplo:

```
$ tshark -i eth0
```

Puede controlar la información que muestra para cada paquete. Por ejemplo, con las siguientes opciones leerá de los dos interfaces y mostrará un contador del número de paquete capturado y la diferencia de tiempo con el anterior:

```
$ tshark -i eth0 -i eth1 -l -T fields -e frame.number -e frame.time_delta_displayed
```

Para que tenga una visualización más cómoda le vamos a dar una forma sencilla de hacer una gráfica con esos datos.

Modifique las opciones de `ethSend` para que envíe un paquete cada 500ms (simplemente para no tener que esperar 5s entre paquetes).

En otro terminal lance el comando `tshark` con las opciones anteriores, redireccionando la salida a un fichero de nombre `captura.txt`

Ahora en un terminal debería tener `ethSend` enviando, en otro tiene `tshark` capturando y mandando líneas de texto a un fichero. En un tercer terminal lance el programa `gnuplot`:

```
$ gnuplot
```

Este programa es interactivo y con él puede hacer una gráfica a partir de un fichero de texto con los datos en columnas simplemente con:

```
gnuplot> plot "captura.txt" using 1:2
```

La parte de "`using 1:2`" indica que la columna 1 tiene los datos de abscisas y la columna 2 los datos de ordenadas para la gráfica.

Como ese fichero está creciendo a medida que se capturan más paquetes, vamos a automatizar que `gnuplot` rehaga la figura cada segundo. Para ello vamos a hacer lo siguiente:

Cree el fichero `procesa_captura.awk` con el siguiente código:

```
{
      if ($2<0) dato=-$2;
      else dato=$2;

      if (dato<0.4) print $1,dato;
}
```

Esto será un script para el programa `awk`. Si tiene interés, busque información sobre este programa, muy útil para procesar ficheros de texto con datos en columnas.

Abra `gnuplot` y escriba los siguientes comandos (tenga cuidado de no copiar los *prompts*):

```
gnuplot> while (1) {
more> plot "< tail –n 100 captura.txt | awk –f procesa_captura.awk" using
1:($2*1000)
more> set yrange [0:*]
more> pause 1
more>}
```

Como puede intuir por el código, es un bucle infinito donde se dibuja la gráfica, hace una pausa de un segundo y repite. El proceso de dibujo de la gráfica es un poco más complejo porque no se le da directamente el fichero de datos a dibujar, sino que se le pide a `gnuplot` que lance un programa para generar los datos en el momento. Mediante el programa `tail` nos estamos quedando solo con las últimas 100 líneas del fichero de captura, esta salida se está mandando mediante una *pipe* (|) al programa `awk`, el cual está eliminando las diferencias superiores a 0.4 segundos, así como se están haciendo positivas las diferencias negativas (para eso era el fichero `procesa_captura.awk`, que es un script simple para el programa `awk`). También estamos escalando los valores del eje de ordenadas a milisegundos y configurando la escala de dicho eje para que siempre empiece en 0.

Con eso debería tener una gráfica que se va actualizando a medida que aparecen nuevos datos en el fichero. Puede usar esta visualización para los experimentos del resto de la práctica. Si en algún momento borra o sobre-escribe el fichero `captura.txt` es probable que tenga que poner de nuevo el bucle de dibujado en `gnuplot`.

Varíe parámetros del programa `ethSend` (deteniéndolo y volviéndolo a lanzar). En

concreto pruebe con diferentes tamaños de la trama. Debería ver el efecto de dicho tamaño en el retardo sufrido por los paquetes.

Cambie la velocidad de uno de los interfaces a 10Mb/s (si la autonegociación está activada en dichos interfaces el otro extremo debería adecuarse a ella). Emplee para ello `ethtool`:

```
$ sudo ethtool -s eth0 speed 10 duplex full
```

Compruebe el efecto del tamaño del paquete y de la tasa de transmisión en los retardos.

Punto de control 1: Muestre mediante la gráfica las diferencias de retardos al cambiar los parámetros mencionados.

2. Store & forward en un conmutador

Interponga un conmutador entre los dos interfaces del PC que está empleando. Es decir, construya algo similar a la Figura 2.

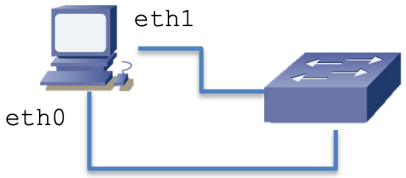

Figura 2 – 2 interfaces del mismo PC al mismo switch

En el conmutador D-Link los puertos 1 al 16 negocian velocidad (10, 100 ó 1000 Mb/s) mientras que los puertos 17 a 24 están forzados a 10Mb/s. Recuerde que las NICs de los PCs son a 10/100 Mb/s.

Empiece probando con un salto por un switch donde ambos enlaces son a 100Mb/s y variando el tamaño de los paquetes para ver su efecto. Es posible que tenga que volver a configurar el interfaz para que anuncie todas las velocidades que soporta:

```
$ sudo ethtool –s eth0 advertise 0x00f
```

Si le interesa, puede ver que significa el 0x00f con `man ethtool`.

Después, modifique uno de los dos enlaces para que sea a 10Mb/s (empleando un puerto del switch a 10Mb/s o forzando los 10Mb/s en el interfaz del PC). De nuevo vea el efecto de la velocidad y el tamaño del paquete.

Finalmente ponga los dos enlaces a 10Mb/s y repita los experimentos.

3. Store & forward con dos conmutadores

Interponga dos conmutadores entre los dos interfaces, de forma que pueda ver el efecto de la tasa de transmisión en cada uno de ellos (Figura 3).

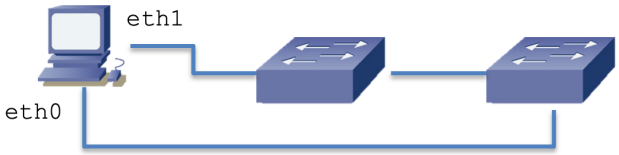

Figura 3 – 2 interfaces del mismo PC con un camino por 2 switches

Puede emplear el switch0. Todos sus puertos son 10/100 y deberían negociar la velocidad según la del otro extremo del enlace.

Según cual sea el fabricante del switch 0 de su armario, esos equipos pueden estar enviando tráfico esporádicamente. Este tráfico puede introducir ruido en las medidas de tiempos ya que nuestros paquetes pueden sufrir encolamiento en los conmutadores.

Por lo menos, podemos filtrar el tráfico que capturamos para no verlo en la gráfica aplicando un filtro de captura en `tshark`. Deberemos indicar un filtro para cada interfaz. Si ve paquetes que prefiere ignorar, decida el filtro a aplicar. El formato de filtro es BPF.

```
$ tshark -i eth0 -f "<capture filter>" -i eth1 -f "<capture filter>" -l -T fields -e frame.number -e frame.time_delta_displayed >> captura.txt
```

Repita los experimentos.

Punto de control 2: Muestre mediante la gráfica las diferencias de retardos al cambiar el escenario.

4. Pérdidas de paquetes

En este apartado vamos a ver pérdidas en el conmutador, dándole más paquetes para reenviar por un puerto de los que pueden salir en un breve periodo de tiempo.

Repita la configuración de la Figura 2. Consiga que uno de los enlaces funcione a 100Mb/s y el otro a 10Mb/s. Emplee `ethSend` para enviar un flujo continuo de tramas por el interfaz configurado a 100Mb/s. Queremos que estas tramas se reenvíen por el interfaz de 10Mb/s. Como está enviando por un puerto a 100Mb/s puede enviar paquetes a una tasa promedio superior a 10Mb/s. Escoja un tamaño de paquete y un tiempo entre paquetes para conseguir una tasa sostenida que esté por ejemplo entre los 15 y los 20Mb/s.

Mida con `wireshark` la tasa promedio del flujo que envía por un interfaz y el que recibe por el otro. Cuente en un intervalo de tiempo la cantidad de paquetes enviados y la cantidad recibida.