Universidad de O'Higgins

Proyecto semestral

Streaming de video en Redes IP

Nicolás Araya, Felipe Gómez y Manuel Muñoz

COM4102 – Redes

Escuela de Ingeniería, Universidad de O'Higgins

22, 12, 2023

***Abstract*— En este trabajo se abordó la simulación de una red en GNS3 para la transmisión de video en tiempo real bajo el paradigma cliente-servidor. La red simulada se conformó por un cliente, tres routers y un servidor, todos conectados por una red ethernet. La simulación del cliente y del servidor se implementó en una máquina virtual de Virtualbox con el sistema operativo Ubuntu. El streaming fue realizado mediante la aplicación VLC con el protocolo UDP. Además, dada la falta de mecanismos de control de flujo y congestión de UDP se implementó también el uso de Traffic Shaping para suplir estas carencias. Los resultados mostraron que la gran limitación del Traffic Shaping empeoró el desempeño de la transmisión debido a la limitación de la cantidad de datos enviados en la red. Además, se realizó una captura de los paquetes enviados en la red mediante la aplicación Wireshark para su posterior análisis. Asimismo, se implementó en Python un analizador de archivos de Wireshark para determinar de manera categórica si se usó o no Traffic Shaping en el lapso de tiempo medido. Los resultados mostraron que UDP funciona de mejor manera sin el uso de Traffic Shaping. Sin embargo, aún existen deficiencias en cuanto al audio. Además, el detector funcionó de manera exitosa en cada uno de los datos probados.**

***Keywords*— GNS3, Streaming, Traffic Shaping, UDP, Wireshark.**

I. Introducción

El servicio de streaming de video es una parte importante de los servicios de entretenimiento al día de hoy como puede ser ChromeCast. Por esta razón, es crucial contar con tecnologías que permitan la optimización de la transmisión de video en tiempo real. Algo que es vital de explorar para tener el conocimiento del funcionamiento de las tecnologías y lógica que estos manejan.

El objetivo de este informe es mostrar la implementación de un streaming de video mediante el protocolo UDP, configurando en las interfaces de los routers las rutas estáticas entre las terminales de los nodos que tendrá la red mediante un protocolo de ruteo estático para permitir el envío de datos entre estos dispositivos permitiendo la medición de estos mismos y observar la cantidad de datos enviados junto con el protocolo correspondiente observable en los frames, finalmente realizando un análisis de desempeño de la red mediante la toma de métricas con el enfoque adecuado. Para la realización de esto, se utilizará GNS3, VLC, Wireshark y VirtualBox.

Para una emulación adecuada del objetivo es requerida la utilización de una arquitectura de red adecuada. El escenario de red consta de tres routers, un servidor de video y un servidor de cliente, pudiendo de esta manera medir el tráfico de entrada en una interfaz de un router para obtener la curva de llegada del tráfico entrante a los routers.

Una vez realizado todo esto, el siguiente objetivo corresponde a la realización de distintas pruebas que permitan encontrar cierta lógica para la detección de traffic shaping, pues será de vital importancia para este experimento, ya que es requerido conocer los efectos de este sobre la transmisión de un video entre dos dispositivos endpoints. Conociendo la definición de traffic shaping, que corresponde a la técnica utilizada para controlar el flujo de tráfico en una red, es posible gestionar y dar forma a la velocidad de transmisión de datos modificando seguramente los datos que reciba el endpoint objetivo y afectando de algún modo la calidad del video a transmitir.

II. Marco Teórico

2.1 Software técnico

2.1.2 GNS3

GNS3 es una herramienta que facilita la creación y simulación de topologías de red. Esta plataforma ofrece un entorno virtual donde los usuarios pueden diseñar y probar configuraciones de red de manera eficiente. GNS3 se destaca por proporcionar una interfaz intuitiva y flexible que permite la configuración de dispositivos de red virtuales y su interconexión.

2.1.3 Wireshark y sniffing

Cuando nos referimos a packet sniffing es obtener el tráfico de los paquetes que pasan a través de una red. En particular, Wireshark nos permite analizar el tráfico de una red, interceptando y convirtiéndose en un formato legible para las personas.

2.1.4 Virtualbox

Virtualbox es un programa que nos permite emular máquinas virtuales, en particular una máquina virtual es un software de emulación de hardware que permite ejecutar un sistema operativo y aplicaciones como si estuvieran corriendo en hardware físico real.

2.1.5 VLC

VLC es un reproductor multimedia de código abierto que permite la reproducción de una amplia variedad de formatos de vídeo y audio sin requerir la instalación de códecs adicionales. Además, tiene la capacidad de reproducir DVD y Blu-ray.

2.2 Endpoints y Routers

En el ámbito de las conexiones locales, como en el entorno empresarial, el Router emerge como un componente de vital importancia. Este dispositivo despliega su función al posibilitar la conexión entre diversos dispositivos, conocidos como endpoints. Estos endpoints abarcan tecnologías en manos de los usuarios, tales como computadoras, teléfonos móviles, entre otros. Asimismo, la creación de una red se materializa mediante la habilidad del enrutamiento y la interconexión de múltiples routers, propiciando así la comunicación entre estos endpoints. La esencia de estas conexiones, aunque a menudo pasada por alto, encierra un tejido que conecta no solo máquinas, sino también experiencias y facilita la interacción de un universo digital que, en su simplicidad aparente, nutre el trasfondo de nuestras interacciones cotidianas.

2.2.1 Traffic Shaping

En la figura a continuación se observan algunas estrategias que se emplean sobre los routers con el fin de gestionar el tráfico de la red hasta llevarlo a un punto en donde es estable. En este trabajo se abordará la alternativa del Traffic Shaping con el fin de revisar su desempeño empírico respecto a la calidad del video recibido por el cliente. Como tal, el Traffic Shaping se puede implementar mediante distintos algoritmos [4], pero la idea es siempre la misma, esta es la de generar un retardo controlado a los paquetes recibidos para mantener un flujo continuo en la transferencia de paquetes.

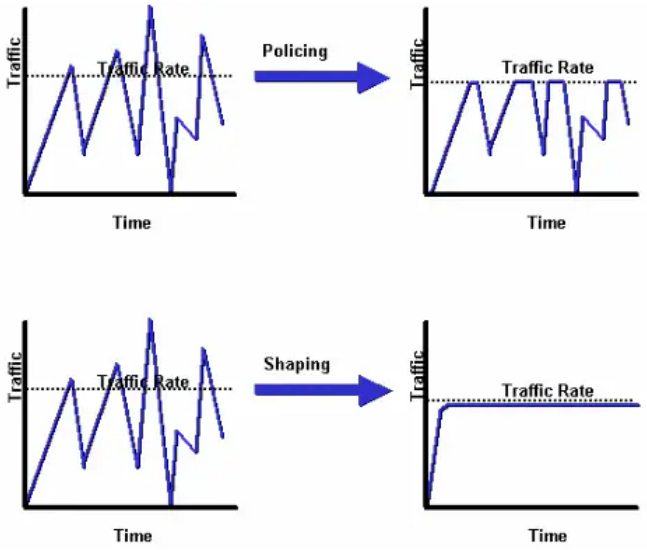

Fig 1. Estrategias de gestión de paquetes.

2.3 Métricas Detector Traffic Shaping

2.3.1 Desviación Estándar

En el marco del desarrollo del detector de traffic shaping fue utilizada la medida de desviación estándar, por lo cuál es crucial explicitar una definición con el fin de evitar ambigüedades.

La desviación estándar es una medida estadística que indica la cantidad de dispersión o variabilidad de un conjunto de datos. Mientras más baja sea esta, indica que la variabilidad de los datos es menor, en cambio una desviación estándar alta indica que esta variabilidad de los datos es alta.

2.3.2 Packet's Size

En la construcción del detector de traffic shaping, es fundamental comprender qué elementos extraídos del sniffer son esenciales para este propósito. En este contexto, se ha determinado la utilidad de la constante "Packet's Size", que representa el tamaño promedio. Este valor se obtiene mediante la división de la banda ancha entre el tamaño total de los paquetes registrados durante el sniffing con Wireshark. Este enfoque busca capturar la relación entre la cantidad de datos transmitidos y la velocidad de transmisión, proporcionando así un indicador clave para la detección de prácticas de traffic shaping en la red.

2.3.3 Media aritmética.

La media aritmética es una medida que provee el valor promedio de los datos analizados obtenidos desde el sniffing de wireshark.

La fórmula para calcular la media aritmética es la siguiente:

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

2.4 Modelo TCP/IP

El Modelo TCP/IP (Transmission Control Protocol/Internet Protocol) es un conjunto de protocolos de comunicación utilizado para conectar dispositivos en redes, incluyendo Internet. Está formado como un modelo de capas que simplifica y facilita la transmisión de datos entre diferentes dispositivos o procesos. El modelo está compuesto por cinco capas diferentes.

Cada capa del Modelo TCP/IP cumple una función específica y trabaja en conjunto con las otras para permitir la comunicación fluida y eficiente a través de redes de computadoras a nivel mundial. Este modelo es fundamental para el funcionamiento de Internet y las redes modernas.

2.4.1. Capa de Aplicación.

Esta capa gestiona los protocolos a nivel de interfaz de usuario, permitiendo a las aplicaciones acceder a los servicios de red. Algunos ejemplos de protocolos pertenecientes a esta capa son HTTP, SMTP o FTP.

2.4.2. Capa de Transporte

Responsable de la transferencia de datos entre puntos finales y proporciona control de flujo, manejo de errores y garantía de entrega. Los protocolos más conocidos aquí son TCP (orientado

a la conexión y confiable) y UDP (sin conexión y más ligero).

2.4.3. Capa de red

Esta capa se encarga de la dirección y el enrutamiento de paquetes a través de diferentes redes. IP (Internet Protocol) es el protocolo principal, asignando direcciones únicas (IP) a cada dispositivo y determinando la ruta óptima para que los datos lleguen a su destino.

2.4.4. Capa de enlace

También conocida como la capa de enlace de datos, gestiona la conexión física y lógica al hardware de red. Esta capa se ocupa de aspectos como la dirección MAC, el enlace Ethernet, y la conexión Wi-Fi, asegurando que los datos lleguen al dispositivo correcto en una red local.

2.4.5. Capa Física

La capa más baja se encarga de la transmisión de datos brutos a través de medios físicos como cables, fibra óptica, o señales inalámbricas. Incluye la especificación de dispositivos, medios de transmisión y tecnologías como Ethernet o protocolos inalámbricos.

2.5. Protocolos TCP y UDP.

2.5.1. TCP

El protocolo TCP o Transmission Control Protocol es un protocolo de la capa de transporte utilizado para establecer un canal de comunicación seguro para el intercambio de información entre dos procesos.

La comunicación se establece mediante un proceso de tres simples pasos conocido como "Three way handshake". Primero, el cliente envía un mensaje con la flag SYN (synchronize) al servidor para solicitar el inicio de una conexión, iniciando una secuencia de números que llamaremos "A". Luego el servidor responde con un mensaje que contiene las banderas SYN y ACK (acknowledgment) junto con un A+1 y una nueva secuencia que llamaremos B, confirmando la recepción del SYN del cliente y simultáneamente solicitando su propia conexión. Finalmente, el cliente envía un mensaje con la bandera ACK al servidor y B+1, reconociendo la respuesta del servidor, completando así el establecimiento de una conexión TCP fiable y bidireccional entre el cliente y el servidor.

El beneficio de establecer una conexión primero, es que esto permite [inserte beneficios de TCP]

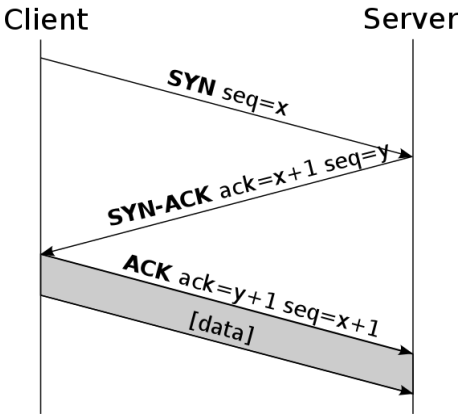

Fig 2. Three way handshake de tcp

2.5.2. UDP

El UDP o Protocolo de Datagramas de Usuario, a diferencia de TCP, es un protocolo de capa de transporte que funciona sin establecer una conexión. UDP es conocido por su simplicidad y velocidad, enviando "datagramas" sin establecer primero comunicación ni asegurar la entrega u orden de paquetes. Esto lo hace ideal para aplicaciones que requieren transferencias rápidas y eficientes, como transmisión de video o juegos en línea, pero que sufren de menor confiabilidad y carecen de mecanismos inherentes de verificación de errores de TCP.

III. Metodología

3.1 Herramientas

3.1.1 GNS 3

Para la emulación de la arquitectura de red se utilizó GNS 3 que es un simulador gráfico de red que contiene diversos dispositivos correspondientes a todas las capas de red del modelo OSI. Para implementar la red requerida, se debe instalar los templates referentes a cada dispositivo que se mencionan a continuación.

3.1.2 Cisco series c3700

Los routers Cisco en esta simulación permiten 3 tipos de conexiones, 2 a través de ethernet (f0/0, f0/) y una serial. No se utiliza la conexión serial, pues requerimos que los 3 routers están conectados de forma física a través de ethernet. Estos fueron instalados a través de la importación del template para cisco c3725 encontrado en el sitio oficial de GNS3.

Los cisco series 3700 [3] tienen dos módulos llamados EtherSwitch, el primero es un NM-ESW-16 10/100 de 16 puertos diseñado para proporcionar funciones de conmutación/switching y el segundo corresponde a un módulo de servicio de alta densidad HDSM 10/100 Etherswitch NMD-36-ESW de 36 puertos que permite el manejo de un alto número de puertos o conexiones. Los routers cisco pueden ofrecer datos, voz y video mediante soluciones inalámbricas fijas, de Ethernet Switching y de IP Routing, y funciones de gateway de voz.

3.1.3 Virtual Machine

Se opta por la instalación de máquinas virtuales. Ambas poseen Ubuntu, pues Ubuntu cumple con los requerimientos debido a su Kernel, que es el mismo que contiene LUbuntu y similares basados en Debian. Ubuntu, al igual que otros sistemas operativos permite la configuración de la ip del dispositivo a través del software de configuración. Además, gracias a que estos OS son compatibles con GNU nos permite instalar versiones más antiguas de VLC en caso de ser necesario.

Se crean dos máquinas virtuales que aporten como cliente y servidor conectadas a través de ethernet a los router R1 y R2 (ver Fig. 2).

Se configuran para que servidor tenga la ip 192.168.50.50 y cliente 192.168.60.60 para la comunicación y envío de mensajes mediante VLC media player.

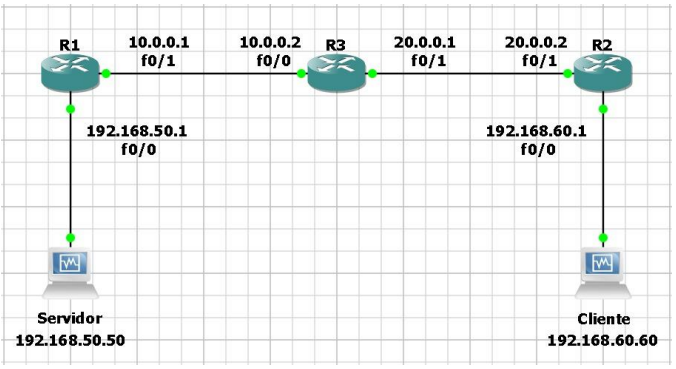

Fig. 3: Arquitectura de red implementada. Consta de 3 routers cisco series c3700 conectados mediante ethernet y de endpoints máquinas virtuales con Ubuntu OS.

3.1.4 VLC Media Player

VLC Media Player es un reproductor de vídeos open source que puede ser utilizado para el streaming de los mismos y es lo más relevante, pues con este software se realizará el streaming de video que nos permitirá evaluar la red y los mensajes que envían estas aplicaciones.

3.2 Configuración red y Máquinas Virtuales

Una vez armada la arquitectura de red en GNS3 se procede a configurar los router para permitir las conexiones a través de solarwind que es una consola que permite asignar ip a cada router a través de PuTTY.
El primer paso corresponde a definir la dirección ip de cada router por conexión que tengan y definir las rutas estáticas según se especifica en la arquitectura de la red (ver Fig. 3).

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface f0/0
R1(config-if)#ip address 192.168.50.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interfac
*Mar  1 00:02:47.143: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:02:48.143: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config)#interface f0/1
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#exit
```

Fig. 4: Configuración de ip Router 1. Se realiza el mismo procedimiento para los otros Router según arquitectura de red.

3.3 Detector de traffic shaping

Se desarrolló un detector de traffic shaping en Python mediante el uso de la librería Scapy. Esta implementación aprovecha las funciones de Scapy para leer archivos de sniffer con formato pcapng, extrayendo así todos los paquetes capturados y la información sobre el ancho de banda. Esta herramienta posibilita analizar archivos pcapng generados por sniffer de Wireshark, permitiendo determinar si, durante un periodo específico en la red, se aplicó o no control sobre los paquetes enviados.

La metodología se basa en el establecimiento de un umbral, definido por la fórmula:

$$\text{Threshold} = K * \text{mean}(\text{Packet's Size})$$

Donde K representa el límite de paquetes tolerados para determinar la presencia o ausencia de control.
La elección de K es crucial: si es mayor, se incrementa la tolerancia, lo que podría resultar en falsos negativos.
Si es menor, se aumenta la sensibilidad, aumentando la probabilidad de falsos positivos. En este caso, se optó por un valor de K igual a 3.

La variable adicional considerada es el tamaño de los paquetes, que representa la magnitud de todos los paquetes en un intervalo de tiempo determinado. La media de este tamaño es crucial para el cálculo del umbral, ya que cambios abruptos pueden indicar la presencia de traffic shaping.

Código disponible en: https://github.com/Felipe1401/Proyecto-redes.

IV. Resultados

Una vez realizado el experimento práctico se obtuvieron los siguientes resultados según el streaming de video y la detección de traffic shapping.

4.1.0 Transmisión UDP

Se transmitió a través de la opción de Streaming ubicada en Media/Stream posteriormente se selecciona el protocolo para la transmisión y se asigna la dirección ip objetivo del dispositivo endpoint de destino. Una vez realizada la configuración del streaming en el servidor, es necesario configurar el cliente para recibir la transmisión. Para ello, basta con ir a Convert/Save ubicado en Media y en la vista de Network asignar la url correspondiente, que en este caso corresponde a udp://:1234.
El contenido enviado corresponde a un video músical de Rick Astley llamado Never Gonna Give You Up (ver Fig. 5) en formato mp4 debido a su popularidad y a la calidad de imagen con un estilo de los 80's para realizar una recepción más adecuada y viable siendo la resolución y calidad un problema que se descarte gracias a esto.

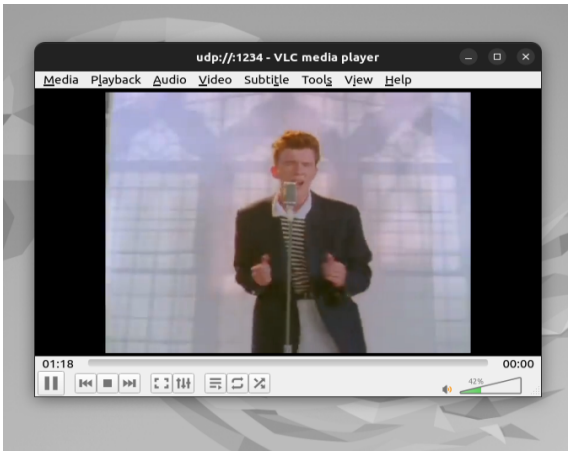

Fig. 5: Frame de clip de la transmisión de Never Gonna Give you Up de Rick Astley.

4.1.1 Sin Traffic Shaping

La transmisión en cuestión, visto desde el cliente, no presentó discontinuidad ni baja de calidad de ningún tipo al haber sido transmitido sin Traffic Shaping, esto se debe a que no hay limitaciones para los datos que son filtrados por el Router R1, evitando así la pérdida de datos esenciales para la calidad de este y la cantidad de paquetes recibidos por wireshark supera los 10000 (ver Fig. 6).

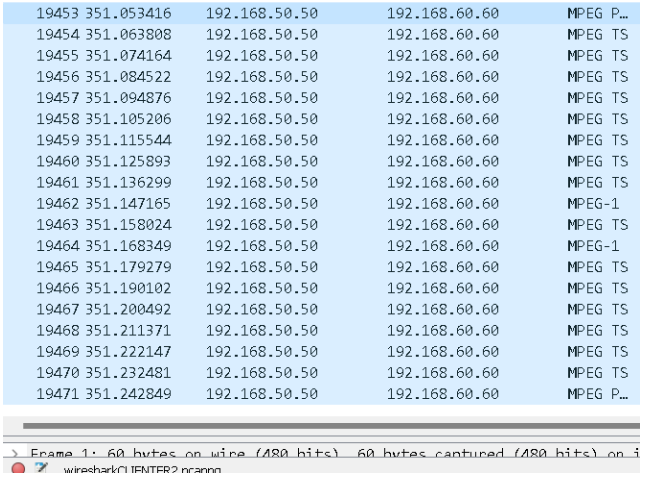

Fig. 6: Datos finales obtenidos sin la implementación de traffic shaping en el Router R1.

4.1.2 Con Traffic Shaping

Una vez aplicado traffic shaping en el Router R1, la transmisión inmediatamente comenzó a tener problemas de audio y posteriormente de imagen, pues, el agregar traffic shaping causó que se descartaran paquetes relevantes referentes a la calidad del video. Además, se notó este descarte de paquetes al ver a través del sniffer que dejó en claro que se reducen en gran medida los datos conservando aproximadamente el 10 % o menos de todo lo enviado (ver Fig. 7).

Los datos obtenidos demuestran que el uso de Traffic Shaping puede tener un impacto negativo en la calidad de una transmisión. En este caso, la transmisión se realizó sin Traffic Shaping y fue exitosa. Esto sugiere que el uso de Traffic Shaping debe realizarse con precaución, ya que puede afectar la calidad de la transmisión.

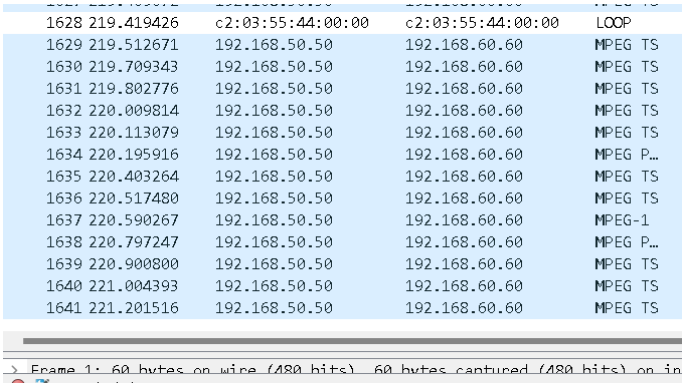

Fig. 7: Datos finales obtenidos al implementar traffic shaping con rate de 80000 en el router R1.

4.2 Detector de traffic shaping

Una vez desarrollado el detector de traffic shaping, se demostró su eficacia en la detección de este. La capacidad del detector radica en su habilidad para identificar el traffic shaping mediante la observación de la diferencia general en la cantidad de paquetes. Este enfoque se revela como una métrica principal y fácilmente observable, contribuyendo a la efectividad del detector.

```
hm, size = read_pcapng(sys.argv[1])
pn, bw = [*hm.values()], [*size.values()]
th = 3 * np.mean(np.float32(bw) / np.float32(pn))

if np.std(np.array(bw)) < th:
    print("Traffic Shaping")
else:
    print("No Traffic Shaping")
```

Fig.8: Fragmento de código que corresponde a la detección del traffic shaping.

Para utilizar el detector, primero se copia la dirección del archivo que se utilizará para la detección (ver Fig. 8) y posteriormente se agrega al momento de ejecutar el código como un string a través de la entrada estándar de la consola y finalmente el algoritmo realiza la comparación calculando el threshold.

V. Análisis

5.1 Uso de GNS3

La implementación de GNS3 fue de gran ayuda para la emulación de la red. Los router cisco series c3700 realizaron un buen trabajo Los resultados indican que a través de la transmisión UDP el streaming es más fluido a pesar de las minúsculas pérdidas de paquetes que este protocolo implica.

5.2 Desempeño de UDP y Traffic Shaping

El streaming obtenido utilizando UDP funcionó de manera correcta en cuanto a las imágenes transmitidas y en cuanto a sonido, pues no se presenció ninguna interrupción de estos salvo en los primeros segundos del video. En el caso del uso de Traffic Shaping, se obtuvieron peores resultados, pues el video simplemente no tenía imagen que mostrar y tampoco se escuchaba a pesar de ver que los datos si los recibía. Puede ser debido a la limitación del tamaño de los paquetes enviados causado por el mismo traffic shaping. Sin embargo, configurando de mejor manera esta limitación sería posible obtener un mejor rendimiento del streaming manteniendo un control del flujo de la red.

En este trabajo, no se implementó el envío de paquetes a través del protocolo TCP para poder realizar comparaciones directas de los desempeños de ambos protocolos, sin embargo, es posible realizar un análisis superficial entendiendo cómo funciona TCP. Para TCP se esperaría un peor desempeño en cuanto a la transmisión de audio y video en tiempo real dadas todas las validaciones de pérdidas de paquetes que caracterizan su robustez. Esto queda respaldado con lo que se comenta dentro de la misma página oficial de VLC, donde se menciona "TCP no es apropiado para muchas aplicaciones como, por ejemplo, aplicaciones en tiempo real." [5].

5.2 Desempeño del detector de Traffic Shaping

El detector de traffic creado a través de la utilización de un umbral que permite comparar las mediciones referentes a la clasificación binaria que detecta los casos en los cuales se es utilizado el traffic shaping y cuando no, mostró ser completamente eficiente en cada uno de los archivos en los cuales fue probado. Los archivos se compusieron de distintos puntos de la red medidos realizando sniffing con la herramienta de Wireshark en distintas conexiones de fast ethernet que componen la red priorizando aquellas que se encuentran direccionadas desde el router R1 hasta el endpoint del cliente para rescatar repercusiones del traffic shaping.

VI. Conclusiones generales

En este trabajo se ha estudiado la eficiencia del protocolo utilizado, siendo este UDP. Debido a la naturaleza de dicho protocolo, la latencia existente referente a la velocidad de transmisión de los datos enviados en el streaming es bastante baja permitiendo que el dispositivo receptor muestre el contenido de manera entendible. La razón, es que el

protocolo TCP tiene un retraso asociado a la retransmisión de paquetes en caso de que no lleguen o hayan llegado de forma corrupta al destino, lo cual no ocurre con el protocolo UDP.

Respecto a los dispositivos router se concluye que existen dos formas de medición de tasa de tráfico en los routers:

-**Throughput**, que corresponde a la cantidad real de datos que se pueden transmitir a través de una conexión en un periodo de tiempo.

-**Bandwidth** siendo la capacidad máxima teórica de transmisión de datos en una conexión.

El principal factor que influye en el tráfico de la red del ambiente emulado corresponde al protocolo utilizado en la transmisión de los segmentos. En este caso, al usar UDP, se prescindió de mecanismos de congestión y control de flujo, por lo cuál se requieren de técnicas como el traffic shaping para suplir esta carencia.

Por otro lado, se propone como una forma práctica de optimización referente al diseño de la topología de red emulada, la implementación de la conexión de los dos endpoints a través de un router de mejor calidad utilizando Local Area Network (LAN). Permitirá que ambos endpoints reciban de manera directa la transmisión de datos y, por lo tanto, el streaming se vea más fluido.

Otra forma de optimizar la red emulada, pero esta vez referente al recibimiento de los datos corresponde a la gestión de Jitter, el cuál es una métrica de variación de retardo en la transferencia de los datos medido a través de la desviación estándar de los tiempos de llegada de los paquetes.

Dentro de la implementación, una vez aplicado traffic shaping en el router 1 se descubrió que el efecto en la transmisión del video fue una pérdida de datos que no permitió una observación ni audición del contenido enviado. Respecto al envío de paquetes, se pudo notar una disminución de estos, siendo una cantidad máxima de 800 paquetes aproximadamente versus 17000 con y sin traffic shaping respectivamente.

Además, es posible concluir que una manera efectiva de generar mayor tráfico de video en la red emulada, con el fin de analizar la congestión, es con la generación de una mayor cantidad de servidores que hagan envío en simultáneo de streaming de diferentes videos en una mayor definición.

VII. References

[1] J. Postel, "Transmission Control Protocol", RFC Editor, sep. 1981. doi: 10.17487/rfc0793.

[2] Cisco Systems, Inc., "Compare Traffic Policy and Traffic Shape to Limit Bandwidth," 2023. [En línea]. Disponible en: https://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-policing/19645-policevsshape.html.

[3] Cisco Systems, Inc., "Preguntas frecuentes sobre EtherSwitch Network and Service Modules on 2600/3600/3700 Series Routers," 2007. [En línea]. Disponible en: https://www.cisco.com/c/es_mx/support/docs/interfaces-modules/network-modules/60565-etherswitch-FAQ.pdf.

[4] S. Xiong, A. D. Sarwate, y N. B. Mandayam, "Network traffic shaping for enhancing privacy in IoT systems," en IEEE/ACM Transactions on Networking, vol. 30, no. 3, pp. 1162-1177, 2022.

[5] VideoLAN Wiki, "TCP," VideoLAN Wiki, [En línea]. Disponible en: https://wiki.videolan.org/TCP/.