



## **I. TEMA: EVALUACION DEL CONTROL DE CONGESTIONAMIENTO EN TCP**

## **II. OBJETIVOS**

Al finalizar la presente práctica el estudiante:

1. Comprende y explica los mecanismos de control de congestionamiento del protocolo TCP.
2. Utiliza la versión del protocolo TCP más adecuada según el entorno de red.

## **III. TRABAJO PRELIMINAR**

Para el presente laboratorio, es necesario que el estudiante esté familiarizado con conceptos y habilidades tales como:

1. Conceptos teóricos de los mecanismos de control de congestionamiento del protocolo TCP
2. Manejo de la plataforma Linux
3. Manejo de máquinas virtuales

## **IV. MATERIALES Y EQUIPOS.**

Los materiales que utilizaremos en los trabajos de laboratorio son:

1. LiveCD WANem
2. Sistema operativo Linux para terminal de red.
3. 02 cables UTP cruzados.
4. Computador con 02 interfaces de red Ethernet para la ejecución de WANem
5. 02 terminales de red



## V. MARCO TEORICO.

### CONTROL DE CONGESTIONAMIENTO EN TCP

TCP es un protocolo adaptivo, que ajusta sus parámetros de operación de acuerdo al desempeño que presenta la red de datos.

Por ejemplo, monitorea y reajusta el RTT (Round Trip Time) para cada segmento confirmado para modificar el RTO (Retransmission TimeOut), parámetro que controla el tiempo que el transmisor espera antes de dar por perdido un segmento y retransmitirlo.

De acuerdo al RFC 6298, para el cálculo del RTO, se utilizan las siguientes fórmulas:

$$SRTT = (1 - \alpha) \times SRTT + \alpha \times RTT'$$

$$RTTVar = (1 - \beta) \times RTTVar + \beta \times |SRTT - RTT'|$$

$$RTO = SRTT + \max(G, k \times RTTVar)$$

Donde,

SRTT	-	RTT suavizado
RTTVar	-	Varianza de RTT
RTT'	-	RTT muestreado
$\alpha$	-	1/8
$\beta$	-	1/4
G	-	Granularidad de reloj de interrupciones en segundos (1 segundo)
k	-	4

Al iniciar una conexión (cuando se envía el primer SYN), RTO se establece en 1 segundo. Cuando regresa la primera confirmación (SYN – ACK), su valor RTT se almacena en SRTT y la varianza se establece en RTT/2. A partir de ese punto, el RTO se calcula mediante las fórmulas indicadas.

Para evitar el congestionamiento de la red, TCP mantiene una ventana de congestionamiento (CongWin) cuyo tamaño limita la cantidad de bytes que el emisor puede enviar al host destino. El tamaño de esta ventana se ajusta dinámicamente.

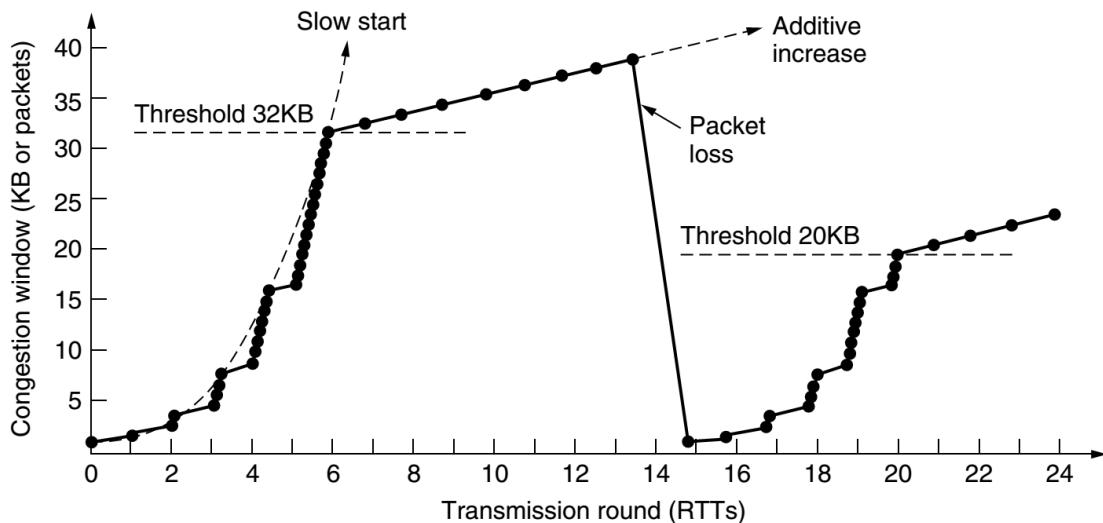
Para determinar la tasa de transmisión de segmentos, al inicio de una conexión, TCP utiliza el algoritmo Slow start: La tasa de transmisión inicial se establece en 1MSS/RTT, CongWin se inicia en 1 MSS, luego el emisor duplica la cantidad de datos en cada transmisión (crecimiento exponencial) hasta que ocurre un timeout o la ventana de congestión supera el slow start threshold.

Cuando ocurre un timeout el SST se iguala a CongWin/2.



Cuando se supera el SST, TCP pasa de slow start a additive increase (Incremento Aditivo). En este modo CongWin se incrementa en un MSS en cada RTT.

Cuando se reciben tres ACK duplicados, se asume que se perdió un segmento. En este caso se retransmite el segmento perdido sin esperar a que venza el temporizador (Fast Retransmission), se establece  $SST = \text{CongWin}/2$  y  $\text{CongWin} = 1\text{MSS}$ . El siguiente gráfico ilustra este algoritmo (Tanenbaum, 2011):



Esta implementación se denomina TCP Tahoe.

Además de TCP Tahoe existen variaciones como TCP Reno, TCP Vegas, TCP BIC, etc.

### GESTIÓN DEL PROTOCOLO TCP EN LINUX

En Linux, se encuentran disponibles diversos algoritmos de control de congestionamiento, los cuales se incluyen en número diverso según la distribución.

Una ventaja de la pila TCP/IP de Linux, es que podemos cambiar el algoritmo de acuerdo a nuestras necesidades, en tiempo de ejecución e incluso podemos seleccionar protocolos independientes para cada socket (vía programación).

La lista de protocolos de control de congestionamiento activos se puede obtener consultando la variable `tcp_available_congestion_control` en el directorio `/proc` (que mantiene las variables que controlan la operación del sistema operativo Linux)

```
Terminal - user@pointlinux: ~
File Edit View Terminal Tabs Help
root@pointlinux:/# cat /proc/sys/net/ipv4/tcp_available_congestion_control
bic cubic reno
```

La lista que incluye protocolos disponibles aun no cargados se puede consultar con:



# UNIVERSIDAD ANDINA DEL CUSCO

## ADMINISTRACIÓN DE SERVICIOS DE REDES

### GUIA DE LABORATORIO

ECP 4 de 16

```
root@pointlinux:/# ls /lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp*
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_bic.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_diag.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_highspeed.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_htcp.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_hybla.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_illinois.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_lp.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_scalable.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_vegas.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_veno.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_westwood.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_yeah.ko
```

## SIMULADOR DE REDES DE AREA AMPLIA WANEM

WANem, es un liveCD que permite simular una WAN.

Algunas de sus características son que puede permitirnos configurar parámetros como el retardo, la pérdida, la fluctuación, etc.

Una ventaja de simular una WAN con WANem, es que las condiciones de prueba se pueden controlar en mayor grado que en una red real, lo que a su vez ayuda a validar los resultados de las pruebas de forma consistente.

WANem puede descargarse libremente desde su sitio: <http://wanem.sourceforge.net/>

The screenshot shows the WANem configuration interface. At the top, there's a header with the TATA Consultancy Services logo, the WANem title, and links for Home, About, WANalyzer, Basic Mode (selected), Advanced Mode, Save/Restore, and Help.

The main area contains several configuration tables:

- Interface:** eth0
- Bandwidth:** Choose BW (Other) - Other: Specify BW(Kbps) 0
- Delay:** Delay time (ms) 0, Loss(%) 0, Duplication(%) 0, Reordering(%) 0, Corruption(%) 0
- Random Disconnect:** Type none, Idle timer Disconnect, Random Disconnect, Random connection Disconnect
- Random Connection:** Type none, MTTF Low, MTTF High, MTTR Low, MTTR High
- IP source address:** any, IP source subnet, IP dest address, any, IP dest subnet, Application port if any

At the bottom, there are buttons for Add a rule set, Apply settings, Reset settings, Refresh settings, and a checkbox for Display commands only, do not execute them.



## VI. PRACTICAS DE LABORATORIO.

1. Evalúe el comportamiento del RTT y de la ventana de congestiónamiento del protocolo TCP durante la transferencia de un archivo de 100MB a través de una red:
  - a) Sin retardo ni pérdida.
  - b) Con retardo de 20 ms y 10% de pérdida

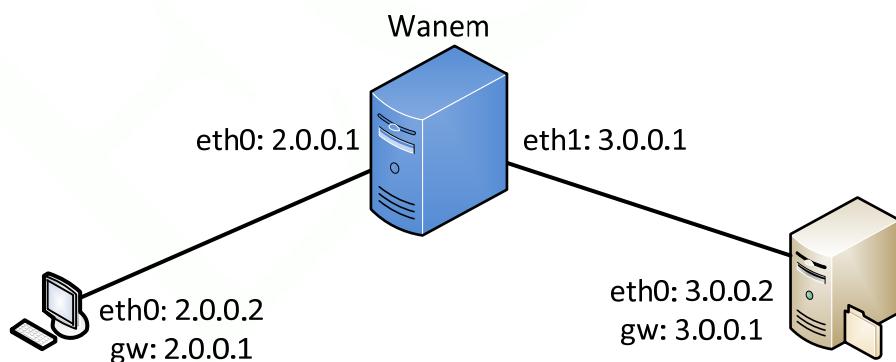
### SOLUCIÓN

Configuraremos la red de pruebas utilizando WANem para la simulación de la WAN y dos terminales Linux.

La red en mención tiene las siguientes especificaciones:



Detallamos los parámetros de configuración de la red que utilizaremos para simular la red:



### CONFIGURACION DE LAS MAQUINAS VIRTUALES PARA LA RED

La máquina virtual para ejecutar WANem debe contar con dos tarjetas de red, las cuales, debido a que todos los equipos de la red se ejecutarán en la misma máquina física, deben configurarse como redes internas (VirtualBox):

La interfaz eth0 se conectará a la red interna lanHost

La interfaz eth1 se conectará a la red interna lanServer



# UNIVERSIDAD ANDINA DEL CUSCO

## ADMINISTRACIÓN DE SERVICIOS DE REDES

### GUIA DE LABORATORIO

ECP 6 de 16

La máquina virtual para ejecutar el cliente se configura con tarjeta de red interna y se asocia con la red interna lanHost

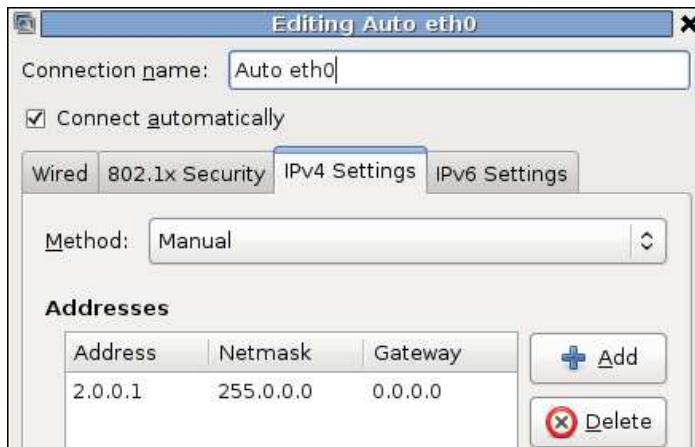
La máquina virtual para ejecutar el servidor se configura con tarjeta de red interna y se asocia con la red interna lanServer

#### CONFIGURACION DE WANEM

En WANem asignamos las direcciones IPv4 de la siguiente manera:

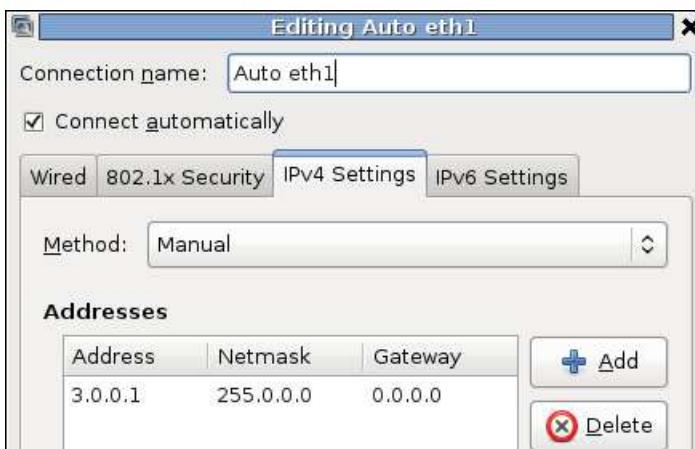
Interfaz eth0:

2.0.0.1/255.0.0.0



Interfaz eth1:

3.0.0.1/255.0.0.0



En la opción de configuración avanzada de WANem configuramos las características de la red de la siguiente manera:



# UNIVERSIDAD ANDINA DEL CUSCO

## ADMINISTRACIÓN DE SERVICIOS DE REDES

### GUIA DE LABORATORIO

ECP 7 de 16

Interfaz eth0:

delay: 10ms  
pérdida: 5%

The screenshot shows the WANem interface for emulating network conditions. The 'Basic Mode' tab is selected. Under the 'Interface' dropdown, 'eth0' is chosen. The 'Delay' section has a value of 10ms. The 'Loss' section has a value of 5%.

Interfaz eth1:

delay: 10ms  
pérdida: 5%

The screenshot shows the WANem interface for emulating network conditions. The 'Basic Mode' tab is selected. Under the 'Interface' dropdown, 'eth1' is chosen. The 'Delay' section has a value of 10ms. The 'Loss' section has a value of 5%.

## CONFIGURACION DEL SERVIDOR

eth0: 3.0.0.2/8 – gw: 3.0.0.1

```
Terminal - user@pointlinux: ~
File Edit View Terminal Tabs Help
root@pointlinux:/var/ftp# ifconfig eth0 3.0.0.2 netmask 255.0.0.0 up
root@pointlinux:/var/ftp# route add default gw 3.0.0.1 eth0
```

## CONFIGURACION DEL CLIENTE

eth0: 2.0.0.2/8 – gw: 2.0.0.1

```
root@slitaz:/home/tux# ifconfig eth0 2.0.0.2 netmask 255.0.0.0 up
root@slitaz:/home/tux# route add default gw 2.0.0.1 eth0
```



# UNIVERSIDAD ANDINA DEL CUSCO

## ADMINISTRACIÓN DE SERVICIOS DE REDES

### GUIA DE LABORATORIO

ECP 8 de 16

#### CONFIGURACION DE TCP

Para verificar el protocolo de control de congestionamiento activo en Linux utilizamos el comando

```
cat /proc/sys/net/ipv4/tcp_congestion_control
```

```
[root@slitaz ~]# cat /proc/sys/net/ipv4/tcp_congestion_control
cubic
```

```
[root@pointlinux ~]# cat /proc/sys/net/ipv4/tcp_congestion_control
vegas
```

Para ver la lista de protocolos de control de congestionamiento soportados por la distribución de Linux que esté utilizando, utilice el comando:

```
ls /lib/modules/<version_del_kernel>/net/ipv4/tcp*
```

```
[root@pointlinux ~]# ls /lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp*
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_bic.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_diag.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_highspeed.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_htcp.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_hybla.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_illinois.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_lp.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_scalable.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_vegas.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_veno.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_westwood.ko
/lib/modules/3.16.0-4-amd64/kernel/net/ipv4/tcp_yeah.ko
```

Para cambiar el protocolo de control de congestionamiento actual por uno diferente, utilizar el comando:

```
echo <protocolo> > /proc/sys/net/ipv4/tcp_congestion_control
```

Por ejemplo:

```
echo cubic > /proc/sys/net/ipv4/tcp_congestion_control
```

```
[root@pointlinux ~]# echo cubic > /proc/sys/net/ipv4/tcp_congestion_control
[root@pointlinux ~]# cat /proc/sys/net/ipv4/tcp_congestion_control
cubic
```

#### CONFIGURACION DEL SERVICIO DE TRANSFERENCIA DE ARCHIVOS

Para verificar el desempeño de los mecanismos de control de congestionamiento configuraremos un servidor de transferencia de archivos en el equipo servidor.



# UNIVERSIDAD ANDINA DEL CUSCO

## ADMINISTRACIÓN DE SERVICIOS DE REDES

### GUIA DE LABORATORIO

ECP 9 de 16

Si utiliza una distribución de Linux basada en Debian, instalamos un servidor FTP utilizando el comando

```
apt-get install vsftpd
```

```
root@pointlinux:/home/user# apt-get install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  dialog
The following NEW packages will be installed:
  dialog vsftpd
0 upgraded, 2 newly installed, 0 to remove and 81 not upgraded.
Need to get 406 kB of archives.
After this operation, 1,824 kB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

Creamos la carpeta /var/ftp, la cual se utilizará para almacenar los archivos que se transferirán

```
root@pointlinux:/home/user# mkdir /var/ftp
root@pointlinux:/home/user# ■
```

En la carpeta /var/ftp creamos un archivo de 100MB, para lo cual podemos utilizar el comando

```
truncate -s 100M cienMegas.txt
```

```
Terminal - user@pointlinux: ~
File Edit View Terminal Tabs Help
root@pointlinux:/var/ftp# truncate -s 100M cienMegas.txt
root@pointlinux:/var/ftp# ls -ls
total 0
0 -rw-r--r-- 1 root root 104857600 Oct 17 02:12 cienMegas.txt
```

Para permitir las sesiones anónimas, realizamos las siguientes modificaciones al archivo de configuración de vsftpd, ubicado en /etc/vsftpd.conf:

```
GNU nano 2.2.6          File: /etc/vsftpd.conf

listen=NO
listen_ipv6=YES
anonymous_enable=YES
local_enable=YES
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
secure_chroot_dir=/var/run/vsftpd/empty
anon_root=/var/ftp
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO
```



# UNIVERSIDAD ANDINA DEL CUSCO

## ADMINISTRACIÓN DE SERVICIOS DE REDES

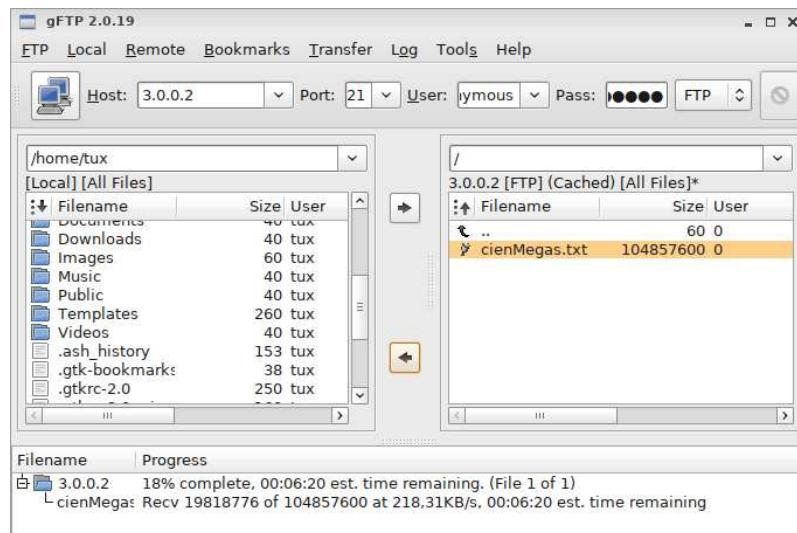
### GUIA DE LABORATORIO

ECP 10 de 16

Terminada la configuración de vsftpd, reiniciamos el servicio con la orden:

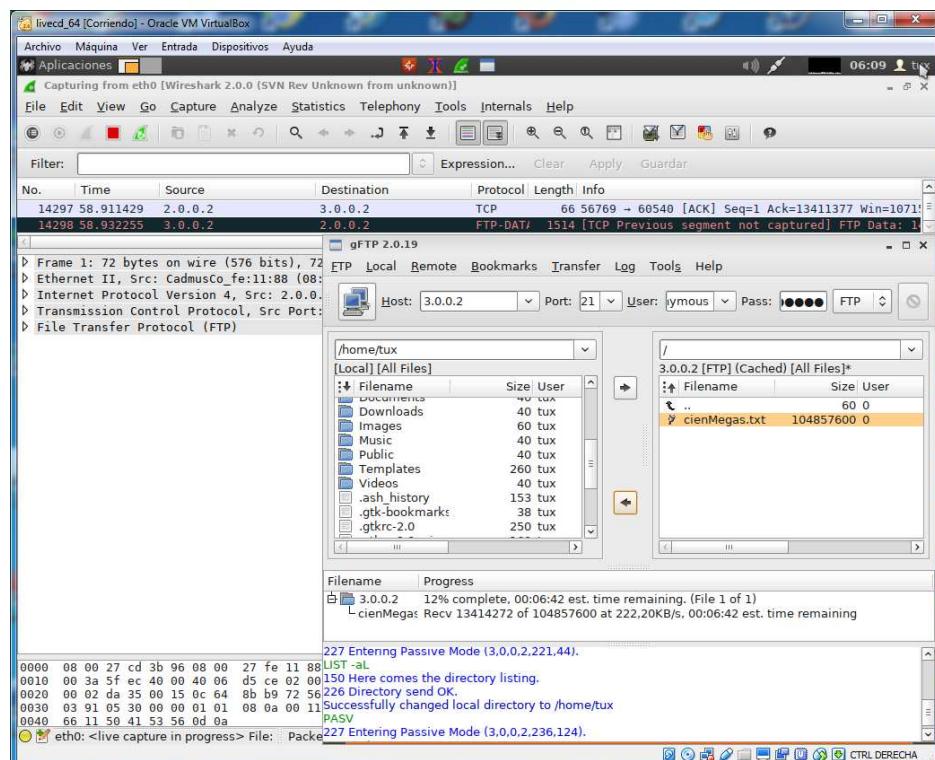
```
/etc/init.d/vsftpd reload
```

En la máquina cliente, debemos instalar un cliente FTP. En este caso utilizaremos la aplicación gFTP:



## WIRESHARK

En la máquina cliente instalamos Wireshark para capturar datos que luego utilizaremos para analizar el desempeño de TCP





# UNIVERSIDAD ANDINA DEL CUSCO

## ADMINISTRACIÓN DE SERVICIOS DE REDES

### GUIA DE LABORATORIO

ECP 11 de 16

Seleccionamos la interfaz que monitorearemos (eth0) en el cliente, iniciamos la captura de datos e inmediatamente iniciamos la transferencia mediante el cliente gFTP

No.	Time	Source	Destination	Protocol	Length	Info
45347	186.8318180	3.0.0.2	2.0.0.2	FTP-DAT/	1514	FTP Data: 1448 bytes
45348	186.83181819	2.0.0.2	3.0.0.2	TCP	66	56769 → 60540 [ACK] Seq=1 Ack=41702401 Win=1092:
45349	186.83181830	3.0.0.2	2.0.0.2	FTP-DAT/	1514	FTP Data: 1448 bytes
45350	186.83181832	2.0.0.2	3.0.0.2	TCP	66	56769 → 60540 [ACK] Seq=1 Ack=41703849 Win=1081:
45351	186.83181836	3.0.0.2	2.0.0.2	FTP-DAT/	1514	FTP Data: 1448 bytes
45352	186.83181838	2.0.0.2	3.0.0.2	TCP	66	56769 → 60540 [ACK] Seq=1 Ack=41705297 Win=1069:
45353	186.852846	3.0.0.2	2.0.0.2	FTP-DAT/	1514	[TCP Previous segment not captured] FTP Data: 1
45354	186.852856	2.0.0.2	3.0.0.2	TCP	78	[TCP Window Update] 56769 → 60540 [ACK] Seq=1 Ack=41705297 Win=1069:
45355	186.852868	3.0.0.2	2.0.0.2	FTP-DAT/	1514	FTP Data: 1448 bytes
45356	186.852869	2.0.0.2	3.0.0.2	TCP	78	[TCP Dup ACK 45352#1] 56769 → 60540 [ACK] Seq=1 Ack=41705297 Win=1069:
45357	186.873796	3.0.0.2	2.0.0.2	FTP-DAT/	1514	FTP Data: 1448 bytes
45358	186.873806	2.0.0.2	3.0.0.2	TCP	78	[TCP Dup ACK 45352#2] 56769 → 60540 [ACK] Seq=1 Ack=41705297 Win=1069:
45359	186.873817	3.0.0.2	2.0.0.2	FTP-DAT/	1514	FTP Data: 1448 bytes
45360	186.873819	2.0.0.2	3.0.0.2	TCP	78	[TCP Dup ACK 45352#3] 56769 → 60540 [ACK] Seq=1 Ack=41705297 Win=1069:
45361	186.894829	3.0.0.2	2.0.0.2	TCP	1514	[TCP Out-Of-Order] 60540 → 56769 [ACK] Seq=41705297 Ack=1 Win=1047:
45362	186.894845	2.0.0.2	3.0.0.2	TCP	66	56769 → 60540 [ACK] Seq=1 Ack=41712537 Win=1047:
45363	186.915834	3.0.0.2	2.0.0.2	FTP-DAT/	1514	FTP Data: 1448 bytes
45364	186.915835	2.0.0.2	3.0.0.2	TCP	66	56769 → 60540 [ACK] Seq=1 Ack=41712537 Win=1047:

Los datos capturados los almacenamos en formato pcapng, para su posterior análisis.

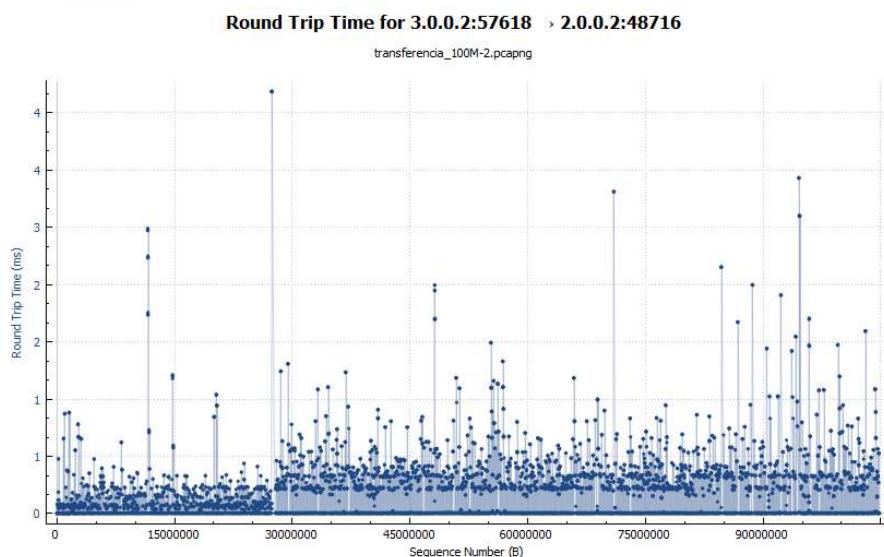
Repetimos las pruebas cambiando correspondientemente los parámetros de operación de WANem.

## ANALISIS DE RESULTADOS

Con la ayuda de Wireshark, mostramos nuestros resultados en forma gráfica

- a) Retardo 0 ms, Pérdida: 0%

RTT





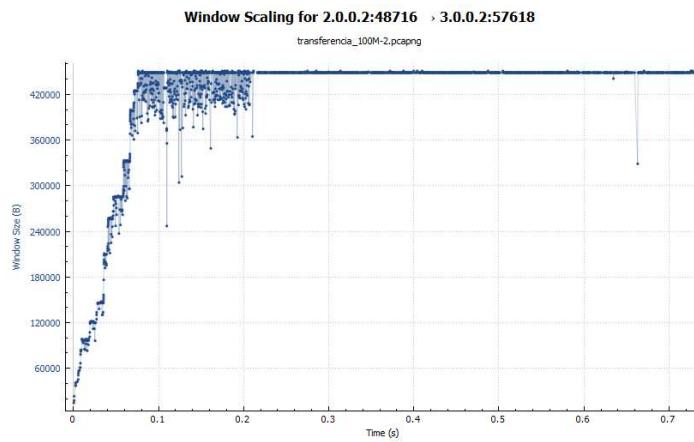
# UNIVERSIDAD ANDINA DEL CUSCO

## ADMINISTRACIÓN DE SERVICIOS DE REDES

### GUIA DE LABORATORIO

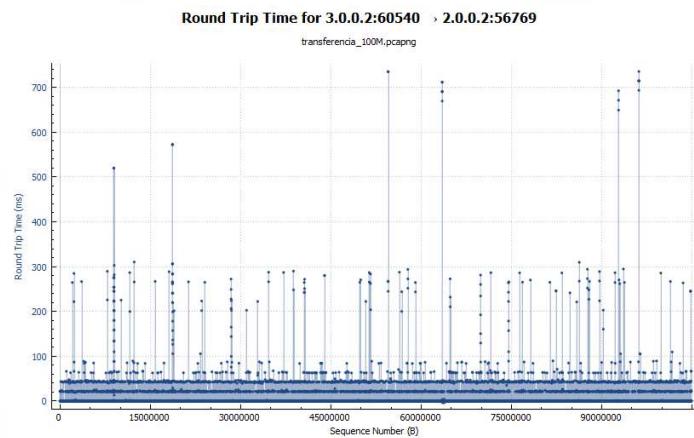
ECP 12 de 16

#### Ventana de congestión

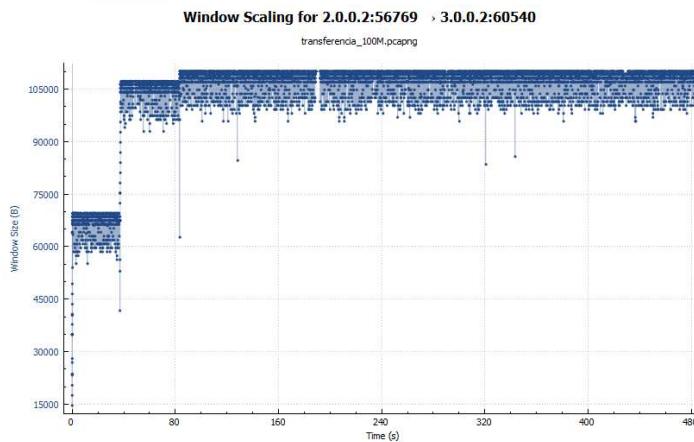


- b) Retardo 10 ms, Pérdida: 5%

#### RTT



#### Ventana de congestión





**UNIVERSIDAD ANDINA DEL CUSCO  
ADMINISTRACIÓN DE SERVICIOS DE REDES  
GUIA DE LABORATORIO**

ECP 13 de 16

---

## CONCLUSIONES

1. Cuanto mayor es el retardo en la red, el RTT toma valores mayores, lo que hace que la transferencia sea más lenta. En las pruebas, la transferencia tomó más de 6 minutos cuando la red tenía retardos de 10 ms en cada interfaz.
2. La ventana de congestionamiento pudo escalar hasta más de 420KB en una red sin errores ni retardos, mientras que cuando la red presentaba errores y retardos, la ventana solo pudo alcanzar un poco más de 105KB.



## **VII. TRABAJOS DE LABORATORIO.**

- 1) Muestre el comportamiento del RTO para los casos analizados en la presente guía.
- 2) Seleccione dos mecanismos de control de congestionamiento (Tahoe, Illinois, Cubic, New Reno, BIC, Highspeed, HTCP, Hybla, LP, Scalable, Veno, Westwood, Yeah, BBR, DCTCP, Vegas, NV, CDG, Sack) de los disponibles en la distribución de Linux utilizada en la asignatura y compare su comportamiento en la transferencia de archivos de 20MB y 60MB. Configure la WAN con 120 ms de retardo, 5% de pérdida y 4 de fluctuación de retardo (jitter). Para la evaluación del desempeño de TCP puede utilizar iPerf3 (<https://iperf.fr/>), NetPerfMeter (<https://www.uni-due.de/~be0001/netperfmonitor/>) u otra herramienta.

Explique los resultados de la comparación.



**UNIVERSIDAD ANDINA DEL CUSCO**  
**ADMINISTRACIÓN DE SERVICIOS DE REDES**  
**GUIA DE LABORATORIO**

ECP 15 de 16

### **VIII. CRITERIO DE EVALUACION**

La evaluación de las actividades realizadas en la presente guía de práctica se hará en función de la siguiente tabla:

ACTIVIDAD	PROCEDIMENTAL	
	SESION 01	SESION 02
Resolución del ejercicio propuesto 01	--	06
Resolución del ejercicio propuesto 02	--	14
<b>TOTAL</b>	<b>--</b>	<b>20</b>



## **IX. REFERENCIAS**

1. Comer “*Internetworking With Tcp/Ip Volumen 1*”. Ed. Prentice Hall 4° edición.
2. Comer “*Internetworking With Tcp/Ip Volumen 2*”. Ed. Prentice Hall 3° edición.
3. Hall et. al. “*Windows Sockets. An Open Interface For Network Programming Under Microsoft Windows*” Version 1.1 1993.
4. Kili A. “*How to Test Network Throughput Using iPerf3 Tool in Linux*”  
<https://www.tecmint.com/test-network-throughput-in-linux/>
5. Lopez , Novo; “*Protocolos De Internet*” Ed. AlfaOmega
6. Nambiar M. “*WANem Setup Guide*” TATA Consultancy Services 2007 (tomado de <http://ufpr.dl.sourceforge.net/project/wanem/Documents/WANemv11-Setup-Guide.pdf>, fecha de visita: 26/10/2016).
7. Tanenbaum A., Wetherall D. “*Computer Networks*”. Prentice Hall 5° edición 2011.
8. Valentine. “*Testing Network Speed with iPerf3*”  
<https://community.time4vps.com/discussion/175/testing-network-speed-with-iperf3>