



ESCUELA SUPERIOR POLITECNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“Estudio sobre las Redes Definidas por Software (SDN) para mejorar la aplicación del Streaming de video en las Redes de Distribución de Contenidos (CDN) en el Ecuador”

EXAMEN DE GRADO

Previo a la obtención del título de

MAGISTER EN TELECOMUNICACIONES

Presentado por:

ALEX FERNANDO GARCÉS NÚÑEZ

GUAYAQUIL-ECUADOR

AÑO: 2020

AGRADECIMIENTO

Al Dr. Álvaro Suarez por su guía y seguimiento en el avance del proyecto, de igual forma a todos los docentes que participaron en mi formación académica durante la maestría, gracias a todo ese conocimiento fue posible la realización del presente proyecto.

Alex Garcés

DEDICATORIA

Este trabajo se encuentra dedicado en primer lugar a Dios, por brindarme la salud y la vida para llegar hasta este momento tan importante de mi vida, a mi familia que siempre estuvieron apoyándome y alentándome a seguir adelante a pesar de cualquier obstáculo existente en el camino.

Alex Garcés

TRIBUNAL DE EVALUACION



Msc. Verónica Soto
PROFESOR EVALUADOR

MARIA Digitally signed
ANTONIETA by MARIA
ALVAREZ ANTONIETA
VILLANUEVA ALVAREZ
VILLANUEVA VILLANUEVA

Dra. María Antonieta Álvarez
PROFESOR EVALUADOR

DECLARACION EXPRESA

“La responsabilidad y autoría del contenido de este Trabajo de Titulación, me corresponde exclusivamente; y doy mi consentimiento que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de proveer la consulta, difusión y uso público de la producción intelectual”



GARCÉS NÚÑEZ ALEX FERNANDO

RESUMEN

Entre las aplicaciones utilizadas en internet, la transmisión de video se ha caracterizado por ser uno de los servicios con mayor índice de crecimiento a lo largo de los últimos años, este incremento ha propiciado un aumento considerable en el tráfico de la red. Esta gestión se ha manejado a través de los proveedores de servicios de internet e Internet Service Provider (ISP), buscando una mejora en los servicios ofertados, y una reducción de tráfico en la red troncal. Por todo aquello se ha creado nuevos métodos de distribución de contenidos, que buscan reducir los efectos producidos por la gran cantidad y demanda de servicios multimedia a varios usuarios de forma simultánea.

El presente proyecto se enfoca en el estudio de la situación actual que vive el país con relación al uso de tecnologías para el streaming de video, mediante simulación se pretende estudiar el comportamiento de na red CDN, con la aplicaciones de SDN para verificar el rendimiento de la red con este tipo de tecnología.

Las redes definidas por software (SDN) se han convertido en un paradigma necesario de estudio y profundización, debido a que separan el plano de control del plano de datos en los equipos de red como switches y routers, a diferencia de las redes tradicionales, permitiendo tener un equipo centralizado llamado controlador pueda desarrollar aplicaciones y diversos servicios que optimicen las redes de datos.

INDICE GENERAL

| | |
|---|----|
| RESUMEN..... | I |
| INDICE GENERAL..... | II |
| ÍNDICE DE FIGURAS..... | IV |
| ÍNDICE DE TABLAS | V |
| CAPITULO 1 | 1 |
| 1. PLANTEAMIENTO DE PROBLEMA | 1 |
| 1.1. Descripción del problema | 1 |
| 1.2. Justificación | 1 |
| 1.3 Objetivos | 2 |
| 1.3.1 Objetivo General..... | 2 |
| 1.3.2 Objetivos Específicos..... | 2 |
| CAPITULO 2 | 3 |
| 1. FUNAMENTOS TEORICOS | 3 |
| 2.1 Participación del mercado del internet fijo en Ecuador..... | 3 |
| 2.2 La suscripción de Streaming de datos en el Ecuador | 3 |
| 2.3 Posición de Ecuador en el Mundo | 4 |
| 2.4 Red de distribución de contenidos (CDN)..... | 5 |
| 2.4.1 La Relevancia de la tecnología CDN | 6 |
| 2.4.2 Arquitectura de CDN..... | 6 |
| 2.4.3 Funcionalidad de CDN..... | 8 |
| 2.5 Redes definidas por software | 9 |
| 2.5.1 Arquitectura SDN | 9 |
| 2.6 Openflow | 11 |
| 2.6.1 Switch OpenFlow | 11 |
| 2.6.2 Controlador Openflow | 13 |
| 2.7 Mininet..... | 14 |

| | |
|--|----|
| 2.8 POX..... | 14 |
| CAPITULO 3..... | 16 |
| 3. Simulación de la topología en Mininet..... | 16 |
| 3.1 Desarrollo de la simulación virtual de la red CDN | 16 |
| 3.1.1 Simulación de video streaming pasó a paso enfocado en el ancho de banda..... | 17 |
| 3.2 Video streaming simultaneo con un ancho de banda constante..... | 19 |
| 3.2.1 El Indie de Similitud Estructural (SSIM) | 19 |
| 3.2.2 Retardo en el paquete | 20 |
| CAPITULO 4..... | 22 |
| 4 CONCLUSIONES Y RECOMENDACIONES | 22 |
| 4.1 Conclusiones..... | 22 |
| 4.2 Recomendaciones..... | 23 |
| BIBLIOGRAFIA..... | 24 |
| ABREVIATURAS | 25 |
| ANEXOS | 26 |
| ANEXO A | 26 |
| ANEXO B | 29 |
| ANEXO C | 30 |

INDICE DE FIGURAS

| | |
|---|----|
| Figura 2.1 Porcentaje de participacion de los proveedores ISP en Ecuador.... | 3 |
| Figura 2.2. Indices de crecimiento de abonados en servicios de streaming..... | 4 |
| Figura 2.3. Indices de crecimiento de trafico CDN para streaming..... | 6 |
| Figura 2.4. Forma de funcionamiento de CDN a nivel global..... | 7 |
| Figura 2.5. Funcion del switch mediante OpenFlow..... | 11 |
| Figura 2.6. Estructura de OpenFlow mediante capas..... | 12 |
| Figura 2.7. Funciones y componentes de OpenFlow..... | 12 |
| Figura 3.8. Topologia de Red para la simulacion..... | 16 |
| Figura 3.9. Comando en mininet virtual para establecer la QoS..... | 16 |
| Figura 3.10. Test del comando Iperf..... | 16 |
| Figura 3.11. Red usada para el video streaming | 17 |
| Figura 3.12. Grafica del indice de similitud vs el ancho de Banda..... | 18 |
| Figura 3.13. Red SDN con incremento de hosts en cada switch..... | 19 |
| Figura 3.14. Promedio de SSIM en relacion con el streaming multiple..... | 20 |
| Figura 3.15. Promedio de la tasa de bits en el streaming multiple..... | 20 |
| Figura 3.16. Promedio de retardo en el streaming multiple de paquetes..... | 21 |

INDICE DE TABLAS

| | |
|---|---|
| Tabla 2.1. Escala de posición de Ecuador en el ranking a nivel global..... | 4 |
| Tabla 2.2. Ubicación de Ecuador en relación los demás países de Sudamérica..... | 5 |

INTRODUCCIÓN

La continua mejora en las redes de telecomunicaciones, permite ofrecer diversas prestaciones en relación con la cantidad y calidad de datos, permitiendo un constante crecimiento y desarrollo, así como un despliegue de modernos servicios con la finalidad de cubrir los requerimientos de los usuarios, y a la vez proveer y dotar de un valor añadido al momento de hacer uso de dichas redes. El incremento de consumo en internet sobre el streaming de video no es algo nuevo, plataformas como youtube y netflix han propiciado una mayor demanda en torno al desarrollo y consumo de estos servicios. Estas mejoras en torno a los servicios de streaming en internet y en las redes que las proveen han permitido que estos servicios tengan cada vez más éxito [1].

Tomando en cuenta todas estas mejoras, todavía aparecen ciertos inconvenientes de congestión de tráfico en estos servicios, debido al aumento de usuarios. Siendo el tráfico de datos en video unos de los desencadenantes para que exista un cuello de botella en internet, debido a todo esto se han enfocado los recursos en optimizar los recursos de la red usando protocolos que ayuden al momento del streaming.

Las redes definidas por software (SDN) son utilizadas para la creación de redes que serán administradas y serán gestionadas a través de un controlador. Este nuevo reto en las redes ayuda a un crecimiento acelerado en los centros de datos, mediante las redes SDN por la flexibilidad que estas brindan frente a los centros de datos tradicionales con hardware, es factible crecer de una manera más acelerada y con una menor inversión [2].

Con este estudio se pretende quiere mostrar los beneficios que se tendrán al no depender de equipos físicos, la flexibilidad que brinda tanto en la administración como en las soluciones siendo más rápidas y eficaces que en los diferentes escenarios que se puedan presentar, esto es posible al poder realizar el trabajo de manera remota sin tener que dirigirse hasta el equipo que presenta la afectación, debido a que los usuarios consumen los contenidos de dos maneras, ya sea contenido en vivo, como pueden ser transmisiones de eventos deportivos, o contenido bajo demanda, como es el caso de las películas o series de televisión.

Es necesario entender cuáles son las limitantes de cada medio, y entender lo que el usuario espera al recibir estos servicios. Por lo tanto, no solo es necesario crear nuevos métodos de distribución de contenidos multimedia (CDN), sino que también es necesario entender como el usuario percibe esos servicios, y encontrar las métricas necesarias para mejorar los servicios existentes en función de la satisfacción del usuario final [3].

CAPITULO 1

1. PLANTEAMIENTO DE PROBLEMA

1.1. Descripción del problema

La complejidad de las redes de telecomunicaciones ha ido en aumento, por lo mismo diferentes tecnologías se han desarrollado con la finalidad de mejorar los componentes y elementos de red, y que estos sean capaces de soportar más servicios y por ende necesariamente deben integrar más funcionalidades para un mejor desempeño y configuración por parte del administrador de la red, ya que los equipos de conmutación tradicionales poseen sistemas específicos o propietarios, creando un incremento en gastos en relación con la necesidad de creación de configuraciones individualizadas.

Como lo menciona [4]. “Los videos y otras formas de contenido multimedia se han ido popularizando con el pasar de los últimos años mostrando un importante incremento en la actualidad”, esto ha ido de la mano con las mejoras en la infraestructura de los proveedores de internet que han incrementado el ancho de banda en las comunicaciones, las redes de distribución de contenidos (CDN) se utilizan para el almacenamiento de datos y contenidos que provee internet. Las peticiones de contenidos por parte de los usuarios son redirigidas hacia las CDN, en especial los videos de alta calidad en tiempo real (streaming) han ido utilizando una importante parte del tráfico en la red, convirtiéndose este en un desafío para el manejo de esta cantidad de información por parte de las CDN, para este efecto los diferentes algoritmos de distribución de carga son esenciales.

1.2. Justificación

En este estudio se investigara la integración de la tecnología de redes definidas por software (SDN) en conjunto con la infraestructura de las redes de distribución de contenidos (CDN), además de realizar y analizar una simulación con la finalidad de verificar el enfoque establecido en relación con el bienestar del cliente al hacer uso del streaming de video de forma eficaz. Esta nueva perspectiva tecnológica permitirá un crecimiento rápido de los datacenter, dado que ya no se requerirá de un incremento en el equipamiento, sino la creación de

más aplicaciones relacionadas con lo que se desee implementar o mejorar dentro de la red.

Para el desarrollo del estudio se pretende mostrar las diversas ventajas que conlleva la disminución de dispositivos físicos y su mayor flexibilidad a la hora de administrar la red haciendo uso de lo antes mencionado, brindando mayores soluciones en menor tiempo en diferentes escenarios, siendo posible y viable realizarlo de forma remota hasta el dispositivo que se encuentra afectado.

1.3 Objetivos

1.3.1 Objetivo General

- Analizar, de manera teórica, y proponer un estudio sobre la aplicabilidad de las Redes Definidas por Software (SDN) en las Redes de Distribución de Contenidos (CDN), para una mejora en el streaming de video hacia los usuarios de la red.

1.3.2 Objetivos Específicos

- Estudiar las técnicas utilizadas para la distribución de contenido multimedia, en especial el streaming de video, así como los estándares que tenemos actualmente.
- Analizar los parámetros necesarios en las redes (SDN) para brindar una mejor calidad a los usuarios, se recolectará la mayor cantidad de información disponible relacionada con el tema de estudio.
- Descripción y análisis de las simulaciones sobre las redes definidas por software, en busca de una optimización de los recursos de red cuando se realiza una distribución de contenido multimedia.

CAPITULO 2

2. FUNAMENTOS TEORICOS

2.1 Participación del mercado del internet fijo en Ecuador

El porcentaje de participación que han tenido los principales proveedores de internet a nivel nacional ha ido en aumento con relación a la demanda que estos han sufrido en los últimos años, según el informe anual presentado por Arcotel a finales del año 2019 se observa que CNT E.P. mantiene un 42,73% del mercado en general, seguido por MEGADATOS, que se maneja con el nombre de Netlife con un 17,06%; seguido por SETEL S.A. con el nombre de TVCable con el 11,11%; en cuarta posición se encuentra Conecel con nombre comercial Claro con el 10,95% del mercado nacional, en la figura 2.1 se muestra los porcentajes de participación [5].

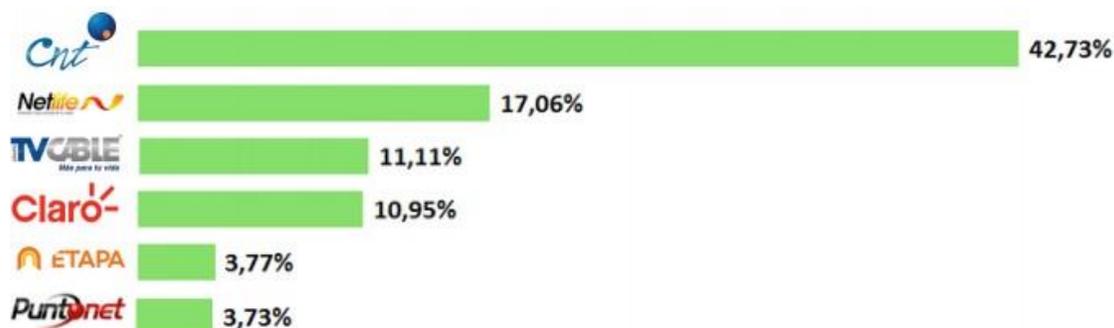


Figura 2.1. Porcentaje de participación de los proveedores ISP en Ecuador

2.2 La suscripción de Streaming de datos en el Ecuador

Según [5]. La suscripción de audio y video es un servicio brindado por medio de imágenes, sonidos y datos multimedia, consignados a un público específico de abonados. Según índices reportados por la ARCOTEL a nivel nacional hasta finales del año 2019 el número de suscriptores creció significativamente hasta alcanzar un total estimado de 1.107.328 abonados, como se lo muestra en la figura 2.2.

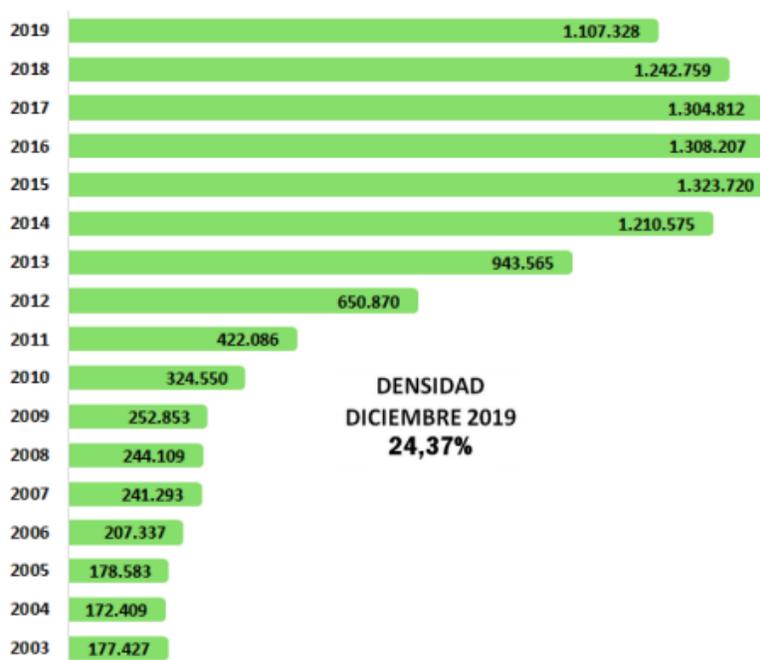


Figura 2.2. Índices de crecimiento de abonados en servicios de streaming

2.3 Posición de Ecuador en el Mundo

Con base en una reconocida fuente a nivel mundial como es el “Ranking Networked Readiness Index”, el mismo que está basado en el informe global de tecnología publicado en el Foro Económico Mundial a finales del año 2018 [6]. Tomando como variables el marco político, la infraestructura existente, el uso de las tics y el impacto relacionado con cada una de estas tecnologías. Esta visión tiene por finalidad estimular al gobierno y la sociedad sobre su involucración en el desarrollo de las TIC, en la tabla 2.1 se muestran estos indicadores.

| Ranking | Posición respecto al mundo |
|-----------|----------------------------|
| 2012-2013 | 116 de 134 países |
| 2013-2014 | 114 de 133 países |
| 2014-2015 | 108 de 138 países |
| 2015-2016 | 96 de 142 países |
| 2016-2017 | 91 de 144 países |
| 2017-2018 | 82 de 148 países |

Tabla 2.1. Escala de posición de Ecuador en el ranking a nivel global.

Con relación al desempeño de Ecuador en América del Sur, está ubicado por sobre países como Venezuela, Bolivia, Paraguay, Perú y Argentina, como lo indica la tabla 2.2.

| Ranking | Pais | Valor |
|---------|-----------|-------|
| 35 | Chile | 4,61 |
| 56 | Uruguay | 4,22 |
| 63 | Colombia | 4,05 |
| 69 | Brazil | 3,98 |
| 82 | Ecuador | 3,85 |
| 88 | Guyana | 3,77 |
| 90 | Peru | 3,73 |
| 100 | Argentina | 3,53 |
| 102 | Paraguay | 3,47 |
| 106 | Venezuela | 3,39 |
| 120 | Bolivia | 3,21 |

Tabla 2.2.Ubicación de Ecuador en relación los demás países de Sudamérica

2.4 Red de distribución de contenidos (CDN)

Tiene por objetivo el permitir una adecuada distribución de contenido de una forma totalmente descentralizada, permitiendo el acceso a la información desde el nodo más cercano, de esta forma se tiene un acceso más eficiente a la información y al contenido, de esta manera la CDN se ha vuelto una parte importante en las redes de telecomunicaciones.

Tomando en cuenta la popularidad relacionada con un sitio web, la cantidad de tráfico y el número de visitantes de cada sitio, estas búsquedas de información siguen una distribución zipf. Todo esto se refiere a que en la red han un sinnúmero de sitios con una menor circulación de tráfico, pero de igual forma existen otros con una alta tasa de tráfico, como se muestra en la figura, teniendo como desenlace que la concentración de mucho tráfico se da en pocos sitios, dando lugar a problemas de retardo, saturación y disponibilidad de la red. Como una solución viable apareció el concepto y uso de CDNs. Según [5]. Este tipo de red permite mejorar el rendimiento de la red y el uso de banda ancha, mejorando la accesibilidad y manteniendo intacto todo el contenido ofrecido, para lo cual las CDNs figuran de una mezcla de:

- Content-delivery: Este radica en el uso de varios servidores perimetrales (denominados subrogantes) que proveen copias del contenido al usuario final.
- Request-routing infraestructura: Es responsable encaminar la adecuada petición al cliente correcto, actualizando toda la información de la cache.

- **Distribution infraestructure:** traslada el contenido del servidor principal a los servidores CDN y confirma la existencia de toda la información en las caches.
- **Accounting infraestructure:** Continúa manteniendo logs con los que los clientes acceden y almacena el uso que cada uno lo realiza.

2.4.1 La Relevancia de la tecnología CDN

Según las estadísticas para el año 2014 eran que más del 39 por ciento del tráfico total del internet se sustentaba en las CDNs. En el año 2019 más del 60 por ciento del tráfico de la web esta soportado por las CDNs en relación con el tráfico total de la red.

Según reportes en el 2019 el tráfico de las CDNs fue de alrededor de los 139 exabytes, superando por mucho la cifra alcanzada en el 2014 [6].

En las CDNs el tráfico se irá incrementando con relación al tráfico total, como se lo muestra en la figura 2.3.

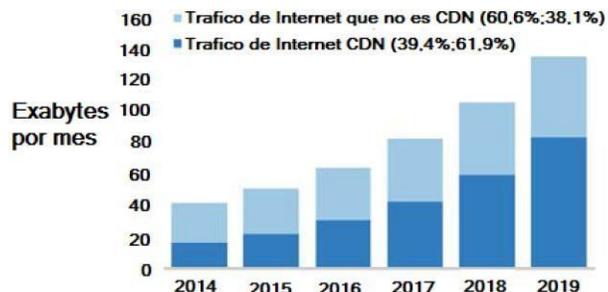


Figura 2.3. Índices de crecimiento de tráfico CDN para streaming

2.4.2 Arquitectura de CDN

En esta parte se explicara la estructura de la CDN. En la figura 2.4 se muestra de forma global como funciona CDN.

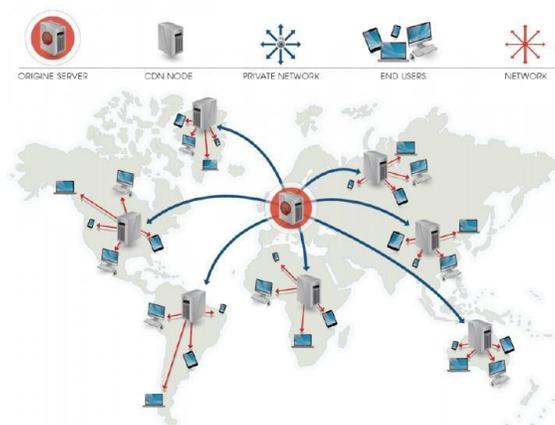


Figura 2.4. Forma de funcionamiento de CDN a nivel global

Los componentes que están involucrados en las CDNs son:

Cientes: A través de la CDN los usuarios acceden por medio del servidor a la información distribuida.

Servidores de cache: Se encargan de la replicación de la información y el contenido del servidor origen.

Servidor origen: Es el proveedor y dueño de la información y los contenidos.

Sistema de tarificación: Son servidores que arregla y contabiliza la información entre los usuarios, las CDN y los encargados de brindar el servicio.

Sistema de tarificación: Son servidores que por medio de las CDn y proveedores de información y contenido contabiliza y adecua los contenidos entre los clientes.

Sistema de encaminamiento: Es una de las partes más importantes de CDN, que se encarga de recibir las peticiones de los usuarios, las procesa y las reenvía hacia el servidor de cache, de una forma rápida, para que los usuarios detecten un retardo menor. En la actualidad, la forma que encamina la información es a través del DNS.

Sistema de distribución: Se destaca por mover el contenido entre los servidores donde se origina los datos hasta los servidores de cache.

Sistema de monitorización: Como tarea principal tiene el controlar y resumir los datos de la monitorización en forma estadística [7].

2.4.3 Funcionalidad de CDN

Las relaciones existentes entre los componentes se muestran a continuación:

- 1) El servidor principal envía los espacios de los nombres y de los objetos que se van a repartir en CDN, al sistema de encaminamiento por peticiones
- 2) El servidor origen se encarga de la publicación de los contenidos que deben ser distribuidos por la CDN en el sistema delegado de la distribución de contenidos.
- 3) El sistema CDN moviliza los contenidos hacia los servidores cache, y permite la adecuada interacción en la selección del servidor con el cache más adecuado.
- 4) El usuario solicita un contenido de lo que el percibe como servidor origen, debido a la entrega en los espacios de los nombres, dichas peticiones se envían hacia el encaminamiento.
- 5) El encaminamiento redirige las peticiones del usuario hacia el servidor cache más próximo que forma parte de la red CDN.
- 6) El servidor cache seleccionado brinda la información contenida al usuario, el servidor cache además envía la información al sistema que realiza la monitorización.
- 7) El sistema que realiza la monitorización procesa los datos para obtener la tarificación, enviando un feedback al segmento encargado del encaminamiento.
- 8) EL sistema de tarificación emplea diversos registros, para finalizar la prestación del servicio a clientes involucrados en la distribución de contenidos.

Para el objetivo de este estudio, la funcionalidad más relevante es la técnica de encaminamiento de CDN que se detalla a continuación.

Actualmente en las redes CDN, se aplican mecanismos que permiten encaminar las peticiones de los clientes hacia los servidores cache, los mecanismos más adecuados son:

Redirección HTTP: Se hace uso de cabeceras del protocolo HTTP, el usuario es dirigido hacia otra URL ubicada en el servidor cache, este método crea una carga extra en el servidor origen.

Redirección DNS: Es un mecanismo desapercibido para el cliente y en la actualidad es más utilizado, la mayoría de proveedores CDN como Akamai

emplea esta forma de redirección. El servidor DNS se encarga de convertir en direcciones IIP los nombres de máquinas, basándose en espacios de nombres distribuidos, estos nombres se agrupan de forma jerárquica, la infraestructura CDN controla la autorización de los servidores CDN, el administrador CDN principal es el que distribuye todas las peticiones a los demás servidores CDN [8].

2.5 Redes definidas por software

Son un conglomerado de técnicas referentes en las redes de computadoras, que como principal objetivo pretende facilitar la implementación de servicios de red de una manera determinista, escalable y además estable, de esta manera se evita que el administrador gestione la información y servicios de bajo nivel, y estas características se conciben por medio de una separación del plano de datos y el plano de control., específicamente sobre el firmware encargado del plano de control que posibilita al administrador a configurar, modificar y supervisar la red de una forma central [10].

SDN está en desarrollo y crecimiento, debido a sus características y beneficios en la programación, brindando una mayor garantía en relación con la seguridad, además provee una redundancia que es posible implementar, sin dejar de lado los beneficios que se mencionaron con anterioridad para el administrador y el cliente.

Las redes SDN necesariamente requieren de ciertos métodos y la utilización de protocolos, que les permitan tener un manejo del plano de control y la gestión sobre el plano de datos, un mecanismo utilizado es denominado openFlow, este concepto es confundido con el conjunto que forma a SDN, Open Networks Foundation ha sido la encargada de promover la utilización de SDN [10].

2.5.1 Arquitectura SDN

Según la definición establecida respecto a la organización ONF, la misma que es un arquitectura que se ha venido desarrollando en los últimos años, siendo flexible, manejable, adaptable y rentable. Teniendo como visión la óptima utilización en entornos con un gran ancho de banda y de naturaleza dinámica, esta arquitectura depara de forma directa el plano de control de la red, con esto se procura la abstracción de los servicios y aplicaciones sobre ella. Este

protocolo OpenFlow se ha convertido en el elemento principal en la construcción de soluciones SDN [11].

Las propiedades que conllevan la arquitectura SDN son:

- 1) *Directamente programable*: El control es programable y este se encuentra desacoplado de las funciones de encaminamiento que se realiza a nivel subyacente.
- 2) *Ágil*: Al momento de enviar los paquetes, se facilita la tarea al administrador para ajustar de forma dinámica todo el flujo de tráfico por toda la red para satisfacer todas las necesidades cambiantes.
- 3) *Gestión centralizada*: En la red la inteligencia está centralizada por medio de controladores que se basan en la arquitectura de software SDN manteniendo una visión general de toda la red, gracias a todo esto se considera algo similar al cerebro de la red, siendo el motor de aplicaciones y políticas.
- 4) *Configuración programable*: El administrador de la red configura, administra, optimiza y asegura los recursos de la red de forma rápida, los programas SDN se escriben por parte del administrador, evitando la utilización de software propietario.
- 5) *Basada en estándares abiertos e independientes de fabricantes*: Al hacer uso de los estándares abiertos, las instrucciones en el controlador SDN, son entendidas por diversas aplicaciones y dispositivos compatibles con el estándar analizado.
- 6) *La capa de aplicación*: está ligada a las aplicaciones de negocio que sean requeridas por el usuario, la utilización de APIs es necesario para el acoplamiento con el software de control.
- 7) *La capa de control*: Permite definir el tratamiento de los flujos de datos, a través de un protocolo estandarizado, OpenFlow en cambio se centra en el plano de control.
- 8) *La capa de infraestructura*: Esta se compone por los dispositivos físicos de comunicación, como lo son los switches y routers. Las tablas de flujo son administradas por los dispositivos de red, el plano de datos esta relacionado con esta capa [11].

2.6 Openflow

Considerada como un estándar por la Open Networking Foundation (ONF), para ser implementada en los equipamientos de red de SDN. El protocolo OpenFlow es el encargado de definir la interfaz entre el controlador y el dispositivo de conmutación.

2.6.1 Switch OpenFlow

Se caracteriza como un dispositivo sea este hardware o software capaz de enviar paquetes dentro de una red SDN, la forma de funcionamiento esta ideada en función del aprovechamiento de las tablas de flujo que poseen los routers y switches, permitiendo implementar firewalls, QoS, entre otras. Además OpenFlow hace uso de ciertas tablas de flujo que tienen un conjunto de funciones en común como se indica en la figura 2.5, que están asociadas con cada entrada de flujo [12].

La programación efectuada en el dispositivo de conmutación puede ser la siguiente:

- Basándose en varios encabezados y campos del paquete se encarga de identificarlos y categorizarlos
- Se encarga del procesamiento de paquetes incluyendo la modificación y el encabezado.
- Desde el controlador OpenFlow recepta y envía paquetes a través de un puerto de salida.

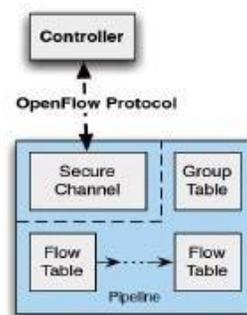


Figura 2.5. Funcion del switch mediante OpenFlow

Las diferentes instrucciones que se transmiten desde el controlador hacia el equipo de conmutación se denominan “flujos”, estos flujos después son organizados en forma de tablas, un paquete que ingrese será procesado por un flujo múltiple antes de que se dirija al puerto de salida [12].

La arquitectura de OpenFlow está separada por tres capas:

- Uno o varios físico o virtuales switches corriendo openflow
- Uno o varios controladores OpenFlow
- Uno o varias aplicaciones OpenFlow, como se muestra en la figura 2.6.

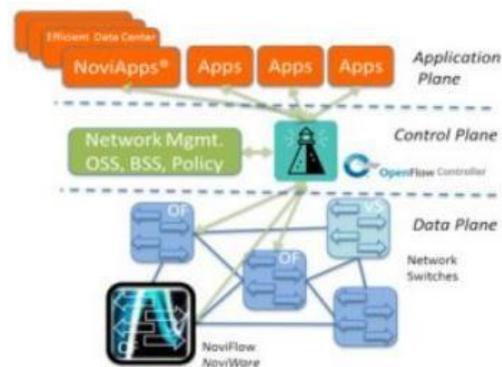


Figura 2.6. Estructura de OpenFlow mediante capas

La tabla de flujo se considera un elemento de vital importancia en los switches OpenFlow, siendo la tabla de flujo una serie de entradas de flujo como se indica en la figura 2.7.

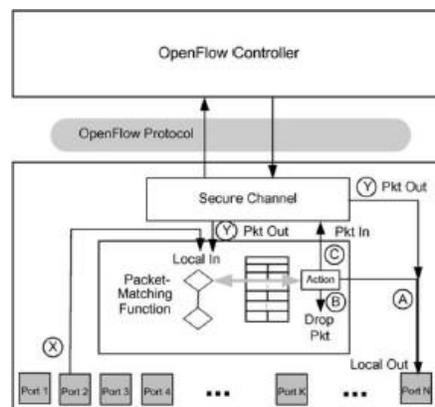


Figura 2.7. Funciones y componentes de OpenFlow

Los campos que corresponden con las entradas de flujo pueden realizar el matching de los paquetes que recibe, para después aplicar las instrucciones requeridas, siendo los principales campos los siguientes:

- *Match Fields*: Está enfocado en la comprobación y la coincidencia de los paquetes, consiste en la entrada y las diferentes cabeceras de los paquetes.
- *Counters*: Contadores que se actualizan al ubicar un paquete coincidente.
- *Instructions*: Este permite una modificación de la acción.

A pesar de que el estándar ha ido evolucionando, cuenta con lo siguiente:

- *Priority*: Al existir coincidencia proporciona prioridad para un paquete.
- *Timeouts*: Es el tiempo requerido para que el flujo se borre del switch.
- *Cookie*: Es un dato elegido y que se ha seleccionado en el controlador, que se lo usa para filtrar ciertos flujos afectados o modificados.
- *Flags*: Su forma se ve alterada debido a los flujos de entrada que han sido gestionados.

2.6.2 Controlador Openflow

Es el que mediante una interfaz de programación nos ofrece tareas de gestión, así como nuevas funciones, un controlador SDN se podría comparar con un sistema operativo de red, ofreciendo:

Gestión integral de la red, haciendo uso de bases de datos, sirviéndose estas como un repositorio de la información y demás elementos gestionables en la misma, así como el estado actual de red y de la topología que posee dicha red.

Incluye un sistema que permite calcular otras rutas y costes en los saltos de red, de igual forma permite el descubrimiento de mecanismos, dispositivos y topologías de red.

Mediante varias APIs, es posible realizar una exposición de las aplicaciones y servicios brindados por el controlador, facilitando en general la interacción misma con las aplicaciones, siendo esta interfaz representada a partir de diferentes modelos de datos descritos como servicios y funciones que caracterizan al controlador [13].

2.7 Mininet

Según [14]. Es un software ampliamente utilizado en la creación de redes y escenarios virtuales, con la característica que los switches simulados soportan el protocolo OpenFlow, permitiendo una amplia experimentación con redes SDN. Para lo cual se utiliza un sistema operativo Linux, con lo cual se puede crear varios nodos compuestos por switches o controladores que se encuentran ejecutándose en una máquina virtual. Cada uno de los hosts puede ejecutar las diferentes aplicaciones que se instalaron en el sistema Linux, gran parte del código que se ejecuta en Mininet se lo escribió en Python y una menor cantidad en C.

Las principales características son:

- Permite la simulación de forma virtual de diferentes topologías de red.
- Se puede hacer uso de APIs de Python de forma simple para permitir la creación de nuevas redes y su simulación.
- Posee una interfaz de comandos CLI mediante los cuales se puede enviar órdenes a los diferentes nodos, permitiendo realizar todo tipo de pruebas sobre las redes creadas.
- Es capaz de ejecutar archivos o un script de diversas topologías de red.
- En mininet se permite el acoplamiento de una gran variedad de controladores basados en protocolos, como es el caso de OpenFlow.

2.8 POX

Este sistema operativo de red permite una visualización y control de los switches OpenFlow que su configuración está basada en Python, este controlador permite la manipulación de switches virtuales y físicos que manipulen este protocolo, además de usarse en la exploración de prototipos con una virtualización de red, entre las principales características tenemos:

- Posee varios componentes que podrían usarse en el descubrimiento de topología y una mejorada selección del camino de la red.

- Tiene una idéntica interfaz gráfica, además de las herramientas que manejan los programas NOX que son trazados en Python, por lo mismo la herramienta mas usada para la programación del controlador es POX
- Al tener un ambiente de desarrollo basado en Python se ofrece la interfaz denominada “Pythonic”.
- Es ampliamente compatible con Windows, Linux y Mac OS [13].

CAPITULO 3

3. Simulación de la topología en Mininet

3.1 Desarrollo de la simulación virtual de la red CDN

Se toma como base un ejemplo mostrado en la página oficial de mininet, debido a que los paquetes de Open vSwitch ya vienen instalados por defecto en mininet, el ejemplo de la red utilizarse se muestra en la figura 3.8.

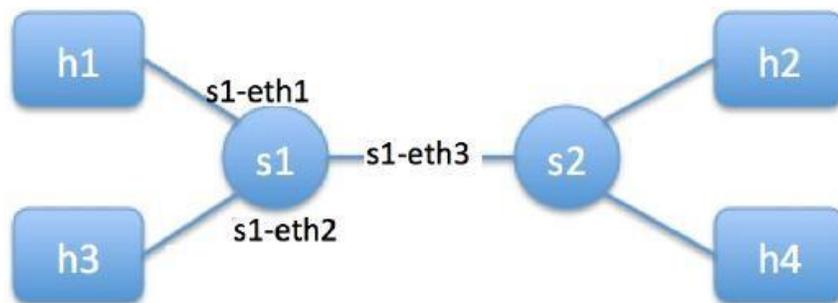


Figura 3.8. Topología de Red para la simulacion

Como se muestra en la figura 3.9, la red fue configurada con 2 switches y cada switch fue conectado a 2 hosts. Para lo cual se procede con el envío de un tráfico desde h1 hacia h2 a una tasa de transmisión de datos de 100 Mbps, por otro lado desde h3 hacia h4 el tráfico ira a 4 Mbps. A continuación se ejecuta el comando sobre el switch virtual

```

alex@alex-VirtualBox: ~
alex@alex-VirtualBox ~$ sudo ovs-vsctl -- set Port s1-eth3 qos=@newqos -- \
> --id=@newqos create QoS type=linux-htb other-config:max-rate=1000000000 queues=0=@q0,1=@q1 -- \
> --id=@q0 create Queue other-config:min-rate=1000000000 other-config:max-rate=1000000000 -- \
> --id=@q1 create Queue other-config:min-rate=4000000 other-config:max-rate=4000000
2efc7e91-d92f-4505-9595-9cb3f5e78437
8f37d7be-bd92-40c1-84cb-394190b5dc21
9e750406-63e2-489c-9284-fdf040a4bfa4
alex@alex-VirtualBox: ~$
  
```

Figura 3.9. Comando en mininet sobre un switch virtual para establecer la QoS

Para poder verificar el ancho de banda es necesaria la aplicación de una política de calidad de servicio, esta instancia es creada mediante el Linux-htb type, el mismo que hace uso de un algoritmo denominado Token Bucket Filter pudiéndose ser implementada dicha política de calidad de servicio con un rango máximo de 1 Gbps. La primera parte se envía tráfico multimedia en la cola

denominada q0 de transmisión sin ninguna restricción, en la segunda cola de transmisión q1 se envía tráfico multimedia con una restricción de 4 Mbps.

Para obtener los valores medidos de los anchos de banda entre (h1, h2) y (h3, h4), se utiliza el comando iperf el que se muestra en la figura 3.10.

```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['95.6 Mbits/sec', '111 Mbits/sec']
mininet> iperf h3 h4
*** Iperf: testing TCP bandwidth between h3 and h4
*** Results: ['3.83 Mbits/sec', '5.47 Mbits/sec']
mininet>
```

Figura 3.10. Test del comando Iperf

3.1.1 Simulación de video streaming pasó a paso enfocado en el ancho de banda

La simulación se realiza con la ayuda de 2 switches conectados entre sí y cada uno es conectado con 2 hosts como se indica en la figura 3.11, para lo cual h1, h2 se conectan al switch S1 y h3, h4 están conectados al switch 2, con dirección 10.0.0.2 y 10.0.0.4, esta simulación se la realiza algunas veces con la reducción progresiva del ancho de banda de 10 Mbps a 0.5 Mbps.

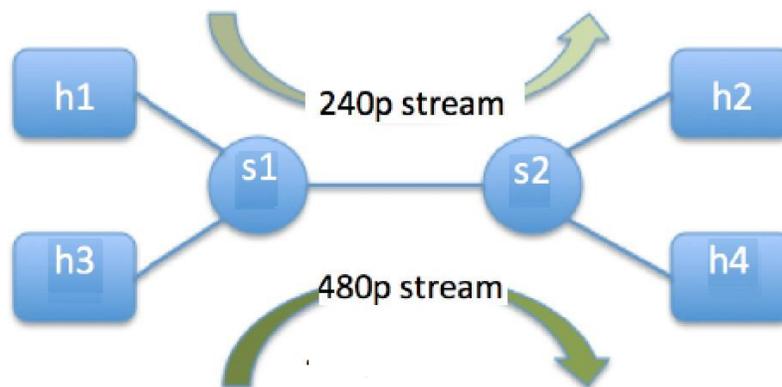


Figura 3.11. Red usada para el video streaming

La simulación llevada a cabo en mininet se llevó a cabo de la siguiente forma:

Solo se permite un video a la vez para realizar el streaming, el video que se envía entre h1 y h2 posee una resolución de 360x240p, siendo el ancho de banda utilizado de 10 Mbps, esta simulación se la efectúa varias veces con una reducción progresiva del ancho de banda como se mencionó anteriormente. La misma configuración se la realiza entre h3 y h4 pero con un video de 720x480p,

estas medidas obtenidas. bLa grafica muestra el comportamiento de la gráfica trazada, entendiéndose que el video de 240p se satisface más fácilmente con el ancho de banda otorgado a diferencia del video de 480p que no se satisface con el mencionado ancho de banda, estas funciones del ancho de banda se muestran divididas en 3 funciones paa el video de 240p y el video de 480p, todas estas graficas se muestran en la figura 3.12.

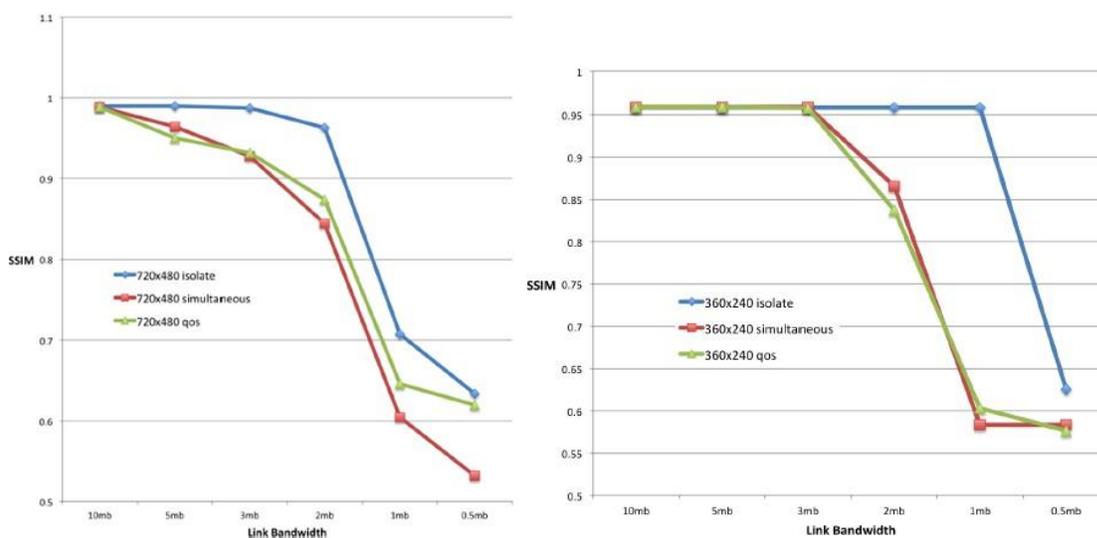


Figura 3.12. Grafica del indice de similitud vs el ancho de Banda

Se muestra que al aplicar la política de calidad de servicio, el video streaming mejora significativamente en el video de 480p, no pudiéndose decir lo mismo del video de 240p. Esto se sobre entiende que el video de 240p necesita menor cantidad de ancho de banda en comparación con el video de 480p, la red se encarga de mejorar el ancho de banda para el video de 480p, y al mismo tiempo evita deteriorar la calidad del video de menor resolución que es de 240p.

3.2 Video streaming simultaneo con un ancho de banda constante

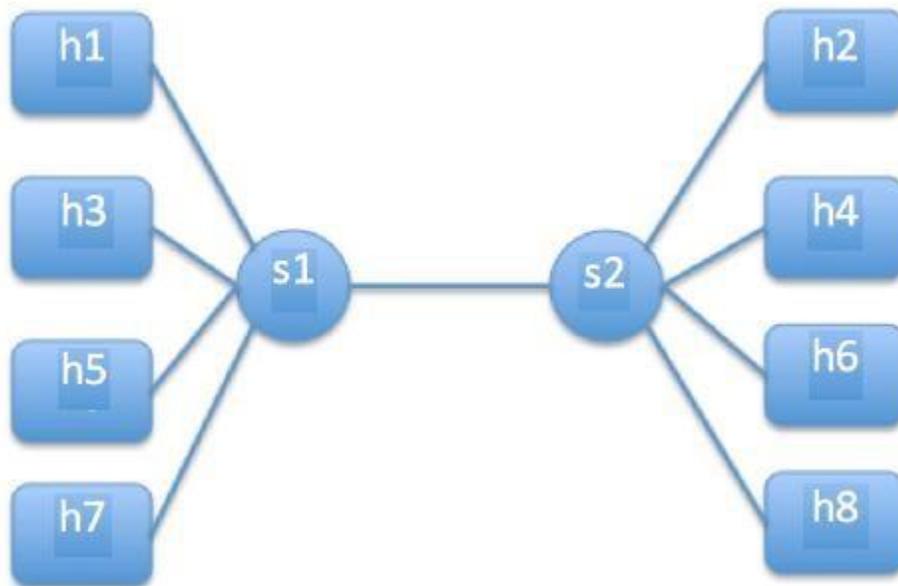


Figura 3.13. Red SDN con incremento de hosts en cada switch. Esta simulacion se inicia

Tomando de base la topologia de red anteriormente descrita con un incremento de 2 hosts en cada switch, que se visualiza en la figura 3.13, teniendo 4 por lado con un ancho de banda de 10 Mbps entre los switches, haciendo uso de dos flujos de video de 480p y 240p. Las funciones utilizadas seran las mismas que en el experimento previo, por lo tanto el ancho de banda sera una constante entre los switchs y este sera de 10 Mbps, la distribucion de cotenido sera la variante debido al incremento de los hosts en 2 por cada switch, este incremento puede iniciarse con 4, 8,16,32,..., de igual forma con los flujos de forma simultanea 2, 4,..., 16.este flujo simultaneo de datos es el analizado en la simulacion

3.2.1 El Indie de Similitud Estructural (SSIM)

Es graficado en el eje y frente al numero de hosts graficados en el eje x, en la figura 3.14, se muestra los valores del SSIM con relacion a la calidad de servicio ofrecida en el streaming de video realizado.Las estadisticas se obtuvieron atravez del analisis con wireshark y se graficaron en el eje x la cantidad de hosts, frente al promedio de la tasa de kbps recibidos por parte de los otros

hosts como se muestra en la figura 3.15, mostrandose una mejora en la calidad del servicio.

3.2.2 Retardo en el paquete

En el eje x se muestran los hosts que seran utilizados con un promedio de retraso en los paquetes de milisegundos, frente a los otros hosts que reciben la informacion, esto se obtuvo mediante wireshark. La mejora en el streaming realizado se ve reflejada debido a las politicas de calidad de servicio de la red, como se muestra en la figura 3.16.

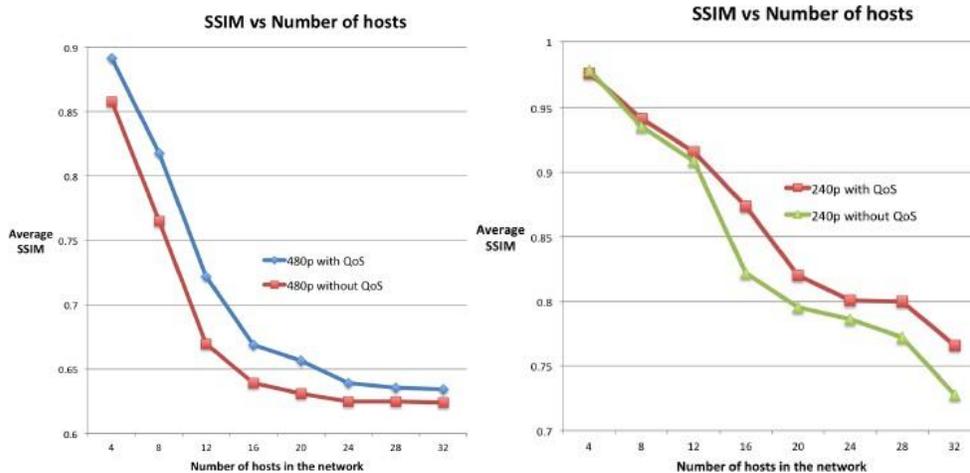


Figura 3.14. Promedio de SSIM en relacion con el streaming multiple

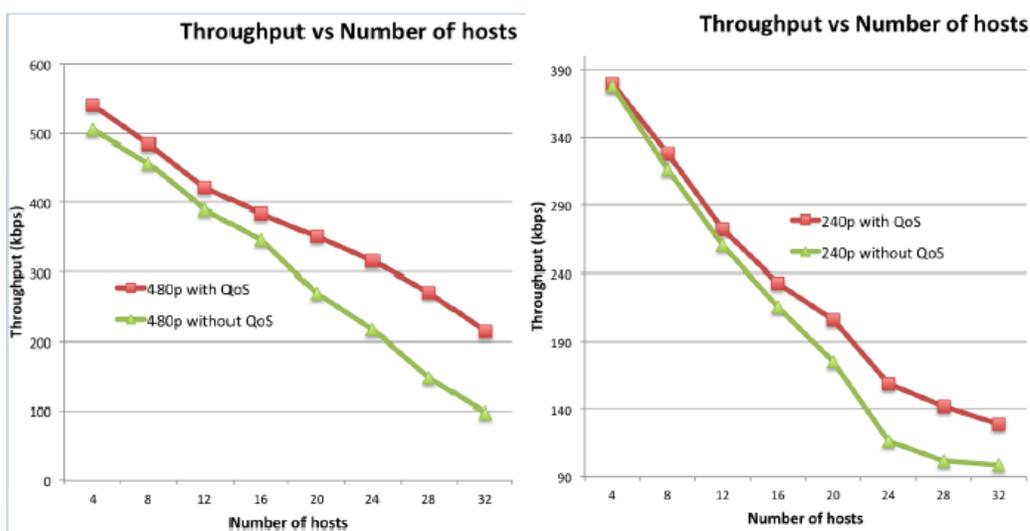


Figura 3.15. Promedio de la tasa de bits en el streaming multiple

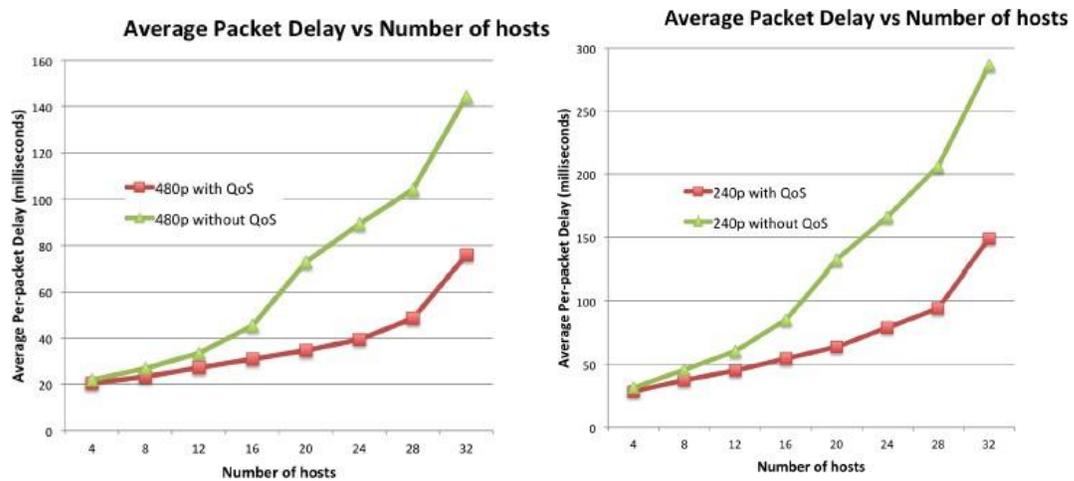


Figura 16. Promedio de retardo en el streaming multiple de paquetes

CAPITULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Este proyecto se centró en el análisis del streaming de video en redes CDN, mediante el uso de SDN que se han ido incrementando a nivel global, por lo cual la delantera que esta tecnología ofrece por medio de controladores específicos como es el caso de OpenFlow, permite visualizar el panorama que se avecina con el crecimiento exponencial de la internet.

Ecuador ha entrado en una nueva era tecnológica con una extensión de servicios de proveedores y cada vez más suscriptores para el servicio de internet y con el streaming de video siendo este último el principal flujo de información que es solicitado por los abonados, el servicio de las redes CDN van a influenciar a las empresas ISP de manera estratégica, permitiendo una optimización de recursos y una mejora en el servicio que estas ofrecen.

Se analizó la importancia de las redes CDN basadas en SDN debido a la gran demanda de contenido multimedia, en especial el video streaming el mismo que ha venido en aumento con el pasar de los últimos años y ha tenido un vertiginoso ascenso, que en el futuro con la llegada de nuevas tecnologías digitales será más solicitado por la población en general, la principal ventaja que refleja CDN es la de potenciar los servicios de distribución de contenidos de manera eficiente.

Mediante la realización de este proyecto se deja ver nuevos campos de investigación, que profundicen en el estudio del mundo relacionado con las redes CDN para un mejor desarrollo de estas en el país y su relevancia que tendrá en el futuro, relacionándose directamente con el software, hardware y el manejo de estos.

Se ha abordado conceptos básicos que engloban a las redes SDN, razones por las cuales la industria está realizando un cambio de cara al panorama que se avecina debido al aumento de abonados con requerimientos de streaming de video y las ventajas que posee al separar el plano de datos del plano de control

que esto supone, permitiendo analizar la arquitectura y el trabajo que realiza una red SDN.

.4.2 Recomendaciones

Se recomienda el estudio de las redes CDN debido a que está presente y seguirá una evolución con el pasar del tiempo, razón por la cual Latinoamérica todavía se consideraría con un retraso en la implementación de esta tecnología frente a otros lugares del mundo.

Se recomienda al momento de realizar una implementación física de una red CDN basada en SDN revisar las diferentes alternativas que puede brindar los distintos fabricantes, con la finalidad de tener una valoración de las potencialidades que estas pueden tener frente a diversos protocolos y tecnologías modernas.

Con mira en el futuro se recomienda la investigación en que otras áreas las redes CDN pueden ser explotadas, al igual que las redes SDN como sería el caso de la tecnología 5G.

BIBLIOGRAFIA

- [1] Global content delivery network (cdn) market insights, opportunity analysis, market shares and forecast 2016 - 2022. March 2016.
- [2] Software Defined Networks (SDN). Leaving network state for rendezvous services.
<http://yle.smu.edu/~nair/ftp/sdn/alto-sdn.pdf>
- [3] Vakali, A., & Pallis, G. Content Delivery Networks: status and trends. IEEE Internet Computing, 7(6), 69-74.
- [4] Diego R. Lopez. "Redes basadas en software, como se aplican ahora y se aplicaran en el futuro". ETSIIT. Master en Ingenieria de Telecomunicacion, 2016
- [5] ARCOTEL, Agencia Nacional de Regulacion de las Telecomunicaciones.
<http://arcotel.gob.ec>
- [6] World Economic Forum. The Global Information Technology Report 2018. World Economic Forum, Davos, Suiza, 2018
- [7] Cihat Çetinkaya ; Müge Sayıt , "ALTO-assisted CDN-based video streaming over SDN", 25th Signal Processing and Communications Applications Conference (SIU), 2017.
- [8] Ahmed Khalid, Ahmed H. Zahran, Cormac J. Sreenan con el tema "mCast: An SDN-Based Resource-Efficient Live Video Streaming Architecture with ISP-CDN Collaboration", IEEE 42nd Conference on Local Computer Networks (LCN), 2017.
- [8] Lawrence Harte. "Introduction to Data Multicasting, IP Multicast Streaming for Audio and Video Media Distribution." Althos, 2008.
- [9] SDN is DNS for Packet Switching
<http://devcentral.f5.com/articles/sdn-is-dns-for-packet-switching>
- [10] Enrique Hernandez-Valencia, Steven Izzo, and Beth Polonsky. "How will nfv/sdn transform service provider open Network", IEEE, 29(3):60-67, 2015.
- [11] Molina Moreno, B. Análisis y desarrollo de una red de distribución de contenidos y su algoritmo de redirección de usuarios para servicios web y streaming, Universidad Politécnica de Valencia, Valencia, España, 2013
- [12] Open Networking Foundation. Software-defined networking: Architecture overview. ONF White Paper, 2015
- [13] POX community. POX Openflow.
<http://openflow.stanford.edu/display/ONL/POX+Wiki>, 2015.
- [14] Diego R. López. Redes basadas en software. Que son, como se aplican ahora y en el futuro. ETSSIIT, 2016. Master en Ingeniería de Telecomunicaciones.

ABREVIATURAS

| | |
|-------------|-------------------------------------|
| API | Application Programing Interface |
| CDN | COntend Delivery Network |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name System |
| IP | Internet Protocol |
| ISP | Internet Service Provider |
| OPEX | Operating Expense |
| PC | Personal Computer |
| QoS | Quality of Service |
| SDN | Software Defined Network |
| SSIM | Structural Similarity Index |
| TCP | Transmission Control Protocol |

ANEXOS

ANEXO A

Código Python de ejecución en mininet

```

1 #!/usr/bin/python
2 '''
3 Se presenta el codigo de la red creada para realizar la simulacion
4   h1-          -h2
5     s1 ----- s2
6
7 if n=4,
8   h1-          -h2
9     s1 ----- s2
10  h3-          -h4
11
12 maximo llegaria hasta n
13
14 el codigo escrito es general y se engloba con las
15 variables acorde a la simulacion realizada
16 '''
17
18
19 import sys
20 import time
21 from threading import Thread
22
23 from mininet.topo import Topo
24 from mininet.net import Mininet
25 from mininet.node import CPULimitedHost
26 from mininet.link import TCLink
27 from mininet.util import dumpNodeConnections
28 from mininet.log import setLogLevel
29 from mininet.cli import CLI
30
31 from functools import partial
32 from mininet.node import RemoteController
33 import subprocess
34 from os.path import isfile, join
35 import os
36
37 '''
38 maximo de 32 hosts , trabajando en esta topologia se la realizo con Ubuntu 16.04
39 '''
40 n = -1 # number of hosts
41 bw = 10.0 # link bandwidth in mbps (all links have the same bandwidth)
42 qos = 1 # 1 -> if QoS needs to be applied | 0 -> no QoS
43
44 qos_k = 2
45 workingDir = os.getcwd()
46 capture_script = join(workingDir, 'capture.sh') # '/home/sumanth/mininetDir/capture.sh'
47 sd_flow_filepath = join(workingDir, 'testVideos/360x240_2mb.mp4') # '/home/sumanth/sample/360x240_2mb.mp4'
48 hd_flow_filepath = join(workingDir, 'testVideos/720x480_5mb.mp4') # '/home/sumanth/sample/720x480_5mb.mp4'
49 stream_time = 40 # wait for 40 seconds before shutting down (acomodate for the max time of a video safely)
50 savedStreamsDir = join(workingDir, 'savedStreams') # '/home/sumanth/teststorage/'
51 capturedTracesDir = join(workingDir, 'capturedTraces') # '/home/sumanth/mininetDir/capture_traces'
52
53 class SimpleTopo(Topo):
54     global n
55     # 2 switches and n hosts (n/2 hosts per switch), a link between 2 switches
56     def __init__(self, **opts):
57         Topo.__init__(self, **opts)
58
59         # Adding switches
60         s1 = self.addSwitch('s1')

```

```

60     s1 = self.addSwitch('s1')
61     s2 = self.addSwitch('s2')
62
63     # 'dummy' is added to not use the zero index
64     h = ['dummy'] # list of hosts
65
66
67     # Adding hosts
68     for i in range(n+1)[1:]:
69         h.append(self.addHost('h{0}'.format(i)))
70         if (i%2)==1:
71             self.addLink(h[i], s1)
72         else:
73             self.addLink(h[i], s2)
74
75     self.addLink(s1, s2)
76
77 def stream(src, dst, input_filename, output_filename, dstIP):
78     global stream_time
79     local_stream_time = stream_time * (n/2)
80
81     # src, dst are host objects obtained from net.get('<host>')
82     print 'Executing command on client %s <- %s'%(dst.name, src.name)
83     client_command = 'cvlc rtp://@:5004 --sout \
84         "#transcode{vcodec=h264,acodec=mpga,ab=128,channels=2,samplerate=44100}:\
85         std{access=file,mux=mp4,dst=%s}" \
86         --run-time %d vlc://quit &'%(output_filename, local_stream_time)
87     result2 = dst.sendCmd(client_command)
88     # print client_command
89     # result2 = dst.cmd('sleep 5')
90
91     # time.sleep(5)
92
93     print 'Executing command on server %s -> %s'%(src.name, dst.name)
94     server_command = 'cvlc -vvv %s --sout \
95         "#transcode{vcodec=h264,acodec=mpga,ab=128,channels=2,samplerate=44100}:\
96         duplicate{dst=rtp{dst=%s,port=5004,mux=ts}}" \
97         --run-time %d vlc://quit'%(input_filename, dstIP, local_stream_time)
98     result1 = src.sendCmd(server_command)
99     # print server_command
100
101     # print result1
102
103     return (src, dst)
104
105     # print 'streaming finished !!! - thread-{}'.format(threadIdx)
106
107
108 def vlcStream_working(net):
109     # sample testing method to send a vlc video stream from host h1 to h2
110
111     h1, h2 = net.get('h1', 'h2')
112
113     print 'Executing command on h2'
114     result2 = h2.cmd('cvlc rtp://@:5004 --sout \
115         "#transcode{vcodec=h264,acodec=mpga,ab=128,channels=2,samplerate=44100}:std{access=file,mux=mp4,dst=output.mp4}" \
116         --run-time 40 vlc://quit &')
117     # result2 = h2.cmd('sleep 5')
118
119     # time.sleep(5)
120
121     print 'Executing command on h1'
122     result1 = h1.cmd('cvlc -vvv test.mp4 --sout \
123         "#transcode{vcodec=h264,acodec=mpga,ab=128,channels=2,samplerate=44100}:duplicate{dst=rtp{dst=10.0.0.2,port=5004,mux=ts}}" \
124         --run-time 40 vlc://quit ')
125     # result1 = h1.cmd('sleep 5')
126

```

```

127 # result1wo = h1.waitoutput()
128
129
130
131 print result1
132 print result2
133 print 'comandos en h1, h2 finalizados'
134
135 def initiatecapture(h):
136
137 |
138 |   for i in range((n/2)+1)[1:]:
139 |     if i%2==1:
140 |       inFilepath = sd_flow_filepath
141 |     else:
142 |       inFilepath = hd_flow_filepath
143 |
144 |     outFile = inFilepath.split('/')[-1] # gets the actual file name, from the full path name
145 |     outFile = outFile.split('.')[0] + '_%dmb_link_%bw + 'h%d_to_h%d'%(2*i-1, 2*i) + ('qos_' if qos else 'congest_') + '%dhosts'%n + '.pcap'
146 |     outFile = join(capturedTracesDir, outFile)
147 |
148 |     capture_time = stream_time + (n-2)*(stream_time/2)
149 |
150 |     interface_name = 'h%d-eth0'%(2*i)
151 |
152 |     command = 'bash %s %s %d %s'%(capture_script, outFile, capture_time, interface_name)
153 |     h[2*i].cmd(command) # doesnt wait for the command to finish, if it is a blocking command
154 |
155 def getOutFilePath(inFilepath, i):
156 |   outFile = inFilepath.split('/')[-1] # gets the actual file name, from the full path name
157 |   outFile = outFile.split('.')[0] + '_%dmb_link_%bw + 'h%d_to_h%d'%(2*i-1, 2*i) + ('qos_' if qos else 'congest_') + '%dhosts'%n + '.mp4'
158 |   outFile = join(savedStreamsDir, outFile)
159 |
160 |   return outFile # returns a filepath

```

ANEXO B

Captura de wireshark para realizar las graficas

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|-----------------------------|-------------|----------|--------|--|
| 1 | 0.00000000 | fe80::640b:d2ff:fe3ff02::16 | | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 2 | 0.00001800 | fe80::640b:d2ff:fe3ff02::2 | | ICMPv6 | 70 | Router Solicitation from 66:0b:d2:34:b2:25 |
| 3 | 0.00781900 | fe80::640b:d2ff:fe3ff02::16 | | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 4 | 0.10099600 | fe80::640b:d2ff:fe3ff02::fb | | MDNS | 107 | Standard query PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question |
| 5 | 0.16037400 | fe80::640b:d2ff:fe3ff02::fb | | MDNS | 327 | Standard query ANY harekrsna [66:0b:d2:34:b2:25]._workstation._tcp.local, "QM" question ANY harek |
| 6 | 0.17313900 | fe80::640b:d2ff:fe3ff02::fb | | MDNS | 178 | Standard query response PTR _workstation._tcp.local PTR harekrsna [66:0b:d2:34:b2:25]._workstation |
| 7 | 0.40730600 | fe80::5c3e:d7ff:fe1ff02::16 | | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 8 | 0.41072600 | fe80::640b:d2ff:fe3ff02::fb | | MDNS | 327 | Standard query ANY harekrsna [66:0b:d2:34:b2:25]._workstation._tcp.local, "QM" question ANY harek |
| 9 | 0.41180900 | fe80::640b:d2ff:fe3ff02::16 | | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 10 | 0.49589100 | fe80::640b:d2ff:fe3ff02::16 | | ICMPv6 | 110 | Multicast Listener Report Message v2 |
| 11 | 0.66163900 | fe80::640b:d2ff:fe3ff02::fb | | MDNS | 327 | Standard query ANY harekrsna [66:0b:d2:34:b2:25]._workstation._tcp.local, "QM" question ANY harek |
| 12 | 0.86204000 | fe80::640b:d2ff:fe3ff02::fb | | MDNS | 309 | Standard query response TXT, cache flush AAAA, cache flush fe80::640b:d2ff:fe34:b2:25 PTR, cache f |
| 13 | 0.90947900 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 14 | 0.90965600 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 15 | 0.91089800 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 16 | 0.91254900 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 17 | 0.91420300 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 18 | 0.91583700 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 19 | 0.91748000 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 20 | 0.91911400 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 21 | 0.92075700 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 22 | 0.92240100 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 23 | 0.92405400 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 24 | 0.92569500 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 25 | 0.92733900 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 26 | 0.92897300 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 27 | 0.93062100 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 28 | 0.93226200 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |
| 29 | 0.93391000 | 10.0.0.3 | 10.0.0.4 | UDP | 1370 | Source port: 34546 Destination port: avt-profile-1 |

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|-----------------------------|-----------------|----------|--------|--|
| 1 | 0.00000000 | fe80::e8fb:22ff:fe4ff02::16 | | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 2 | 0.02005400 | fe80::fc09:29ff:fe9ff02::16 | | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 3 | 0.07609700 | fe80::fc09:29ff:fe9ff02::16 | | ICMPv6 | 110 | Multicast Listener Report Message v2 |
| 4 | 0.14169500 | fe80::fc09:29ff:fe9ff02::fb | | MDNS | 327 | Standard query ANY harekrsna [fe:09:29:98:a5:ee]._workstation._tcp.local, "QM" question ANY harek |
| 5 | 0.39208200 | fe80::fc09:29ff:fe9ff02::fb | | MDNS | 327 | Standard query ANY harekrsna [fe:09:29:98:a5:ee]._workstation._tcp.local, "QM" question ANY harek |
| 6 | 0.59267600 | fe80::fc09:29ff:fe9ff02::fb | | MDNS | 309 | Standard query response TXT, cache flush AAAA, cache flush fe80::fc09:29ff:fe98:a5:ee PTR, cache f |
| 7 | 0.83361700 | fe80::fc09:29ff:fe9ff02::fb | | MDNS | 107 | Standard query PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question |
| 8 | 1.01191300 | fe80::fc09:29ff:fe9ff02::fb | | MDNS | 178 | Standard query response PTR _workstation._tcp.local PTR harekrsna [fe:09:29:98:a5:ee]._workstation |
| 9 | 1.20412000 | 0.0.0.0 | 255.255.255.255 | DHCP | 342 | DHCP Discover - Transaction ID 0x2b93f93f |
| 10 | 1.70075300 | fe80::fc09:29ff:fe9ff02::fb | | MDNS | 309 | Standard query response TXT, cache flush AAAA, cache flush fe80::fc09:29ff:fe98:a5:ee PTR, cache f |
| 11 | 2.38930000 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 12 | 2.38942800 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 13 | 2.39212700 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 14 | 2.39541100 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 15 | 2.39869900 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 16 | 2.40198700 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 17 | 2.40527400 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 18 | 2.40856400 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 19 | 2.41185400 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 20 | 2.41516400 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 21 | 2.41847300 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 22 | 2.42172300 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 23 | 2.42501800 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 24 | 2.42831100 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 25 | 2.43160000 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 26 | 2.43488800 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 27 | 2.60269000 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 28 | 2.60292700 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 29 | 2.60552600 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 30 | 2.60880000 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 31 | 2.61209100 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |
| 32 | 2.79605400 | 10.0.0.1 | 10.0.0.2 | UDP | 1370 | Source port: 54176 Destination port: avt-profile-1 |

ANEXO C

Ejemplo de visualización de la salida de video de streaming

