

ANÁLISIS DE DESEMPEÑO DEL EMULADOR DE CANAL NETEM SOBRE EL PROTOCOLO TCP

Luis Paredes Caicedo

e-mail: luchoparedes55@hotmail.com

Gonzalo Olmedo Cifuentes

e-mail: gfolmedo@espe.edu.ec

Carlos Romero Gallardo

e-mail: cgromero@espe.edu.ec

ESCUELA POLITÉCNICA DEL EJÉRCITO DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA SANGOLQUÍ – ECUADOR

RESUMEN

En el presente proyecto se realizó un análisis de desempeño del emulador de canal Netem bajo el protocolo TCP, mediante la implementación de escenarios de pruebas con la ayuda de ordenadores, en las cuales se realizaron varias inyecciones de tráfico mediante D-ITG con cada uno de los parámetros individualmente que proporciona el emulador, así también una inyección de tráfico representando un posible escenario real con todos los parámetros en conjunto, bajándose a esta experiencia se procede a realizar guías prácticas de laboratorio las mismas que ayudaran al uso correcto del emulador en futuras investigaciones; luego de este proceso se puede decir que NetEm es una herramienta muy útil para representar los factores que afectan a las transmisiones de redes inalámbricas porque proporciona las opciones estadísticas necesarias para emular la repuesta de una red real.

INTRODUCCIÓN

Aunque en sus comienzos Internet se entendía como una infraestructura de investigación, hoy en día se ha convertido en un entorno operativo y es más difícil de usar como plataforma de pruebas. Muchas universidades y departamentos de I+D que trabajan en redes y protocolos necesitan herramientas y entornos para verificar su funcionamiento en condiciones reales. En ocasiones, las pruebas son muy difíciles de realizar debido al gran número de máquinas y entornos que se requieren, existen varias soluciones posibles para este problema, una pasa por el uso de herramientas de simulación que permiten realizar medidas controladas y repetibles a bajo coste. Tienen dos inconvenientes: el primero es la carga computacional, especialmente cuando el escenario de red incluye un gran número de máquinas; el segundo se debe al hecho de que los simuladores simplifican el sistema a medir, alejándolo de la realidad, usando implementaciones específicas de los protocolos, y no

permitiendo el uso de cualquier aplicación o sistema operativo (SO).

Una opción es utilizar hardware real, que incluye la pila de protocolos completa del SO, esta puede ser una buena solución en muchos casos, pero también se han desarrollado muchos *testbed* y emuladores para facilitar las pruebas y reducir el coste. Algunos son híbridos, combinando las ventajas de la simulación, la emulación y las pruebas con equipos reales.

NETEM

NetEm provee la funcionalidad de emulación de red para realizar pruebas de protocolos, emulando las propiedades de una red WAN. La actual versión puede emular demora, pérdidas, duplicación, corrupción y reordenamiento de paquetes [3].

NetEm se controla a través de la línea de comando, con el comando *tc*, que es parte del paquete de herramientas de *iproute2*.

NetEm es una mejora reciente a las funcionalidades de control de tráfico de Linux. Se construye utilizando las funcionalidades existentes en Linux de QoS y Servicios Diferenciados (DiffServ).

La motivación que hay detrás de NetEm es la de proveer una manera de reproducir grandes redes en un ambiente de laboratorio. Primeramente se utilizó para evaluar nuevas mejoras a TCP en Linux.

DISEÑO NETEM

NetEm consiste de dos partes, un pequeño modulo del kernel de Linux para disciplina de colas y una herramienta, *tc*, de línea de comando para configurarlo. El modulo está integrado desde la versión 2.6.8 y 2.4.28 del kernel, y la herramienta es parte del paquete *iproute2* [1].

La Figura 1 muestra la arquitectura básica de NetEm. Las disciplinas de cola existen entre la salida del protocolo y el dispositivo de red [2].

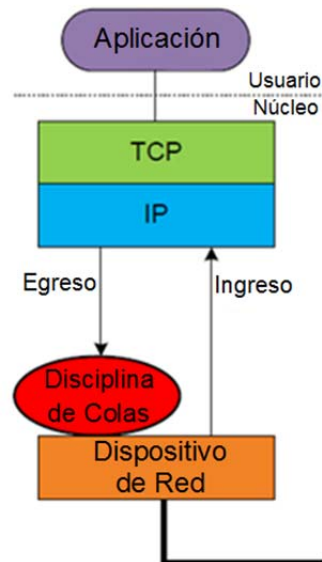


Figura 1 NetEm Básico

PARÁMETROS NETEM

El usuario especifica los parámetros de emulación a través de la herramienta *tc*. Sin parámetros, NetEm se comporta como una cola TFIFO sin demora, pérdida, duplicación o reordenamiento de paquetes.

Demora

Las redes de computadoras no muestran siempre la misma demora, ésta varía en base a otros flujos de tráfico que viajan por el mismo camino.

Pérdidas

La pérdida de paquete está implementada como la eliminación de un cierto porcentaje de paquetes al azar antes de ser encolados.

Duplicación

En redes con hardware confiable no duplican paquetes, pero con routers redundantes y hardware real la duplicación puede ocurrir.

Corrupción

Ruido aleatorio en los enlaces causa que algunos paquetes lleguen corruptos a su destino.

Reordenamiento

Reordenamiento de paquete ocurre cuando paquetes atraviesan caminos con diferentes demoras. Algunos routers de alta velocidad usan múltiples buses y procesadores que internamente crean caminos alternativos. Los paquetes son reordenados cuando diferentes procesadores y buses tienen diferente demora.

ESCENARIO DE PRUEBAS

Para la ejecución de NETEM se implementa el escenario de pruebas que se muestra en la Figura 2, para la

inyección de tráfico es necesario que tanto el PC emisor como el PC Receptor posean software de inyección de tráfico D-ITG.



Figura 2 Escenario de Pruebas

PC A

eth0 => ip estática (192.168.0.1)

PC B

eth0 => ip estática (192.168.0.4)

NETEM BOX

eth0 y eth1 => ip dinámicas

La configuración de parámetros de inyección de tráfico se detalla en la Tabla 1 para *Define Flow* y Tabla 2 para *Settings* en el emisor. Los parámetros restantes se dejan con la configuración por defecto.

Tabla 1 Parámetros de Inyector de Tráfico (*Define Flow*) – Emisor

Stream Option	
Meter	One-Way-Delay
Duration	60 (1minute)
Start Delay	0 (default)
Randon Seed	0 (default)
Headers Options	
Target host	192.168.0.4
Protocol	TCP
Destination Port	Default
Sourse Port	(Auto)

Tabla 4.2 Parámetros de Inyector de Tráfico (Settings) – Emisor

Sender Option	
Local Sender log	Local
Logging Server	-
Remote Receiver	Remote
Logging Server	192.168.0.4
Local Receiver Option	
Logging Type	None
Logging Server	-

En la Tabla 3 se muestra la configuración de parámetros de *Settings* en el receptor, no es necesario configurar *Define Flow* en el receptor ya que este hace la función de escuchar y no transmite.

Tabla 3 Parámetros de Inyector de Tráfico (Settings) – Receptor

Sender Option	
Local Sender log	None
Logging Server	-
Remote Receiver	None
Logging Server	-
Local Receiver Option	
Logging Type	Local
Logging Server	-

APLICACIÓN DE PARÁMETROS DE UN POSIBLE ESCENARIO REAL EN EL EMULADOR

En esta sección se aplica una configuración con todos los parámetros al mismo tiempo que permite usar el emulador NetEm, en la Tabla 4 se tiene los porcentajes y valores muy cercanos a

los de un posible escenario real que se configurara en la emulación.

Tabla 4 Parámetros de un posible escenario real en caja NetEm

PARÁMETRO \ MEDIDA	%	ms
Demora	-	100
Perdida	10	-
Duplicación	15	-
Corrupción	10	-
Reordenamiento	20	-

Para configurar el emulador se digita el comando que se muestra a continuación lo que provocara en la tarjeta de red eth1 una demora de 100ms, una pérdida de 10%, una duplicación del 15%, una corrupción del 10% y el reordenamiento del 20%, todos estos parámetros actuaran simultáneamente.

```
tc qdisc add dev eth1 root
netem delay 100ms loss 10%
duplicate 15% corrupt 10%
reorder 20%
```

ANÁLISIS DE RESULTADOS

Para evaluar el desempeño de Netem en un posible escenario real bajo el protocolo TCP se aplica simultáneamente todos los parámetros que nos proporciona el emulador, una vez realizado la inyección de tráfico con D-ITG, en la Figura 3 se muestra un Throughput relativamente bajo y muy cambiante sin sobrepasar los 900 Kbit/s, en

algunos instantes llega hasta ser nulo, con 200 Kbit/s de promedio en toda la inyección.

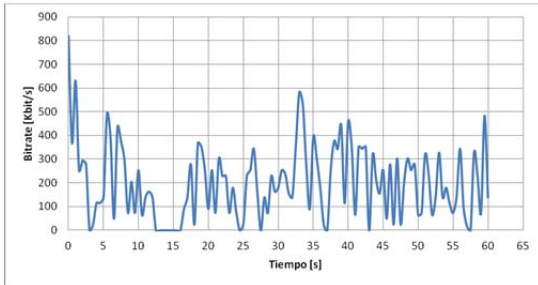


Figura 3 Throughput en función del tiempo de un posible escenario real.

El retraso que presenta la Figura 4 no solamente está relacionado por la configuración del parámetro de Demora, ya que otros parámetros como el de pérdidas también influyen, el desempeño general de la transmisión nos muestra un retardo promedio de 1.92 s, con un pico máximo de 7.73 s y mínimo de 0.22 s con lo que se puede afirmar que los parámetros configurados en la caja NetEm están desempeñando de manera eficiente.

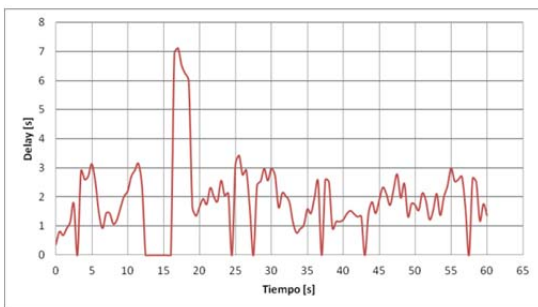


Figura 4. Retardo en función del tiempo de un posible escenario real.

Con un porcentaje de corrupción de 10% y considerando la influencia de los demás parámetros configurados en el emulador se observa en la Figura 5 un jitter promedio de 0.04 s.

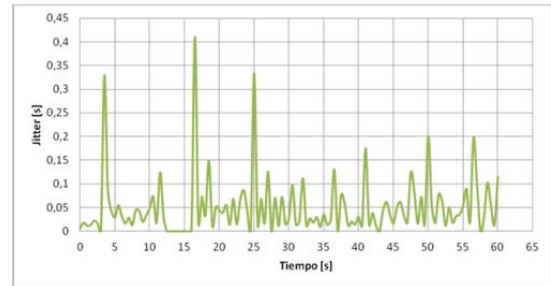


Figura 5. Jitter en función del tiempo de un posible escenario real.

En la Tabla 5 se presenta con detalle los valores máximos, mínimos y promedio de las mediciones efectuadas.

Tabla 5 Archivo de resultados de inyección de tráfico de un posible escenario real.

```
ITGDec version 2.8.0-rc1 (r457:458)
Compile-time options:
-----
Flow number: 1
From 192.168.0.1:33336
To 192.168.0.4:8999
-----
Total time = 60.937482 s
Total packets = 2987
Minimum delay = 0.220986 s
Maximum delay = 7.726704 s
Average delay = 1.923996 s
Average jitter = 0.040376 s
Delay standard deviation = 1.200314 s
Bytes received = 1529344
Average bitrate = 200.775477 Kbit/s
Average packet rate = 49.017450 pkt/s
Packets dropped = 0 (0.00 %)
Average loss-burst size = 0.000000 pkt
-----
```

***** TOTAL RESULTS *****

```
-----
Number of flows = 1
Total time = 60.937482 s
Total packets = 2987
Minimum delay = 0.220986 s
Maximum delay = 7.726704 s
Average delay = 1.923996 s
Average jitter = 0.040376 s
Delay standard deviation = 1.200314 s
Bytes received = 1529344
Average bitrate = 200.775477 Kbit/s
Average packet rate = 49.017450 pkt/s
Packets dropped = 0 (0.00 %)
Average loss-burst size = 0 pkt
Error lines = 0
-----
```

CONCLUSIONES

NetEm es una herramienta muy útil para representar los factores que afectan a las transmisiones de redes inalámbricas bajo el protocolo TCP. Proporciona las opciones estadísticas necesarios para emular la respuesta de la red de escenarios reales.

Con el inyector de tráfico D-ITG se generó el tráfico unidireccional necesario en el lado del emisor, el mismo que atravesó el emulador de canal NetEm, en el cual se configuro cada parámetro con valores y porcentajes altos que alteran el trafico inyectado, al llegar al receptor se pudo conseguir los valores y archivos de mediciones por medio del analyzer que es una parte de DITG, para posteriormente generar las gráficas de Bitrate, Delay y Jitter para su respectivo análisis.

Las guías prácticas de laboratorio se desarrollaron basadas en la experiencia de investigación e implementación de los diferentes escenarios de pruebas que se realizó con NetEm durante el presente proyecto de tesis.

Para la implementación del escenario de pruebas donde se ejecutó NetEm no requirió incurrir en gastos mayores, ya que se utilizó software libre y ordenadores existentes, solamente se adquirió una tarjeta de red que se instaló en el ordenador que actuó como caja NetEm para poder realizar el puente de red.

En el presente proyecto de tesis se realizó un análisis de cada uno de los diferentes parámetros que permite

configurar NetEm como así también utilizando todos los parámetros al mismo tiempo, para poder comprender su funcionamiento, no se efectuó emulaciones de casos específicos de comportamiento de alguna red.

REFERENCIAS

- [1] CodeAlias. (s.f.). Netem, tc and qdisc loss and delay emulation examples. Recuperado el 5 de Junio de 2012, de CodeAlias: http://www.codealias.info/technotes/network_delay_and_packet_loss_emulation_using_netem
- [2] Hemminger, S. (2005). *Network Emulation with NetEm*. Open Source Development Lab.
- [3] Saldaña, J. M., Murillo, J., Fernández-Navajas, J., Ruiz-Mas, J., Viruete, E. A., & Aznar, J. I. (2010). *EMULACIÓN DE ESCENARIOS DE RED MEDIANTE UN TESTBED*. Zaragoza: Universidad de Zaragoza.

BIOGRAFÍA

Luis Edison Paredes Caicedo



Nació en Ambato, Ecuador el 20 de junio de 1985.

Sus estudios primarios los realizo en la Escuela Particular “Santo Domingo de Guzmán” y sus estudios secundarios en el Colegio “San Alfonso”, obteniendo el título de Bachiller en ciencias, especialidad Físico

Matemático, sus estudios superiores los realizó en la Escuela Politécnica del Ejército en la carrera de Ingeniería Electrónica en Telecomunicaciones.

Gonzalo F. Olmedo C. recibió el título de Ingeniero en Electrónica y Telecomunicaciones de la Facultad de Ingeniería Electrónica de la Escuela Politécnica del Ejército, en 1998, el grado de Master en Ciencias en Ingeniería Eléctrica y Computación en el área de Telecomunicaciones y Telemática, Universidad Estatal de Campinas (UNICAMP), Campinas - Brasil, 2003. Doctor en Ingeniería Eléctrica y Computación en el área de Telecomunicaciones y Telemática, Universidad Estatal de Campinas

(UNICAMP), Campinas - Brasil. Sus intereses investigativos incluyen las Ad Hoc networks, sistemas de comunicación Sem Fio CDMA, Desempeño de las redes de Telecomunicaciones.

Carlos G. Romero G. Ingeniero en Electrónica y Telecomunicaciones en la Escuela Politécnica del Ejército (2002) y Especialista en Proyectos de Investigación Científica y Tecnológica en la Universidad Complutense de Madrid (2006). Candidato a PhD Universidad Nacional de la Plata. Es profesor de la Escuela Politécnica del Ejército. Sus áreas de interés e investigación son Networking con TCP/IP e Implementación de servicios y aplicaciones con software libre.