



# Análisis de QoS para IPTV en un entorno de redes definidas por *software*\*

Juan Camilo Caviedes Valencia\*\*  
Wilmar Campo-Muñoz\*\*\*  
Gabriel Elías Chanchi Golondrino\*\*\*\*

Recibido: 2018/10/18 • Aceptado: 2019/07/09  
<https://doi.org/10.22395/riium.v19n36a2>

## Resumen

El despliegue masivo de servicios soportados en la tecnología del video *streaming*, trae consigo la expansión de redes de datos y el crecimiento del ancho de banda necesario para transportar la información. Esto, a su vez, demanda mayores capacidades de control y gestión de red que permitan garantizar calidad de servicio (QoS) al usuario final. Tal demanda ha impulsado la innovación de las arquitecturas de red. Surgen así las redes definidas por *software* (SDN), paradigma de arquitectura de red donde se desacoplan los planos de datos y control. Este artículo analiza para uno de los servicios de video *streaming*, como lo es la televisión sobre el protocolo IP (IPTV), las métricas de QoS sobre una red SDN real y emulada. Los resultados demuestran que es posible igualar la QoS de una arquitectura convencional, ofreciendo otras ventajas a la red como la lógica centralizada y programabilidad desde el plano de gestión.

Palabras clave: IPTV; QoS; SDN; video *streaming*.

\* Artículo derivado de investigación titulada Banco de pruebas para el despliegue del servicio IPTV sobre una red WiFi, con código 839 de la Universidad del Quindío.

\*\* Magíster en Ingeniería de Telecomunicaciones. Consultor Telecom, Everis Colombia Ltda., Bogotá, Colombia. Correo electrónico: [jccaviedesv@uqvirtual.edu.co](mailto:jccaviedesv@uqvirtual.edu.co). Orcid: <http://orcid.org/0000-0003-2842-2515>

\*\*\* Doctor y magíster en Ingeniería Telemática. Profesor asistente, Universidad del Quindío, Armenia, Colombia. Correo electrónico: [wycampo@uniquindio.edu.co](mailto:wycampo@uniquindio.edu.co). Orcid: <http://orcid.org/0000-0001-8585-706X>

\*\*\*\* Doctor y magíster en Ingeniería Telemática. Profesor auxiliar, Universidad de Cartagena, Cartagena, Colombia. Correo electrónico: [gchanchig@unicartagena.edu.co](mailto:gchanchig@unicartagena.edu.co). Orcid: <http://orcid.org/0000-0002-0257-1988>

## QoS Analysis for IPTV in a Software Defined Network Environment

### **Abstract**

The massive deployment of services supported on the video streaming technology brings with itself the expansion of the data networks and the growth of the bandwidth necessary for carrying information. This demands greater control capabilities and network management that guarantees the quality of the service (QoS) for the final user. Such a demand has pushed the innovation of network architectures. Generating thus software-defined networks (SDN), network architecture paradigm where the data and control blueprints are uncoupled. This article analyzes the QoS metrics of a real and emulated SDN network for one of the video streaming services such as the television over the IP protocol (IPTV). The results show the possibility of equaling the QoS of a conventional architecture while offering other advantages for the network such as centralized logic and programmability at the management level.

**Keywords:** IPTV; QoS; SDN; video streaming.

## Análise de QoS para IPTV em um ambiente de redes definidas por software

### **Resumo**

O desenvolvimento massivo de serviços apoiados na tecnologia de streaming de vídeo traz consigo a expansão de redes de dados e o crescimento da largura da banda necessário para transportar a informação. Por sua vez, isso requer maior capacidade de controle e gestão de rede que permitam garantir a qualidade de serviço (QoS) ao usuário final. Essa demanda vem impulsionando a inovação das arquiteturas de rede. Assim, surgem as redes definidas por softwares (SDN), paradigma de arquitetura de rede em que se desassocia os planos de dados e controle. Este artigo analisa para um dos serviços de streaming de vídeo, como é a televisão sobre o protocolo IP (IPTV), as métricas de QoS sobre uma rede SDN real e similar. Os resultados demonstram que é possível igualar a QoS de uma arquitetura convencional, oferecendo outras vantagens à rede, como a lógica centralizada e a programação a partir do plano de gestão.

**Palavras-chave:** IPTV; QoS; SDN; *streaming* de vídeo

## INTRODUCCIÓN

El aumento del tráfico en las redes de datos, asociado a servicios digitales con contenido audiovisual, ha llevado a investigar cómo se reenvía la información con el propósito de mejorar, entre otras cosas, la eficiencia en el uso del canal [1-2]. La evolución de las arquitecturas de redes ha generado una evaluación de las nuevas tecnologías y arquitecturas, de manera que estas se adapten a las condiciones impuestas por los servicios y las expectativas del usuario durante su consumo. Un servicio en el que se ha centrado la atención es el video *streaming* debido a su alta demanda de ancho de banda [3]. Se espera que esta demanda siga creciendo de tal forma que, según [4], para el año 2022 de todo el tráfico en redes móviles, el 75 % corresponderá a información de video.

Las arquitecturas de red convencionales han sido el medio de transporte para servicios como el video *streaming*. Sin embargo, estas construcciones vuelven rígida la red porque cada fabricante crea su propio sistema operativo y *firmware*, lo que limita la visión global de la red y su flexibilidad debido a su lógica distribuida [5-6]. Por tal motivo, el despliegue masivo de servicios ha incentivado la investigación de las redes, cómo ejecutan las funciones de reenvío, control y gestión de la información. De esta situación surge un concepto que aprovecha los avances tecnológicos de los últimos años: las redes definidas por *software* SDN (*Software Defined Networking*). En general, SDN es un paradigma que promete romper la integración vertical de las redes tradicionales, separando la lógica de control de la red de los enrutadores y conmutadores subyacentes, promoviendo la centralización (lógica) del control de red e introduciendo la capacidad para programarla [2].

Los beneficios de SDN son múltiples, entre ellos destaca la lógica centralizada que permite desde un único controlador, decidir la forma como se reenviará la información en los dispositivos del plano de datos, otorgando a los gestores una visión global de la topología de red. Esta centralización y visión global, sumada a la programabilidad de la red, permite generar soluciones que aumenten la eficiencia de la red a partir de, por ejemplo, asignación dinámica de recursos que reducen el sobreaprovisionamiento sin desmejorar la QoS ofrecida [7]. Del mismo modo, el dinamismo en la red permite dar soluciones dedicadas a servicios de video *streaming*. En [8] se realiza una transmisión dinámica adaptativa a través de HTTP (DASH), donde se asigna el ancho de banda según condiciones de red de los terminales solicitantes del servicio bajo el protocolo TCP, mejorando el rendimiento a partir de mecanismos de asistencia de adaptación desplegados sobre SDN. Por otra parte, el uso de un protocolo común que comunique los planos de datos y control independientemente del proveedor de equipos, elimina los límites de interoperabilidad existentes en la implementación de redes de datos usando

técnicas de la década pasada. También, la capacidad intrínseca de la arquitectura SDN para detectar fallas de manera remota da una mejor orientación en función de la capacidad de respuesta ante comportamientos anómalos de la red [9-10].

Este artículo pretende demostrar cómo el SDN iguala la QoS que ofrece una red convencional mientras brinda ventajas a nivel de control y gestión como las mencionadas en el párrafo anterior. Mediante la implementación de tres escenarios de experimentación, formados tanto por redes convencionales como redes SDN (reales y emuladas), se despliega un servicio de video *streaming* como IPTV, usando dos protocolos de transmisión: RTSP (*Real Time Streaming Protocol*) y RTMP (*Real Time Messaging Protocol*). La elección de estos protocolos se basa en que son los más usados para el despliegue de servicios de video *streaming* [11]. Durante la transmisión con cada protocolo, se extrae información necesaria para calcular cinco métricas de QoS que son: retardo, *jitter*, rendimiento (*throughput*), tasa de pérdida de paquetes (PLR) y nivel de ruido aditivo blanco gaussiano (AWGN). A partir de los valores obtenidos en estas métricas, para cada escenario de experimentación se establecen las diferencias a nivel de QoS entre las arquitecturas convencionales y SDN. Además, se señalan las ventajas que ofrece SDN a nivel de control y gestión, mediante el análisis de las características de los elementos de red usados para su implementación. Adicionalmente, se evalúa la fiabilidad del entorno de emulación, al momento de emular SDN mediante el contraste de resultados obtenidos en este escenario y los obtenidos en la SDN real.

Finalmente, el artículo se divide en cuatro secciones. La primera presenta al SDN como paradigma de arquitectura de red y los escenarios de experimentación propuestos con los elementos de red que los componen, también pone en contexto IPTV y los elementos clave de esta tecnología que son tratados durante el desarrollo de la investigación; la sección 2 expone la implementación de los escenarios de experimentación; la sección 3 muestra los resultados obtenidos al desplegar el servicio sobre cada escenario y la sección 4, presenta las conclusiones obtenidas.

## 1. MATERIALES Y MÉTODOS

La arquitectura SDN genérica está conformada por un plano de gestión, uno de control y otro de datos, que se coordinan para que la red tenga dinamismo y facilidad de control. Esta arquitectura se muestra en la figura 1.

Por otra parte, la implementación física de una arquitectura SDN tiene en cuenta un elemento de red por cada plano, esto es, existe un dispositivo en el plano de datos que se comunica con un elemento de control, normalmente un *software*, el cual, a su vez, es orquestado y programado mediante aplicaciones de red ubicadas en el plano de gestión. En consecuencia, el dispositivo ubicado en el plano de datos debe soportar

OpenFlow, ya que mediante este protocolo, el dispositivo es capaz de comunicarse con el controlador de red. Así mismo, las arquitecturas basadas en SDN requieren aplicaciones de gestión que interactúen con el controlador de manera que se programen tareas que simplifiquen operaciones de reenvío, como el trazo de rutas óptimas y/o la habilitación de servicios bajo demanda. En adición, el controlador también se comunica con aplicaciones que permiten el diseño de tablas de flujo, la extracción de estadísticas de red y la visualización de la topología [2].

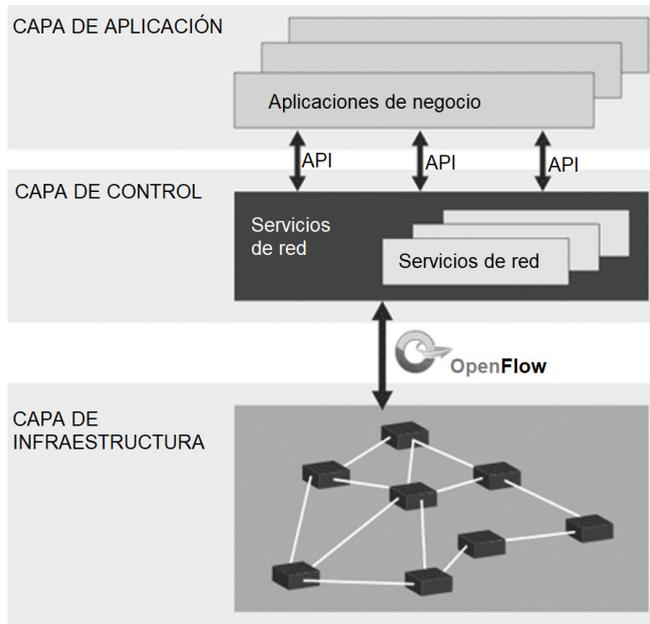


Figura 1: Arquitectura SDN [12].

Fuente: Adaptado de ONF, Software-Defined Networking (SDN) Definition [12]

Para la implementación de los escenarios SDN, específicamente la SDN real, se usa el conmutador Zodiac FX en el plano de datos, el controlador OpenDaylight (ODL) en el plano de control y la aplicación OpenFlow Manager (OFM) en el plano de gestión. Hay que destacar que el Zodiac FX es un conmutador de bajo costo diseñado con fines académicos e investigativos en el ámbito de SDN; este se encuentra habilitado para soportar el protocolo OpenFlow en sus versiones 1.0 a 1.4 [13]. De hecho, el controlador ODL soporta OpenFlow en cualquier versión y es compatible con la aplicación OFM [14-15]. Específicamente, OFM es una parte del controlador SDN de Cisco (Open SDN Controller) que permite el diseño de tablas de flujo, recopilación de estadísticas, obtención de una visión global de la topología de red y gestión de recursos a partir de los tipos de servicios que se despliegan en la red. En otra instancia, para el escenario SDN emulado se usa Mininet, conectado a ODL y OFM como entorno de emulación [16].

## 1.1 IPTV

Es una tecnología digital de distribución y difusión de televisión y video/audio bajo demanda sobre redes de computadores de banda ancha, que alcanza como mínimo las condiciones de QoS de la televisión tradicional por cable. Específicamente, IPTV está soportada sobre el protocolo de Internet IP para la distribución de canales de televisión, películas, texto, gráficos, datos o contenido de video y audio bajo demanda en una red propietaria. Se ha demostrado que usando SDN como tecnología subyacente, es posible garantizar y dinamizar la QoS de extremo a extremo sin perder de vista la supervisión de recursos. Esto logra brindar soluciones a nivel práctico, garantizando los valores de referencia de las métricas de QoS establecidos por la UIT (Unión Internacional de Telecomunicaciones) para el transporte de televisión digital [17-18].

La tabla 1 muestra los valores de referencia establecidos por la UIT para tres de las cinco métricas de QoS consideradas en esta investigación. Las otras dos métricas no tienen un punto de comparación, dado que en la práctica son dependientes del acuerdo de nivel de servicio, como es el caso del rendimiento, o pueden estar relacionadas con el proveedor de contenido, como lo es el nivel de ruido. Sin embargo, una variación del rendimiento puede afectar el valor de las métricas referenciadas, mientras que una fluctuación del nivel de ruido puede inferirse de la variación en la pérdida de paquetes. Hay que esclarecer que el nivel de ruido es una métrica medida a nivel de aplicación.

Tabla 1. Valores de referencia para las métricas de QoS [19]

<i>Métrica de QoS</i>	<i>Objetivo de desempeño</i>	<i>Valor de referencia</i>
Retardo	Límite superior del promedio de valores del retardo	100 ms
<i>Jitter</i>	Límite superior en el mayor cuartil menos el mínimo valor	50 ms
PLR	Límite superior de la pérdida de paquetes	0,1 %

Fuente: elaboración propia

## 2. ESCENARIOS DE EXPERIMENTACIÓN

Los escenarios construidos se basan en una infraestructura de red genérica compuesta por un servidor, un dispositivo de reenvío y dos clientes que consumen el servicio y generan contienda por el acceso al medio. La figura 2 muestra la arquitectura funcional genérica de experimentación.

A continuación, se presentan y describen cada una de las arquitecturas desplegadas en los escenarios de experimentación.

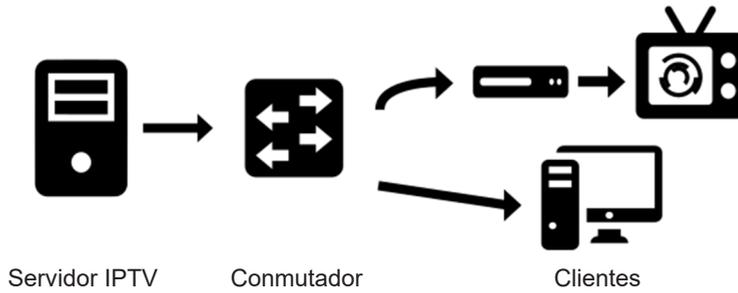


Figura 2. Arquitectura funcional genérica de experimentación  
Fuente: elaboración propia

## 2.1 Red convencional

Las infraestructuras de redes convencionales poseen dispositivos de reenvío cuyos planos de datos y control se encuentran en el mismo dispositivo. La figura 3 muestra el escenario de experimentación de red convencional.

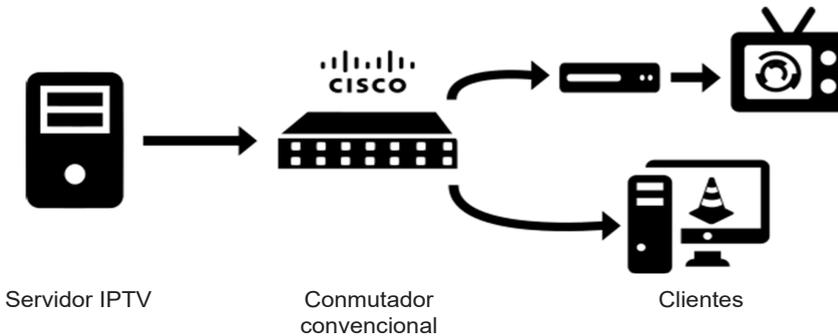


Figura 3. Escenario de red convencional  
Fuente: elaboración propia

El despliegue del servicio IPTV sobre este escenario proporciona el punto de comparación para los resultados obtenidos con SDN. En detalle, el escenario de red convencional está compuesto por el servidor Wowza, un conmutador Cisco Catalyst 2950-24 controlado por un *firmware* Cisco IOS c2950-c3h2s y dos clientes: un computador y un STB (Set-Top Box) en el caso de RTSP y dos computadores en el caso de RTMP.

## 2.2 Red SDN real

En el plano de datos se usa un conmutador habilitado para OpenFlow; para el plano de control un dispositivo que hospeda el *software* de control y está conectado con el

conmutador, el plano de gestión requiere de un *software* (puede estar hospedado en el mismo dispositivo que hospeda el de control) que contiene las aplicaciones de gestión de red para orquestar su comportamiento.

Como se menciona en la sección 1, el controlador de red que integra los escenarios de red SDN (real y emulado) es ODL, mientras que la aplicación de gestión es OFM; ambos comunicados con el Zodiac FX. La figura 4 muestra los elementos de red necesarios para el despliegue del servicio IPTV en la SDN real. La principal diferencia entre un entorno de red convencional y uno SDN es la abstracción del plano de control que se hace con SDN en un dispositivo remoto y centralizado denominado controlador de red. Por consiguiente, en este escenario se logra observar que el controlador y la aplicación de gestión se encuentran desacoplados del conmutador Zodiac FX.

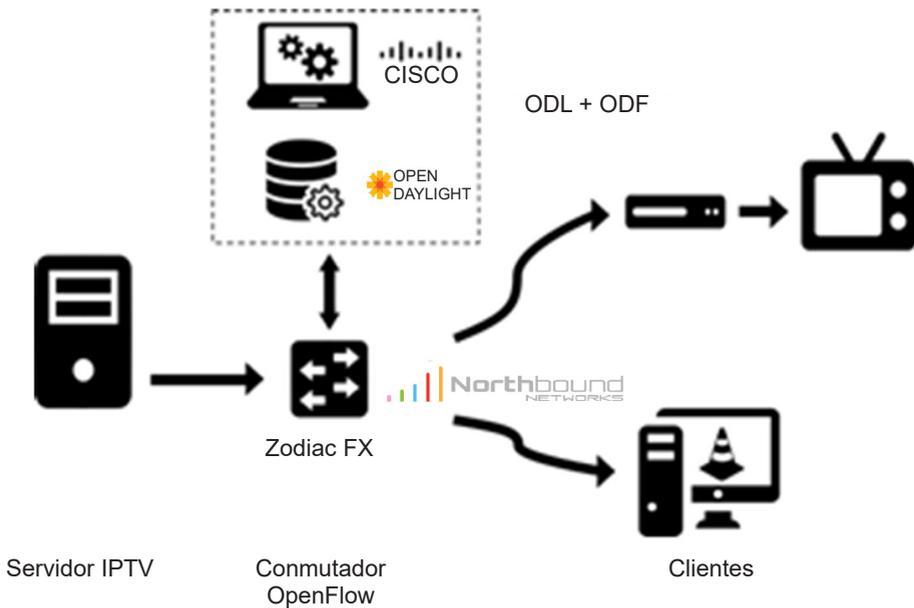


Figura 4. Escenario de red SDN real

Fuente: elaboración propia

De hecho, el transporte adecuado de la información requiere la instalación de tablas de flujo en el conmutador de manera que sus puertos logren comunicarse entre sí de forma bidireccional. Esta función es realizada en OFM mediante el diseño de las tablas de flujo que ODL instala en el Zodiac FX. Básicamente, la arquitectura SDN se implementa en dos dispositivos: el Zodiac FX y el computador que hospeda los *softwares* de control y gestión.

La figura 5 muestra cómo desde ODL se puede ver la topología de red en el plano de datos, así como características de los dispositivos que la componen (dirección IP, MAC y etiquetas de ODL). Desde el plano de gestión, la figura 6 evidencia la conexión de OFM con el Zodiac FX y con las estadísticas recopiladas tales como paquetes transmitidos y recibidos; la figura 7 muestra una de las tablas de flujo diseñadas en OFM e instaladas en el Zodiac FX para lograr la conexión bidireccional entre sus puertos.

Statistics	Host		
<b>openflow:123917682136665</b> <b>[World's smallest OpenFlow switch!][Zodiac-FX Rev.A]</b>			
Device	Id	PacketsTransmitted	PacketsReceived
openflow:1 23917682 136665	openflo w:1239 17682 13666 5:1	43066	11593
openflow:1 23917682 136665	openflo w:1239 17682 13666 5:3	21716	34121
openflow:1 23917682 136665	openflo w:1239 17682 13666 5:2	46453	7951

Figura 5. Topología de red cargada desde ODL

Fuente: elaboración propia

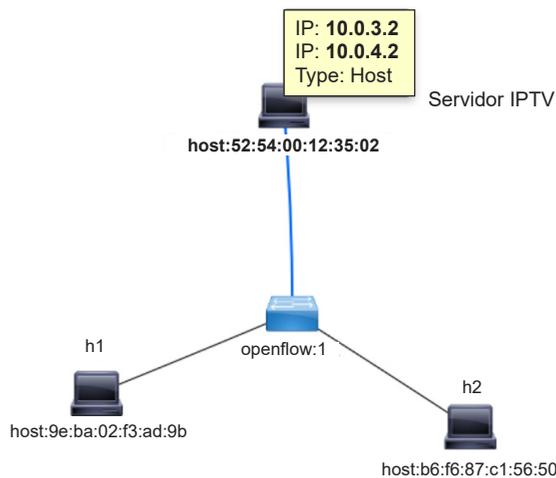


Figura 6. Estadísticas de la red SDN real desde OFM

Fuente: elaboración propia

```
Capri(openflow)# show flows
-----
Flow 1
Match:
  In Port: 1
Attributes:
Table ID: 0          Cookie:0x0
Priority: 32750      Duration: 12198 secs
Hard Timeout: 0 secs  Idle Timeout: 0 secs
Byte Count: 4820542  Packet Count: 11456
Last Match: 00:05:02
Instructions:
Apply Actions:
  Output: FLOOD
```

Figura 7. Tabla de flujo instalada en el conmutador OpenFlow  
Fuente: elaboración propia

### 2.3 Red SDN emulada

Este escenario se emula con Mininet [9]. La figura 8 muestra los elementos de red que componen la red emulada. Esta red contempla la implementación de ODL y OFM con el propósito de conservar la equivalencia entre escenarios. Justamente, la imposibilidad de emular un STB por parte de Mininet, conlleva al uso de dos computadores emulados como clientes como se observa en la figura 8.

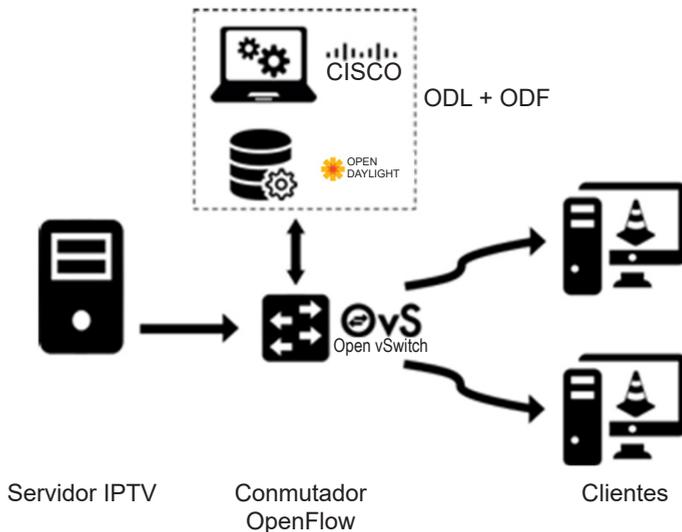


Figura 8. Escenario de red SDN emulada  
Fuente: elaboración propia

Inicialmente, los clientes emulados carecen de conexión con cualquier otro dispositivo diferente al controlador por defecto de Mininet (NOX) y otros clientes. En el caso de la red emulada, los clientes deben conectarse con ODL y OFM para conservar la equivalencia entre escenarios. Para lograrlo, se diseña un código en Python que usa librerías y paquetes disponibles luego de la instalación de Mininet. El código se presenta en la figura 9.

```
#!/usr/bin/python

from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController
from mininet.cli import CLI
from mininet.link import Intf
from mininet.log import setLogLevel, info

def myNetwork():

    net = Mininet ( topo=None,
                  build=False)

    info< '** Agregar controlador **\n' )
    net.addController(name='c0', controller=RemoteController, ip='192.168.1.12', port=6633)

    info< '** Agregar conmutador **\n' )
    s1 = net.addSwitch<'s1' )
    Intf( 'eth1', node=s1 )
    Intf( 'eth2', node=s1 )

    info< '** Agregar hosts **\n' )
    h1 = net.addHost('h1' , ip='0.0.0.0')
    h2 = net.addHost('h2', ip='0.0.0.0')

    info< '** Agregar links **\n' )
    net.addLink(h1, s1)
    net.addLink <h2, s1)

    info( '** Iniciando la red **\n' )
    net.start0
    h1.cmdPrint('\dhcpclient '+h1.defaultIntf().name)
    h2.cmdPrint('\dhcpclient '+h2.defaultIntf().name) CLI(net)
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' ) myNetwork()
```

Figura 9. Código fuente de la red SDN emulada en Mininet

Fuente: elaboración propia

En general, el código le indica a Mininet que los clientes emulados se conectan a un controlador remoto, alojado en una dirección IP determinada. Luego, se crea el conmutador virtual, los clientes y los enlaces entre estos y, por último, se configuran las direcciones IP de los clientes mediante un servidor DHCP (*Dynamic Host Configuration Protocol*). En contraste, la conexión entre OFM y ODL se garantiza mediante la configuración de un script de la aplicación de gestión. De este modo se asegura el puente entre el plano de gestión y el plano de datos.

Con los escenarios de experimentación implementados, se procede a desplegar el servicio IPTV en cada uno de ellos con el propósito de extraer, evaluar y comparar las métricas de QoS tomadas a consideración.

### 3. RESULTADOS

En cada uno de los escenarios de experimentación se desplegó el servicio IPTV usando como archivo de prueba *Big Buck Bunny* en formato mp4 y resolución 512x288 [21]. Entre las características más importantes a considerar está el rendimiento (*throughput*) promedio requerido cuando se transmite el video y el número de tramas que lo componen. La primera característica tiene un valor teórico de 640 kbps y la segunda de 19,039 tramas. Considerando lo anterior, el resto de esta sección presenta los resultados obtenidos al analizar cada métrica de QoS individualmente, basados en el escenario de experimentación y protocolo de transmisión usado.

#### 3.1 Retardo

La medición del retardo usando RTMP muestra uniformidad en los tres escenarios de experimentación. En cada uno de estos, el retardo varía entre los 33 y 34 ms específicamente. Hay que destacar que a diferencia de RTSP, RTMP es un protocolo orientado a conexión, por lo cual existe la posibilidad de reenviar paquetes del servidor si alguno se pierde durante el transporte. La figura 10 muestra el retardo en los tres escenarios de experimentación.

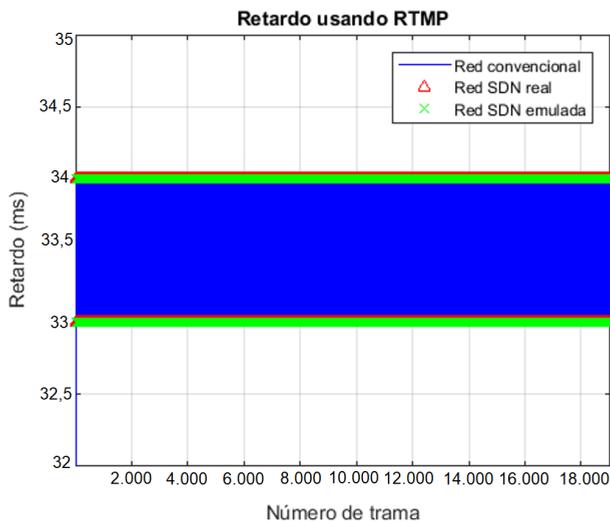


Figura 10. Retardo en escenarios con RTMP

Fuente: elaboración propia

Se puede notar en la figura 10 que en todos los escenarios de experimentación se cumplen los límites sugeridos por la UIT (ver tabla 1). Así mismo, cuando se hace un acercamiento al retardo, como en la figura 11, se observa un comportamiento repetitivo en las tramas de video: dos tramas poseen un retardo de 33 ms seguidas de una tercera trama con un retardo de 34 ms.

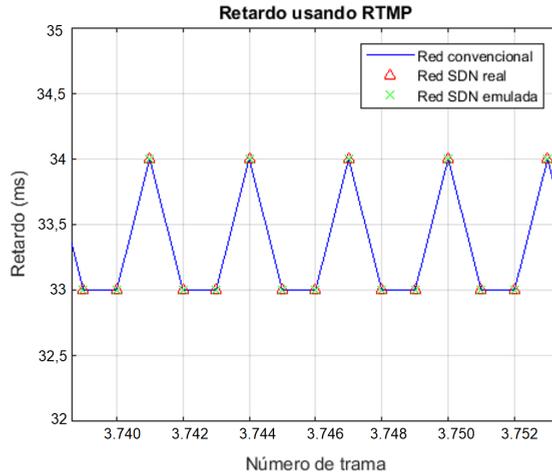


Figura 11. Acercamiento al retardo usando RTMP

Fuente: elaboración propia

Este resultado es esperado puesto que el video fue grabado a 30 cuadros por segundo. Dado que el video completo dura 634,6 segundos, este está compuesto por 19.039 cuadros aproximadamente, número que coincide con la cantidad de tramas de video que son enviadas por el servidor. Así pues, se puede decir que por cada trama de video hay un cuadro que se visualiza en pantalla en un instante de tiempo. Para mostrar 30 cuadros en un segundo, cada trama de video debe llegar con un retardo promedio de 33,3 ms. Al analizar el patrón de llegada de RTMP, el retardo promedio es un ponderado entre las dos tramas que se retardan 33 ms y una tercera que se retarda 34 ms. Por consiguiente,  $33(2/3) + 34(1/3) = 33,3$  ms, resultado que concuerda con el esperado teóricamente.

De igual forma, el despliegue del servicio IPTV sobre RTSP muestra un comportamiento no uniforme en el retardo medido por cada trama recibida. Para el escenario de red convencional, no existen tramas que superen el valor referenciado de 100 ms propuesto por la UIT. En la red SDN real, el 0,07222 % de las tramas superan el umbral de 100 ms. En la red SDN emulada, cerca del 0,3934 % de las tramas recibidas están por encima del umbral. Este último resultado es 5,5 veces mayor al obtenido en la red SDN real. La figura 12 muestra el retardo medido en cada escenario usando RTSP.

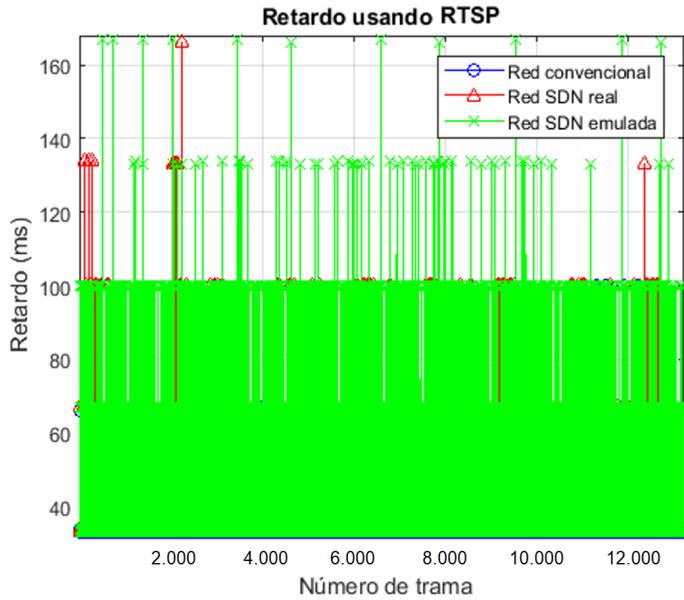


Figura 12. Retardo en escenarios con RTSP  
Fuente: elaboración propia

Cuando se hace un acercamiento a la figura 12 se observa que el comportamiento del retardo no es uniforme y presenta una variación de más de 100 ms en algunas tramas, especialmente de la SDN emulada, como se muestra en la figura 13.

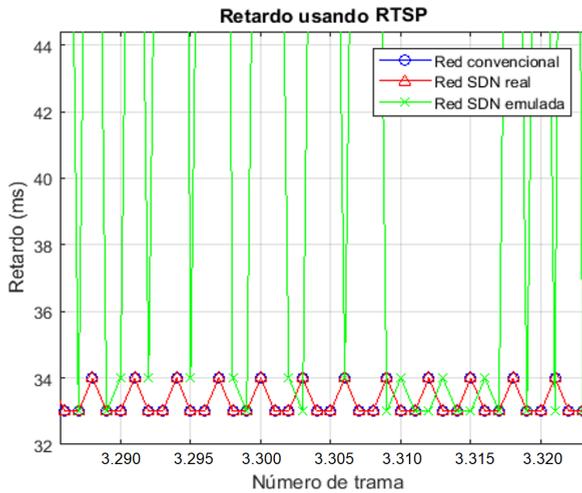


Figura 13. Acercamiento gráfico al retardo usando RTSP  
Fuente: elaboración propia

Sin embargo, la variación al retardo o *jitter*, no representa un problema en la reproducción del video en pantalla siempre y cuando no exceda los 1.200 ms. Este valor es configurado en el *búfer* receptor del reproductor multimedia de cada cliente en todos los escenarios. Se infiere así que el *búfer* absorbe los efectos del *jitter*.

### 3.2 Variación del retardo

La medición del *jitter* cuando se despliega el servicio IPTV usando RTMP, es uniforme para cada escenario. De lo expuesto en las figuras 10 y 11, se deduce que el máximo *jitter* obtenido en cualquier escenario es de 1 ms, valor que está por debajo de los 50 ms propuestos por la UIT [19]. Por otra parte, el *jitter* promedio obtenido en cada escenario se calcula a partir de la ecuación (1).

$$\overline{Jitter} = \frac{jitterSum}{rxPackets - 1} \quad (1)$$

En la ecuación (1), *jitterSum* representa la suma del *jitter* de todas las tramas de video recibidas y *rxPackets* es el número de paquetes recibidos. Con base en lo anterior, la tabla 2 muestra el *jitter* promedio en cada escenario.

Tabla 2. *Jitter* promedio usando RTMP

<i>Escenario</i>	<i>Jitter promedio (ms)</i>
Convencional	0,667
SDN real	0,6664
SDN emulada	0,6667

Fuente: elaboración propia

La tabla 2 demuestra uniformidad en el comportamiento del *jitter* para cada escenario. En complemento a la medición promedio, la figura 14 muestra el histograma de los valores obtenidos para el *jitter* en cada escenario.

De la figura 14 se observa que el 66,6 % de las tramas recibidas tienen un *jitter* de 1 ms mientras que las tramas restantes carecen de variación al retardo. Es de anotar que en todos los escenarios se obtiene el mismo resultado.

Con RTSP el panorama es diferente. En el escenario convencional, el 0,1944 % de las tramas superan los 50 ms sugeridos por la UIT. En el escenario SDN real, la cifra asciende a 0,975 % que, comparado con la red convencional, representa una diferencia de 148 tramas tomando como base las 19.039 tramas enviadas por el servidor. Del mismo modo, en la red SDN emulada el 13,6708 % de las tramas no cumplen con el

valor sugerido por la UIT, valor 14 veces mayor que el obtenido en la red SDN real. La tabla 3 muestra el *jitter* promedio en cada escenario.

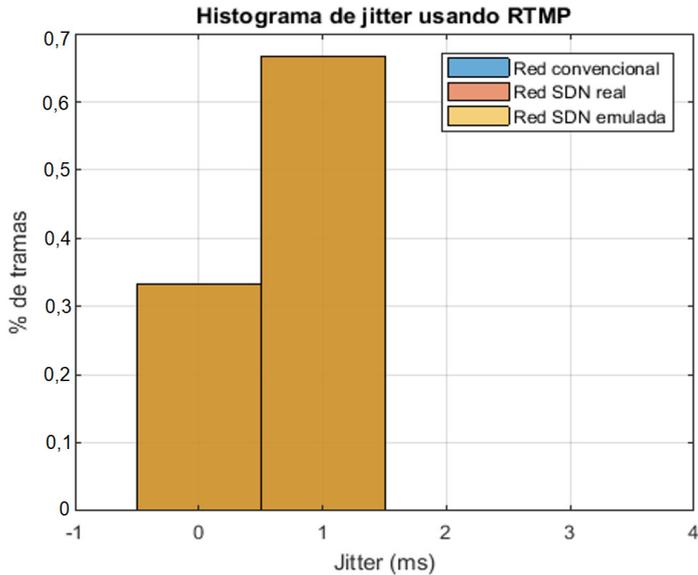


Figura 14. Histograma de *jitter* RTMP medido en cada escenario

Fuente: elaboración propia

Tabla 3. Jitter promedio usando RTSP

Escenario	Jitter promedio (ms)
Convencional	1,0072
SDN real	1,6022
SDN emulada	22,4619

Fuente: elaboración propia

Nuevamente, el *jitter* promedio no brinda mucha información que permita deducir el comportamiento del tráfico. Por esto, en la figura 15 se muestra el histograma calculado para cada escenario de experimentación.

Se observa cómo la red SDN emulada presenta un mayor porcentaje de tramas por encima del umbral sugerido, llegando a una variación del retardo de hasta 100 ms. Sin embargo, independientemente del escenario de experimentación, el *búfer* absorbe los efectos del *jitter*.

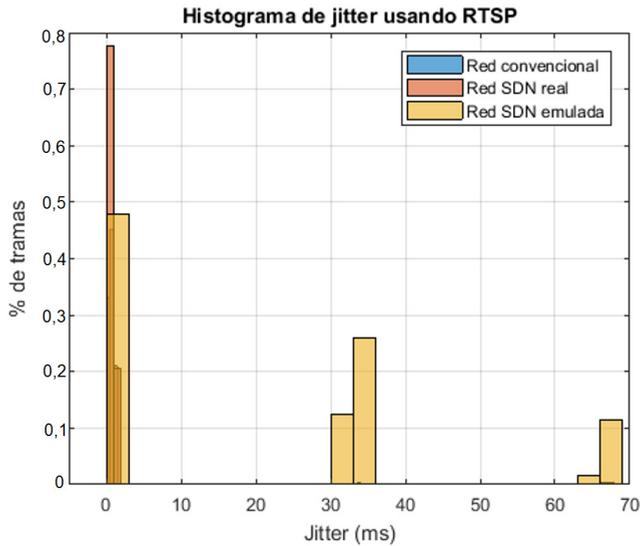


Figura 15. Histograma de *jitter* RTSP medido en cada escenario  
Fuente: elaboración propia

### 3.3 Rendimiento

El rendimiento no es una métrica reglamentada o sugerida por la UIT, pero su variación puede afectar la QoS recibida en un servicio de video *streaming*. Cuando se usa RTMP el rendimiento obtenido en todos los escenarios es semejante, como se evidencia en la figura 16.

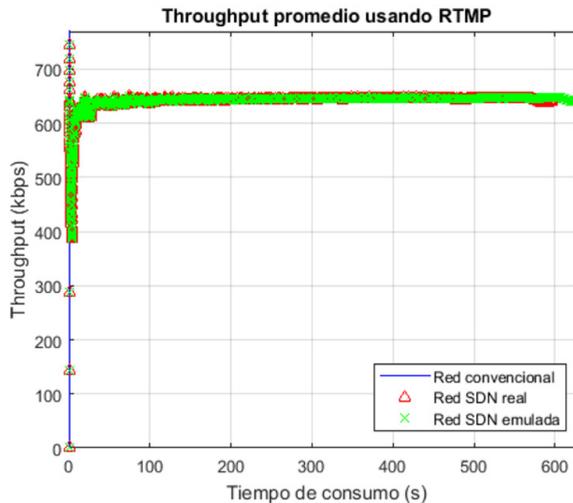


Figura 16. Rendimiento promedio usando RTMP  
Fuente: elaboración propia

Los tres escenarios de experimentación convergen al mismo valor para el rendimiento promedio, 640 kbps, que coincide con el valor teórico de transmisión del video de prueba usado. No obstante, con RTSP se nota una diferencia mayor entre escenarios, como muestra la figura 17 ante la medición realizada.

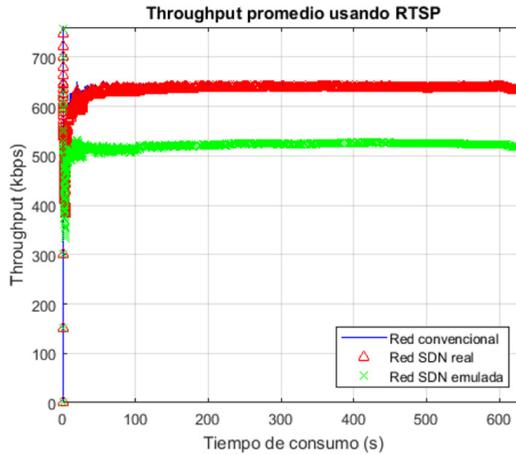


Figura 17. Rendimiento promedio usando RTSP

Fuente: elaboración propia

Claramente, el escenario de peor desempeño es la SDN emulada. Este sufre una degradación del rendimiento debido a la alta PLR que presenta, como se explica en la sección 4.4.

### 3.4 PLR

Al observar la tabla 1 vemos que el valor de referencia de la PLR es de 0,1 %. Al medir la PLR de cada uno de los escenarios, se obtiene como resultado los valores presentados en la tabla 4.

Tabla 4. PLR medida en cada escenario

Escenario	Protocolo	PLR (%)
Convencional	RTMP	0
	RTSP	0,5568
SDN real	RTMP	5,8617
	RTSP	1,4129
SDN emulada	RTMP	0
	RTSP	30,5741

Fuente: elaboración propia

En todos los escenarios se evidencia una diferencia en la PLR medida de acuerdo al protocolo usado. Es importante aclarar que RTMP es un protocolo orientado a conexión, pues usa TCP como protocolo de transporte. En cambio, RTSP usa UDP que es un protocolo no orientado a conexión. Esa diferencia genera una ventaja sustancial a nivel de PLR en RTMP, que se refleja en dos de los escenarios: convencional y SDN emulada. En SDN real, se obtiene un valor de 5,8617 % que supera el valor de referencia sugerido. A pesar de esta ventaja hay que considerar una diferencia a nivel de *hardware* con los demás escenarios: el conmutador Zodiac FX está diseñado para fines académicos e investigativos. Cuando ambos se disputan el acceso al medio, las características del Zodiac FX degradan la QoS. Durante la experimentación se comprobó que cuando se desconecta uno de los clientes de sus puertos para evitar esto, la PLR disminuye hasta un 0,08 % en el mejor de los casos. Visualmente, en RTSP, la pérdida de paquetes se manifiesta como una degradación de la imagen mostrada en pantalla. La figura 18 muestra el efecto mencionado.

Sumado a esto, la PLR de 30,5741 % en la SDN emulada usando RTSP es el peor de los casos. Esta cifra indica que solo se recibieron 13.218 tramas de video de las 19.039 enviadas. El resultado derivado es poco más de cinco veces mayor al obtenido en la red SDN real, exponiendo el impacto en el uso de RTSP, igualmente el rendimiento disminuye proporcionalmente al aumento de la PLR. Lo anterior se explica en el hecho de recibir menos paquetes en el mismo periodo de tiempo comparado con los demás escenarios.



Figura 18. Efectos de la pérdida de paquetes

Fuente: elaboración propia

### 3.5 Nivel de ruido

La medición del nivel de ruido en cada uno de los escenarios se hace considerando la siguiente condición: para que los resultados de la medición en las imágenes emitidas y recibidas sean equivalentes entre escenarios, se selecciona un intervalo de 20 segundos sobre los cuales se estima el nivel de ruido en 600 fotogramas. Este intervalo es una

muestra que determina el comportamiento global del ruido en el corto animado. Cabe destacar que el nivel de ruido se estima haciendo uso del algoritmo de Tanaka [22]. Así pues, la figura 19 muestra el nivel de ruido en los escenarios usando RTMP.

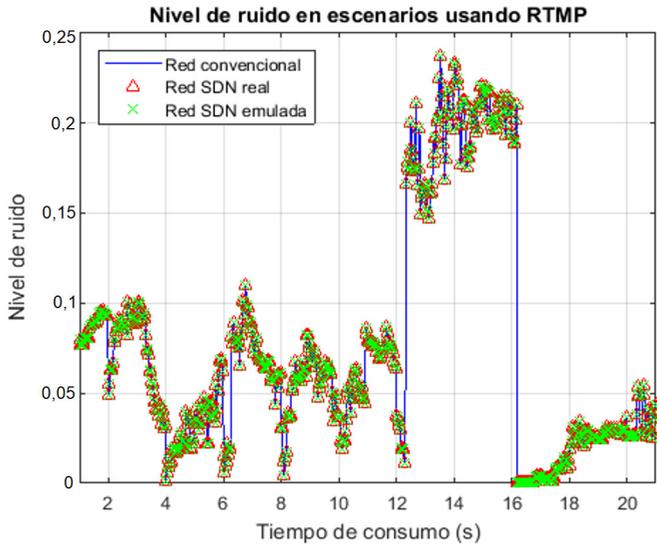


Figura 19. Nivel de ruido en escenarios de experimentación con RTMP  
Fuente: elaboración propia

Cuando se usa RTMP, la variación del nivel de ruido en las imágenes es independiente de la arquitectura de red como se muestra en la figura 19, donde las gráficas se encuentran superpuestas sobre una misma curva. Por otra parte, la figura 20 muestra el nivel de ruido en los escenarios usando RTSP.

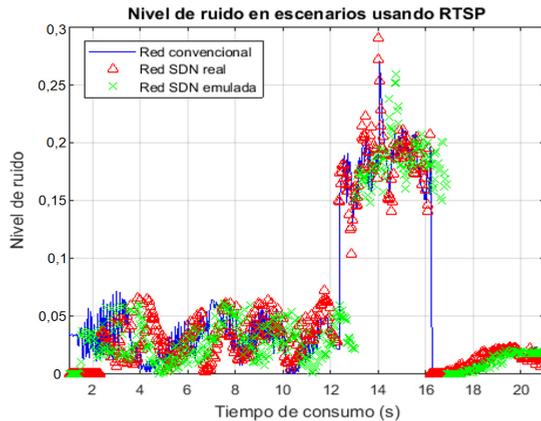


Figura 20. Nivel de ruido en escenarios de experimentación con RTSP  
Fuente: elaboración propia

Como se muestra en la figura 20, el nivel de ruido depende de la arquitectura de red. Se puede observar cómo entre los primeros 11 segundos del consumo del servicio, el nivel de ruido se mantiene en niveles inferiores a 0,1, caso que no sucede cuando se usa RTMP. De igual manera, el máximo nivel de ruido alcanzado al usar RTMP es inferior al máximo nivel alcanzado usando RTSP. En este último caso se alcanza un valor próximo a 0,3, mientras que con RTMP no se supera en cualquier escenario el 0,25.

#### 4. CONCLUSIONES

Entre los protocolos de transmisión usados, RTMP ofrece un mayor control sobre el envío de la información. Esto se refleja en métricas de QoS como el retardo y el *jitter*, donde los resultados obtenidos independientes de la arquitectura de red, demuestran que con este protocolo se logra una mayor estabilidad en la entrega de contenido multimedia. De igual modo, esa estabilidad se observa en la PLR y el nivel de ruido de las imágenes recibidas. El primer caso demuestra la importancia del desempeño de los dispositivos de red usados ya que, si bien la red convencional y la SDN emulada presentaron una PLR de 0 %, la red SDN real se vio expuesta a las desventajas del Zodiac FX cuando se disputa el acceso al medio. En el segundo caso el nivel de ruido no varía entre escenarios de experimentación usando RTMP.

En RTSP se generan inconvenientes en cuatro métricas de QoS. El retardo de la transmisión en los escenarios SDN superó el umbral de 100 ms establecido por la UIT en menos del 1 % de las tramas recibidas. En la red convencional no hubo tramas que superaran el umbral. También, el *jitter* se vio afectado por los valores atípicos obtenidos en el retardo. Por un lado, en la red convencional se alcanzaron valores superiores a los 50 ms de referencia en por lo menos 0,1944 % de las tramas recibidas. Por otra parte, en la red SDN real se obtuvieron valores mayores a dicho umbral en el 0,975 % de las tramas. Sin embargo, estos porcentajes representan tramas cuyo *jitter* es cerca de nueve veces inferior a los 1.200 ms configurados en el *búfer* receptor. Lo anterior indica que los efectos del *jitter* durante la reproducción del video son absorbidos. Una variación más significativa puede apreciarse con la PLR y el nivel de ruido. En la PLR se logró ver cómo ningún escenario está dentro de los límites sugeridos por la UIT, siendo la red convencional la de mejor desempeño. Consecutivamente, en el nivel de ruido se observa que en promedio, con RTSP se consiguieron valores inferiores a los obtenidos con RTMP pero con un valor pico mayor.

En el caso del rendimiento, se deduce que esta métrica es independiente de la arquitectura de red cuando se usa RTMP. Para RTSP existe una diferencia significativa entre los escenarios convencional y SDN real, con la SDN emulada. Lo anterior relacionado con la alta PLR obtenida durante la experimentación en el escenario emulado. Cabe

añadir que una variación en el rendimiento depende en mayor medida de la calidad que desea recibir el usuario y la capacidad de servicio del proveedor. En otras palabras, el rendimiento se relaciona explícitamente con la tasa de codificación del video, tamaño de la imagen, protocolo de transmisión usado por el servidor, características del equipo receptor, velocidad de transmisión de información y ancho de banda disponible.

Finalmente, respecto a las arquitecturas de red y sus desempeños en general, SDN equipara las prestaciones ofrecidas por una arquitectura convencional a nivel de QoS, es decir, no mejora, pero tampoco empeora el comportamiento del servicio si se tienen en cuenta únicamente las métricas de QoS consideradas en esta investigación. Empero, SDN ofrece otras ventajas a nivel de control y gestión. Como puede detallarse en la sección 3 de resultados, es posible tener una visión global de la topología red y de las características de los elementos que la componen.

## REFERENCIAS

- [1] J. Bailey y S. Stuart, “Faucet: Deploying SDN in the enterprise”, *Commun. ACM*, [En línea], vol. 60, n.º 1, pp. 45–49, Dec. 2016. DOI: <https://dl.acm.org/citation.cfm?doid=3028256.3009828>
- [2] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, y S. Uhlig, “Software-Defined Networking: A Comprehensive Survey”, *Proc. IEEE*, [En línea], vol. 103, n.º 1, pp. 14–76, Jan. 2015. DOI: 10.1109/JPROC.2014.2371999
- [3] Cisco, “Cisco Visual Networking Index: Forecast and Methodology, 2015-2020”, *Forecast Methodol.*, p. 22, 2015.
- [4] Ericsson, “Ericsson Mobility Report November 2018”, 2018. Estocolmo: Ericsson. Disponible: <https://www.ericsson.com/en/mobility-report>
- [5] R. Jain y S. Paul, “Network virtualization and software defined networking for cloud computing: A survey”, *IEEE Communications Magazine*, [En línea], vol. 51, n.º 11, pp. 24–31, Nov-2013. DOI: 10.1109/MCOM.2013.6658648
- [6] A. Tong y K. Wade, *NFV and SDN Guide for Carriers and Service Providers*, 1st ed. Blueplanet, 2017, 36 p.
- [7] D. L. C. Dutra, M. Bagaa, T. Taleb, and K. Samdanis, “Ensuring End-to-End QoS Based on Multi-Paths Routing Using SDN Technology”, in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, [En línea], pp. 1–6, 2017, DOI: 10.1109/GLOCOM.2017.8254076
- [8] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, “Delivering stable high-quality video: an SDN architecture with DASH assisting network elements”, in *Proceedings of the 7th International Conference on Multimedia Systems - MMSys '16*, [En línea], pp. 1–10, 2016, DOI: <https://dl.acm.org/citation.cfm?doid=2910017.2910599>
- [9] S. Ramakrishnan, X. Zhu, F. Chan, and K. Kambhatla, “SDN Based QoE Optimization for HTTP-Based Adaptive Video Streaming”, in *2015 IEEE International Symposium on Multimedia (ISM)*, [En línea], pp. 120–123, 2016, DOI: 10.1109/ISM.2015.53

- [10] J. Wang, C. de Laat, and Z. Zhao, “QoS-aware virtual SDN network planning”, in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, [En línea], pp. 644–647, 2017, DOI: 10.23919/INM.2017.7987350
- [11] P. L. Agudelo, W. Y. Campo, A. Ruíz, J. L. Arciniegas, and W. J. Giraldo, “Architectonic Proposal for the Video Streaming Service Deployment Within the Educational Context”, Springer, Cham, [En línea], pp. 313–326, 2017, DOI: [https://link.springer.com/chapter/10.1007/978-3-319-66562-7\\_23](https://link.springer.com/chapter/10.1007/978-3-319-66562-7_23)
- [12] D. Maldonado, *Diseño e implementación de una aplicación de red bajo la arquitectura SDN*, Bogotá:Universidad Javeriana, 2014.
- [13] J. Alcorn, S. Melton, and C. E. Chow, “SDN On-The-Go (OTG) physical testbed”, in *2017 IEEE Conference on Dependable and Secure Computing*, 2017, [En línea], pp. 202–208. DOI: 10.1109/DESEC.2017.8073808
- [14] M. Aziz, H. A. Fazely, G. Landi, D. Gallico, K. Christodoulopoulos, and P. Wieder, “SDN-enabled application-aware networking for data center networks”, in *2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2016, [En línea], pp. 372–375. DOI: 10.1109/ICECS.2016.7841210
- [15] L. Wang, H. Ruan, H. Li, and Q. Liu, “Combining Neutron and OpenDaylight for Management of Networking”, in *2017 IEEE 41st Annual Computer Software and Applications Conference (Compsac)*, 2017, [En línea], pp. 457–462. DOI: 10.1109/COMPSAC.2017.89
- [16] M. Erel, E. Teoman, Y. Ozcevik, G. Secinti, and B. Canberk, “Scalability analysis and flow admission control in mininet-based SDN environment”, in *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, 2015, [En línea], pp. 18–19. DOI: 10.1109/NFV-SDN.2015.7387396
- [17] Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani, “Internet Protocol Television (IPTV): The Killer Application for the Next-Generation Internet”, *IEEE Commun. Mag.*, [En línea], vol. 45, n.º 11, pp. 126–134, Nov. 2007, DOI: 10.1109/MCOM.2007.4378332
- [18] R. Mohammadi, R. Javidan, and M. Keshtgari, “OpenIPTV: a comprehensive SDN-based IPTV service framework”, *Multimed. Syst.*, [En línea], vol. 24, n.º 3, pp. 313–325, Jun. 2018, DOI: <https://doi.org/10.1007/s00530-017-0553-x>
- [19] UIT, “Network performance objectives for IP-based services”, [En línea], Disponible: <https://www.itu.int/rec/T-REC-Y.1541/en>, 2011.
- [20] UIT, “IPTV Focus Group Proceeding”, [En línea], Disponible: <http://www.itu.int/pub/T-PROC-IPTVFG-2008/en>, 2008.
- [21] Fundación Blender, “Big Buck Bunny”, [En línea], acceso 23 de septiembre, 2017; Disponible: <http://www.bigbuckbunny.org>
- [22] X. Liu, M. Tanaka, and M. Okutomi, “Noise Level Estimation Using Weak Textured Patches of a Single Noisy Image”, in *2012 19th IEEE International Conference on Image Processing*, [En línea], pp. 665–668, 2012, DOI: 10.1109/ICIP.2012.6466947