UNIVERSIDAD DE O'HIGGINS

Streaming de Video en Redes IP

Proyecto Final
Redes
COM4102-1

Integrantes:
Cristóbal Lagos V.
Bastián Rubio M.
Cristian Herrera B.

Profesor(a): Alfonso Ehijo B.
14 de Diciembre 2023

Contents

Abstract

Este informe se enfoca en la implementación de una red IP para la transmisión de video, utilizando herramientas como GNS3, Wireshark, VirtualBox y VLC, junto con sistemas operativos como Ubuntu y Lubuntu. Se examina la configuración de routers, asignación de direcciones IP, enrutamiento y transmisión de video a través del protocolo UDP.

El estudio incluye la exploración del "traffic shaping" para controlar el flujo de datos y su impacto en la eficiencia de la red. Mediante capturas de Wireshark y análisis detallado, se investiga cómo diferentes configuraciones de traffic shaping afectan el rendimiento, permitiendo comprender mejor su influencia en la eficiencia y flujo de datos en la red.

1 Introducción

En el complejo entorno de las redes informáticas, asegurar una comunicación fluida, una sólida seguridad y un flujo eficiente de datos sigue siendo un desafío constante. Este informe se enfoca en la configuración y análisis de un entorno de red de pruebas, utilizando herramientas esenciales como GNS3, Wireshark, VirtualBox, VLC y sistemas operativos como Ubuntu y Lubuntu.

La metodología adoptada abarca desde la creación y configuración de una red dentro de GNS3, haciendo énfasis en elementos tales como la asignación de direcciones IP, la configuración de routers para el enrutamiento y la validación de conexiones mediante protocolos de control de mensajes. Además, el informe explora la transmisión de video utilizando VLC a través del protocolo UDP.

Un aspecto fundamental para poder evaluar el rendimiento de la red se basa en el análisis del "traffic shaping", una técnica crucial para controlar el flujo de datos y optimizar el uso del ancho de banda. Al experimentar con diversas configuraciones de este parámetro y examinar su impacto en la transmisión, este informe utiliza capturas de Wireshark y análisis detallados para explicar cómo diferentes configuraciones de traffic shaping afectan el rendimiento de la red.

Por último, este informe tiene como objetivo proporcionar una perspectiva general sobre la configuración y análisis de redes, destacando la importancia de herramientas específicas y técnicas para mantener un rendimiento óptimo en entornos de comunicación. El exhaustivo examen del traffic shaping contribuye a una comprensión profunda de su influencia en la eficiencia y el flujo de datos dentro del contexto de una red.

2 Marco Teórico

A continuación se definirán los conceptos necesarios para poder llevar a cabo dicho proyecto.

- **GNS3**: Simulador de red de código abierto que permite emular redes complejas utilizando dispositivos virtuales. Permite configurar y probar redes sin necesidad de hardware físico, lo que lo convierte en una herramienta valiosa para aprender sobre networking, realizar pruebas y desarrollar configuraciones de red.

- **Wireshark**: Herramienta que permite capturar y analizar el tráfico de red en tiempo real. Desde paquetes de datos hasta detalles sobre protocolos específicos, Wireshark muestra información detallada que puede ser útil para solucionar problemas de red, analizar seguridad o simplemente entender cómo funcionan las comunicaciones en una red.

- **VirtualBox**: Software de virtualización de código abierto que permite crear y ejecutar máquinas virtuales en una computadora. Con VirtualBox, se puede instalar y ejecutar múltiples sistemas operativos simultáneamente, como Windows, Linux, macOS y otros, dentro del sistema operativo principal.

- **VLC**: Reproductor multimedia de código abierto y multiplataforma que ofrece una amplia compatibilidad con una diversidad de formatos de archivos de audio y video. Este software se destaca por su versatilidad, permitiendo

la reproducción de contenido local, streaming y la capacidad de convertir archivos multimedia entre distintos formatos.

- **Ubuntu**: Sistemas operativos de código abierto más populares basados en Linux. Es reconocido por su facilidad de uso, estabilidad y gran comunidad de usuarios y desarrolladores.

- **Lubuntu**: Variante oficial de Ubuntu, un sistema operativo basado en Linux. Está diseñado para ofrecer un entorno ligero y de bajo consumo de recursos, ideal para computadoras más antiguas o con especificaciones limitadas.

- **TCP (Transmission Control Protocol)**: Protocolo de comunicación de la capa de transporte en redes de computadoras. Es uno de los componentes fundamentales del modelo TCP/IP, utilizado en internet y en muchas redes locales.

- **UDP (User Datagram Protocol)**: Protocolo de comunicación en redes de computadoras que pertenece a la capa de transporte del modelo OSI. A diferencia del TCP (Transmission Control Protocol), UDP es un protocolo sin conexión, lo que significa que no establece una conexión antes de enviar datos y no garantiza la entrega o el orden de los paquetes.

- **Firewall**: Barrera de seguridad que se utiliza para controlar y filtrar el tráfico de red. Puede ser un dispositivo físico, software o una combinación de ambos, diseñado para proteger una red privada al controlar el flujo de datos que entra y sale de esta red.

- **Traffic shapping**: Técnica utilizada en redes de computadoras para controlar el flujo de datos, priorizar ciertos tipos de tráfico sobre otros y optimizar el uso del ancho de banda disponible.

- **Direcciones IP**: Identificadores numéricos únicos asignados a cada dispositivo conectado a una red de computadoras que utiliza el protocolo de Internet (IP). Estos identificadores permiten que los dispositivos se comuniquen entre sí a través de la red.

- **MPEG TS**: (MPEG Transport Stream) es un protocolo de transporte de datos ampliamente utilizado para la transmisión de video, audio y datos en entornos de difusión y almacenamiento. Fue desarrollado por el grupo Moving Picture Experts Group (MPEG) y se ha convertido en un estándar fundamental para la distribución de contenido multimedia.

- **MPEG PES**: (Packetized Elementary Stream) es una parte esencial del proceso de codificación y transporte de datos multimedia dentro del estándar MPEG. Se enfoca en la fragmentación y empaquetado de flujos de datos de audio o video en paquetes elementales, lo que facilita su transporte y manipulación en diferentes aplicaciones multimedia.

3 Metodología

3.1 Creación y Configuración de la red en GNS3

Antes de comenzar con la transmisión del video mediante los distintos protocolos, lo que debemos hacer es crear nuestra red en GNS3. Para eso, utilizaremos los routers modelo c3725, luego lo que haremos será inicializar dos sistemas operativos, el primero será Lubuntu de 64-bits y un Ubuntu también de 64 bits. Luego abriremos GNS3, nos dirigimos a Edit y agregamos las dos máquinas anteriormente descritas. Nos debería quedar de la siguiente manera:

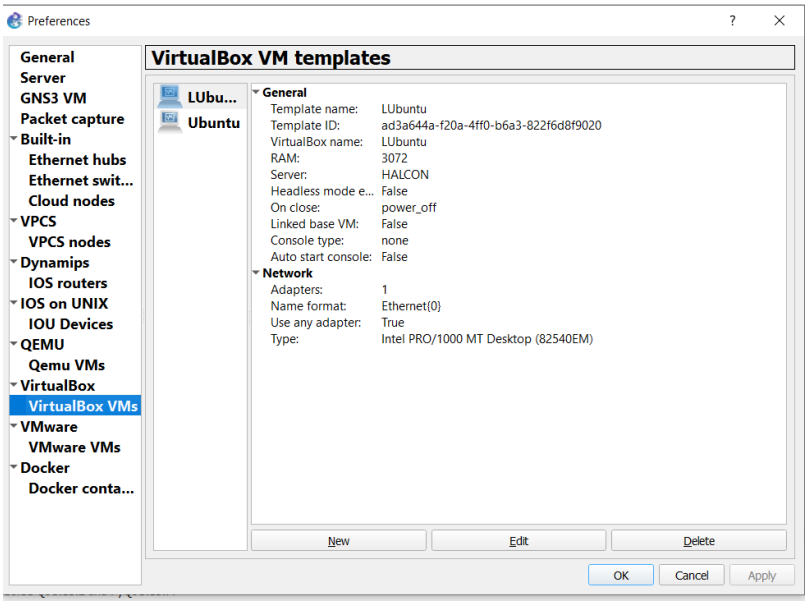


Figure 1: Asignación de las máquinas virtuales en GNS3

Una vez hecho esto, procedemos a armar nuestra red, con tres routers y las máquinas mencionadas anteriormente, que serán nuestro cliente y nuestro servidor. Quedando la red de esta forma:

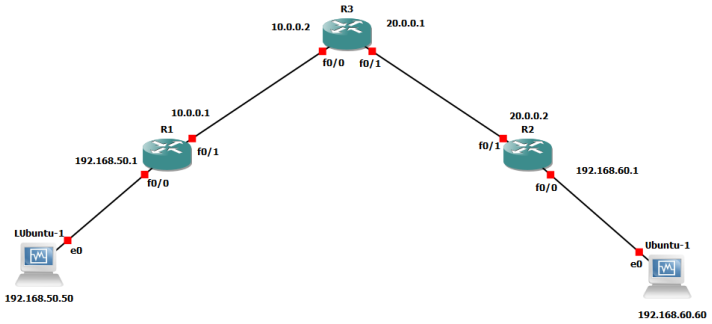


Figure 2: Red para la transmisión

Una vez creada y asignada la red mediante sus interfaces correspondientes, lo que haremos será ejecutar todos los nodos, el resultado esperado es que las dos máquinas virtuales se ejecuten y en GNS3 aparezca el siguiente estado:

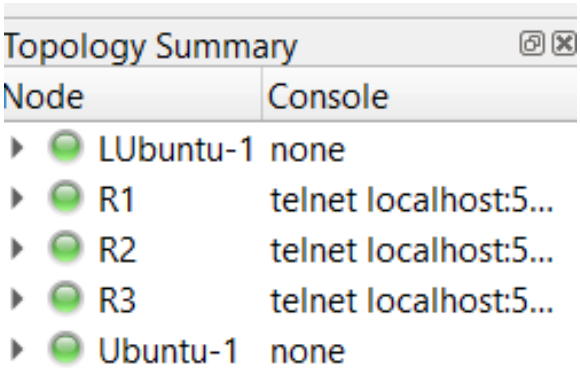


Figure 3: Estado de los nodos

Una vez configurado lo anterior y que los nodos estén corriendo pasamos al siguiente paso.

3.1.1 Configuración de Routers

Como bien sabemos, la red no es inteligente, por lo que debemos establecer las direcciones IP a cada router, para ello lo que haremos será abrir las consolas y ejecutar configure terminal seguido de los siguientes comandos:

```
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)#
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
```

(a) Asignación IP Router 1

```
R2(config)#interface f0/0
R2(config-if)#ip address 192.168.60.1 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#
R2(config)#interface f0/1
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
R2(config-if)#exit
```

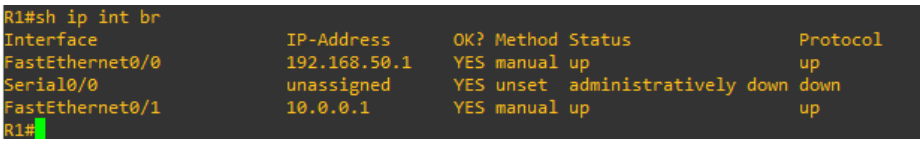
(b) Asignación IP Router 2

```
R3(config)#interface f0/0
R3(config-if)#ip address 10.0.0.2 255.0.0.0
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
R3(config)#interface f0/1
R3(config-if)#ip address 20.0.0.1 255.0.0.0
R3(config-if)#no shutdown
R3(config-if)#exit
```

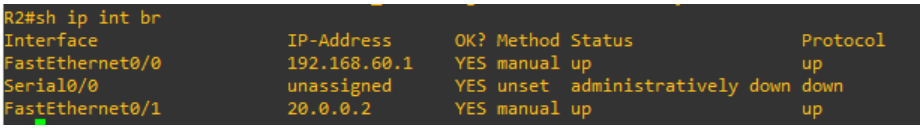
(c) Asignación IP Router 3

Figure 4: Asignación IP Routers

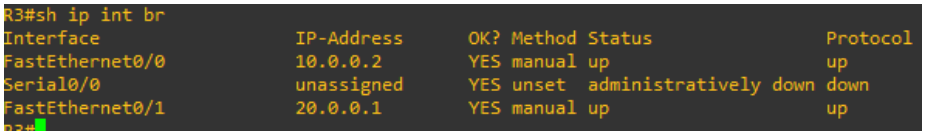
Una vez hecho esto, para poder visualizar los estados de los router podemos ejecutar el comando **sh ip int br** que es una abreviatura de "show ip interface brief". Este comando nos mostrará una lista resumida de las interfaces de red IP presentes en el dispositivo y su estado.



(a) Estado Router 1



(b) Estado Router 2



(c) Estado Router 3

Figure 5: Estado de los Routers

Hasta ahora, lo que tenemos es a los routers en un estado de up, y conectados mediante interfaces, entonces lo que haremos será comprobar que están correctamente conectados utilizando el protocolo de control de mensaje con el comando **ping**. Para ello haremos un ping desde R1 hacia R3, y luego de R1 hacia R2.

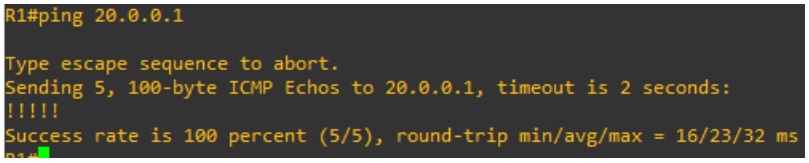


Figure 6: Ping R1 a R3

Podemos observar que llegaron los 5 paquetes que mando el protocolo, por lo tanto, existe una conexión entre R1 y R3, ahora haremos lo mismo con R1 y R2.

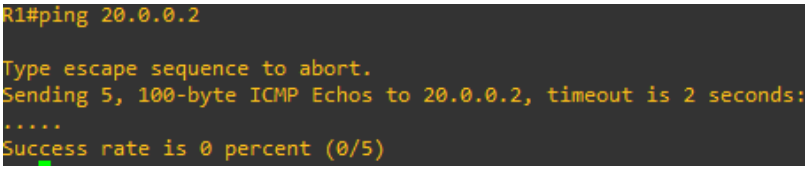


Figure 7: Ping R1 a R2

Podemos observar que al enviar paquetes desde R1 hacia R2, no tenemos una respuesta esperada ya que se pierden los paquetes, ahora para solucionar esto debemos enrutar los routers de la siguiente manera.

```
R1(config)#ip route 192.168.50.0 255.255.255.0 f0/0
R1(config)#ip route 10.0.0.0 255.0.0.0 f0/1
R1(config)#ip route 20.0.0.0 255.0.0.0 f0/1
R1(config)#ip route 192.168.60.0 255.255.255.0 f0/1
```

(a) Asignación rutas Router 1

```
R2(config)#ip route 192.168.60.0 255.255.255.0 f0/0
R2(config)#ip route 20.0.0.0 255.0.0.0 f0/1
R2(config)#ip route 10.0.0.0 255.0.0.0 f0/1
R2(config)#ip route 192.168.50.0 255.255.255.0 f0/0
```

(b) Asignación rutas Router 2

```
R3(config)#ip route 10.0.0.0 255.0.0.0 f0/0
R3(config)#ip route 20.0.0.0 255.0.0.0 f0/1
R3(config)#ip route 192.168.50.0 255.255.255.0 f0/0
R3(config)#ip route 192.168.60.0 255.255.255.0 f0/1
```

(c) Asignación rutas Router 3

Figure 8: Asignación rutas de los Routers

Para verificar que están correctamente asignadas, lo que haremos será ejecutar el comando **show ip route** dentro de configure terminal, y el resultado esperado es el siguiente:

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

S    192.168.60.0/24 is directly connected, FastEthernet0/1
S    20.0.0.0/8 is directly connected, FastEthernet0/1
C    10.0.0.0/8 is directly connected, FastEthernet0/1
C    192.168.50.0/24 is directly connected, FastEthernet0/0
```

(a) Rutas Router 1

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.60.0/24 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, FastEthernet0/1
S    10.0.0.0/8 is directly connected, FastEthernet0/1
S    192.168.50.0/24 is directly connected, FastEthernet0/0
```

(b) Rutas Router 2

```
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

S    192.168.60.0/24 is directly connected, FastEthernet0/1
C    20.0.0.0/8 is directly connected, FastEthernet0/1
C    10.0.0.0/8 is directly connected, FastEthernet0/0
S    192.168.50.0/24 is directly connected, FastEthernet0/0
```

(c) Rutas Router 3

Figure 9: Rutas de los Routers

Con todo lo anterior, volveremos a realizar el mismo ping desde R1 hasta R2

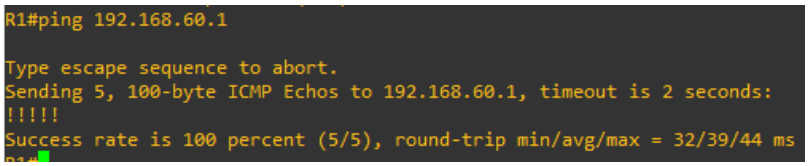
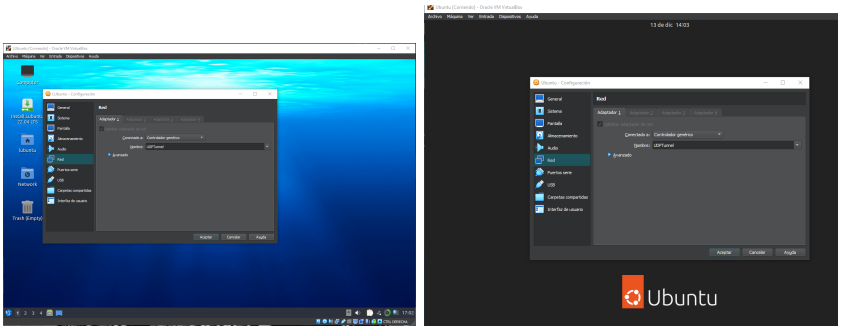


Figure 10: Ping R1 a R2 con rutas

Una vez todo conectado los routers, procederemos a configurar las maquinas virtuales.

3.1.2 Configuración de las maquinas virtuales

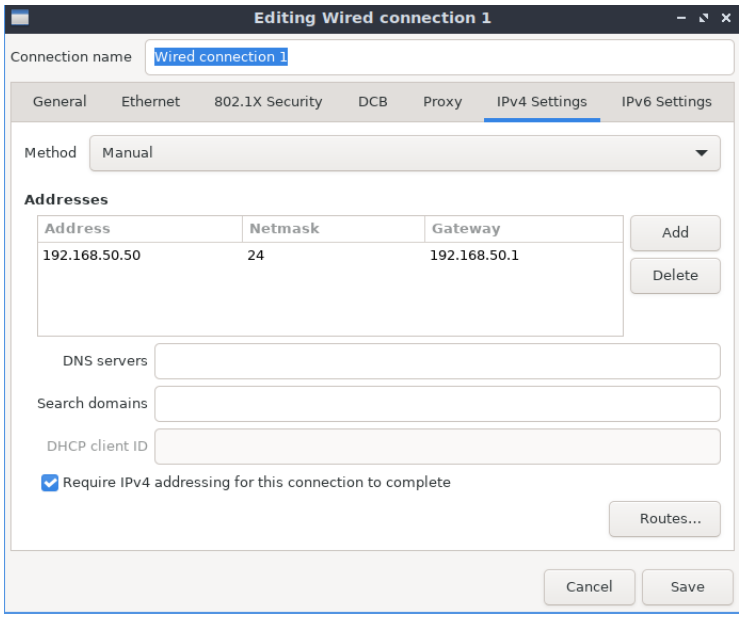
Al iniciar todos los nodos mediante GNS3, también se nos inicializaron las Maquinas virtuales (Lubuntu como ubuntu), ahora lo que debemos hacer es verificar que estén conectadas a la red de GNS3, para ello iremos a la opción de preferencias de red de virtual box y nos aseguremos que esten como se muestra a continuación:



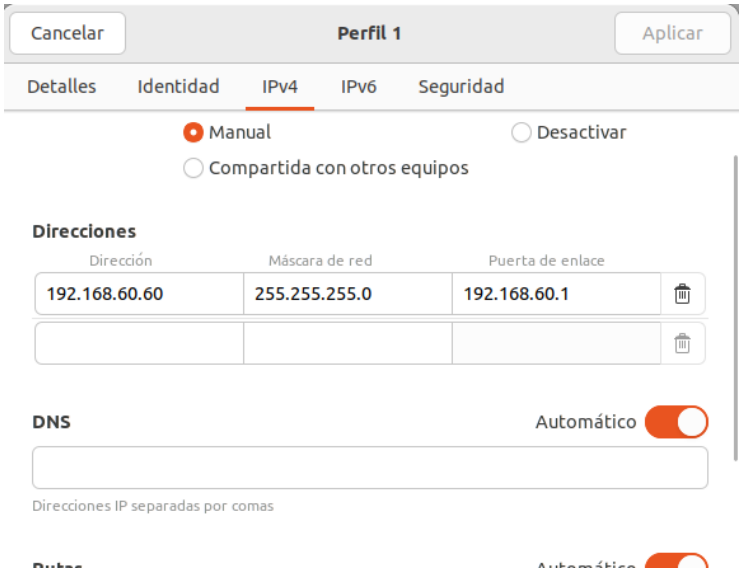
(a) Lubuntu (b) Ubuntu

Figure 11: Preferencias de red

Luego lo que haremos será configurar las conexiones de tal manera que las maquinas virtuales se conecten a los routers de nuestra red de la siguiente forma: Primero en Lubuntu, lo que haremos será ir al apartado de conexiones, luego edit connections y editamos la conexión que encontramos, para ello, apretamos configuración y nos dirigimos a IPv4 Settings y le colocamos la configuración de más abajo. Ahora para Ubuntu, lo que haremos será ir al apartado de red, creamos un nuevo apartado e introducimos la configuración mostrada a continuación.



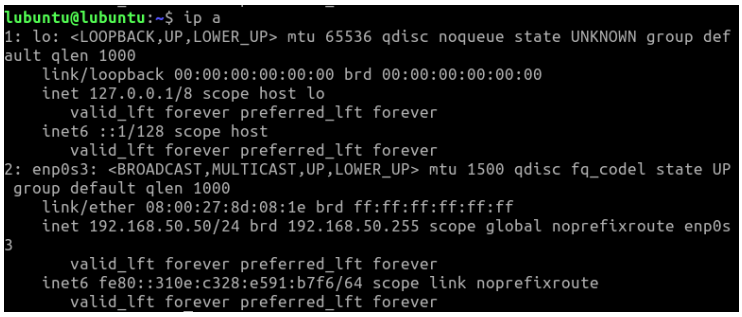
(a) Lubuntu



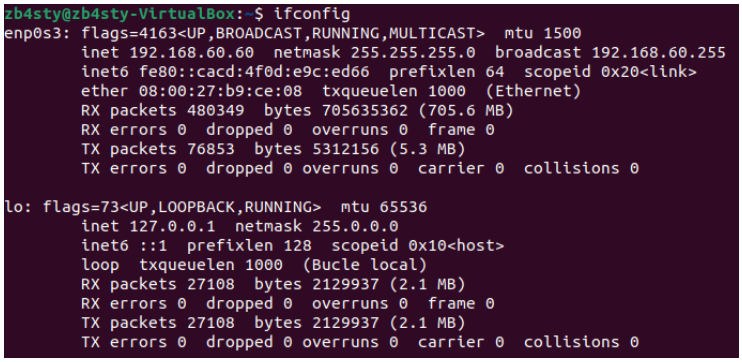
(b) Ubuntu

Figure 12: Asignación de IPs

Por último, comprobaremos la asignación correcta y se debería ver de la siguiente forma:



(a) Lubuntu



(b) Ubuntu

Figure 13: Ips de las máquinas virtuales

Con todo lo anterior tenemos listo para transmitir el video a través de las máquinas virtuales.

3.2 Transmisión de video mediante UDP

Para la transmisión del video a través del protocolo UDP, lo que haremos será lo siguiente:

- **Parte del servidor**: Abrir VLC media player, luego ir a Stream, añadimos el archivo .mp4 que queremos transmitir luego apretamos el botón Stream, Next, cambiamos el formato a UDP (legacy) y apretamos el botón Add, en Address agregamos la dirección a la cuál queremos enviar (En este caso es 192.168.60.60) y dejamos el puerto 1234, apretamos Next. Ahora en transcoding elegimos *Video - H.264 + MP3 (MP4)*, apretamos el botón de Next y por último Stream. **A este punto ya se esta transmitiendo el video**.
- **Parte del cliente**: Abrir VLC media player, ir a Convert, opción Red, y en la URL, colocamos udp://1234 y por último en el combobox, elegimos Play. **Ahora se empezará a mostrar el video transmitido en el servidor**

Lo anterior es una screenshot tomada del video del funcionamiento, video de transmisión por UDP

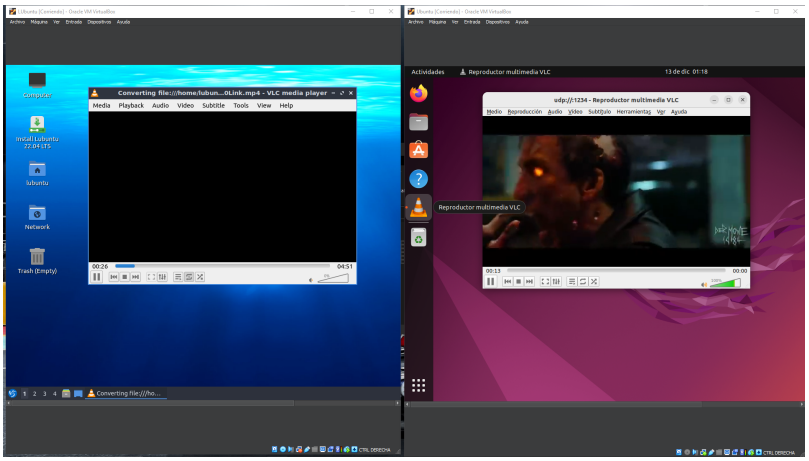


Figure 14: Transmisión con UDP

3.3 Creación de detector de alteraciones de tráfico

Con el fin de analizar el fenómeno del traffic shaping, se llevará a cabo un estudio mediante la captura de paquetes de red con Wireshark durante un lapso de 15 segundos aproximadamente. Se realizarán cuatro capturas distintas para examinar el comportamiento de la red bajo diferentes condiciones de traffic shaping: una captura sin la aplicación de traffic shaping, otra con un traffic shaping de 10,000, una tercera con 80,000 y finalmente una captura con un nivel extremadamente alto de traffic shaping.

Primero lo que haremos será programar el conteo de paquetes de la siguiente forma:

```
import pyshark

# Lista para los conteo de paquetes
npackage = []


# Detectar el envio de paquetes
def detect_network_alteration(pcap_file):
    # Leer la captura
    cap = pyshark.FileCapture(pcap_file)
    sent_packets = 0

    # Contar todos los paquetes
    for pkt in cap:
        try:
            sent_packets += 1

        except AttributeError:
            pass

    return sent_packets


# Llamar a la funci n y pasar la ruta del archivo de captura .
    pcapng y agregarlas a la lista
result = detect_network_alteration('Captura limpia 15s.pcapng')
npackage.append(result)
```

```
result1 = detect_network_alteration('Captura t10000 15s.pcapng')
npackage.append(result1)
result2 = detect_network_alteration('Captura t80000 15s.pcapng')
npackage.append(result2)
result3 = detect_network_alteration('Captura tmuyalto 15s.pcapng')
npackage.append(result3)

# Mostrar la ruta para
print(npackage)
```

Listing 1: Conteo de paquetes

Una vez hecho esto procedemos a crear el código para detectar el traffic shapping

```
import pyshark

# Paquetes capturados de cada uno
result = [1690, 20, 143, 1575]
mean = sum(result) / len(result)

# Mostrar el Threshold
print(f" Threshold : {mean}")

# Funci n para detectar anomalias
def detect_traffic_alteration(pcapng_file):
    # Capturar el paquete
    cap = pyshark.FileCapture(pcapng_file)
    packets = 0

    # Sumar todos los paquetes encontrados
    for pkt in cap:
        try:
            packets += 1

        except AttributeError:
            pass

    # Analizar los paquetes UDP capturados
    threshold = mean  # Promedio de los paquetes

    # Verificar si los paquetes son menor a los threshold
    if packets < threshold:
        return "Posible alteraci n de tr fico detectada en la
    transmisi n UDP"
    else:
        return "No se detecta alteraci n de tr fico en la
    transmisi n UDP"


# Llamar a la funci n y pasar la ruta del archivo de captura .
    pcapng
result = detect_traffic_alteration('Captura tmuyalto 15s.pcapng')
print(result)
```

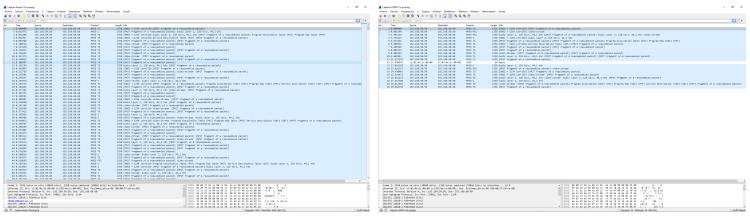
Listing 2: Traffic shapping

4 Resultados

A continuación mostraremos los resultados obtenidos.

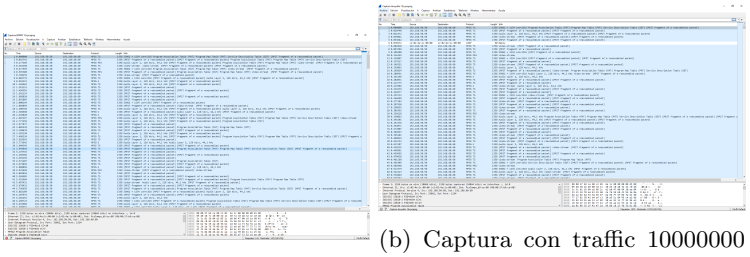
4.1 Traffic Shapping

4.1.1 Capturas de Wireshark



(a) Captura Limpia (b) Capturas de Wireshark

Figure 15: Primero dos capturas



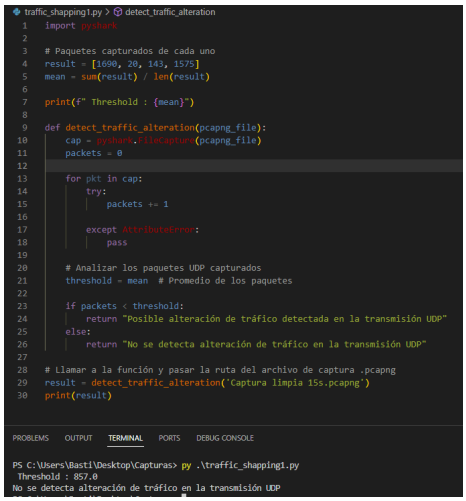
(a) Captura con traffic 80000 (b) Captura con traffic 10000000 (muy alto)

Figure 16: Capturas de Wireshark con un lapso de tiempo de 15s

4.1.2 Capturas analizadas por el código

Se utilizaron los códigos previamente creados para contar el número de paquetes capturados en cada situación. Posteriormente, se calculará un umbral o threshold a partir del promedio obtenido de estos recuentos. Este umbral representará la cantidad esperada de paquetes durante condiciones normales de la red.

El posterior análisis se centrará en la comparación entre el número real de paquetes capturados en cada escenario y el umbral establecido previamente. Esta comparación nos permitirá determinar si existieron alteraciones significativas en el tráfico de red durante las distintas configuraciones de traffic shaping, brindando así una comprensión más profunda sobre el impacto de estas técnicas en el flujo de datos de la red.



(a) Captura Limpia

(b) Captura con traffic 10000

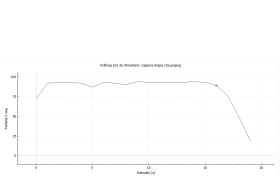
Figure 17: Primero dos capturas

(a) Captura con traffic 80000

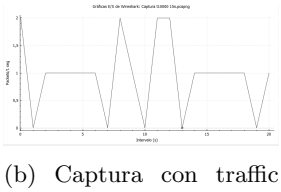
(b) Captura con traffic 10000000 (muy alto)

Figure 18: Segundas dos capturas

4.1.3 Gráficas de entradas y salidas de paquetes

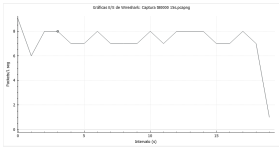


(a) Captura Limpia

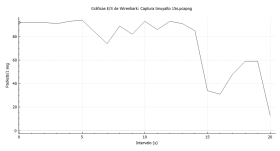


(b) Captura con traffic 10000

Figure 19: Primero dos capturas



(a) Captura con traffic 80000



(b) Captura con traffic 10000000 (muy alto)

Figure 20: Segundas dos capturas

5 Análisis

5.1 Enrutamiento de redes

El enrutamiento de redes y las tablas de enrutamiento son esenciales para el funcionamiento eficiente de cualquier red de computadoras. Estos sistemas permiten que los paquetes de datos se desplacen de manera efectiva entre dispositivos, incluso cuando están en redes separadas, facilitando la conectividad entre diferentes puntos de la red. Como pudimos observar anteriormente, cuando intentamos realizar un ping desde R1 a R3 funcionaba correctamente, pero al hacerlo con R2 no, debido a que, la red de por sí no es inteligente, por lo que se debe almacenar la información sobre las rutas para el tráfico de datos, las tablas de enrutamiento optimizan el flujo, evitando congestiones y retrasos innecesarios. Además, ofrecen redundancia al contener múltiples rutas hacia un mismo destino, garantizando que la red siga operativa en caso de fallos.

5.2 Análisis de Capturas

La medición de la tasa de tráfico en los routers es fundamental para comprender el volumen de datos que atraviesa una interfaz de red en un periodo definido. Esta medida se expresa en bits por segundo (bps), kilobits por segundo (Kbps), megabits por segundo (Mbps) o gigabits por segundo (Gbps), y varía según la escala y la velocidad de transmisión de la red.

Este monitoreo nos permite evaluar la carga de trabajo específica de una interfaz, brindando una visión detallada del flujo de datos procesado por el router. Esta información es esencial para analizar el rendimiento de la red, identificar

posibles congestiones o cuellos de botella y tomar decisiones informadas sobre ajustes en la configuración para optimizar la eficiencia del enrutamiento.

En este contexto de entornos de streaming de video, es común observar el uso de protocolos como MPEG-TS (MPEG Transport Stream) y MPEG PES (Packetized Elementary Stream). Estos protocolos desempeñan un papel crucial en la fragmentación, multiplexación y transmisión eficiente de datos de audio y video. Monitorear la tasa de tráfico en relación con estos protocolos específicos permite comprender mejor la distribución y el flujo de datos multimedia en la red, lo que resulta clave para mejorar la calidad y la eficiencia de la transmisión de video.

El tipo de tráfico emulado es un factor crítico en la determinación del flujo de datos en las interfaces de red de los routers. Por ejemplo, el tráfico de video suele generar considerablemente más datos que el tráfico de texto simple. Siendo la capacidad del ancho de banda disponible en la red emulada algo crucial; si es limitada, puede causar congestión en las interfaces de los routers si se transmite un gran volumen de datos. Los protocolos de red específicos y sus requisitos de ancho de banda también juegan un papel importante, ya que algunos protocolos pueden consumir más recursos de red que otros, lo que afectaría la carga en las interfaces de los routers. Además, la cantidad de dispositivos y usuarios simulados en la red impacta directamente en el tráfico, a medida que aumenta su número, se incrementa la cantidad de datos transmitidos. La implementación de traffic shaping para simular distintas condiciones de red altera significativamente la cantidad de tráfico que atraviesa las interfaces de los routers al limitar o priorizar ciertos flujos de datos. Por otro lado, el comportamiento de las aplicaciones utilizadas en la emulación también influye en el tráfico, dado que aplicaciones de transmisión de video pueden generar picos de tráfico más altos en comparación con aplicaciones que generan tráfico más constante.

Por último, podemos observar que dependiendo de la cantidad de traffic que coloquemos el gráfico de los paquetes entrantes/salientes varia, por ejemplo, la captura limpia podemos ver mayor prolijidad a la hora de los paquetes que salen y entran, mientras que cuando tenemos un traffic de 10000, el gráfico varia llegando a puntos donde no salen paquetes y luego se disparan. Mientras mayor el traffic mayor es el número de paquetes que salen y entran.

5.3 Traffic Shapping en UDP

Como era de esperarse, durante las condiciones de traffic shaping bajo, se observó un menor flujo de paquetes en comparación (alrededor de 80 paquetes en promedio) con los escenarios donde no se aplicó traffic shaping o se implementó a niveles muy altos (Donde hubo más de 1000 paquetes). Este fenómeno se alinea con la naturaleza de estas técnicas, donde la limitación o priorización de ciertos flujos de datos impacta directamente en la cantidad de paquetes transmitidos.

Por otro lado, en ausencia de restricciones (sin traffic shaping o con un valor alto de traffic), se evidenció un aumento significativo en el flujo de paquetes. Esta mayor cantidad de paquetes durante estas configuraciones indica una liberación o priorización sin limitaciones de los datos, lo cual es característico de estos contextos.

El análisis detallado de estos patrones en el flujo de paquetes en diferentes niveles de traffic shaping nos permite comprender de mejor manera cómo estas técnicas influyen en la distribución y el volumen del tráfico de red, proporcio-

nando una visión más completa de su impacto en el rendimiento y la eficiencia de la red.

5.4 Protocolo más óptimo para la transmisión de video

En cuanto a la decisión de usar protocolo UDP o TCP, en el contexto de transmisión de video en tiempo real, es más óptimo escoger UDP por sobre TCP, esto pues, al ser TCP más enfocado en la integridad de los datos enviados (pues implementa métodos como el 3-way-handshaking), repercute negativamente en la latencia causando que el cliente no pueda reproducir el video a una velocidad constante.

5.5 Posibles optimizaciones de la red

En cuanto a las posibles optimizaciones de red que creemos que pueden desencadenar en mejores resultados, destacan las siguientes:

1. Caché local: Almacenar en el buffer segmentos de video para evitar interrupciones durante la reproducción. También se puede implementar almacenamiento en caché en el lado del cliente y del servidor, esto con el objetivo de optimizar la entrega.

2. Buffering inteligente: Implementar mecanismos de buffering adecuados puede ayudar a lidiar con variaciones en la velocidad de la red. Un buffer bien gestionado puede suavizar las fluctuaciones en la velocidad de transmisión y proporcionar una experiencia de visualización más consistente.

3. Priorización de tráfico: Asignar prioridades a diferentes tipos de tráfico puede ser beneficioso, ya que podría dar prioridad a los paquetes de video para asegurar una entrega más rápida y suave.

   En conjunto, estas técnicas permitirían una mayor eficiencia en la reproducción de videos a través de transmisiones mediante internet.

6 Conclusiones generales

La implementación y evaluación de la red propuesta revelaron perspectivas significativas sobre la configuración eficiente y el análisis detallado en entornos de comunicación. La combinación de herramientas como GNS3, Wireshark, VirtualBox, VLC y sistemas operativos Ubuntu y Lubuntu demostró ser útil para comprender y optimizar el rendimiento de la red.

La metodología empleada permitió la configuración efectiva de la red en GNS3, resaltando la asignación de direcciones IP, el enrutamiento de routers y la verificación de conexiones mediante protocolos de control de mensajes. La transmisión de video utilizando VLC mediante el protocolo UDP, en particular, ofreció una visión detallada de los desafíos y las posibilidades en la gestión de flujos multimedia.

El análisis exhaustivo de "traffic shaping" presentó variaciones significativas en el rendimiento de la red al modificar este parámetro, destacando su influencia crítica en el uso del ancho de banda y la optimización del flujo de datos. Las

capturas de Wireshark y el análisis detallado demostraron claramente cómo las diferentes configuraciones afectan la transmisión y la eficiencia general de la red.

Para finalizar, este estudio destaca la importancia de las herramientas específicas y las técnicas avanzadas para mantener un rendimiento óptimo en entornos de comunicación. La comprensión detallada del traffic shaping aporta claridad sobre su papel fundamental en la gestión del flujo de datos en una red. Las futuras investigaciones se podrían enfocar en abarcar la implementación en entornos empresariales complejos, la adaptación a redes definidas por software (SDN), la optimización de la transmisión de multimedia, el desarrollo de políticas de gestión de tráfico específicas, el fortalecimiento de la seguridad de la red, y la integración en sistemas automatizados como IoT y SDN, entre otras.

References

[1] "Gns3 - The software that empowers network professionals. Disponible en: Gns3. Accedido en: 10, Diciembre 2023.

[2] A. S. Tanenbaum, "Computer Networks," Online. Disponible en: Computer Networks. Accedido en: 10, Diciembre 2023.

[3] J. Kurose and K. Ross, "Computer Networking: A Top-Down Approach," Online. Disponible en: Computer networking. Accedido en: 10, Diciembre 2023.

[4] Alfonso Ehijo B., "Capa de Aplicaciones = Video Streaming y sockets UDP-TCP", Redes, Universidad de O'Higgins, Rancagua, 2023.

[5] Alfonso Ehijo B. , "Control en TCP y evolucion de Capa de Transporte", Redes, Universidad de O'Higgins, Rancagua, 2023.

[6] Videolan, "VLC media player," Online. Disponible en: VLC media player. Accedido en: 10, Diciembre 2023.

[7] Oracle, "Oracle VM VirtualBox," Online. Available: VirtualBox. Accedido en: 10, Diciembre 2023.