TALLER Nº7: VERIFICAR LA CAPACIDAD DEL CANAL EN UNA RED (RED ÓPTICA) DE COMUNICACIONES.

REDES DE COMUNICACIONES ÓPTICAS

Castillo Carpio Jorge Eduardo
Juela Banshuy Danny Alexander
Escuela Politécnica Nacional - Ingeniería en Telecomunicaciones
jorge.castillo@epn.edu.ec, danny.juela@epn.edu.ec

Abstract. - As Telecommunications engineering students, it is necessary to understand how a real environment develops in terms of the deployment of conventional or optical networks, for this reason this document shows the necessary needs to simulate a network deployment using all the connection tests necessary to verify its operation and in the same way the use of tools for the generation of traffic and as well as monitors.

Keywords: *iperf, jperg, PRTG, Wireshark, TCP/UDP, SNMP.*

I. INDICACIONES

- *En este taller usted verificará la capacidad (física/lógica) de una red de comunicaciones de a través de la generación y censado de tráfico.*
- *Como primer punto usted debe desplegar una red de comunicaciones entre dos puntos (hosts). El primero presentara el proveedor y el segundo el cliente.*

II. INTRODUCCION

Los sistemas de monitoreo de red incluyen herramientas de software y hardware que pueden hacer un seguimiento de diversos aspectos de la red y su funcionamiento, como el tráfico, el uso de ancho de banda y el tiempo de actividad. Estos sistemas pueden detectar dispositivos y otros elementos que componen o tocan la red, además de proporcionar actualizaciones de estado.[1]

Los administradores de red confían en los sistemas de monitoreo de red para detectar rápidamente las fallas de dispositivos o conexiones, o los problemas como los cuellos de botella de tráfico que limitan el flujo de datos. Estos sistemas pueden alertar a los administradores de los problemas por correo electrónico o mensaje de texto, y enviar informes mediante la analítica de red.[1]

III. DESARROLLO

1. Despliegue en GNS3

La topología propuesta es de tipo anillo, la cual se dispone de tres routers a manera de una red LAN, dentro de la cual el primero nodo involucra un servidor matriz y un administrador que será el encargado de la monitorización de la red, además el punto en el que se verificarán las conexiones entre servidores, equipos y clientes, mientras que en las otras dos estaciones o nodos se tienen dos clientes conectados directamente a la red.

1.1. Características del Software y Hardware

- ***Software de emulación***

GNS3 es un simulador gráfico de red que permite diseñar topologías de red complejas y poner en marcha simulaciones sobre ellos. Con GNS3 los usuarios tendrán la posibilidad de poder escoger cada uno de los elementos que llegarán a formar parte de una red informática. [2]

- ***Sistema operativo***

Ubuntu es un sistema operativo de código abierto basado en Linux. El encargado de su desarrollo es Canonical junto con una comunidad de desarrolladores —con un modelo de gestión meritocrático—. Esta distribución de Linux, basada en Debian, es muy popular en proyectos de cloud computing. Ubuntu se lanzó en octubre de 2004. Cada 6 meses se publica una nueva versión y cada 2 años se publica una versión con soporte técnico extendido (LTS). [3]

- ***Software de monitoreo***

PRTG Enterprise Monitor se ha diseñado para ofrecer a los operadores de grandes infraestructuras informáticas una única solución basada en los servicios de TI a un nivel superior. Esta solución es fácil de usar, permite supervisarlo todo y ofrece una amplia comprensión del entorno que puede generar gran cantidad de información y datos detallados para ciertos aspectos fundamentales del mismo, como el almacenamiento y los servidores.[4]

Wireshark es el analizador de paquetes más conocido y utilizado en todo el mundo. Gracias a este programa, podremos capturar y analizar en detalle todo el tráfico de red que entra y sale de nuestro PC, además, debemos recordar que es multiplataforma, esto significa que está disponible para sistemas operativos Windows, Linux, macOS, Solaris, FreeBSD, NetBSD y otros. Hoy en RedesZone os vamos a enseñar de manera básica, como realizar una captura de tráfico,

y cómo analizar el tráfico de red para ver si hay algún tipo de anomalía. [5]

- ***Routers***

Los Router **Cisco 3700** permiten niveles dramáticamente más altos de integración de la aplicación y del servicio en las sucursales de la empresa. Con conectividad LAN/WAN a bordo, nuevos módulos de servicio de alta densidad y soporte para múltiples módulos de integración avanzada (AIMs), los Cisco 3700 Routers ofrecen densidad de servicio de sucursal en un factor de forma compacto.[6]

1.2. Software de Generación de Trafico

IPerf es un software de medición de la capacidad de tráfico de una dentro de un mismo segmento de red LAN. Esta prueba puede ayudar a realizar el diagnostico de una red, verificando si esta se encuentra con problemas de congestión, degradación o afectación física de alguno de los elementos de red. IPerf es un software que funciona a través de línea de comandas, existe una versión de IPerf con un entorno gráfico que es JPerf. [7]

1.3. Direccionamiento

Para los servidores se ha usado direcciones IP de clase C, esto ya que son ideales para redes pequeñas, mientras que para los routers se ha usado redes de clase B, ya que las conexiones seriales entre los routers se consideran redes un poco más grandes.

Tabla 1. Direccionamiento de la topología a implementar

Disp.	Int	Dirección IP	Máscara de subred	Default Gateway
Administrador	E0	192.168.122.52	255.255.255.0	192.168.122.41
Administrador	E1	192.168.122.84	255.255.255.0	192.168.122.1
Server Matriz	E0	192.168.122.52	255.255.255.0	192.168.122.41
Cliente A	E0	128.16.32.5	255.255.248.0	128.16.32.1
Cliente B	E0	128.16.40.5	255.255.254.0	128.16.40.1
Router_M	F0/0	192.168.122.41	255.255.255.0	-
Router_M	S0/1	128.16.43.1	255.255.255.252	-
Router_M	S0/0	128.16.43.5	255.255.255.252	-
Router_CA	F0/0	128.16.32.1	255.255.248.0	-
Router_CA	S0/1	128.16.43.9	255.255.255.252	-
Router_CA	S0/0	128.16.43.2	255.255.255.252	-
Router_CB	F0/0	128.16.40.1	255.255.254.0	-
Router_CB	S0/1	128.16.43.10	255.255.255.252	-
Router_CB	S0/0	128.16.43.6	255.255.255.252	-

1.4. Topología

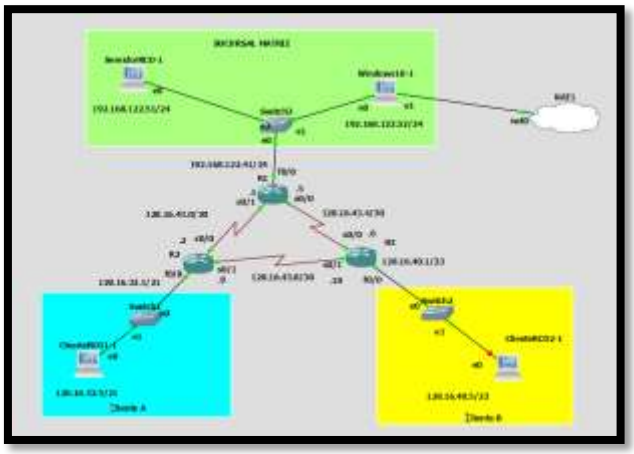


Fig. 1. Topología Implementada.

Como se puede observar en la imagen superior, el desarrollo de la topología muestra la sucursal matriz (verde) junto al equipo administrador, el cual se encuentra instalado en una máquina virtual con sistema operativo Windows 10, el cual ha sido escogido debido a que presenta mayores ventajas como entorno para el software de monitorización PRTG. Es de mencionar que la NAT conectada a un adaptador de red en el administrador se ha usado únicamente para instalar los servicios requeridos como el mismo PRTG, Wireshark o demás complementos a usar. En la parte inferior se muestran los clientes conectados a la red.

1.5. Generación de tráfico con IPERF

Para la realización de las pruebas de generación de tráfico se han usado las interfaces graficas de JPERF a fin de emplear Iperf y generar tráfico desde la sucursal matriz hacia los clientes A y B. De manera similar en las maquinas cliente se ha instalado Wireshark a fin de analizar los paquetes enviados y recibidos.

1.5.1. Generación de Trafico UDP

Como se muestra a continuación, los primeros paquetes a enviar son UDP, por tal motivo se ha configurado la interfaz gráfica de JPERF como servidor en la maquina sucursal matriz, mientras que en la maquina cliente A se ha configurado JPERF como cliente.

Para el primer escenario se han empleado 20 canales paralelos a fin de saturar la capacidad del canal, obteniendo los siguientes resultados. Adicionalmente se adjuntan capturas de la configuración y graficas del tráfico generado y capturado.

Tabla 2. Escenario 1.

Tráfico UDP	
Capacidad Total	137 Kbytes/s
Capacidad Individual	7 Kbytes/s
Jitter Individual	6,46 ms
Canales Paralelos	20
Paquetes	17207
Tamaño	1470 Bytes

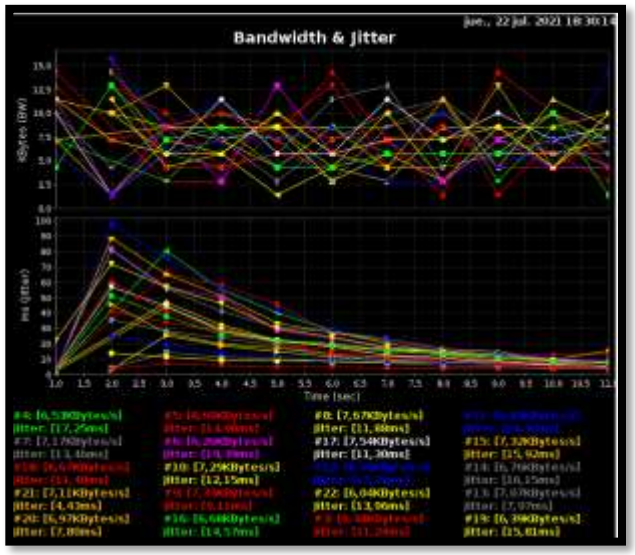


Fig. 2. Servidor Tráfico UDP Kbytes.

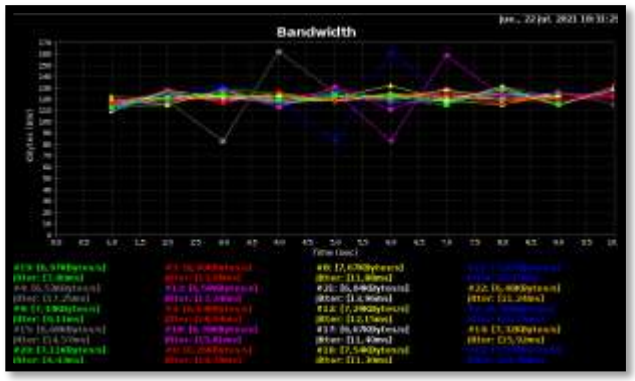
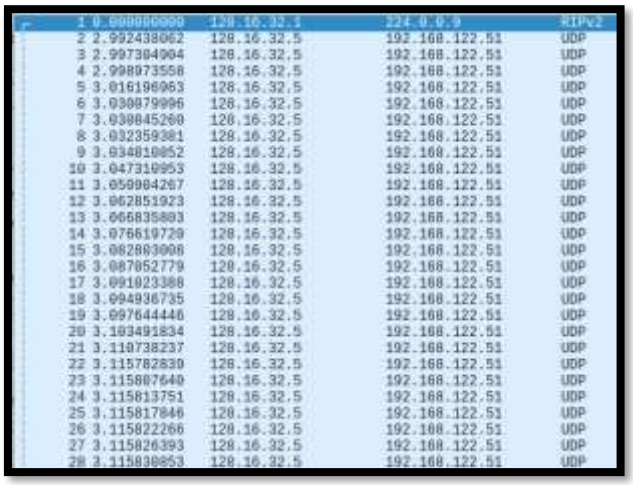


Fig. 3. Cliente Tráfico UDP Parallel=20 Kbytes.



1	0.000000000	128.16.32.1	224.0.0.9	RIPv2
2	2.992438062	128.16.32.5	192.168.122.51	UDP
3	2.997304904	128.16.32.5	192.168.122.51	UDP
4	2.998973558	128.16.32.5	192.168.122.51	UDP
5	3.016196963	128.16.32.5	192.168.122.51	UDP
6	3.030879996	128.16.32.5	192.168.122.51	UDP
7	3.030845260	128.16.32.5	192.168.122.51	UDP
8	3.032359381	128.16.32.5	192.168.122.51	UDP
9	3.034810852	128.16.32.5	192.168.122.51	UDP
10	3.047310953	128.16.32.5	192.168.122.51	UDP
11	3.050904267	128.16.32.5	192.168.122.51	UDP
12	3.062851923	128.16.32.5	192.168.122.51	UDP
13	3.066835803	128.16.32.5	192.168.122.51	UDP
14	3.076619720	128.16.32.5	192.168.122.51	UDP
15	3.082803008	128.16.32.5	192.168.122.51	UDP
16	3.087852779	128.16.32.5	192.168.122.51	UDP
17	3.091023388	128.16.32.5	192.168.122.51	UDP
18	3.094936735	128.16.32.5	192.168.122.51	UDP
19	3.097644446	128.16.32.5	192.168.122.51	UDP
20	3.103491834	128.16.32.5	192.168.122.51	UDP
21	3.110738237	128.16.32.5	192.168.122.51	UDP
22	3.115782839	128.16.32.5	192.168.122.51	UDP
23	3.115807640	128.16.32.5	192.168.122.51	UDP
24	3.115813751	128.16.32.5	192.168.122.51	UDP
25	3.115817846	128.16.32.5	192.168.122.51	UDP
26	3.115822266	128.16.32.5	192.168.122.51	UDP
27	3.115826393	128.16.32.5	192.168.122.51	UDP
28	3.115830853	128.16.32.5	192.168.122.51	UDP

Fig. 4. Paquetes UDP generados.

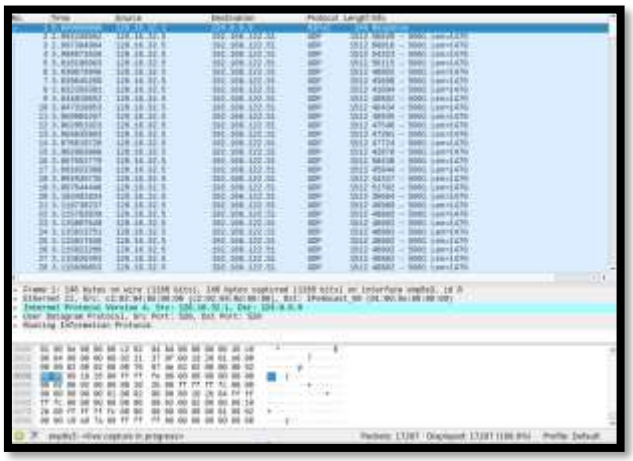


Fig. 5. Captura Wireshark

Para el escenario 2, se ha empleado un solo canal a fin de comprobar la capacidad máxima del enlace. De la misma forma se emplean JPERF en configuración servidor-cliente y se adjuntan los resultados tras emplear las herramientas de generación de tráfico y monitorización.

Tabla 3. Escenario 2.

Tráfico UDP	
Capacidad Total	122 Kbytes/s
Jitter Individual	3,3 ms
Paquetes	858
Tamaño	1470 Bytes

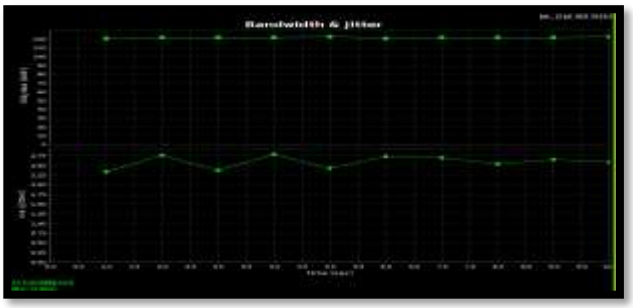


Fig. 6. Servidor Tráfico UDP Kbytes.

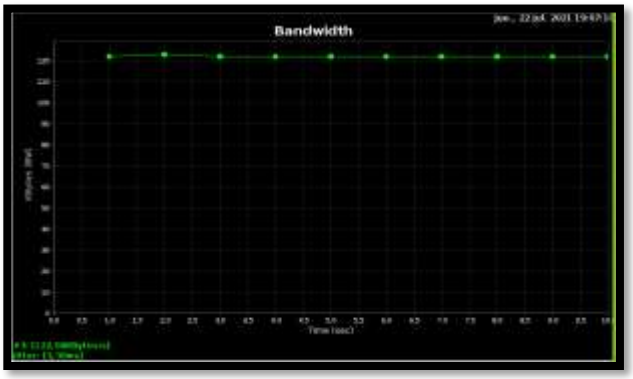
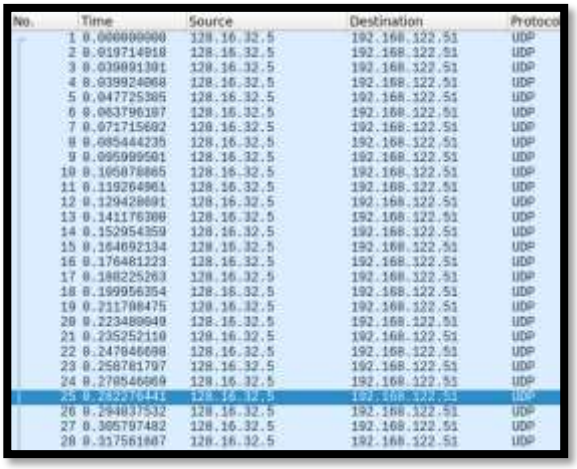


Fig. 7. Cliente Tráfico UDP Kbytes.



No.	Time	Source	Destination	Protocol
1	0.000000000	128.16.32.5	192.168.122.51	UDP
2	0.019714818	128.16.32.5	192.168.122.51	UDP
3	0.039091301	128.16.32.5	192.168.122.51	UDP
4	0.039924068	128.16.32.5	192.168.122.51	UDP
5	0.047725385	128.16.32.5	192.168.122.51	UDP
6	0.063796187	128.16.32.5	192.168.122.51	UDP
7	0.071715682	128.16.32.5	192.168.122.51	UDP
8	0.085444235	128.16.32.5	192.168.122.51	UDP
9	0.095999501	128.16.32.5	192.168.122.51	UDP
10	0.106078865	128.16.32.5	192.168.122.51	UDP
11	0.119264961	128.16.32.5	192.168.122.51	UDP
12	0.129428691	128.16.32.5	192.168.122.51	UDP
13	0.141176308	128.16.32.5	192.168.122.51	UDP
14	0.152954359	128.16.32.5	192.168.122.51	UDP
15	0.164692134	128.16.32.5	192.168.122.51	UDP
16	0.176481223	128.16.32.5	192.168.122.51	UDP
17	0.188225263	128.16.32.5	192.168.122.51	UDP
18	0.199956354	128.16.32.5	192.168.122.51	UDP
19	0.211708475	128.16.32.5	192.168.122.51	UDP
20	0.223480640	128.16.32.5	192.168.122.51	UDP
21	0.235252110	128.16.32.5	192.168.122.51	UDP
22	0.247046088	128.16.32.5	192.168.122.51	UDP
23	0.258781797	128.16.32.5	192.168.122.51	UDP
24	0.270546069	128.16.32.5	192.168.122.51	UDP
25	0.282270441	128.16.32.5	192.168.122.51	UDP
26	0.294037532	128.16.32.5	192.168.122.51	UDP
27	0.305797482	128.16.32.5	192.168.122.51	UDP
28	0.317561607	128.16.32.5	192.168.122.51	UDP

Fig. 8. Paquetes UDP generados.

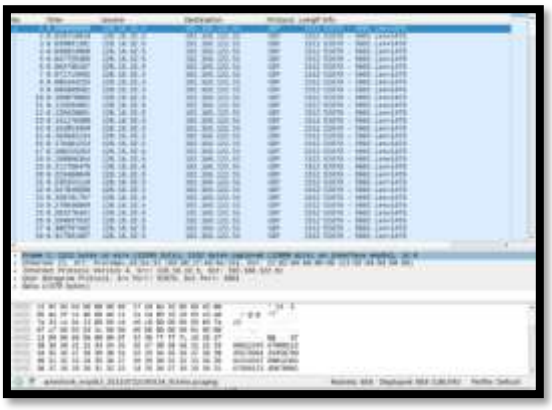


Fig. 9. Captura Wireshark.

1.5.2. Generación de Trafico TCP

Como se muestra a continuación, los paquetes generados y enviados corresponden a TCP, por tal motivo se ha configurado la interfaz gráfica de JPERF como servidor en la maquina sucursal matriz, mientras que en la maquina cliente A se ha configurado JPERF como cliente.

Para el tercer escenario se han empleado 20 canales paralelos a fin de saturar la capacidad del canal, obteniendo los siguientes resultados. Adicionalmente se adjuntan capturas de la configuración y graficas del tráfico generado y capturado.

Tabla 3. Escenario 3.

Tráfico TCP	
Capacidad Total	132 Kbytes
Capacidad Individual	5,21 Kbytes/s
Jitter Individual	0 ms
Canales Paralelos	20
Paquetes	4429
Tamaño Promedio	925 B

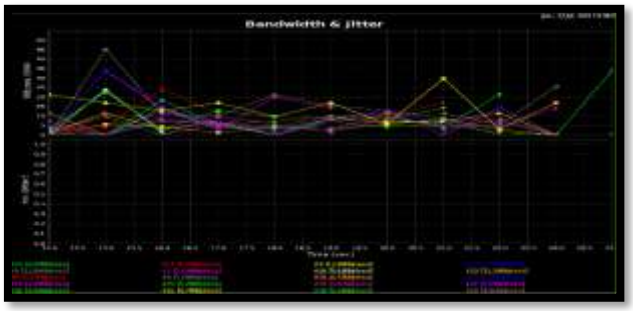


Fig. 10. Servidor Tráfico TCP Kbytes

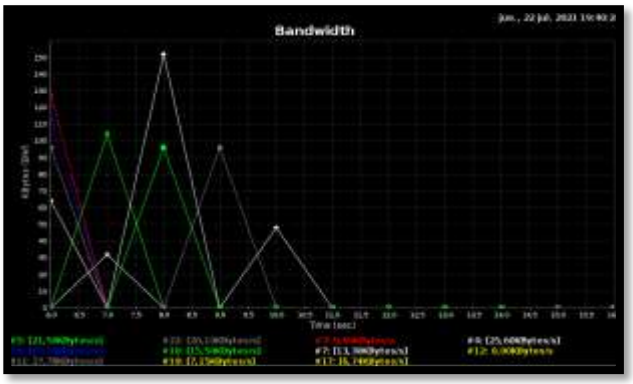


Fig. 11. Cliente Tráfico TCP Parallel=20 Kbytes.

No.	Time	Source	Destination	Protoc
4402	24.140908115	192.168.122.51	128.16.32.5	TCP
4403	24.181478009	192.168.122.51	128.16.32.5	TCP
4404	24.181505389	128.16.32.5	192.168.122.51	TCP
4405	24.181745277	192.168.122.51	128.16.32.5	TCP
4406	24.191765888	128.16.32.5	192.168.122.51	TCP
4407	24.202541851	192.168.122.51	128.16.32.5	TCP
4408	24.202576515	128.16.32.5	192.168.122.51	TCP
4409	24.212633383	192.168.122.51	128.16.32.5	TCP
4410	24.212663839	128.16.32.5	192.168.122.51	TCP
4411	24.212680728	128.16.32.5	192.168.122.51	TCP
4412	24.222816532	192.168.122.51	128.16.32.5	TCP
4413	24.232962921	192.168.122.51	128.16.32.5	TCP
4414	24.243423231	192.168.122.51	128.16.32.5	TCP
4415	24.253566294	192.168.122.51	128.16.32.5	TCP
4416	24.263654949	192.168.122.51	128.16.32.5	TCP
4417	24.284308090	192.168.122.51	128.16.32.5	TCP
4418	24.294283430	128.16.32.5	192.168.122.51	TCP
4419	24.294435738	192.168.122.51	128.16.32.5	TCP
4420	24.305018385	192.168.122.51	128.16.32.5	TCP
4421	24.314970277	192.168.122.51	128.16.32.5	TCP
4422	24.325406584	192.168.122.51	128.16.32.5	TCP
4423	24.335563703	192.168.122.51	128.16.32.5	TCP
4424	24.345098546	128.16.32.5	192.168.122.51	TCP
4425	24.355868201	192.168.122.51	128.16.32.5	TCP
4426	24.355992943	128.16.32.5	192.168.122.51	TCP
4427	24.366696655	192.168.122.51	128.16.32.5	TCP
4428	24.377129541	192.168.122.51	128.16.32.5	TCP
4429	24.377157591	128.16.32.5	192.168.122.51	TCP

Fig. 12. Paquetes TCP generados.

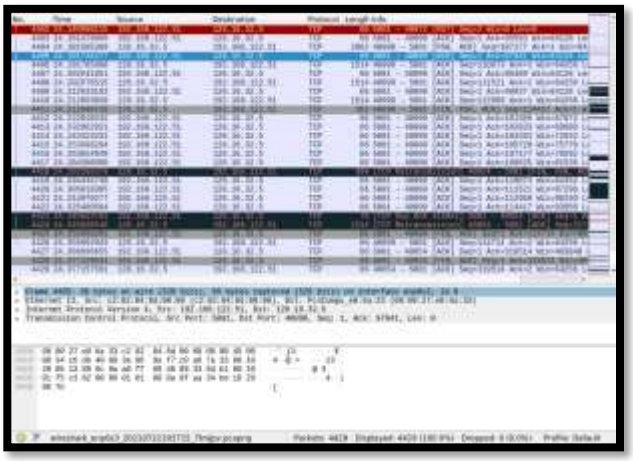


Fig. 13. Captura Wireshark

Statistics

Measurement	Captured	Displayed	Marked
Packets	4429	4429 (100.0%)	—
Time span, s	24.377	24.377	—
Average pps	181.7	181.7	—
Average packet size, B	925	925	—
Bytes	4096384	4096384 (100.0%)	0
Average bytes/s	168 k	168 k	—
Average bits/s	1.344 k	1.344 k	—

Fig. 14. Estadísticas en Wireshark.

Para el escenario 4, se ha empleado un solo canal a fin de comprobar la capacidad máxima del enlace. De la misma forma se emplean JPERF en configuración servidor-cliente y se adjuntan los resultados tras emplear las herramientas de generación de tráfico y monitorización en la transmisión de paquetes TCP.

Tabla 4. Escenario 4

Tráfico TCP	
Capacidad Total	138 Kbytes/s
Jitter Individual	0 ms
Paquetes	2259
Tamaño Promedio	1017 B

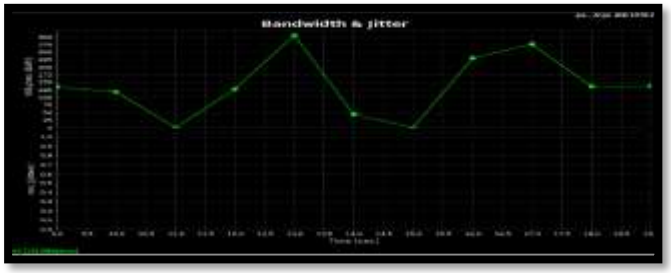


Fig. 15. Servidor Tráfico TCP Kbytes.

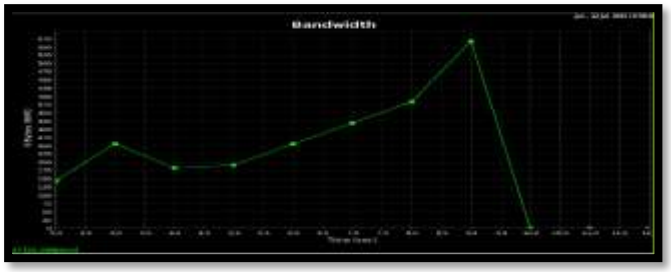


Fig. 16. Cliente Tráfico TCP Kbytes.

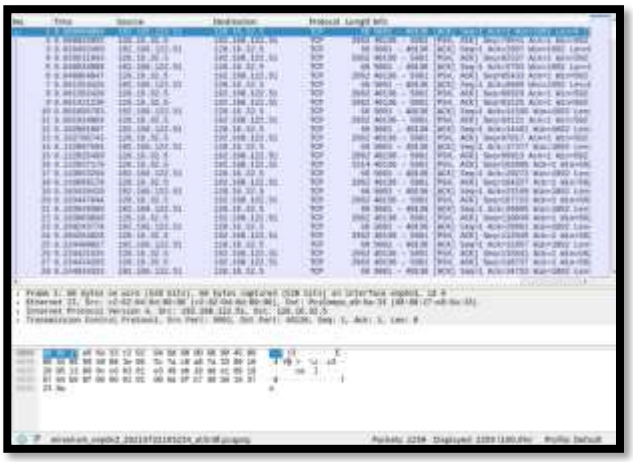


Fig. 17. Captura Wireshark.

No.	Time	Source	Destination	Protocol
1	0.000000000	192.168.122.51	128.16.32.5	TCP
2	0.000025955	128.16.32.5	192.168.122.51	TCP
3	0.020492469	192.168.122.51	128.16.32.5	TCP
4	0.020511943	128.16.32.5	192.168.122.51	TCP
5	0.040934988	192.168.122.51	128.16.32.5	TCP
6	0.040964647	128.16.32.5	192.168.122.51	TCP
7	0.061353426	192.168.122.51	128.16.32.5	TCP
8	0.061382420	128.16.32.5	192.168.122.51	TCP
9	0.061421230	128.16.32.5	192.168.122.51	TCP
10	0.081895703	192.168.122.51	128.16.32.5	TCP
11	0.081914884	128.16.32.5	192.168.122.51	TCP
12	0.102681697	192.168.122.51	128.16.32.5	TCP
13	0.102706743	128.16.32.5	192.168.122.51	TCP
14	0.122897504	192.168.122.51	128.16.32.5	TCP
15	0.122825499	128.16.32.5	192.168.122.51	TCP
16	0.122837179	128.16.32.5	192.168.122.51	TCP
17	0.143653250	192.168.122.51	128.16.32.5	TCP
18	0.143686178	128.16.32.5	192.168.122.51	TCP
19	0.163428420	192.168.122.51	128.16.32.5	TCP
20	0.163447644	128.16.32.5	192.168.122.51	TCP
21	0.183949386	192.168.122.51	128.16.32.5	TCP
22	0.183983699	128.16.32.5	192.168.122.51	TCP
23	0.204243774	192.168.122.51	128.16.32.5	TCP
24	0.204264828	128.16.32.5	192.168.122.51	TCP
25	0.224400827	192.168.122.51	128.16.32.5	TCP
26	0.224421535	128.16.32.5	192.168.122.51	TCP
27	0.224434203	128.16.32.5	192.168.122.51	TCP
28	0.244934333	192.168.122.51	128.16.32.5	TCP

Fig. 18. Paquetes TCP.

Statistics

Measurement	Captured	Displayed	Marked
Packets	2259	2259 (100.0%)	—
Time span, s	13.555	13.555	—
Average pps	166.7	166.7	—
Average packet size, B	1017	1017	—
Bytes	2298200	2298200 (100.0%)	0
Average bytes/s	169 k	169 k	—
Average bits/s	1.356 k	1.356 k	—

Fig. 19. Estadísticas en Wireshark.

1.6. Generación de tráfico TCP/UDP.

Con respecto a la monitorización de la red, se emplean PRTG a fin de observar el tráfico que circula la red, para ello se modifican los sensores dentro del software indicando los equipos a monitorizar, como se muestran a continuación.

Fig. 20. Sensores en R1.

Fig. 21. Sensores en R2.

Fig. 22. Sensores en R3.

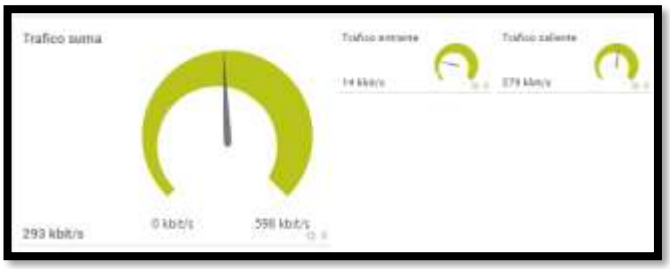


Fig. 23. Tráfico suma en PRTG.

Fig. 24. Parámetros monitorizados.

Fig. 25. Descripción de monitorización.

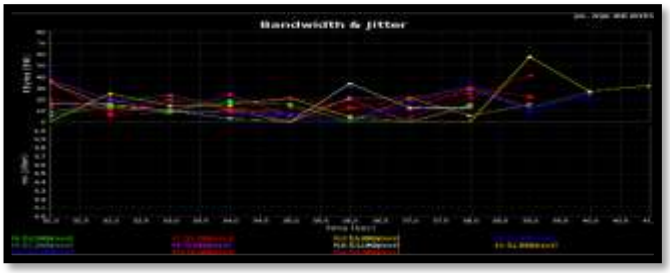


Fig. 26. Tráfico TCP-Servidor.

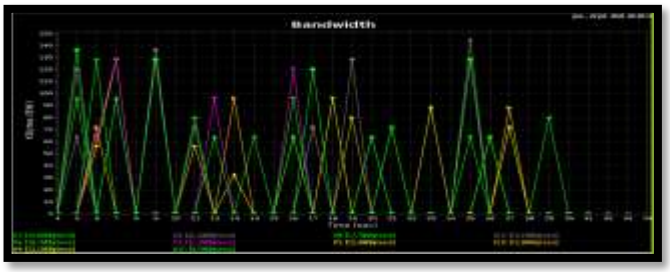


Fig. 27. Tráfico TCP-Cliente.

Una vez realizada la monitorización de los paquetes TCP se emplea el mismo procedimiento para paquetes UDP.

Fig. 28. Tráfico suma en PRTG.

Fig. 29. Parámetros monitorizados.

Fig. 30. Descripción de monitorización.

Monitorización en Router 1:

Fig. 31. Interfaz Fast 0/0.

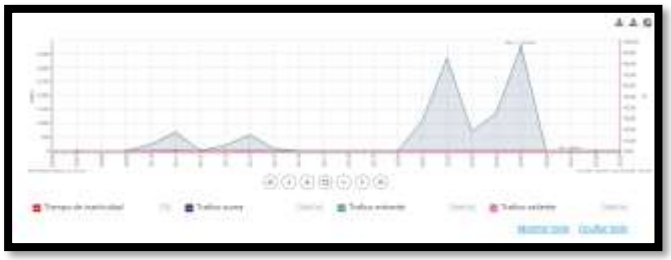


Fig. 32. Interfaz Serial 0/1

Monitorización en Router 2:

Fig. 33. Interfaz Fast 0/0.

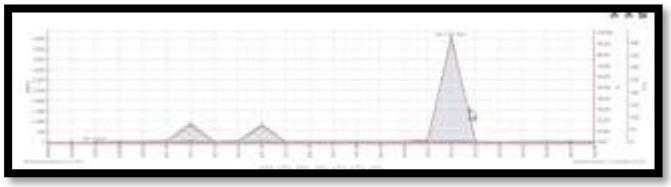


Fig. 34. Interfaz Serial 0/0

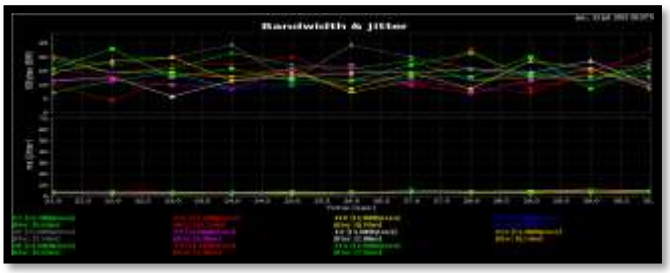


Fig. 35. Tráfico UDP-Servidor.

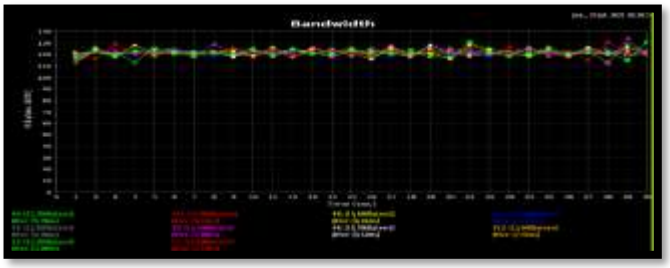


Fig. 36. Tráfico UDP-Cliente.

2. Monitorización con IPV6

En primer lugar, se verifican las direcciones IPv6 en cada computador mediante el comando ipconfig, como se muestra a continuación.

Fig. 37. Verificación de dirección IPv6-Servidor.

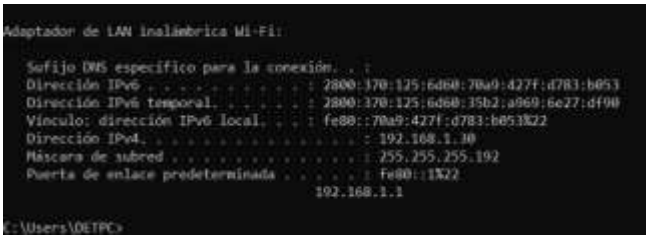


Fig. 38. Verificación de dirección IPv6-Cliente.

Después de la verificación de las direcciones IPv6 se procede a realizar las pruebas de conectividad empleando ping, es necesario recordar que para habilitar la conexión es necesario desactivar el firewall.

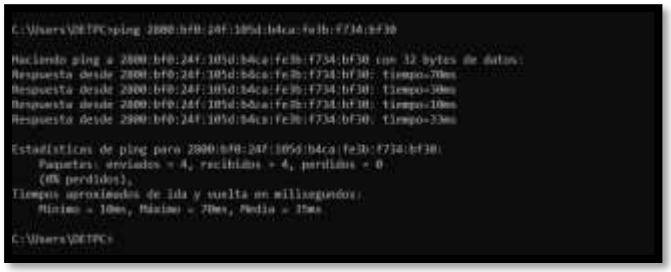


Fig. 39. Ping hacia el servidor.

Fig. 40. Ping hacia el cliente.

Fig. 41. Desactivación del firewall.

Fig. 42. Prueba de capacidad real en Speedtest.

En este momento se procede a realizar la generación de trafico mediante JPERF, los paquetes enviados serán TCP y estos serán monitorizados mediante Wireshark en el lado del servidor.

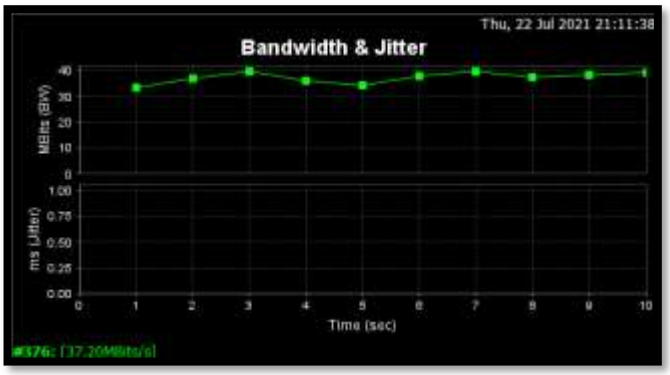


Fig. 43. Tráfico TCP-Servidor.

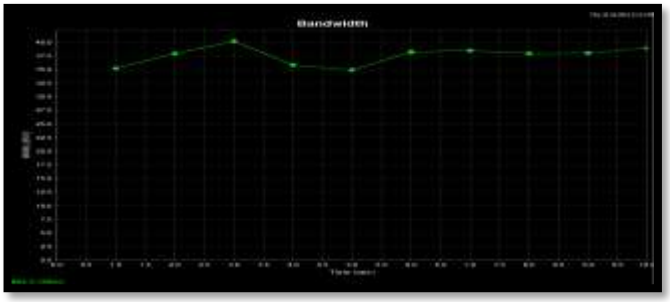


Fig. 44. Tráfico TCP-Cliente.

Fig. 45. Paquetes TCP capturados en Wireshark.

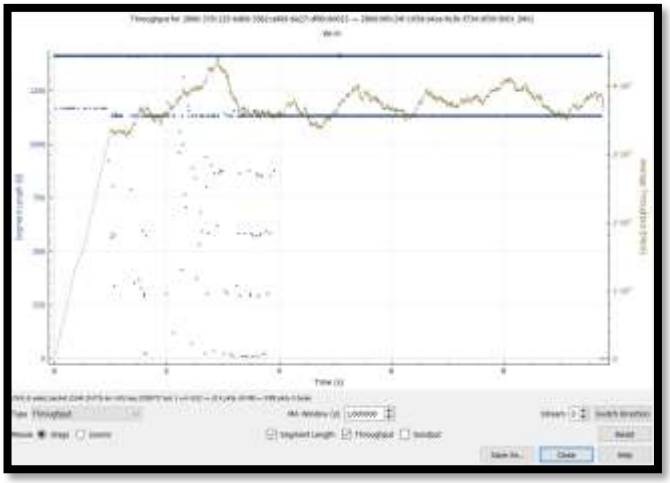


Fig. 46. Troughput obtenido.

Fig. 47. Estadísticas en Wireshark.

En el siguiente ejemplo se muestran las gráficas de generación y captura de tráfico cuando el enlace presenta una baja capacidad, es decir en horas pico.

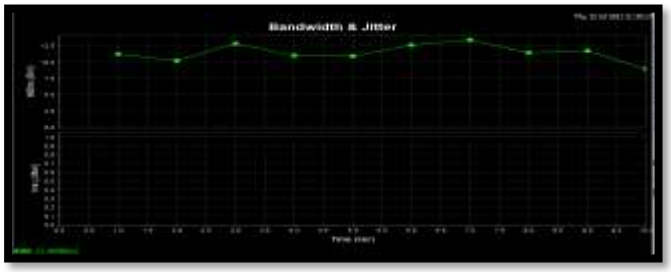


Fig. 48. Tráfico TCP-Servidor.

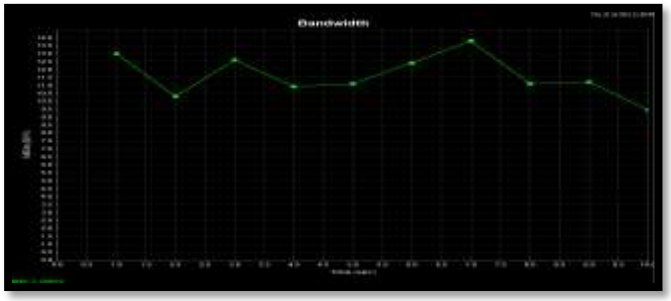


Fig. 49. Tráfico TCP-Cliente.

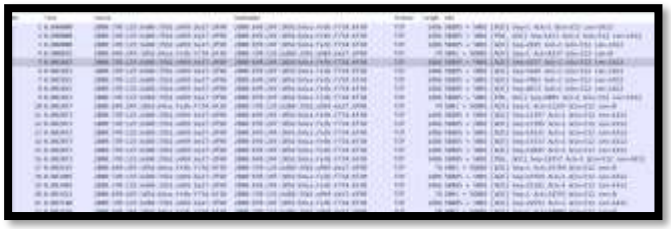


Fig. 50. Paquetes TCP capturados en Wireshark.

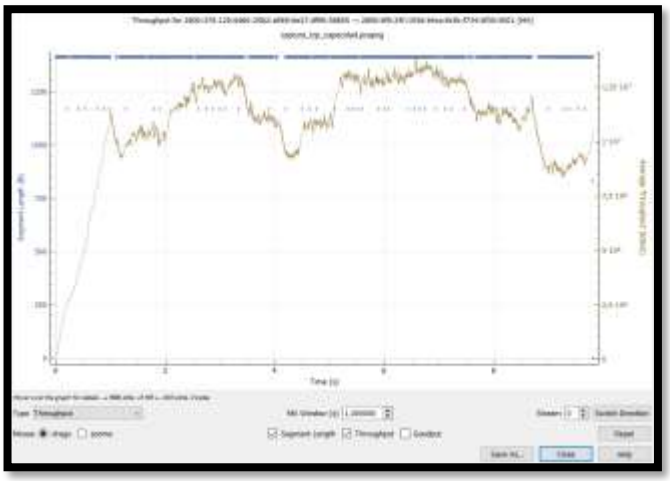


Fig. 51. Troughput obtenido.

Fig. 52. Estadísticas en Wireshark

De la misma forma se comprueban los valores reales en cuanto a capacidad del canal, empleando la herramienta Speedtest.

Fig. 53. Prueba de capacidad real en Speedtest.

Intercambio Cliente Servidor

Para este ejemplo intercambiamos mediante jperf la dirección IPv6 del servidor con la del cliente y generamos tráfico TCP se pudo visualizar que la capacidad máxima es de 42,30 Mbits/s sin embargo al realizar las pruebas de velocidad de speedtest se obtuvo 59,61 Mbps.

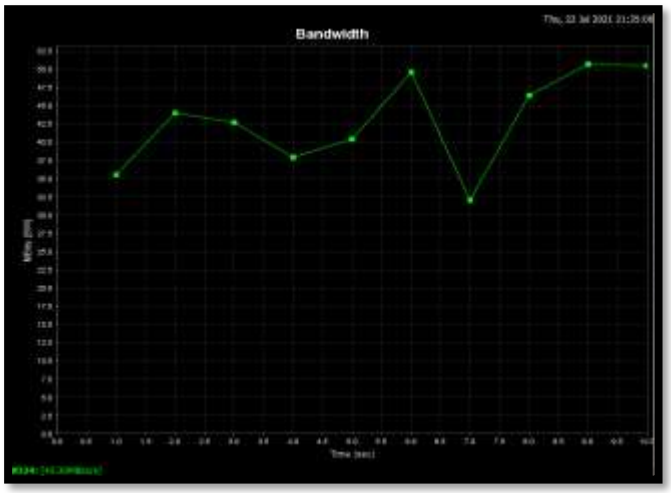


Fig. 54. Tráfico Saliente desde el cliente.

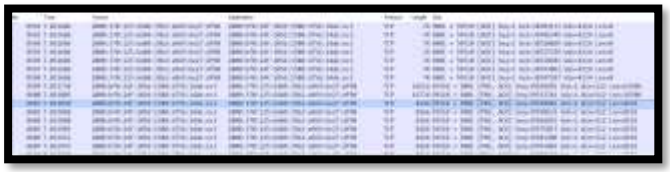


Fig. 55. Paquetes TCP salientes.

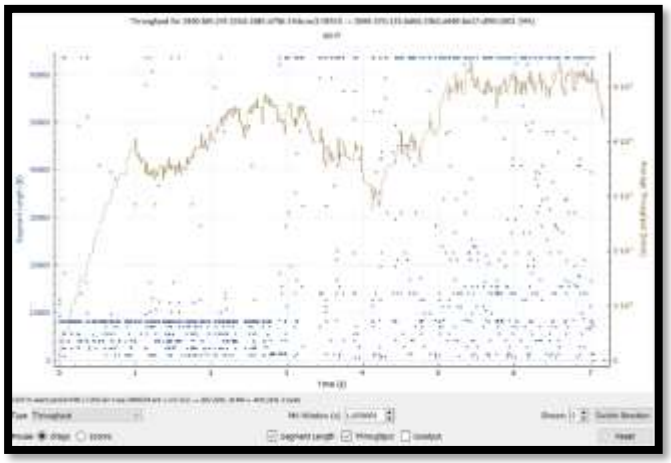


Fig. 56. Troughput de los paquetes TCP.

Fig. 57. Prueba de capacidad real en Speedtest.

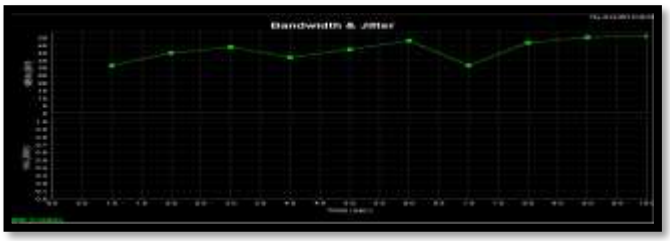


Fig. 58. Tráfico Entrante al servidor-Danny.

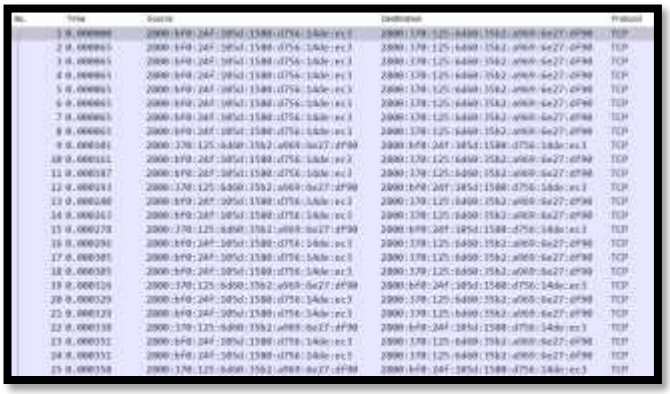


Fig. 59. Paquetes TCP entrantes al servidor.

Tráfico UDP

Para este ejemplo se generó tráfico UDP desde el cliente y se verifico con Wireshark que los paquetes sean enviados sin embargo en el servidor no se recibió los paquetes y en él, la herramienta de JPERF se observó un mensaje de conexión rechazada.

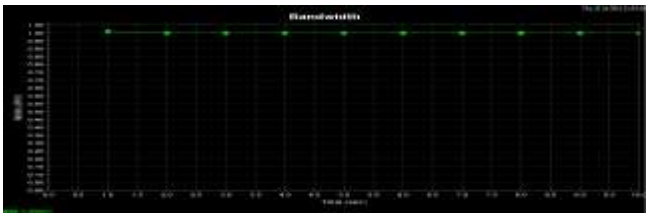


Fig. 60. Tráfico UDP saliente del Cliente

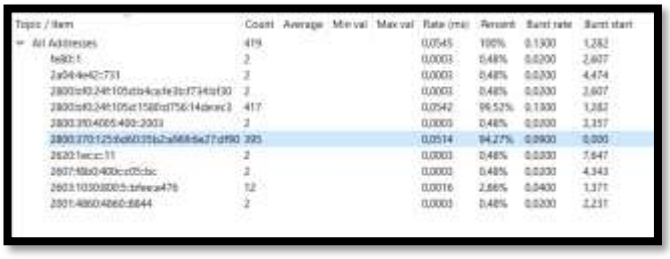


Fig. 61. Paquetes Enviados hacia el servidor

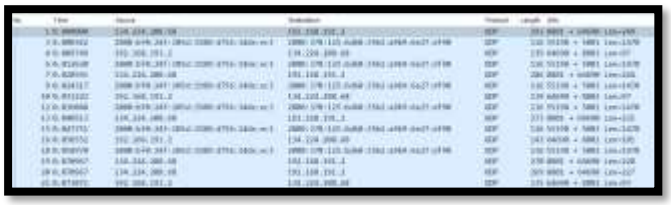


Fig. 62. Paquetes Enviados análisis de wireshark

Fig. 63. Conexión rechazada en el servidor

3. Generación de tráfico mediante consola

3.1. Generación de tráfico con SIPp:

SIPp es una herramienta de prueba/generador de tráfico de código abierto gratuito para el protocolo SIP. Incluye algunos escenarios básicos de agentes de usuario de SipStone (UAC y UAS) y establece y libera múltiples llamadas con los métodos INVITE y BYE. También puede leer archivos de escenarios XML personalizados que describen desde flujos de llamadas muy simples hasta complejos. Presenta la visualización dinámica de estadísticas sobre pruebas en ejecución (tasa de llamadas, demora de ida y vuelta y estadísticas de mensajes), volcados de estadísticas CSV periódicas, TCP y UDP en múltiples sockets o multiplexados con gestión de retransmisión y tasas de llamadas ajustables dinámicamente. Otras características avanzadas incluyen soporte de IPv6, TLS, SCTP, autenticación SIP, escenarios condicionales, retransmisiones UDP, robustez de errores (tiempo de espera de llamada, defensa de protocolo), variable específica de llamada, expresión regular Posix para extraer y reinyectar cualquier campo de protocolo, acciones personalizadas (registro, ejecución de comando del sistema, parada de llamada) al recibir el mensaje, inyección de campo desde un archivo CSV externo para emular a los usuarios en vivo.

Al ejecutar el comando ***sipp -v*** nos permite observar la versión instalada en nuestro equipo.

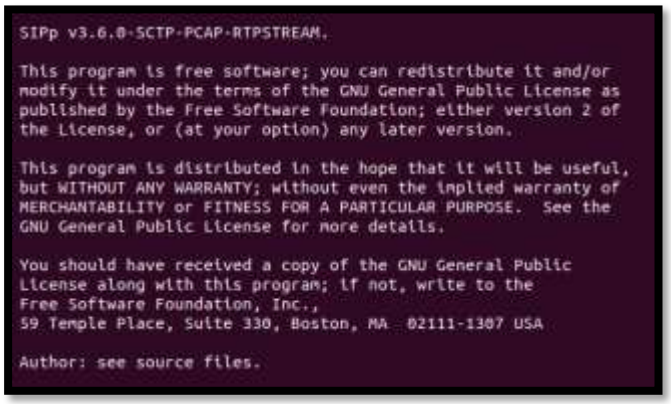


Fig. 64. Versión Sipp

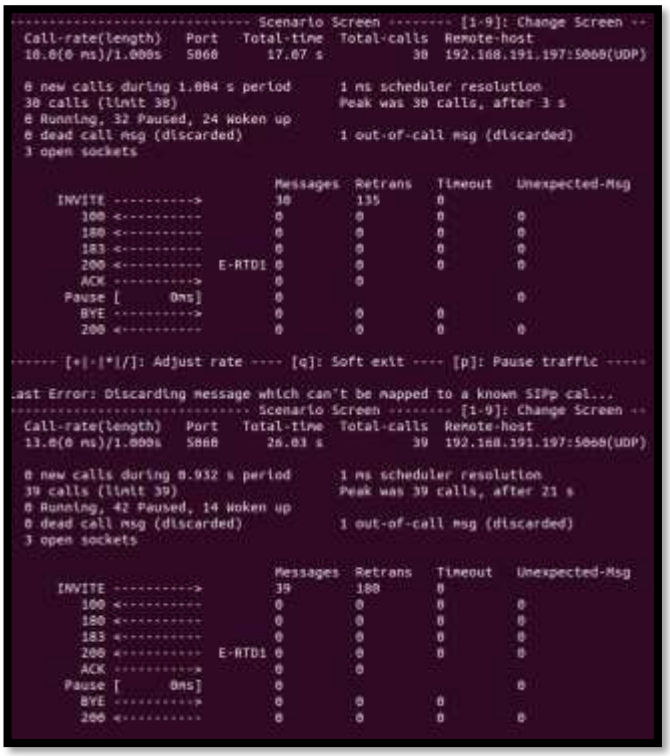


Fig. 65. Interfaz de sipp.

Por medio de los comandos ***sipp -sn uac 192.168.191.197*** se realizó la prueba del envío de paquetes sipp.

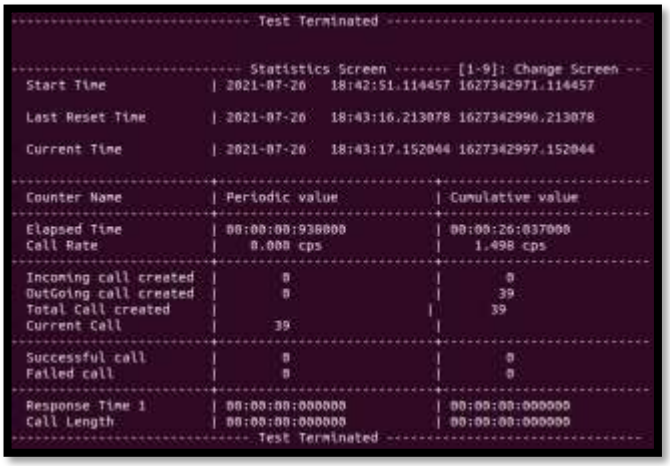


Fig. 66. Test terminado

En el resultado de la prueba final se puede evidenciar que no existen llamadas fallidas, pero si existen llamadas realizadas para verificar su funcionamiento se colocó el sensor en PRTG como se observa en la figura 68, pero no nos permite capturar los ping sipp aunque en nuestra máquina virtual aparezca el PRTG monitor como se observa en la figura 67.

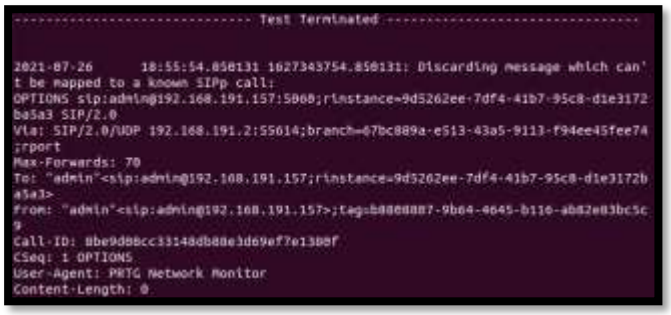


Fig. 67. Test terminado

Aquí se realizo las pruebas con la Ip del host para poder agregar el sensor de sipp en PRTG sin embargo podemos ver que aparece el User-Agent como PRTG Monitor, pero debido a que es una máquina virtual no hay respuesta es posible que sea ya que no posee una tarjeta de red como tal, sino que es virtual.

Fig. 68. Sensor en PRTG Monitor

Mediante Wireshark en la maquina cliente se observa el tráfico capturado mediante la red, los mensajes recibidos corresponden al dominio de service@192.168.191.197.

Fig. 69. Captura de paquetes sip mediante Wireshark.

3.2. Generación de tráfico con iperf en distribuciones de Linux:

Empleando una maquina servidor en Ubuntu, con la dirección IP asociada de 192.168.191.157 se procede a generar trafico desde una maquina cliente Windows con la interfaz gráfica de iperf, Jperf. De esta forma se observa en la siguiente figura la generación de paquetes TCP, los cuales se aprecian en el servidor mediante líneas de comandos.

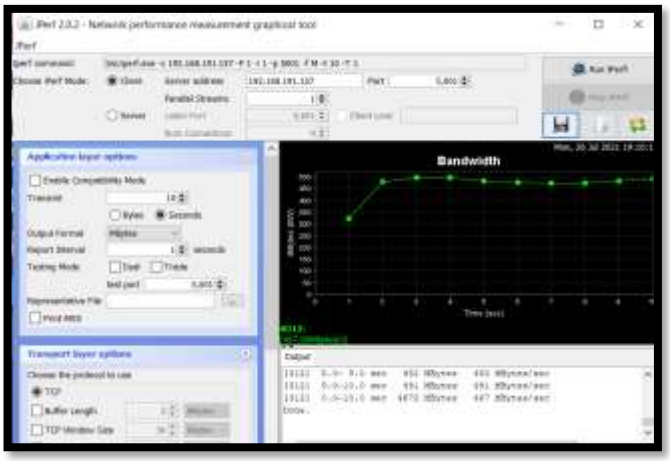


Fig. 70. Generación de paquetes TCP cliente.

```
root@usuario1-VirtualBox:/home/usuario1# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 192.168.191.157 port 5001 connected with 192.168.191.2 port 50971
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.0 sec  4.26 GBytes  3.65 Gbits/sec
[  4] local 192.168.191.157 port 5001 connected with 192.168.191.2 port 50998
[  4]  0.0-10.0 sec  4.34 GBytes  3.73 Gbits/sec
[  4] local 192.168.191.157 port 5001 connected with 192.168.191.2 port 51390
[  4]  0.0-10.0 sec  4.56 GBytes  3.92 Gbits/sec
```

Fig. 71. Recepción de paquetes TCP servidor.

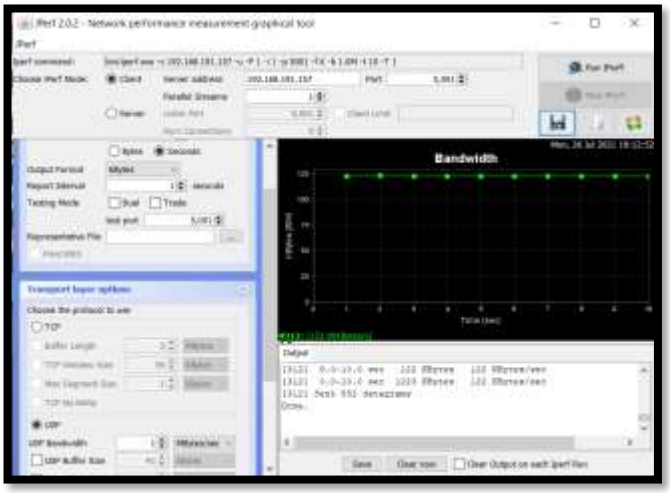


Fig. 72. Generación de paquetes UDP cliente

```
^Croot@usuario1-VirtualBox:/home/usuario1# iperf -s -u
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.191.157 port 5001 connected with 192.168.191.2 port 58388
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0-10.0 sec  1.19 MBytes  1.00 Mbits/sec   0.148 ms    0/  852 (0%)
[  4] local 192.168.191.157 port 5001 connected with 192.168.191.2 port 53707
[  4]  0.0-10.0 sec  1.19 MBytes  1.00 Mbits/sec   0.060 ms    0/  852 (0%)
[  3] local 192.168.191.157 port 5001 connected with 192.168.191.2 port 52929
[  3]  0.0-10.0 sec  1.19 MBytes  1.00 Mbits/sec   0.130 ms    0/  852 (0%)
[  4] local 192.168.191.157 port 5001 connected with 192.168.191.2 port 52936
[  4]  0.0-10.0 sec  1.19 MBytes  1.00 Mbits/sec   0.122 ms    0/  852 (0%)
```

Fig. 73. Recepción de paquetes UDP servidor.

Como se aprecia en las imágenes posteriores, la generación de paquetes y su captura presenta una gran capacidad debido al empleo de una amquina física y una virtual dentro de la misma red. A diferencia de las pruebas anteriores, por estar dentro de la misma red y sin la protección del firewall, los paquetes UDP pueden ser capturados sin problema. A diferencia de emplear IPv6 como se mostrará en el siguiente ejemplo.

3.3. Generación de tráfico con iperf en Windows:

Empleando las direcciones IPv6 es posible establecer la conexión entre maquinas físicas muy apartadas o de diferentes redes, sin embargo, como IPv6 emplea direcciones únicas es posible realizar las pruebas con iperf.

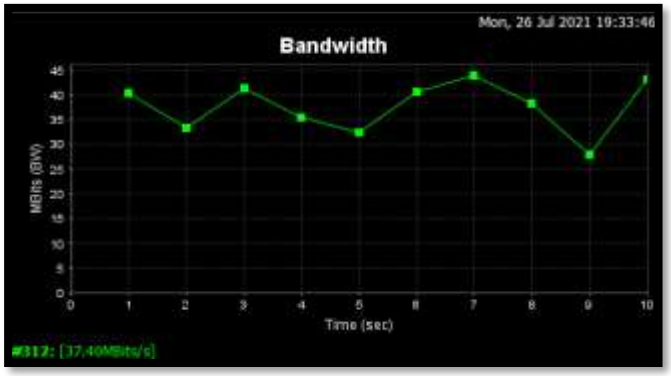


Fig. 74. Tráfico Saliente del cliente.

Al momento de utilizar IPv6 nos acercamos a un entorno real de al medir la capacidad a diferencia de trabajar con máquinas virtuales los valores se acercan a capacidades reales que podemos comprobar mediante páginas web de velocidad.

Comando TCP: iperf.exe -s -P 0 -i 1 -p 5001 -V -f k

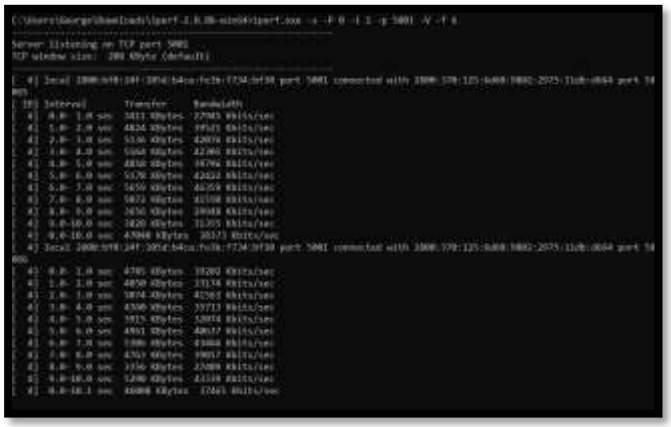


Fig. 75. Tráfico entrante en el servidor.

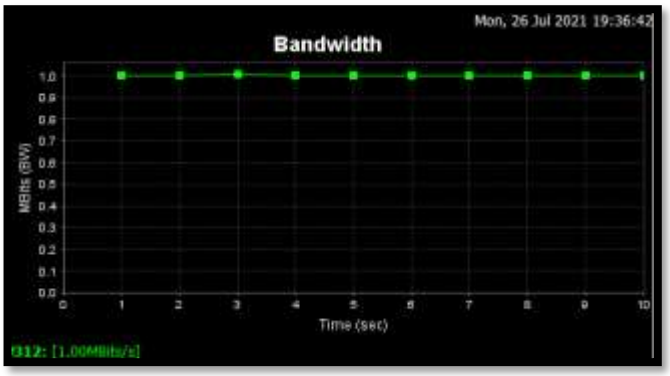


Fig. 76. Tráfico saliente del cliente.

Aquí pudimos notar una gran diferencia con el tráfico UDP ya que al no ser de confianza y los diferentes firewalls que debe atravesar no existe respuesta en nuestro servidor como ser observa en la figura 77. En cambio, en nuestra máquina virtual con nuestro host si fue posible como se puede apreciar en la figura 73.

Comando UDP: iperf.exe -s -u -P 0 -i 1 -p 5001 -f k

Fig. 77. Tráfico entrante en el servidor.

Conclusiones:

- *Se trabajo con 20 canales paralelos para poder evidenciar el particionamiento de la capacidad del canal y se comprobó que al enviar una sola señal se obtiene esa máxima capacidad. Cabe recalcar que al incrementar el numero de canales o caminos paralelos la capacidad del canal se divide, esta seria una buena forma de comprobar que el servicio de internet que se nos proporciona es el indicado.*

- *El PRTG nos permite monitorear equipos de enrutamiento mediante su dirección IP y existe una gran variedad de sensores para una amplia gama de protocolos, en el despliegue de una red real todas estas herramientas podrían ser usadas. Por cuestiones de procesamiento solo se han empleado protocolos simples en la topología diseñada a fin de analizar paquetes TCP Y UDP, además de la habilitación del protocolo SNMP.*

- *Para poder insertar los sensores SNMP y poder verificar el incremento del tráfico en las diferentes interfaces fue necesario configurar SNMP en los routers y en las máquinas virtuales.*

- *Para el monitoreo por medio de PRTG fue necesario implementar la topología en GNS3 ya que así podemos configurar el protocolo SNMP que permite monitorear las interfaces y su tráfico.*

- *Si bien es cierto pudimos haber agregado nuestro Router al PRTG monitor, pero por fines de aprendizaje y ya que los equipos pueden ser sensibles se optó por la opción de utilizar una topología virtual mediante el emulador de redes GNS3.*

- *En la generación de trafico UDP se puede mencionar que al habilitar este trafico desde el cliente, este va a ser visible solo en este sector, pero si se trata de analizar en el servidor los paquetes no serán visibles y la conexión será rechazada, esto puede deberse en esencia a que el nivel de seguridad que manejan estos paquetes es muy bajo. A pesar de inhabilitar el firewall tanto en la maquina cliente como servidor, el despliegue de toda la red presenta aún más filtros de seguridad que impiden el transporte de estos paquetes.*

Recomendaciones:

- *Se recomienda implementar esta topología en un computador con buenas características en memoria y procesamiento.*

- *Para configurar las máquinas virtuales se recomienda que antes de conectarla con GNS3 se configure el adaptador en modo Bridge e instalar todas las aplicaciones necesarias como IPERF, JPERF, Wireshark, PRTG. Y luego se podría clonar las máquinas clientes con diferentes direcciones MAC y solo tendríamos que cambiar las direcciones IP de acuerdo con la topología.*

- *Verificar que todos los protocolos han sido activados en los equipos de red, por ejemplo, en los routers en los cuales al efectuar el comando show running-config se pueden evidenciar las configuraciones realizadas.*

- *Si se emplean direcciones IPv6 en la herramienta de generación de tráfico JPERF es recomendable revisar que la casilla de direcciones IPv6 haya sido habilitada.*

VI.REFERENCIAS

[1] "¿Qué es el monitoreo de red?", Cisco. https://www.cisco.com/c/es_mx/solutions/automation/what-is-network-monitoring.html (consultado jul. 21, 2021).

[2] "UCM-Proyecto de Innovación Software libre para ciencias e ingenierías". https://www.ucm.es/pimcd2014-free-software/gns3 (consultado jul. 21, 2021).

[3] "Ubuntu 20.04 LTS: seguridad y rendimiento mejorados", Stackscale, abr. 30, 2020. https://www.stackscale.com/es/blog/ubuntu-20-04-lts/ (consultado jul. 21, 2021).

[4] "Características de PRTG Enterprise". https://www.paessler.com/es/prtg-enterprise-monitor/features (consultado jul. 21, 2021).

[5] "Wireshark". https://www.redeszone.net/tutoriales/redes-cable/wireshark-capturar-analizar-trafico-red/ (consultado jul. 21, 2021).

[6] "Cisco 3725 and Cisco 3745 - Cisco IOS Release 12.2(15)ZJ", Cisco. https://www.cisco.com/c/en/us/td/docs/ios/12_2/12_2z/release/notes/rn3700zj.html (consultado jul. 21, 2021).

[7] "Jperf 2.0.2: Medidor de capacidad de tráfico de red", Telectrónica. https://www.telectronika.com/descargas/jperf/ (consultado jul. 21, 2021).