



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“ESTUDIO TEÓRICO DE LA QoS DE LAS REDES DE
DISTRIBUCIÓN DE CONTENIDOS MULTIMEDIA (CDN)
USANDO EL PROTOCOLO OPEN FLOW DE SDN Y
SUS AÑADIDOS”**

EXAMEN DE GRADO

Previo a la obtención del Título de:

MAGÍSTER EN TELECOMUNICACIONES

CARLOS ANTONIO GARCÉS MENDOZA

GUAYAQUIL – ECUADOR

AÑO: 2020

AGRADECIMIENTOS

Agradezco a Dios por hacer que cada paso sea acertado en la búsqueda de este objetivo. Agradezco a mi madre Magdalena Mendoza Alcívar por la fuerza y entusiasmo que me dio cada día para terminar triunfante este nuevo logro alcanzado.

DEDICATORIA

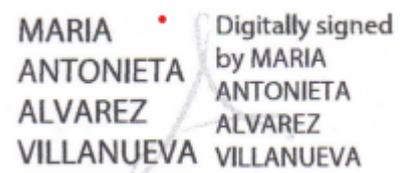
Dedico este trabajo a los Ingenieros que buscamos superarnos cada día mediante el conocimiento y la investigación.

TRIBUNAL DE EVALUACIÓN



Msc. Verónica Soto

PROFESOR EVALUADOR



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

Ph.D. María Antonieta Álvarez

PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me(nos) corresponde exclusivamente; y doy(damos) mi(nuestro) consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

A rectangular box containing a handwritten signature in blue ink. The signature is cursive and appears to read 'Carlos Garcés Núñez'.

CARLOS GARCÉS NÚÑEZ

RESUMEN

Las redes de telecomunicaciones actuales son fundamentales para proveer de aplicaciones y servicios en la web, pero estas crecen cada vez más y toda la infraestructura de red se ve comprometida a una evolución constante, a nivel de hardware y software, por causa de la moderna tecnología que, actualmente, crea en las personas una necesidad de acceder al contenido y los servicios ofrecidos en internet.

Esta constante demanda de los usuarios implica una mejor infraestructura de red en todos sus aspectos, para esto las CDN (redes de distribución de contenido), gracias a su infraestructura, otorgan a los usuarios este servicio de red, de manera eficiente.

Pero esta constante demanda de los usuarios por acceder a los servicios web y aplicaciones de internet ha originado investigaciones y estudios para la aplicación de un nuevo concepto de red, con características similares y mayores ventajas necesarias para estos momentos en los que es de gran importancia el ancho de banda y el encaminamiento de los paquetes en la red, que serían las ya conocidas SDN (redes definidas por software), con un mecanismo novedoso para desarrollar, de mejor manera, el desempeño de la red.

Un nuevo enfoque, basado en dividir el plano de datos y el plano de control, resulta ser la solución a muchas necesidades que afrontan en estos momentos las redes tradicionales, tales como velocidad, costo, inteligencia en su gestión.

El propósito de este trabajo es realizar un estudio teórico de la QoS de CDN si se aplicaría el protocolo OpenFlow de SDN, esto mediante la investigación y el análisis de artículos relacionados encontrados por Web of Science y también por IEEEExplore, acerca de las características de SDN, el funcionamiento de cada una de sus etapas, resultados de aplicar SDN a CDN y estudios relacionados a mejorar la gestión de esta red usando SDN, estableciendo en base a esto, conclusiones acerca de la QoS de CDN con OpenFlow de SDN y

estar en capacidad de usar esta información para ejecutar proyectos futuros y mejorar las redes actuales.

Finalmente, se analizan las potencialidades y las deficiencias de CDN y SDN, respectivamente, las mejoras con SDN y sus añadidos. También se muestra la posibilidad de usar un emulador para destacar el rendimiento de una SDN.

ÍNDICE GENERAL

AGRADECIMIENTOS.....	ii
DEDICATORIA	iii
TRIBUNAL DE EVALUACIÓN	iv
DECLARACIÓN EXPRESA.....	v
RESUMEN.....	vi
ÍNDICE GENERAL.....	viii
CAPÍTULO 1	1
1. LA QoS DE LAS REDES DE DISTRIBUCIÓN DE CONTENIDOS MULTIMEDIA (CDN) USANDO EL PROTOCOLO OPEN FLOW DE SDN.....	1
1.1 Introducción.....	1
1.2 Descripción del problema	1
1.3 Justificación.....	2
1.4 Objetivos.....	2
1.4.1 Objetivo General	2
1.4.2 Objetivos Específicos.....	2
1.5 Marco teórico.....	3
Temas de aplicaciones relevantes para el estudio:.....	3
1.5.1 Intervención de una arquitectura SDN en una red tradicional	3
1.5.2 Aplicabilidad de SDN para mejorar la QoS de CDN.....	5
1.5.3 Bondades del uso de SDN.....	5
1.5.4 QoS de CDN aplicando características de SDN	7
1.5.5 Mejora en el desempeño del tráfico de red en CDN a través de SDN	8
1.5.6 La entrega de contenido en equipos móviles y como se mejora esta con SDN.....	9
1.5.7 El protocolo OpenFlow ayuda a aplicar QoS a una CDN.....	10
1.5.8 SDN y OpenFlow en la entrega del contenido al usuario con mayor QoS.....	12
1.5.9 Aumento de la QoS en servidores de réplicas de CDN con SDN	14
1.5.10 QoS en la entrega de contenido con OpenFlow y sus conmutadores.....	15

CAPÍTULO 2	17
2. DISEÑO DE LA ESTRUCTURA DE ESTUDIO PARA IMPLEMENTAR LAS CARACTERÍSTICAS DE SDN A UNA CDN	17
2.1 Conceptos prioritarios y funcionalidad: Características de las SDN y el protocolo OpenFlow	17
2.1.1 Arquitectura de red SDN	17
2.1.2 Funcionamiento de la arquitectura de red SDN	18
2.1.3 Protocolo OpenFlow para SDN	20
2.2 Conceptos prioritarios y funcionalidad: Características de la tecnología CDN.....	22
2.3 Descripción de las ventajas de aplicar SDN a las CDN	24
2.4 Análisis del mejoramiento de la QoS de las Redes de Distribución de contenidos Multimedia (CDN) usando el protocolo Open Flow de SDN y sus añadidos.....	25
2.5 Trabajos relacionados acerca de mejoramiento de CDN mediante SDN	26
CAPÍTULO 3	28
3. ANÁLISIS DE RESULTADOS DE LA APLICACIÓN DE LAS CARACTERÍSTICAS DE SDN PARA MEJORAR EL RENDIMIENTO DE CDN.....	28
3.1 Resultados de la aplicación del protocolo OpenFlow para CDN.....	28
3.2 Resultados de las características de SDN mediante un emulador.....	29
3.2.1 Características de Mininet para emular una red basada en Software.....	29
3.3 Resultados de análisis de las potencialidades de SDN para mejorar la QoS de CDN en la interacción de ambas tecnologías.....	32
CONCLUSIONES Y RECOMENDACIONES	34
BIBLIOGRAFÍA	35
ANEXO 1	38
ANEXO 2	38
ANEXO 3	39

ÍNDICE DE TABLAS

Tabla 1: Características de una SDN.....	18
Tabla 2: Características del protocolo OpenFlow.....	21
Tabla 3: Comparación del rendimiento de CDN con SDN	33

ÍNDICE DE FIGURAS

Figura 1.1: Etapas y operación de una SDN [11].....	4
Figura 1.2: SDN aplicada al uso de CDN [12].....	5
Figura 1.3: OpenFlow en los switches para la gestión en la red [13]	6
Figura 1.4: OpenFlow en la virtualización con internet de las cosas [14].....	7
Figura 1.5: OpenFlow en la selección del servidor acertado [15].....	8
Figura 1.6: Mayor QoS en la descarga de Streams en CDN con SDN [16] ..	9
Figura 1.7: CDN y SDN en equipos móviles [17]	10
Figura 1.8: SDN en la red de entrega de contenido con sus clientes [18]...	11
Figura 1.9: Rendimiento en una red sin y con SDN [19]	13
Figura 1.10: QoS en réplicas de una CDN con controladores OpenFlow de SDN [20].....	15
Figura 1.11: Como se administra la caché de una manera centralizada con SDN [21].....	16
Figura 2.1: Principio de una arquitectura de Red Definida por Software.....	17
Figura 2.2: Capas de SDN	20
Figura 2.3: Función de OpenFlow en SDN	22
Figura 2.4: La arquitectura de una Red de distribución de contenido	22
Figura 3.1: Red con un controlador diseñada en Mininet.....	30
Figura 3.2: Puerto y dirección ip del controlador en emulador Mininet.....	30
Figura 3.3: Topología ejecutada en Mininet	31

CAPÍTULO 1

1. LA QoS DE LAS REDES DE DISTRIBUCIÓN DE CONTENIDOS MULTIMEDIA (CDN) USANDO EL PROTOCOLO OPEN FLOW DE SDN

1.1 Introducción

Frente a la demanda de ancho de banda en los servicios de telecomunicaciones por parte de los usuarios en la actual convivencia con la tecnología, en que la mayor parte de actividades del ser humano se realizan mediante internet, encontrándonos con conceptos como big data, plataformas ott, whatsapp, internet de las cosas y muchas más aplicaciones usadas a diario, surge la necesidad de incrementar el ancho de banda y eso implica cambio y reestructuración de equipos y no es posible subir el costo al usuario por el incremento del costo de nuevos y modernos equipos. Ante esta situación aparece una nueva arquitectura de red, las SDN o redes definidas por software, las cuales, mediante sus características como separar el plano de control y el plano de datos de los equipos de red, logran abrir el camino al estudio de soluciones de QoS de redes ya implementadas como CDN.

1.2 Descripción del problema

El problema radica en la infraestructura de una red de telecomunicaciones, la cual debe ser cada vez más grande y mejor estructurada en razón del constante crecimiento de los usuarios al acceso a los servicios de internet, lo que involucra entonces aumento de los costos del hardware de la red, mayor velocidad y eficiencia en la entrega de los datos; necesidad de una mayor escalabilidad de los recursos, virtualización de los servicios, control centralizado en la administración, cambios y modificaciones en la programación. También en las CDN, por causa del creciente tráfico dentro de la red, además, se presentan problemas tanto en la implementación como en la gestión. [1]

1.3 Justificación

El estudio realizado en este trabajo es necesario e importante por los siguientes aspectos:

La necesidad de mayor ancho de banda por parte de los servicios y aplicaciones requeridas por las personas, conlleva a mejorar y por ende a realizar cambios en la infraestructura de una red de distribución de contenido (CDN), por lo cual se busca en este trabajo estudiar las características de estas redes y de las SDN junto con el protocolo OpenFlow, con la finalidad de analizar, si es posible mejorar la QoS de una CDN con SDN y exponer todas estas potencialidades que podrían implementarse interactuando ambas tecnologías, como es el caso de no utilizar hardware y reemplazar esto con software, también gestionar de una mejor manera toda la red. [2]

La realización de este estudio tiene como justificación, conocer más acerca de las cualidades que puede ofrecer una SDN para mejorar una CDN y analizar, si es factible, esta implementación, para utilizar esta información en estudios que se realicen posteriormente.

También en este estudio se muestra la existencia de un simulador de red virtual que provea de switches, host, controlador y que en sus switches se pueda interactuar con el Open Flow, obteniendo experiencias sin el uso de hardware.

1.4 Objetivos

1.4.1 Objetivo General

Realizar un estudio teórico acerca de las características de CDN y el protocolo OpenFlow de SDN, mediante el análisis de las cualidades de ambas tecnologías, con la finalidad de conocer como se mejoraría la calidad de servicio (QoS) de CDN con OpenFlow.

1.4.2 Objetivos Específicos

Investigar y dar a conocer como se mejoraría las CDN en cuanto a hardware, software y gestión mediante la arquitectura SDN y sus herramientas.

Explicar el funcionamiento de una CDN y una SDN, exponiendo cada una de sus etapas e identificando elementos propios de CDN que se pueden mejorar con el uso del protocolo OpenFlow de SDN.

Dar a conocer la existencia y utilidad de un emulador para las SDN y el protocolo OpenFlow, indicando sus características.

1.5 Marco teórico

Temas de aplicaciones relevantes para el estudio:

1.5.1 Intervención de una arquitectura SDN en una red tradicional

Una SDN implementada para la estructura de una red tradicional, se describe en este artículo [11] de la siguiente manera:

La red de internet está estructurada de tal manera que los dispositivos y equipos individuales que realizan la comunicación manejan la información de forma controlada bajo el mando de protocolos, pero en base al concepto de SDN se establece crear un protocolo con el objetivo de que los nodos de la red funcionen compatibles hacia este en la comunicación, logrando una red con mayor programabilidad, administrando mejor la red teniendo el control de las aplicaciones, permitiendo diferenciarse de la red tradicional al excluir de tomar decisiones en el envío de paquetes a los enrutadores, otorgando un plano de control de tipo centralizado, facilitando la gestión de la red, esto al realizar la separación de la etapa de control de los equipos o hardware que realiza el reenvío de datos. En la arquitectura SDN la interacción entre la etapa de control y la etapa de datos es posible mediante el mecanismo que otorga OpenFlow, el cual realiza la programación de los equipos de reenvío de datos, es decir, a través de este protocolo se administran y configuran los dispositivos presentes en la red. En la figura 1.1 se muestran las partes y operación de una SDN:

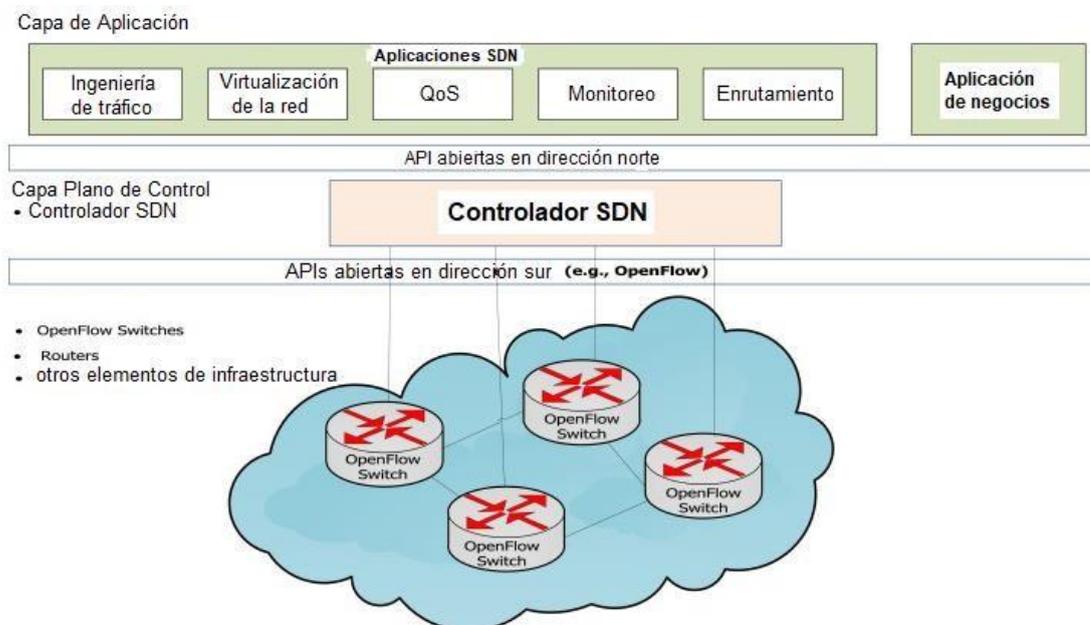


Figura 1.1: Etapas y operación de una SDN [11]

SDN hace referencia a un tipo de red rápida que ofrece QoS al proveer de flexibilidad y programación de un extremo a otro, así como también abstracción en la red, el control y la gestión de la red en los recursos involucrados en su estructura.

En una red tradicional como CDN por ejemplo, los equipos están estructurados verticalmente, que significa que ambos planos, de control y de datos se encuentran en los equipos, lo que dificulta gestionar la red, ya que al realizar alguna modificación necesaria, involucra que el administrador rectifique cada equipo de la red, pero este mecanismo de cambio manual en la red promueve muchos errores. Los investigadores han afirmado que en el campo de las redes se ha acogido, por la flexibilidad para la configuración, la arquitectura SDN, como es en el campo de las telecomunicaciones.

En relación a las CDN, su infraestructura ya funciona con la tecnología SDN, en la entrega de datos, para realizar esto de forma rápida cuando el usuario lo requiera, en la interconexión de los centros de datos situados en varias zonas del mundo, un ejemplo es Google, el cual usa esta arquitectura. Mediante el protocolo OpenFlow los switches gestionan a partir de un sitio

central y según investigadores en una red estructurada con OpenFlow, el problema de la latencia es solucionado de una mejor manera.

1.5.2 Aplicabilidad de SDN para mejorar la QoS de CDN

Según el estudio realizado en el artículo [12], debido al veloz incremento del requerimiento de datos por parte de las redes de los usuarios de casa, las CDN han crecido, por consiguiente, los ISP empiezan a ayudar a estas redes con el fin de mejorar la QoS, proponiendo el uso de SDN y los switches OpenFlow.

Aplicando procedimientos propios de SDN se permite interactuar el controlador con el equipo de red, logrando con esto conceder a los que operan esta CDN con ISP involucrarse en las determinaciones referentes al redireccionamiento de toda la red conjunta. En la figura 1.2 se puede observar la estructura de SDN usada en esta aplicación para el funcionamiento de una CDN: en los equipos usados para la infraestructura donde interviene solamente SDN, la etapa del redireccionamiento, establecida de manera lógica, se puede ubicar apartada del resto, entablando la comunicación mediante las llamadas APIs, enfocadas como aplicaciones externas a los demás componentes.

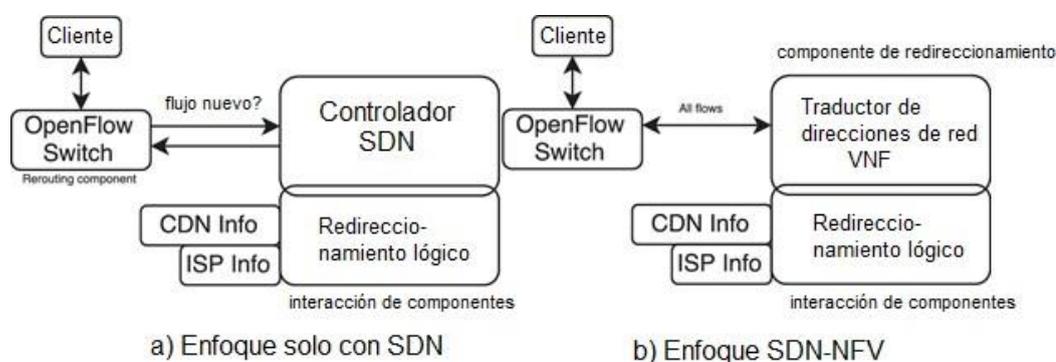


Figura 1.2: SDN aplicada al uso de CDN [12]

1.5.3 Bondades del uso de SDN

En el estudio realizado por [13], acerca de las bondades obtenidas al utilizar SDN, se describe lo siguiente:

Poseedores y también operadores de grandes redes en todo el mundo como Deutsche Telekom, indican que la arquitectura SDN va a contribuir en la implementación de redes mucho más sencillas en su estructura, con programación y una mayor facilidad en la administración, con costos menores que ayuden a incrementar los ingresos, es decir, SDN aparece como un prototipo sustituto de redes convencionales. En la figura 1.3 se muestra la función del protocolo OpenFlow dentro del plano de control formando un grupo apartado de switches conectados entre sí para gestionar la conectividad y poder intervenir en la topología de la red, logrando con OpenFlow programar el plano de datos.

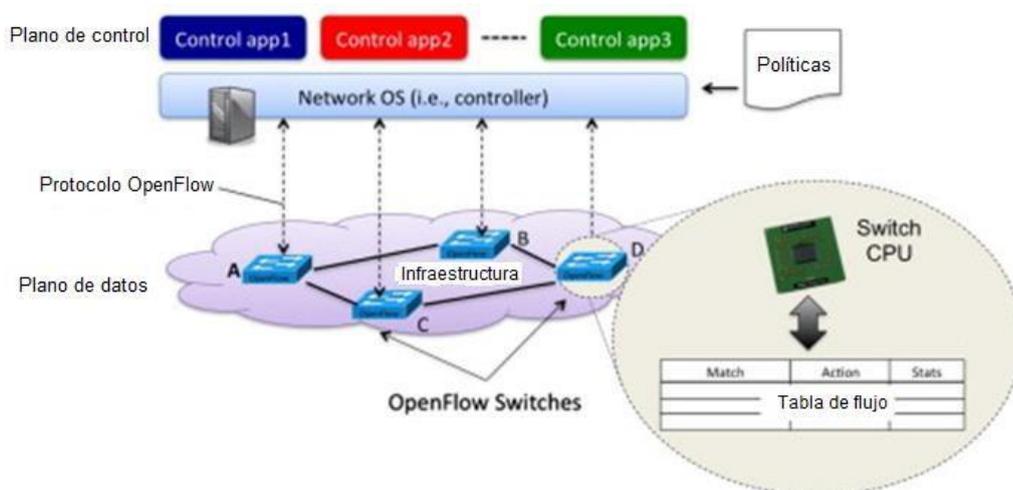


Figura 1.3: OpenFlow en los switches para la gestión en la red [13]

Según el estudio del artículo [14], los resultados de utilizar SDN son los siguientes:

Envío de datos con una gran velocidad y la utilización de los recursos de una forma mayormente eficiente.

Poder gestionar con mayor facilidad la red, al tener los administradores la potestad de controlar la red y modificar las características de esta.

La tecnología SDN permite administrar las redes de tipo rural de mejor manera que una red convencional, porque puede gestionar la red mediante el controlador, al separar las etapas de datos y control. En la figura 1.4 se

observa un esquema de red formada por sensores que operan con OpenFlow y que es factible una conexión con el internet de las cosas, sin inconvenientes, mediante la virtualización.

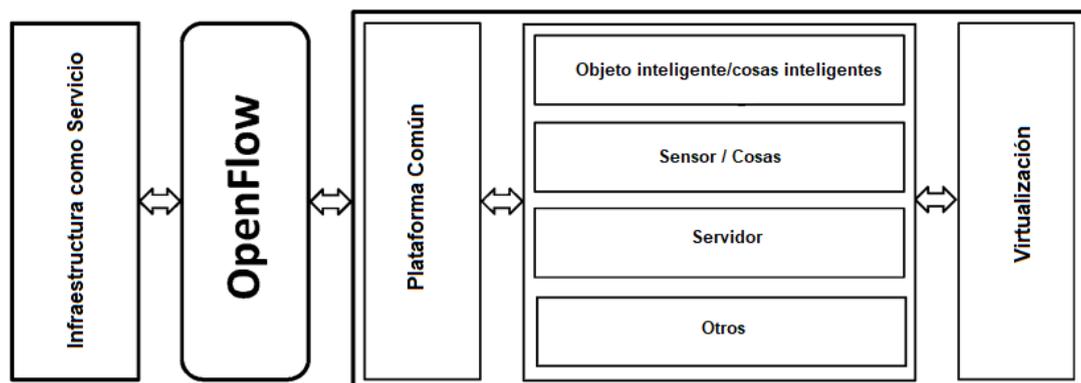


Figura 1.4: OpenFlow en la virtualización con internet de las cosas [14]

1.5.4 QoS de CDN aplicando características de SDN

Una CDN implementada con características otorgadas por SDN, mejora la calidad de servicio, en algunos aspectos. El artículo [15] describe lo siguiente:

Involucrar SDN en la infraestructura de una CDN concede a los encargados de abastecer de CDN o proveedores, decidir de forma centralizada.

Las CDN en su funcionamiento deben escoger al servidor conveniente que otorgue una mayor calidad en el servicio hacia el usuario, pero el problema es que este proceso no se lleva a cabo sin tener, de todo el sistema, una visión general, lo cual es una deficiencia el no conocer el estado y la topología de la cual está estructurada la red. Las SDN, mediante la flexibilidad y programabilidad, se convierten en una solución a esta clase de inconvenientes, ya que en las CDN el no captar la novedad de la condición que se encuentra la red y de su topología, origina disminuir la calidad del servicio al cual el visitante accede. En la figura 1.5 se plantea una red CDN cuyo funcionamiento se basa en una arquitectura SDN, en donde, con sus controladores, permite reunir en tiempo real información estadística de la red, para realizar decisiones acertadas sobre el servidor y al mismo tiempo su

enrutamiento en un Smart Controller Server, que se integra a una CDN y que opera mediante lógica SDN para decidir acertadamente en la búsqueda del servidor y mediante OpenFlow permitir el enrutamiento correcto de solicitudes hacia este.

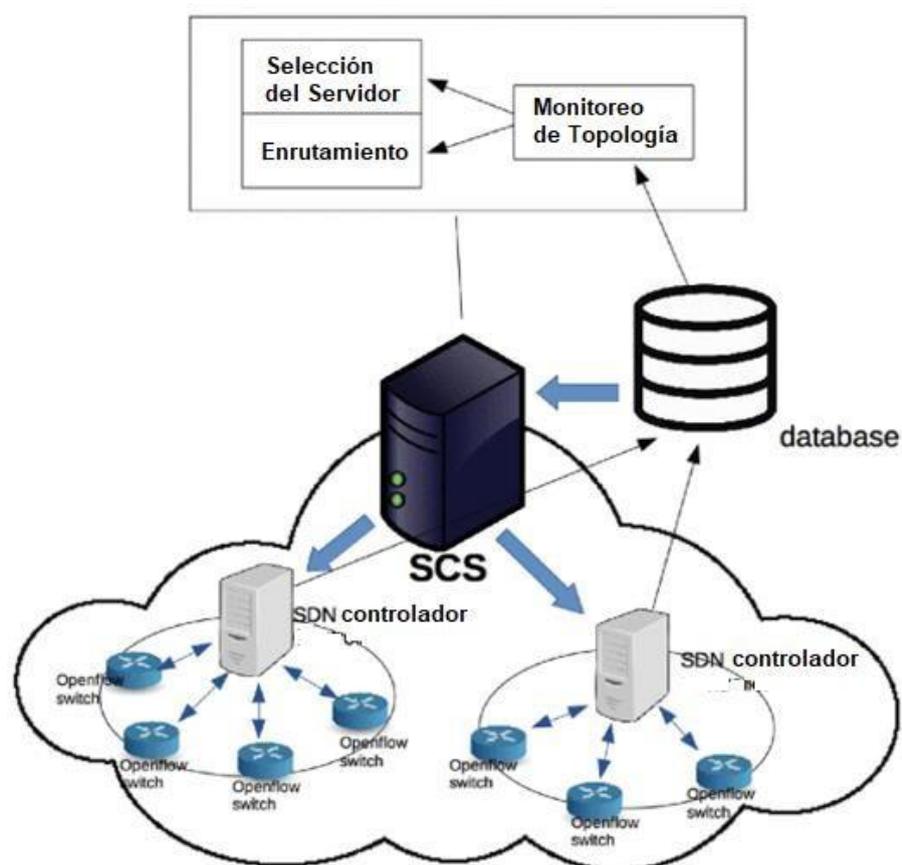


Figura 1.5: OpenFlow en la selección del servidor acertado [15]

1.5.5 Mejora en el desempeño del tráfico de red en CDN a través de SDN

El rendimiento de una CDN puede mejorar mediante SDN; en el artículo [16] se describe lo siguiente:

De acuerdo al estudio de tráfico, implementar redes definidas por software otorga mayor QoS por medio de la flexibilidad en el enrutamiento y la forma de gestionar centralizadamente la red. Redes de distribución de contenido de alto alcance, como por ejemplo Google, tienen en su estructura redes definidas por software, en el caso específico de SD-WAN, para funcionar

eficientemente en relación al enrutamiento. En este artículo también se predice, referente al tráfico de CDN, este llegará al 82% del total que se produce en internet, esto pronosticado para el 2021, en cambio sólo el 25% abarcará para esta misma fecha el tráfico SD-WAN del total de esta red WAN. Se mejora la calidad de servicio para aplicaciones de alto ancho de banda, usando conmutadores con gran flexibilidad. En la figura 1.6 se observa una mayor QoS en una CDN, se mejora el rendimiento al transferir la información, en el caso sin SDN para realizar la descarga de streams en rutas separadas y con SDN la descarga de secuencias de datos de streams de manera simultánea, utilizando una infraestructura equipada con la tecnología VRF (otorga a la red la segmentación sin necesidad de distintos equipos), switches y en este proceso el protocolo OpenFlow. Se muestra en las gráficas el tiempo de la descarga y la tasa de promedio real de los datos que se transfieren, también su rendimiento, tanto en a) sin SDN y en b) con SDN, en donde en esta se retransmiten los streams con mayor velocidad:

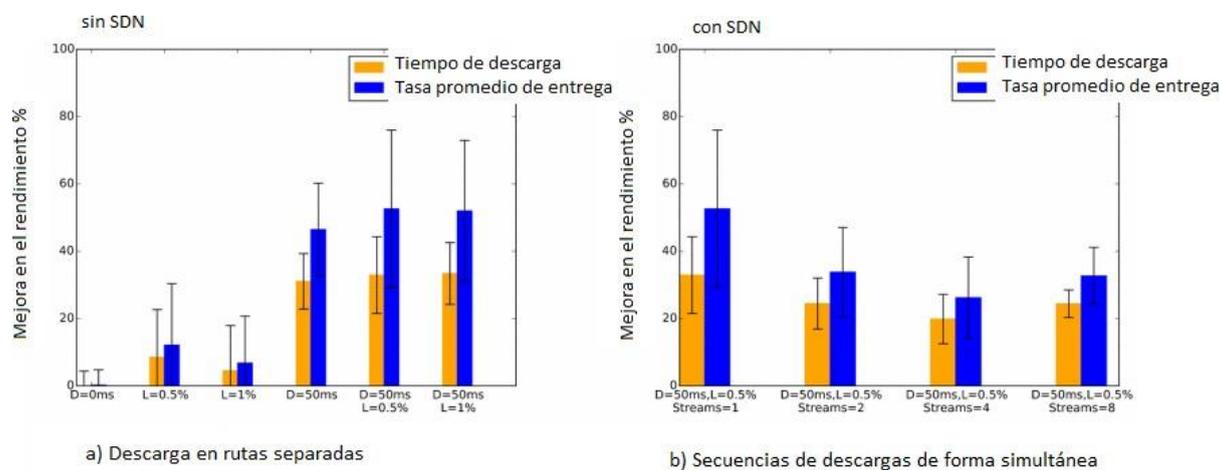


Figura 1.6: Mayor QoS en la descarga de Streams en CDN con SDN [16]

1.5.6 La entrega de contenido en equipos móviles y como se mejora esta con SDN

Por el aumento, cada vez mayor, del uso de equipos móviles se incrementan también los usuarios, pero la infraestructura de la red, para la entrega de

contenido aplicando SDN, puede ser mejorada. En el artículo [17] se explica lo siguiente:

Cuando se entrega contenido, específicamente MSM y se utiliza en el diseño las cualidades de SDN, se obtiene una mejoría en el rendimiento gracias a que el controlador funciona de manera centralizada y en este caso puede realizar la descarga en los equipos móviles aún con limitaciones de recursos.

En la figura 1.7 se puede observar este proceso:

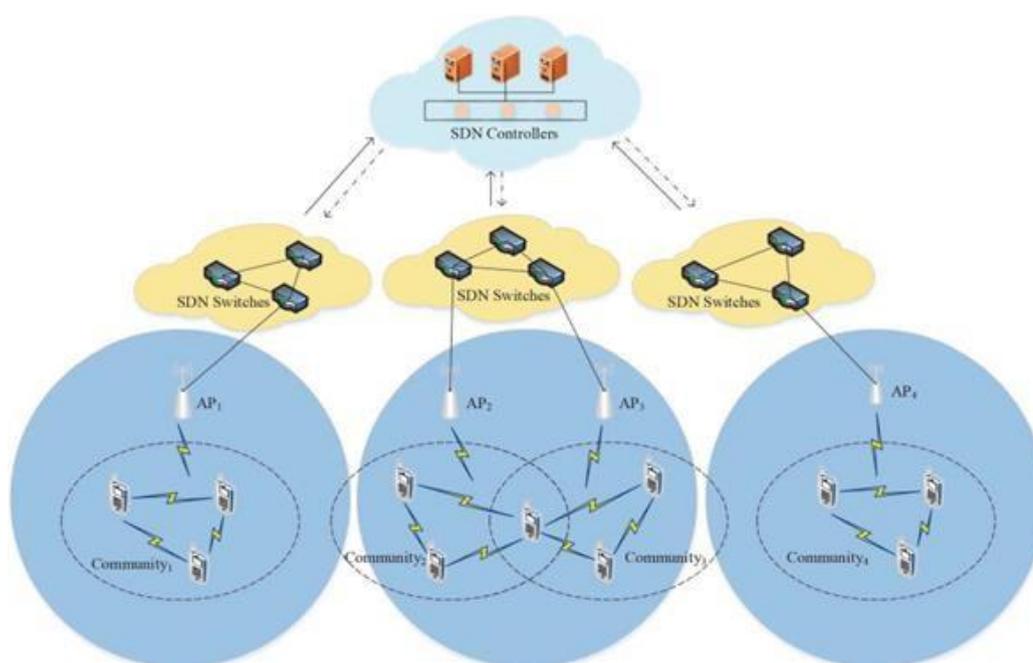


Figura 1.7: CDN y SDN en equipos móviles [17]

1.5.7 El protocolo OpenFlow ayuda a aplicar QoS a una CDN

Referente a las cualidades que puede entregar una infraestructura SDN, el protocolo OpenFlow y sus componentes, en el artículo [18] se menciona lo siguiente:

En la implementación de la infraestructura para una CDN, en donde se necesita para los usuarios el contenido siempre disponible, involucra un alto costo y el requerimiento de difíciles gestiones administrativas para el almacenamiento caché que está incluido en los servidores repartidos en todos los lugares del mundo. Por motivo del enorme tráfico se ve

comprometida la QoS en el funcionamiento de toda la CDN y para mitigar este inconveniente, una opción sería plantear políticas para la conducción de los datos, para lo cual se necesita de una arquitectura moderna que es SDN. El protocolo OpenFlow concede en la red tener una vista integrada y con esto realizar el control de toda la red con menos dificultades, reparando los errores y fallas en diversos puntos de la red. Una SDN involucrada en una CDN logra ajustar todas las funciones del control mediante la lógica que se otorga al servidor, lo que brinda beneficios como entablar comunicación con el servidor DNS por ejemplo, en cada momento que el visitante desee acceder a una información. En la figura 1.8 se muestra la ventaja de utilizar SDN, se tiene un enfoque completo del lugar donde se encuentra el usuario o cliente y el contenido, no se necesita entablar comunicación nuevamente con el proveedor para retomar un contenido, se puede entender entonces que esta arquitectura permite disminuir el retraso en un servicio específico, en la entrega de contenido de la red, lo que ayuda a incrementar la QoS de la CDN.

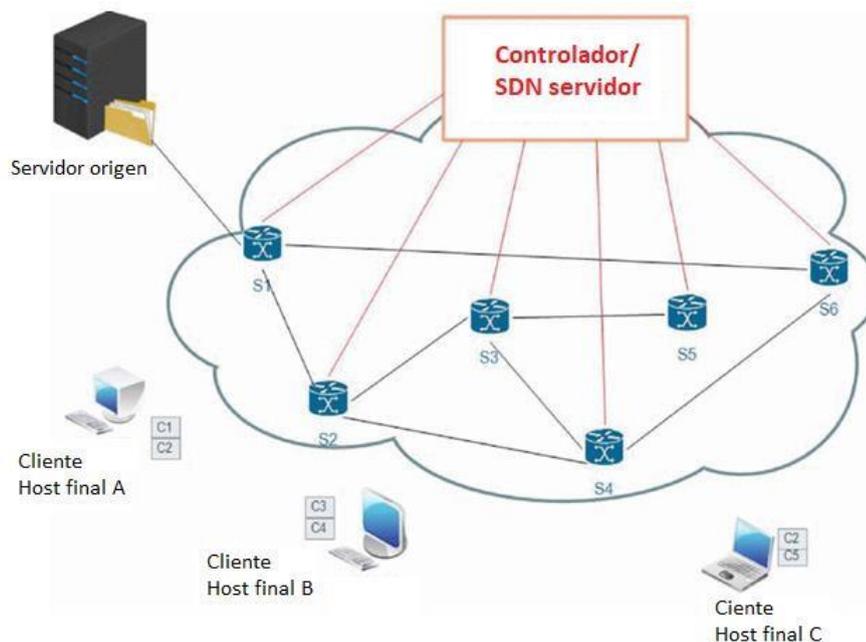


Figura 1.8: SDN en la red de entrega de contenido con sus clientes [18]

1.5.8 SDN y OpenFlow en la entrega del contenido al usuario con mayor QoS

De acuerdo al artículo [19], en una CDN, al entablar la comunicación con el usuario para hacer la entrega del contenido, es complejo el proceso de monitorear como el usuario ha recibido el servicio y al mismo tiempo saber si existe un problema en la red mientras se está entregando este y saber también la QoS hacia el usuario. Frente a esto, es necesario tener una vista general de como se está llevando la conexión de un extremo a otro y también conocer el estado de cada segmento de la red. Ante esta situación presentada en una CDN, la aplicación de OpenFlow y SDN otorga controlar el flujo de forma dinámica de acuerdo a las restricciones que varían en la red y también de sus distintas aplicaciones, para luego entregar una mejor QoS.

Se puede observar la comparación del rendimiento en el transcurso del tiempo real sin y con la intervención de SDN, en la figura 1.9: Mediante prácticas realizadas se ha comprobado que para la congestión del tráfico que se origina en el transcurso de la llamada ruta estática de la red se utiliza SDN, arquitectura en la cual su controlador está apto para elegir superiores posibilidades de enrutamiento para un mejor desempeño de esta, en la figuras a, b y c se muestra la tasa de enrutamiento real promedio (throughput) de la infraestructura de una red para entrega de contenido, en la cual se vincula el plano de control de un extremo a otro extremo accediendo de manera inalámbrica con la tecnología LTE, controlando en una WAN el tráfico de manera dinámica. Con el uso de Mininet se provee las gráficas: a) tasa promedio TCP de la transferencia de video sin utilizar SDN, b) tasa promedio TCP de la transferencia de video utilizando SDN y c) la información del acumulado del número de secuencia de paquete con y sin SDN.

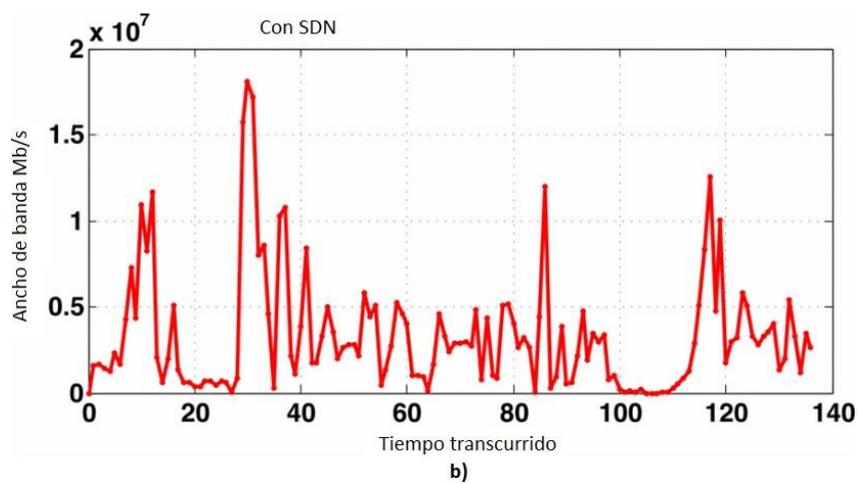
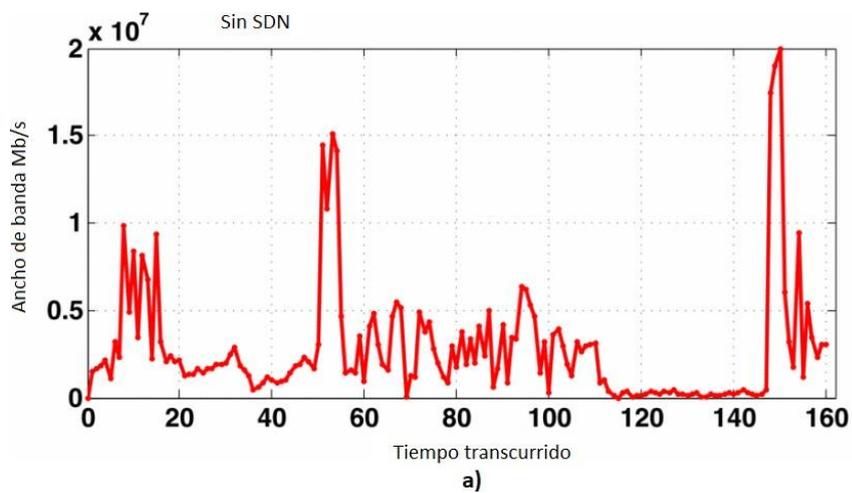


Figura 1.9: Rendimiento en una red sin y con SDN [19]

En la infraestructura CDN en la WAN, para un usuario, sin la utilización de los componentes de SDN, el tiempo en la descarga del video en su totalidad fue de 160 segundos. Para otro usuario con SDN en la infraestructura de red, el tiempo de la descarga del video fue de 132 segundos, lo que evidencia una mejor QoS en la red.

1.5.9 Aumento de la QoS en servidores de réplicas de CDN con SDN

Referente al análisis realizado en el artículo [20], en la figura 1.10 se observa el proceso de una replicación en una CDN en cuya infraestructura están presentes, para su funcionamiento, servidores origen de contenido y de réplica, switches y controladores OpenFlow de SDN, para realizar operaciones de mejora en la tasa de replicación, como ordenar los objetos de contenido de acuerdo a la necesidad del usuario final. En la gráfica acerca de Número de visitas del usuario versus el Incremento del porcentaje de las tasas de réplicas, se indica el Access Rate, que es la tasa de acceso, esta cambia su porcentaje de 7.24 a 11.95; Client Priority, que es la tasa correspondiente para los contenidos que tienen prioridad para el visitante, esta cambia su porcentaje de 6.38 a 9.05 y finalmente, la Even Driven cambia de 3.88 a 6.38 su porcentaje, que es la tasa generada por eventos. Estos valores son mostrados sin SDN y con SDN, indicando una QoS superior en el segundo caso, porque los controladores basados en SDN verifican y examinan en rangos regulares este proceso y con esto se consigue mejorar la disponibilidad del contenido, aumentando la tasa de extracción de datos en la CDN.

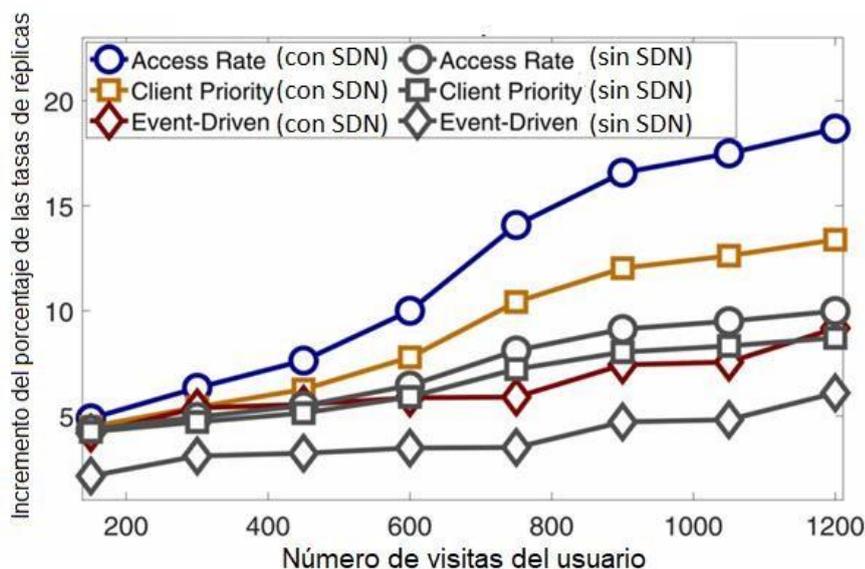


Figura 1.10: QoS en réplicas de una CDN con controladores OpenFlow de SDN [20]

1.5.10 QoS en la entrega de contenido con OpenFlow y sus conmutadores

Para el estudio acerca de la entrega de contenido con OpenFlow en el artículo [21] se señala lo siguiente: La entrega de contenido en internet y su tráfico se torna cada vez más complejo ya sea por aplicaciones o por servicios requeridos por el visitante, para esto la arquitectura que suministra SDN en una infraestructura de red se presenta como una posibilidad en la necesidad de algoritmos nuevos y requerir establecer políticas, esto mediante la programación orientada hacia los flujos de datos en la red. En experimentos realizados en mininet se observa que con SDN se logra una eficiencia superior en la red de entrega de contenido a través de un análisis en el redireccionamiento dinámico y también la mejora en la administración de la caché. En la figura 1.11 se observa la ejecución de la red y como está gestionada la caché, de manera centralizada y cuando no es centralizada. La gráfica en estado normal hace alusión a que no existen servidores de réplicas involucrados ni la instalación tampoco de memoria caché.

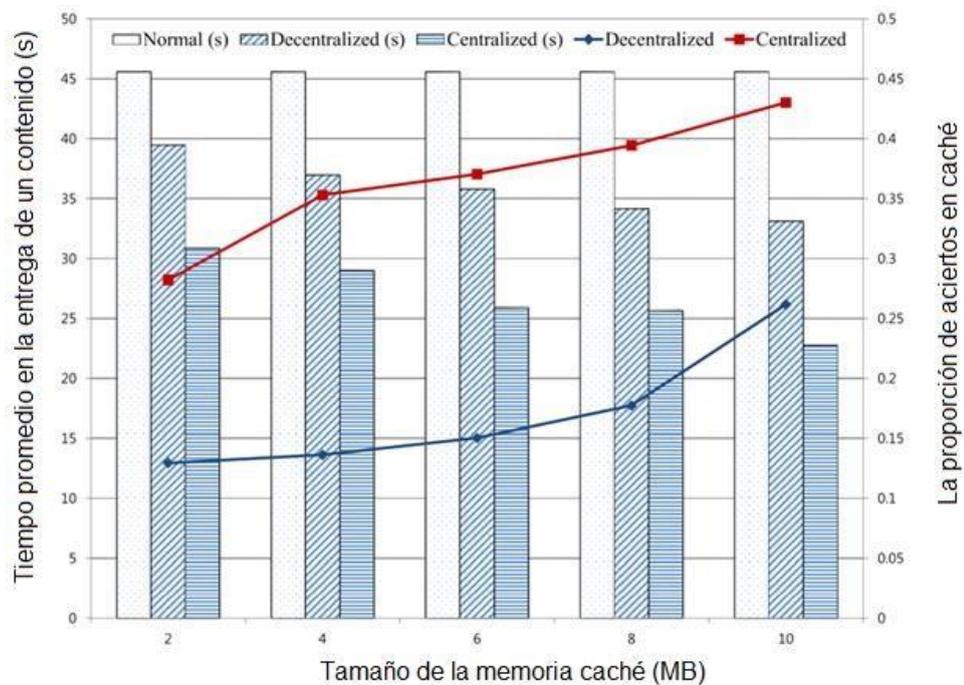


Figura 1.11: Como se administra la caché de una manera centralizada con SDN [21]

De manera centralizada con OpenFlow en SDN se observa que otorga un superior rendimiento dentro de la red, en la entrega a los usuarios de los distintos tipos de contenido.

CAPÍTULO 2

2. DISEÑO DE LA ESTRUCTURA DE ESTUDIO PARA IMPLEMENTAR LAS CARACTERÍSTICAS DE SDN A UNA CDN

2.1 Conceptos prioritarios y funcionalidad: Características de las SDN y el protocolo OpenFlow

2.1.1 Arquitectura de red SDN

Las redes definidas por software, Software Define Network o SDN, se trata de mecanismos programables para virtualizar las redes de comunicaciones, desvinculando en plano de control que es el software, del plano de datos que es el hardware. SDN es una clase de arquitectura de red, en la cual [3] el control puede ser programado, ya que es separado de la parte encargada de redireccionar los puertos y esta independencia de la etapa de control otorga un cambio para dirigir la red de manera virtual.

La figura 2.1 muestra el principio en que se basa la arquitectura de una red definida por software. Con SDN se obtiene el funcionamiento de acuerdo a una programación establecida, la cual consigue un control centralizado, esto por medio de aplicaciones conformadas por APIs, con esta etapa de control integrada en esta arquitectura SDN se logra gestionar tanto la red como sus equipos de una manera más eficiente. [4]

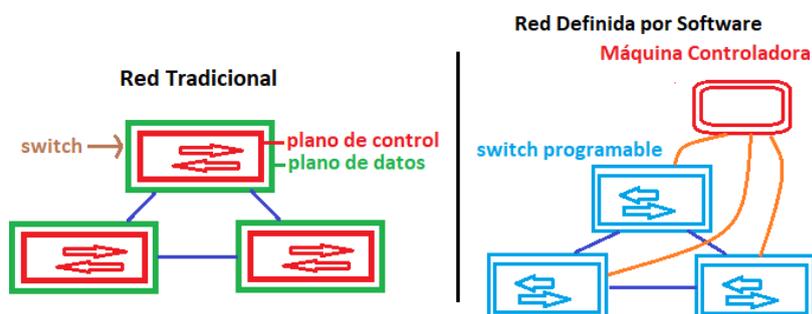


Figura 2.1: Principio de una arquitectura de Red Definida por Software

La tabla 1 nos muestra algunas de las características que otorga a una red la arquitectura SDN:

Reducción de costo	Se disminuye el costo tanto en equipos como en operadores de la red
Red ágil	En menos tiempo se provee a los usuarios de los servicios y aplicaciones
Red programable	La red es controlada mediante software, puede configurar políticas y mejorar la seguridad de la red
Control centralizado	Se logra administrar y controlar eficientemente la red
Desvincula el hardware	Se interactúa con la red sin hardware, ahora mediante API's
Automatización	La programación puede ser modificada de acuerdo a la variación de la infraestructura en la red de comunicación
Virtualización	Los servicios de red se centralizan a través de un software, en lugar de que cada función este asignada a un equipo físico respectivamente dentro la red
Escalabilidad	En los recursos se logra la escalabilidad, porque ya los diferentes equipos no cuentan con una lógica de control individual

Tabla 1: Características de una SDN

2.1.2 Funcionamiento de la arquitectura de red SDN

En el funcionamiento de una SDN están presentes dos conceptos muy importantes, la API y las capas que integran su estructura. La interfaz de programación de aplicaciones es la que logra comunicar al controlador con las aplicaciones externas y las capas de control, datos y aplicación, dirigen como se transportan los datos en los equipos de red, son los equipos físicos y virtuales, los servicios y aplicaciones inmersos en la red, respectivamente.

APIs: Para poder entablar la comunicación en una red definida por software, existe la llamada interfaz de programación de aplicaciones o API, que se sitúa entre el plano de control y el plano de datos, esta interfaz comunica ambos planos, realiza esta comunicación mediante software. La API utilizada en SDN se la puede ejecutar por diferentes tecnologías, como por ejemplo el protocolo OpenFlow, que es el protocolo empleado para crear la arquitectura y el desarrollo de SDN en la actualidad. [5]

Capas de una SDN: Las SDN basan generalmente su funcionamiento en tres capas, la capa de control, la capa de datos y la capa de aplicación, la explicación de cada capa se muestra a continuación:

La capa de control: Esta capa o nivel de control basa su trabajo en un controlador [6], el cual se encarga del resolver la fluidez de los paquetes, de la gestión y configurar la red.

La capa de datos: Esta capa establece la movilidad del paquete de un lugar a otro de la red [6], en esta capa están los equipos de manera física o virtual, que son los llamados nodos en una red. El movimiento del tráfico en las SDN se genera mediante estos nodos de red.

El proceso realizado en esta arquitectura, por la capa de control y la capa de datos, da lugar al establecimiento de una tercera capa, la capa de aplicación:

La capa de aplicación: Dentro de esta capa existen los programas, de los cuales depende las decisiones que se establecen en la red, ya que estas aplicaciones van a interactuar con los requerimientos del usuario y también con las condiciones del servicio y de la administración. [6]

La figura 2.2 muestra la interacción de las tres capas, arquitectura de la red SDN, aquí se observa también las interfaces de programación de aplicaciones (API) ubicadas en la dirección norte, las cuales son utilizadas para admitir la comunicación que se establece con el controlador de SDN y todas las aplicaciones que se efectúan dentro de la red. Las interfaces de programación de aplicaciones ubicadas en la dirección sur, a su vez, se encargan de la habilitación de la comunicación que entabla en controlador SDN con los diferentes nodos presentes en la red. [6]

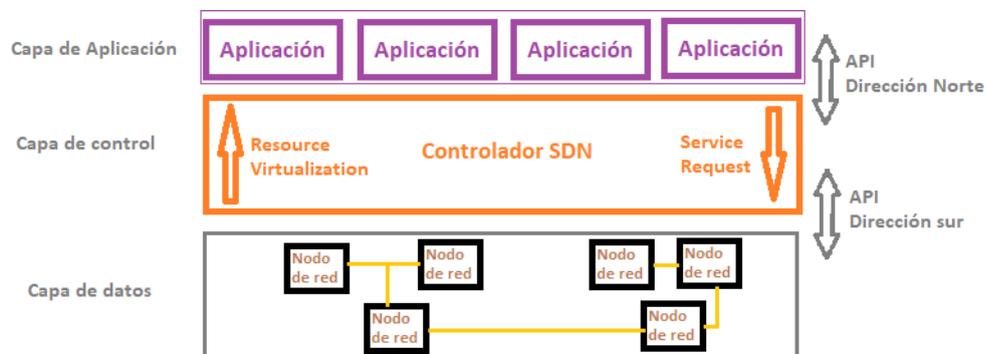


Figura 2.2: Capas de SDN

2.1.3 Protocolo OpenFlow para SDN

Es un protocolo que se encarga de centralizar el envío de los paquetes en los servidores y el control de los datos, en lugar de que cada conmutador tenga su software independiente, para realizar esta actividad en una red.

La ONF, fundación cuyas siglas significan en inglés Open Networking Foundation, es una organización financiada por algunas empresas, que se encarga de fomentar la creación de las redes con arquitectura SDN, el desarrollo de estas y de sus componentes como el conocido OpenFlow, el protocolo que controla los switches y realiza una administración centralizada en la red.

El protocolo OpenFlow mediante su interfaz otorga en la red al operador realizar la interacción de equipos y dispositivos de distintos fabricantes y poder decidir sin afectar en esto, la marca de los equipos involucrados en la red. Esta característica también es utilizada para los encargados de realizar las aplicaciones en la red. [7]

Es importante comprender que en el funcionamiento de una red definida por software, [8] el tráfico generado puede ser otorgado por el administrador, sin tener la necesidad de involucrarse parcialmente con los equipos de red que realizan la conmutación, ya que este se puede ejecutar de una forma centralizada desde la capa de control, teniendo la capacidad de realizar cambios de políticas en los equipos de conmutación dependiendo de la situación que se presente, controlando minuciosamente las características

de los paquetes. El protocolo para SDN, OpenFlow, le da la facultad a su administrador de dirigir tablas de enrutamiento desde algún lugar, esto lo hace recolectando constantemente los datos de los equipos de la red y conociendo su funcionamiento, para luego poder dirigir todo su tráfico, basándose en la difusión de comandos. El protocolo OpenFlow puede desarrollar su funcionamiento con diferentes lenguajes de programación, como el ya conocido Python, logrando con esto poder determinar sus propios programas para basar su comportamiento y ejecutar cambios cuando sea necesario.

La tabla 2 muestra las características que el protocolo OpenFlow otorga a las SDN:

Protocolo OpenFlow de SDN
Trabaja con los switches, permitiendo realizar funciones sin tener que recurrir a los fabricantes de los equipos de red.
Establece la comunicación con el equipo y controlador.
Desvincula el plano de datos del de control, lo que optimiza el rendimiento y mejora la gestión de la red.
Es programable, lo que ayuda a implementar nuevas mejoras en la red.
Funciona de forma directa con el envío de datos en los equipos de red directos o virtuales.

Tabla 2: Características del protocolo OpenFlow

La figura 2.3 muestra la función de la tecnología OpenFlow en la red SDN, en los equipos de red se conserva la separación del plano de datos con el plano de control, pero también interviene otra característica, el plano de control no está en los equipos utilizados para el tráfico de la red, este se encuentra en una sola máquina, para gestionar de manera centralizada las reglas para la conmutación, que intervienen en el funcionamiento de la red.

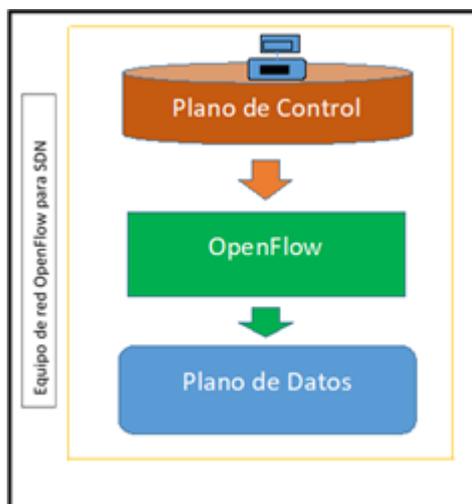


Figura 2.3: Función de OpenFlow en SDN

2.2 Conceptos prioritarios y funcionalidad: Características de la tecnología CDN

La CDN o Content Delivery Network, red de distribución de contenido, se refiere a un conjunto de servidores ubicados en diferentes puntos del mundo, que proveen a los usuarios visitantes contenido web de forma rápida, escogiendo el servidor más cercano, principalmente para descarga de audio y video, logrando de esta manera un servicio eficiente y una experiencia agradable en el visitante.

La figura 2.4 muestra un diagrama de una arquitectura CDN.

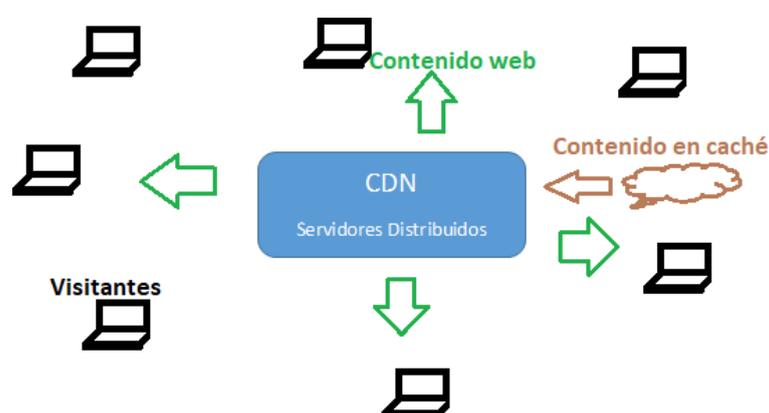


Figura 2.4: La arquitectura de una Red de distribución de contenido

2.2.1 La utilidad que entrega CDN

Los servidores de CDN están ubicados en diferentes lugares y abarcan información como video o imágenes, que es almacenada en más servidores de diferentes lugares del mundo. Por tal motivo con CDN se puede entregar a los usuarios de manera rápida la información requerida obviando dificultades relacionadas al tráfico o la cantidad de veces que se solicita esta, consiguiendo que la red no se sature, principalmente donde se inició la petición. [9]

Algunas características importantes de las CDN se muestran [9] a continuación:

Potenciar el procedimiento de entregar al visitante cuando ha solicitado algún contenido.

Se logra ajustar mejor la velocidad en las páginas web, disminuyendo el tiempo de carga.

Logra disminuir recursos en el servidor que acoge página web, mediante la segmentación.

Maneja de buena forma los mayores picos de tráfico.

2.2.2 Ventajas de utilizar CDN

Las Redes de Distribución de Contenido brindan muchas ventajas a los visitantes de sitios web, como está indicado a continuación: [10]

Administra el espacio del servidor para las elevadas cantidades de tráfico.

Permite almacenar en la memoria caché el contenido estático que se suministra a los usuarios, logrando focalizar al servidor en producir el contenido dinámico.

El rendimiento de la web se optimiza, la CDN efectúa gran parte del proceso, esto provoca disminuir el gasto del ancho de banda de una manera apreciable en el servidor.

Suprime las distancias en el mundo, se puede entregar a los usuarios de manera fácil el contenido requerido, como si estuvieran en el sitio de donde es procedente dicha información.

Resguardan la web de los llamados ataques informáticos.

Con la utilización de CDN disminuye el costo de los requerimientos de los servidores, así como también de la memoria necesaria, algunas veces existe la posibilidad de encontrarse sin costo y proveyendo de efectos excelentes en el rendimiento.

2.3 Descripción de las ventajas de aplicar SDN a las CDN

Las ventajas de aplicar a las redes de distribución de contenidos multimedia (CDN) el protocolo Open Flow de SDN y sus añadidos se basan en que SDN desvincula el plano de control del plano de datos, algunas de estas ventajas son:

El estudio realizado en [11], analizando la figura 1.1 y en [12] la figura 1.2 en el capítulo 1, permite concluir que SDN logra gestionar, de mejor manera, la red de distribución de contenido, ya que teniendo centralizado el control existe un conocimiento general de la red, lo que conlleva a tomar mejores decisiones referentes al control de la red. También mediante la programación se puede implementar nuevos protocolos con mayor grado de eficiencia para el funcionamiento del plano de datos.

El estudio realizado en [14], analizando la figura 1.4 en el capítulo 1, permite concluir que con SDN todos los equipos que están usándose en la red, es decir el hardware de la red se vuelve menos complejo y esto a su vez implica disminuir los costos cuando se necesita hacer modificaciones del mismo.

El estudio realizado en [13], analizando la figura 1.3 en el capítulo 1, permite concluir que con la arquitectura SDN se recurre al controlador cuando se necesita modificar la red, esto hace tener a la red más flexibilidad y a la vez poder ser más fácil su ejecución. Se puede hacer cambios de solo una parte del hardware, la que se desea cambiar, solo la parte necesaria. También no

depender del software de los fabricantes de equipos de la red usados en CDN, los controladores SDN son compatibles con equipos de diferente origen.

El estudio realizado en [21], analizando la figura 1.11 en el capítulo 1, permite concluir que con SDN se logra controlar de forma dinámica todo el tráfico de la red, dependiendo de los cambios requeridos.

El estudio realizado en [15], analizando la figura 1.5 en el capítulo 1, permite concluir que en SDN se pueden programar APIs para cualquier cambio que se requiera en la red.

2.4 Análisis del mejoramiento de la QoS de las Redes de Distribución de contenidos Multimedia (CDN) usando el protocolo Open Flow de SDN y sus añadidos

Las CDN otorgan muchos beneficios al tráfico de la red global, pero también a causa de la demanda del ancho de banda, que es cada vez mayor, existen elementos propios de CDN que necesitan ser mejorados, el análisis de estos se indica a continuación:

En base al estudio efectuado en [19], analizando la figura 1.9 en el capítulo 1, se establece que la dependencia cada vez mayor de nuevos servicios, como: la demanda de video, audio, descargas web por parte de los usuarios es cada vez mayor. Se concluye entonces que, con SDN estos servicios se vuelven más ágiles gracias a su arquitectura, en donde los paquetes se dirigen no al equipo sino al controlador y este da la orden de destino, seleccionando una mejor ruta mediante OpenFlow.

En base al estudio efectuado en [21], analizando la figura 1.11 en el capítulo 1 se establece que los dispositivos, es decir, el hardware de diferentes marcas, puede llegar a convertirse en un problema en el desarrollo de la CDN, porque este crece cada vez más. Se concluye entonces que con SDN se puede administrar, de forma centralizada, la red con software programable, para realizar la gestión de manera óptima y dinámica. También, en lo referente al aumento de probabilidad de error, es decir, al convertirse en una red más grande pueden surgir errores e inconvenientes en la gestión de una CDN. Se

concluye entonces que, si existe un problema en un nodo de alguna parte de la red, se corrige el error de forma más rápida.

En base al estudio efectuado en [18], analizando la figura 1.8 en el capítulo 1, se establece que los nuevos requerimientos de negocios actuales conllevan a la CDN a una compleja implementación. Se concluye entonces que, SDN otorga facilidad para implementar servicios de red, logrando tener escalabilidad y dinamismo.

En base al estudio efectuado en [13], analizando la figura 1.3 en el capítulo 1 se establece que, con SDN se puede disminuir la presencia de virus en un contenido en una CDN, ya que los usuarios o visitantes son susceptibles a virus informáticos. Se concluye entonces que, de acuerdo a las reglas establecidas por OpenFlow en su funcionamiento, para el trayecto del paquete, se pudiera mitigar este problema.

2.5 Trabajos relacionados acerca de mejoramiento de CDN mediante SDN

Existen trabajos relacionados referente a la implementación de redes de distribución de contenidos basadas en la arquitectura SDN y su programación con OpenFlow, el objetivo de estos trabajos se describe a continuación:

Creación de redes mediante emuladores y simuladores, pero se presentan dificultades en la práctica, donde se utilicen ciertos equipos como switches, porque la tecnología de estos aún no se traslada o no se ha acoplado al sistema de SDN y también existe algunas dificultades relacionadas con los controladores de SDN.

Mejorar la funcionalidad de los servidores DNS presentes en una CDN, esto basado en el uso controladores SDN, pero existe problemas de acoplamiento con los equipos de la red.

También se han realizado estudios donde exponen ejemplos de redes tradicionales funcionando con OpenFlow, se plantean infraestructuras con switches SDN para gestionar la conectividad y conocer el estado de las topologías. Estudios acerca del uso de la arquitectura SDN en el proceso de entrega de contenido a usuarios en equipos móviles. Para implementar políticas

en el desempeño de los datos de la red y mitigar la deficiencia de la calidad en la operación de una CDN. Disminuir retrasos en la entrega del contenido a un usuario que accede a los servicios y aplicaciones.

En este trabajo se expone un análisis de importantes aplicaciones de la arquitectura SDN, para el mejor desempeño de una red de entrega de contenido y se plantean conclusiones de sus características esenciales, para el desarrollo de nuevos experimentos futuros.

CAPÍTULO 3

3. ANÁLISIS DE RESULTADOS DE LA APLICACIÓN DE LAS CARACTERÍSTICAS DE SDN PARA MEJORAR EL RENDIMIENTO DE CDN

3.1 Resultados de la aplicación del protocolo OpenFlow para CDN

Los resultados obtenidos para el funcionamiento de una CDN utilizando el protocolo OpenFlow, se describen a continuación:

- El estudio efectuado en [21], de acuerdo a la figura 1.11 en el capítulo 1, permite concluir que, el resultado de montar un entorno de SDN con la programación de sus controladores para analizar el flujo de datos en la red, usando un simulador con máquina virtual, armada una topología con switches y hosts, logra analizar como controlar, de una manera centralizada, el tráfico en la red.
- El estudio efectuado en [16], de acuerdo a la figura 1.6 y en [17] la figura 1.7 en el capítulo 1, permite concluir que, en SDN el protocolo OpenFlow en la capa de control programa los mecanismos para dirigir el tráfico de la red, de esta manera resulta una gestión de superior calidad en la red basada en los equipos que se dispone.
- El estudio efectuado en [19], de acuerdo a la figura 1.9 en el capítulo 1, permite concluir que los switches OpenFlow en la red CDN reciben el paquete y realizan el reenvío hacia otro puerto haciendo cambios, si es necesario, con las características en la tabla de flujo, llamados campos, como son counters y priority.
- El estudio efectuado en [15], de acuerdo a la figura 1.5 en el capítulo 1, permite concluir que los contadores de OpenFlow en el switch de la red CDN verifican problemas en el comportamiento de la red, esto en tiempo real.

- El estudio efectuado en [20], de acuerdo a la figura 1.10 y en [18] con la figura 1.8 en el capítulo 1, permite concluir que en la CDN el DNS se basa en servidores que cambian su identificación por dirección IP, entonces, cuando un visitante solicita un contenido de la web, SDN elige un servidor apropiado y realiza la comunicación con el usuario. Esta implementación hace eficiente a la CDN.

3.2 Resultados de las características de SDN mediante un emulador

Con la utilización de un emulador como Mininet se puede montar una red de distribución de contenido en una arquitectura SDN, reemplazando el equipo tradicional con un solo equipo, una computadora personal. También, teniendo a disposición importante información que existe en internet sobre plataformas virtuales, para experimentar comportamientos de redes con switches, compatibles con OpenFlow para SDN.

3.2.1 Características de Mininet para emular una red basada en Software

Con el emulador Mininet se puede analizar el comportamiento de una SDN mediante las herramientas que este suministra.

Primero se requiere instalar una máquina virtual (ver anexo 1), para lo cual se puede escoger virtual Box.

Luego en la máquina virtual se instala Ubuntu y posteriormente en Ubuntu el emulador Mininet (ver anexo 2).

Se arma una red y se emula su funcionamiento, por ejemplo, como la mostrada en la figura 3.1, realizada en Miniedit, con un controlador, 10 switches y 6 máquinas.

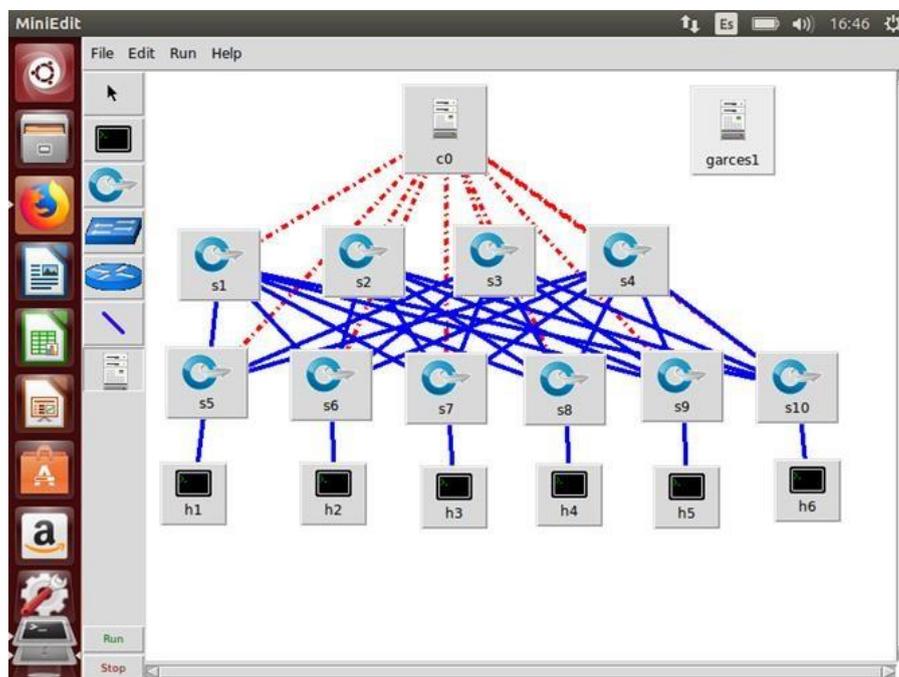


Figura 3.1: Red con un controlador diseñada en Mininet

En esta red se utiliza el controlador floodlight que es de OpenFlow, en el cual se configura la ip de floodlight y su puerto, que es 6353, como se muestra en la figura 3.2:

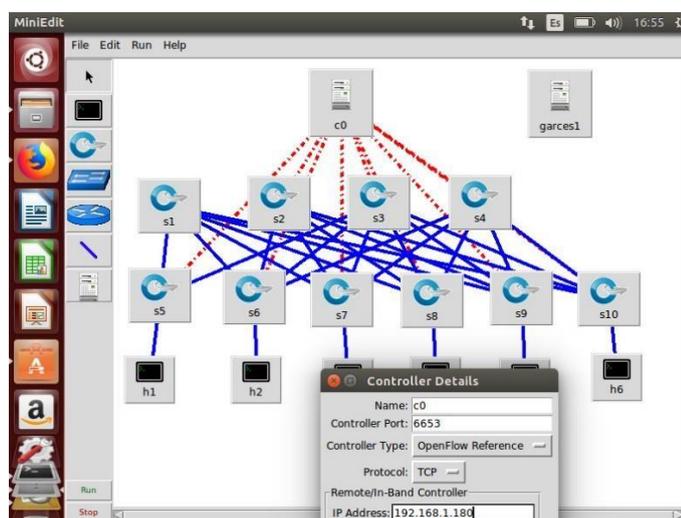


Figura 3.2: Puerto y dirección ip del controlador en emulador Mininet

Se ejecuta la red con run y mediante el comando nodes se verifica los switches y también los hosts de la red, como muestra la figura 3.3:

```

Terminal
garces@garces-VirtualBox: ~/mininet/custom
garces@garces-VirtualBox:~/mininet$ cd custom/
garces@garces-VirtualBox:~/mininet/custom$ ls
cgarcesm9.py  garces14.py  garces7.py  README
cgarcesmen.py  garces21.py  garcescarlos.py  simulacion2.py
cgarcesm.py  garces2.py  garcesexpo.py  simulacion.py
garces10.py  garces4.py  garcesmendozaCarlos.py  topo-2sw-2host.py
garces11.py  garces5.py  garces.py
garces12.py  garces6.py  practicagarces.py
garces@garces-VirtualBox:~/mininet/custom$ sudo python garcesmendozaCarlos.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h3 h1 h6 h5 h4 h2
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 h6 s1 s10 s2 s3 s4 s5 s6 s7 s8 s9
mininet>

```

Figura 3.3: Topología ejecutada en Mininet

En internet, en el navegador se busca con la dirección ip el controlador floodlight, se accede y se verifica los componentes como switches y hosts.

Se puede comprobar que el controlador es el encargado del análisis del tráfico y que OpenFlow selecciona la mejor ruta, ejecutando un ping entre dos hosts de la red, de esta manera los paquetes se dirigen al controlador no a los equipos, cuando se produce un nuevo tráfico, este es enviado inicialmente al controlador y floodlight mediante su aplicación de learning switch, va asignando entradas, para que el controlador pueda asignar flujos que son dirigidos a los switches, posteriormente los siguientes paquetes van a ser redireccionados por los propios switches.

3.3 Resultados de análisis de las potencialidades de SDN para mejorar la QoS de CDN en la interacción de ambas tecnologías.

El análisis realizado en este trabajo, en base a estudios de diferentes aplicaciones de la arquitectura SDN hacia el mejor desempeño de una CDN, permite establecer las siguientes conclusiones finales, referente a la interacción de ambas tecnologías:

La configuración individual que se debe realizar en los switches de diversos fabricantes en una CDN, se reduce con el controlador basado en la arquitectura SDN, permitiendo una gestión globalizada.

Para suplir el ancho de banda requerido por los usuarios en una CDN, la utilidad que aporta SDN es una de las soluciones, ya que los programadores pueden elaborar algoritmos necesarios en su implementación.

Frente a la saturación que está expuesta una CDN, la centralización de SDN juega un papel importante, principalmente en el encaminamiento de la red desde el controlador.

Los costos de equipos y operadores se vuelven mayores al hacerse más compleja la red, con SDN la red no necesita de la misma estructura, ya que la mayor parte de la implementación está basada en software.

Con una cantidad de servidores DNS en una CDN se complica resolver problemas y mantener controlada la red. En una red basada en SDN que utiliza switches con tecnología OpenFlow, el plano de control se encarga de la trayectoria de cada flujo en cada equipo, memorizando las rutas de todos los nodos de la red.

En la tabla 3 se observa un resumen del rendimiento comparativo entre las dos arquitecturas:

Comparación del rendimiento de CDN con y sin la utilización de SDN	
Sin SDN	Con SDN
La necesidad de mayor escalabilidad en la red y también la falta de la flexibilidad en el mecanismo de operación de los paquetes de variado contenido, dificultan la operación de los dispositivos de red con respecto a su rendimiento, a causa del tráfico que crece cada vez más.	Se suministra el contenido con mayor rapidez y de una manera más eficiente, logrando utilizar mucho mejor los recursos que cuenta la red y con esto poder mitigar el tráfico presente en la red.
Cuando un usuario solicita un contenido se necesita realizar el proceso de buscar al servidor DNS.	Con el protocolo OpenFlow, el controlador de SDN, se tiene el detalle completo de donde está situado el contenido .
Se reduce la QoS en la red, a causa de atender un mayor número de paquetes en un pequeño lapso de tiempo, que a su vez produce inestabilidad en otros puntos de la red.	Se implementa políticas para la conducción de los datos en la parte del software, mas no en el hardware de la red.
El acceso a las redes sociales mediante dispositivos móviles crece cada vez más y esto produce que consuma los recursos reducidos de los equipos móviles.	SDN y su arquitectura proveen a la transmisión de un rendimiento más eficiente, ya que las decisiones las realiza el controlador de manera centralizada, en lo que respecta al enrutamiento y no el equipo móvil.
Disminuye la QoS a los visitantes cuando desean acceder a las distintas aplicaciones, por causa del congestionamiento de la red.	Al tener los niveles de control y de reenvío de datos por separado propuestos por la arquitectura SDN, se espera conseguir de forma dinámica información a partir de la etapa de control, para obtener control del flujo del tráfico, mejorando así la QoS.

Tabla 3: Comparación del rendimiento de CDN con SDN

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

De acuerdo al estudio realizado en este trabajo acerca de la QoS de una CDN con OpenFlow de SDN, se puede llegar a las siguientes conclusiones:

Las redes definidas por software proponen, mediante su funcionamiento, mejorar el rendimiento a las redes tradicionales, principalmente a las redes de distribución de contenido, que cada vez se torna más compleja su infraestructura, producto del aumento de visitantes que acceden a servicios y aplicaciones.

Entre las características más importantes está el control centralizado, el uso de la programación como eje fundamental en su estructura, una menor dependencia de hardware, una reducción de costos en el equipamiento cuando se necesita aumentar los servicios, una mejor gestión programable con el uso controladores estandarizando los equipos la configuración.

El protocolo OpenFlow provee la inteligencia entre el hardware, que es el nivel de datos y el software, que es el nivel de control, en la separación de estos dos planos en una red de distribución de contenido, la corrección de errores se puede realizar con mayor rapidez en una destinada área de la red, mediante este trabajo realizado se puede conocer mejor las cualidades de OpenFlow y SDN para futuras implementaciones.

RECOMENDACIONES

Existen muchos estudios que se han realizado acerca de la ejecución de la arquitectura SDN hacia redes actuales como CDN, con simuladores y emuladores, pero todavía falta más experimentación principalmente en el desempeño del hardware y en la seguridad, por lo cual se recomienda profundizar el análisis de resultados en estos dos aspectos, con equipos reales y emuladores, verificando también, en más estudios teóricos, cada una de las características que se espera que otorgue SDN para el futuro desempeño de las redes de telecomunicaciones.

BIBLIOGRAFÍA

- [1] Scielo. (2016, Septiembre 19). Conceptualización de SDN. [En línea]. Disponible en: http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-67122016000100029.
- [2] Digital guide ionos. (2019, Junio 12). SDN. [En línea]. Disponible en: <https://www.ionos.es/digitalguide/servidores/know-how/software-defined-network/>.
- [3] Logiscali Now. (2013, Marzo). Software Defined Network. Autor Carlos Spera. [En línea]. Disponible en: <https://www.la.logicalis.com/globalassets/latin-america/logicalisnow/revista-20/lnow20-nota-42-45.pdf>.
- [4] Ciena. (2017, Septiembre 09). SDN. [En línea]. Disponible en: https://www.ciena.com.mx/insights/what-is/What-is-SDN_es_LA.html.
- [5] Arsys. (2020, Marzo 18). Qué son las redes definidas por software. Autor Tao Yuan. [En línea]. Disponible en: <https://www.arsys.es/blog/sdn/>.
- [6] Pandorafms. (2018, Junio 28). Cómo las redes definidas por Software cambian nuestra visión sobre las redes. Autor Alexander La rosa. [En línea]. Disponible en: <https://pandorafms.com/blog/es/redes-definidas-por-software/>.
- [7] Revista Channel news. (2012, Mayo). OpenFlow/SDN. [En línea]. Disponible en: <http://www.emb.cl/channelnews/articulo.mvc?xid=2048&ni=openflow/sdn-la-innovacion-llega-a-la-red#:~:text=Para%20superar%20estas%20dificultades%20se,revolucionarias%20de%20los%20%C3%BAltimos%20a%C3%B1os..>
- [8] Mira telecomunicaciones. (2019, Septiembre 30). Software-defined networking. [En línea]. Disponible en: <https://miratelecomunicacions.com/blog/noticias-sector/sdn-software-defined-networking-que-es-y-como-funciona/>.
- [9] MarTech Forum. (2019, Julio 02). Qué son las Redes de Distribución de Contenido. [En línea]. Disponible en:

-<https://www.martechforum.com/articulo/distribucion-de-contenido-cdn/>.

[10] Nerion. (2020). CDN. [En línea]. Disponible en:

-<https://www.nerion.es/blog/utilizar-cdn-mejora-el-rendimiento-de-tu-web/>.

[11] K. Nisar, I. Welch, R. Hassan, A. H. Sodhro, and S. Pirbhulal, "A Survey on the Architecture, Application, and Security of Software Defined Networking," *Internet of Things*, p. 100289, 2020, doi: 10.1016/j.iot.2020.100289.

[12] J. Lai, Q. Fu, and T. Moors, "Using SDN and NFV to enhance request rerouting in ISP-CDN collaborations," *Comput. Networks*, vol. 113, pp. 176–187, 2017, doi: 10.1016/j.comnet.2016.12.010.

[13] M. Kobayashi et al., "Maturing of OpenFlow and Software-defined Networking through deployments," *Comput. Networks*, vol. 61, pp. 151–175, 2014, doi: 10.1016/j.bjp.2013.10.011.

[14] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and OpenFlow: From concept to implementation," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014, doi: 10.1109/COMST.2014.2326417.

[15] H. A. Tran, S. Souihi, D. Tran, and A. Mellouk, "MABRESE: A New Server Selection Method for Smart SDN-Based CDN Architecture," *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 1012–1015, 2019, doi: 10.1109/LCOMM.2019.2907948.

[16] D. Bhat, J. Anderson, P. Ruth, M. Zink, and K. Keahey, "Application-based QoE support with P4 and OpenFlow," *INFOCOM 2019 - IEEE Conf. Comput. Commun. Work. INFOCOM WKSHPs 2019*, pp. 817–823, 2019, doi: 10.1109/INFOCOMW.2019.8845180.

[17] F. Li et al., "Software-Defined Networking-Assisted Content Delivery at Edge of Mobile Social Networks," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8122–8132, 2020, doi: 10.1109/jiot.2020.3009854.

[18] J. Chandrakanth, P. Chollangi, and C. H. Lung, "Content Distribution Networks Using Software Defined Networks," *Proc. - 2nd Int. Conf. Trust. Syst. Their Appl. TSA 2015*, pp. 44–50, 2015, doi: 10.1109/TSA.2015.18.

[19] H. Nam, D. Calin, and H. Schulzrinne, "Intelligent content delivery over wireless via SDN," 2015 IEEE Wirel. Commun. Netw. Conf. WCNC 2015, pp. 2185–2190, 2015, doi: 10.1109/WCNC.2015.7127806.

[20] M. Bala Krishna and P. Lorenz, "Proactive replication scheme for resilient content delivery in software defined networks," 2019 IEEE Glob. Commun. Conf. GLOBECOM 2019 - Proc., pp. 1–6, 2019, doi: 10.1109/GLOBECOM38437.2019.9013441.

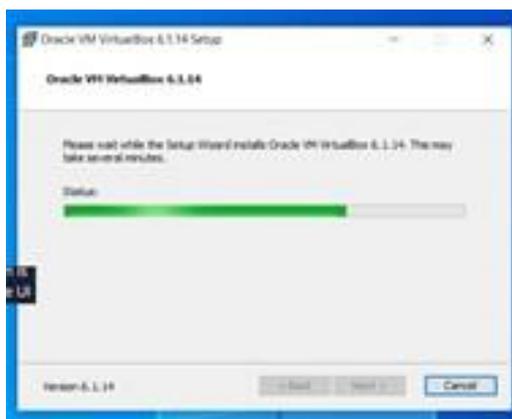
[21] D. Chang, M. Kwak, N. Choi, T. Kwon, and Y. Choi, "C-flow: An efficient content delivery framework with OpenFlow," Int. Conf. Inf. Netw., pp. 270–275, 2014, doi: 10.1109/ICOIN.2014.6799480.

ANEXO 1

La instalación de una máquina virtual

Para descargar el programa de virtual Box se accede a la página:
<https://www.virtualbox.org/>

Luego se escoge el VirtualBox 6.1.14 platform packages, para Windows, dependiendo el requerimiento de la pc a utilizar, como se muestra en la imagen:



ANEXO 2

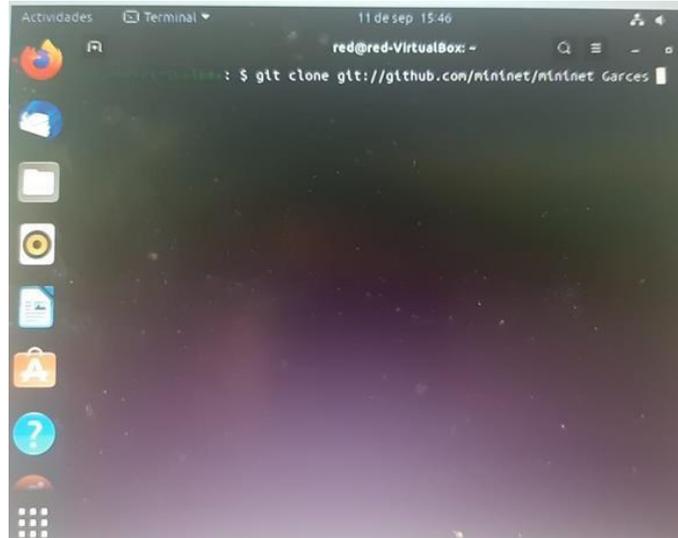
La instalación de Ubuntu y Mininet

Para descargar el programa del sistema operativo Ubuntu se accede a la página: <https://ubuntu.com/download/desktop#download>

Luego de escoger download Ubuntu Desktop, se lo instala dentro de virtual Box.

Finalmente se instala Mininet en Ubuntu, se accede a la página: <http://mininet.org/download/> , en la imagen se muestra el inicio de la

instalación, donde se copia los comandos desde la página de Mininet hacia el Terminal de Ubuntu:



ANEXO 3

Uso de los comandos básicos de Mininet

Se puede buscar el archivo .py guardado en custom y ejecutar la red mediante sudo python nombre.py.

```

terminal
garces@garces-VirtualBox: ~/mininet/custom
[sudo] password for garces:
MiniEdit running against Mininet 2.2.1
topo=None
garces@garces-VirtualBox:~/mininet/examples$ cd ..
garces@garces-VirtualBox:~/mininet$ ls
bin          custom  doc      LICENSE  mininet.egg-info  mnexec.1  setup.
build       debian  examples Makefile  mn.1             mnexec.c  util
CONTRIBUTORS dist    INSTALL  mininet  mnexec           README.md
garces@garces-VirtualBox:~/mininet$ cd custom/
garces@garces-VirtualBox:~/mininet/custom$ ls
cgarcesmen.py  garces2.py  garces4.py  garces.py  README  topo-2sw-2host.py
garces@garces-VirtualBox:~/mininet/custom$ sudo python garces4.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h2 h3 h1
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet>

```

Con el comando net se observa la red que se crea en Mininet:

```
Terminal
garces@garces-VirtualBox: ~/mininet/custom
CONTRIBUTORS dist INSTALL mininet mnexec README.md
garces@garces-VirtualBox:~/mininet$ cd custom/
garces@garces-VirtualBox:~/mininet/custom$ ls
cgarcesmen.py garces2.py garces4.py garces.py README topo-2sw-2host.py
garces@garces-VirtualBox:~/mininet/custom$ sudo python garces4.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h2 h3 h1
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet> net
h2 h2-eth0:s1-eth3
h3 h3-eth0:s2-eth2
h1 h1-eth0:s1-eth2
s2 lo: s2-eth1:s1-eth1 s2-eth2:h3-eth0
s1 lo: s1-eth1:s2-eth1 s1-eth2:h1-eth0 s1-eth3:h2-eth0
c0
mininet>
```

Con el comando nodes se comprueba que están habilitados los nodos que se crearon:

```
garces@garces-VirtualBox: ~/mininet/custom
cgarcesmen.py garces2.py garces4.py garces.py README topo-2sw-2host.
garces@garces-VirtualBox:~/mininet/custom$ sudo python garces4.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h2 h3 h1
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet> net
h2 h2-eth0:s1-eth3
h3 h3-eth0:s2-eth2
h1 h1-eth0:s1-eth2
s2 lo: s2-eth1:s1-eth1 s2-eth2:h3-eth0
s1 lo: s1-eth1:s2-eth1 s1-eth2:h1-eth0 s1-eth3:h2-eth0
c0
mininet> nodes
available nodes are:
c0 h1 h2 h3 s1 s2
mininet>
```

El comando Dump muestra las direcciones ip y otras características de cada uno de los nodos:

```
Terminal
garces@garces-VirtualBox: ~/mininet/custom
64 bytes from 10.0.0.3: icmp_seq=19 ttl=64 time=0.043 ms
64 bytes from 10.0.0.3: icmp_seq=20 ttl=64 time=0.051 ms
64 bytes from 10.0.0.3: icmp_seq=21 ttl=64 time=0.087 ms
64 bytes from 10.0.0.3: icmp_seq=22 ttl=64 time=0.106 ms
64 bytes from 10.0.0.3: icmp_seq=23 ttl=64 time=0.108 ms
64 bytes from 10.0.0.3: icmp_seq=24 ttl=64 time=0.092 ms
64 bytes from 10.0.0.3: icmp_seq=25 ttl=64 time=0.090 ms
64 bytes from 10.0.0.3: icmp_seq=26 ttl=64 time=0.069 ms
64 bytes from 10.0.0.3: icmp_seq=27 ttl=64 time=0.056 ms
64 bytes from 10.0.0.3: icmp_seq=28 ttl=64 time=0.048 ms
64 bytes from 10.0.0.3: icmp_seq=29 ttl=64 time=0.064 ms
64 bytes from 10.0.0.3: icmp_seq=30 ttl=64 time=0.088 ms
^C
--- 10.0.0.3 ping statistics ---
30 packets transmitted, 30 received, 0% packet loss, time 29002ms
rtt min/avg/max/mdev = 0.043/0.183/2.596/0.460 ms
mininet> dump
<Host h2: h2-eth0:10.0.0.2 pid=2499>
<Host h3: h3-eth0:10.0.0.3 pid=2502>
<Host h1: h1-eth0:10.0.0.1 pid=2504>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None pid=2491>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=2494>
<Controller c0: 127.0.0.1:6633 pid=2483>
mininet>
```

Se comprueba conectividad mediante h1 ping -c 1 h2: se realiza entre ambos hosts un ping de manera controlada con un solo paquete:

```
Terminal
garces@garces-VirtualBox: ~/mininet/custom
--- 10.0.0.3 ping statistics ---
30 packets transmitted, 30 received, 0% packet loss, time 29002ms
rtt min/avg/max/mdev = 0.043/0.183/2.596/0.460 ms
mininet> dump
<Host h2: h2-eth0:10.0.0.2 pid=2499>
<Host h3: h3-eth0:10.0.0.3 pid=2502>
<Host h1: h1-eth0:10.0.0.1 pid=2504>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None pid=2491>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=2494>
<Controller c0: 127.0.0.1:6633 pid=2483>
mininet> h1 ps -a
  PID TTY          TIME CMD
 2478 pts/0        00:00:00 sudo
 2479 pts/0        00:00:00 python
 2580 pts/4        00:00:00 controller
 3048 pts/25       00:00:00 ps
mininet> h1 ping -c 1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=10.9 ms
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 10.931/10.931/10.931/0.000 ms
mininet>
```

En la red creada, se puede desactivar la comunicación entre el switch 1 y el host 1 con el comando: link s1 h1 down, así como también volver a entablar la comunicación usando link s1 h1 up:

```

Terminal
garces@garces-VirtualBox: ~/mininet/custom
mininet> h1 ps -a
  PID TTY          TIME CMD
 2478 pts/0    00:00:00 sudo
 2479 pts/0    00:00:00 python
 2580 pts/4    00:00:00 controller
 3048 pts/25   00:00:00 ps
mininet> h1 ping -c 1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=10.9 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 10.931/10.931/10.931/0.000 ms
mininet> pingall
*** Ping: testing ping reachability
h2 -> h3 h1
h3 -> h2 h1
h1 -> h2 h3
*** Results: 0% dropped (6/6 received)
mininet> pingpair
h2 -> h3
h3 -> h2
*** Results: 0% dropped (2/2 received)
mininet>

```

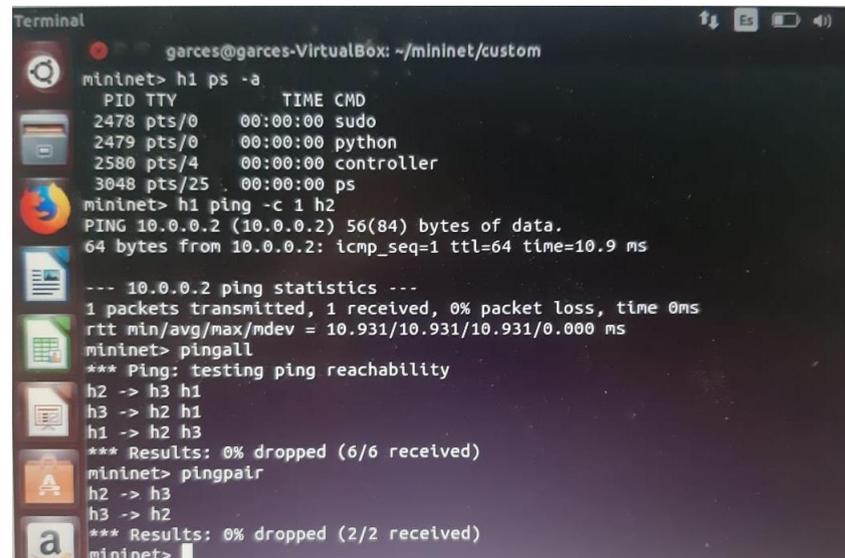
El ancho de banda se lo puede verificar con el uso del comando iperf, el cual mide esta característica:

```

Terminal
garces@garces-VirtualBox: ~/mininet/custom
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.059 ms
[64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.086 ms
[64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.077 ms
[64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.046 ms
[[64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.054 ms
[64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=0.042 ms
^[64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=0.039 ms
c64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=0.075 ms
^[64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=0.076 ms
^C
--- 10.0.0.2 ping statistics ---
29 packets transmitted, 29 received, 0% packet loss, time 2800ms
rtt min/avg/max/mdev = 0.035/0.150/2.052/0.375 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h2 and h1
*** Results: ['22.1 Gbits/sec', '22.1 Gbits/sec']
mininet>

```

Se puede comprobar comunicación entre dos hosts, con la utilidad del comando pingall y también con el comando pingpair:



```
Terminal
garces@garces-VirtualBox: ~/mininet/custom
mininet> h1 ps -a
  PID TTY          TIME CMD
 2478 pts/0        00:00:00 sudo
 2479 pts/0        00:00:00 python
 2580 pts/4        00:00:00 controller
 3048 pts/25       00:00:00 ps
mininet> h1 ping -c 1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=10.9 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 10.931/10.931/10.931/0.000 ms
mininet> pingall
*** Ping: testing ping reachability
h2 -> h3 h1
h3 -> h2 h1
h1 -> h2 h3
*** Results: 0% dropped (6/6 received)
mininet> pingpair
h2 -> h3
h3 -> h2
*** Results: 0% dropped (2/2 received)
mininet>
```

ABREVIATURAS Y SÍMBOLOGÍA

SDN	Software Define Network
CDN	Content Delivery Network
QoS	Quality of Service
DNS	Domain Name System
API	Application Programming Interface
ISP	Internet Service Provider
WAN	Wide Area Network
SD-WAN	Software Defined Wide Area Network
VRF	Virtual Routing and Forwarding
LTE	Long Term Evolution
NFV	Network Function Virtualization
SCS	Smart Controller Server