ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

"POLÍTICAS DE QoS EN REDES EMPRESARIALES PARA EL ANÁLISIS DE RENDIMIENTO, EN ENTORNOS CONVENCIONALES Y SDN"

MARCELO DANIEL CRIOLLO BUSTAMANTE

Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito parcial para la obtención del grado de:

MAGÍSTER EN INTERCONECTIVIDAD DE REDES

Riobamba – Ecuador

Marzo - 2020

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN

EL TRIBUNAL DE TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, denominado: "POLÍTICAS DE QoS EN REDES EMPRESARIALES PARA EL ANALISIS DE RENDIMIENTO, EN ENTORNOS CONVENCIONALES Y SDN", de responsabilidad del Sr. Marcelo Daniel Criollo Bustamante, ha sido prolijamente revisado y se autoriza su presentación.

Tribunal:

Ing. Marco Vinicio Ramos Valencia; Mag.
PRESIDENTE

Ing. Ruth Genoveva Barba Vera; Mag.
DIRECTOR

Ing. Norma Viviana Aimacaña Toledo; Mag.
MIEMBRO DEL TRIBUNAL

Ing. Jonny Israel Guaiña Yungan; Mag.
MIEMBRO DEL TRIBUNAL

Riobamba, marzo 2020

DERECHOS INTELECTUALES

Yo, Marcelo Daniel Criollo Bustamante, soy responsable de las ideas, doctrinas y resultados expuestos en el Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, y que el patrimonio intelectual generado del mismo pertenece exclusivamente a la Escuela Superior Politécnica de Chimborazo.

CRIOLLO BUSTAMANTE.
N° Cédula 0602892432

DECLARACIÓN DE AUTENTICIDAD

Yo, Marcelo Daniel Criollo Bustamante, declaro que el presente proyecto de investigación, es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como Autor, asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación de Maestría.

MARCELO DANIEL CRIOLLO BUSTAMANTE.

N°. Cédula: 0602892432

DEDICATORIA

A mis padres Ana y Humberto por su apoyo incondicional y amor brindado día a día.

Marcelo.

AGRADECIMIENTO

A la Escuela Superior Politécnica de Chimborazo, al Instituto de Postgrado, a los miembros del tribunal, a la coordinación del programa de Maestría de Interconectividad de Redes; por haber sido entes fundamentales en el transcurso de la Maestría. A los amigos y docentes que me acompañaron durante las largas jornadas de estudio. A Dios y a mis padres por su incondicional amor.

Marcelo Criollo.

TABLA DE CONTENDO

ÍNDICE DE TABLAS.

ÍNDICE DE FIGURAS

ÍNDICE DE ABREVIATURAS

AAA. Autenticación, Autorización y Contabilidad.

API. (Applicacion programming interfaz), la interfaz de programación de aplicaciones es un conjunto de librerías que ofrecen acceso a ciertos servicios para la comunicación entre componentes de software.

AVAYA. Industria de telecomunicaciones pionera en productos SDN.

DPCTL. Es una utilidad de línea de comando, útil para la interacción con switches Open Flow

EPN. Envolved Programmable Network.

IPS. Instruder prevention system (Sistema de prevención de intrusos), protege los sistemas computacionales de ataques.

IETF. (Internet Engineering Task Force) es una organización internacional de normalización que tiene por objetivo contribuir con la ingeniería de internet. Creado en los EEUU en 1986.

JAVA. Es un lenguaje de programación multiplataforma que usa el paradigma WORA (write once, run anywhere). Donde el código puede ser ejecutado en varias plataformas.

NFV. (Netwok Function Virtualization), virtualización de funciones de red. Esto está relacionado con la arquitectura de redes, ya que virtualiza funciones de nodo de red en bloques que se pueden interconectar y crear servicios de comunicación.

OPEN STACK. Es un sistema operativo de código abierto, con la estructura de computación en la nube, el que es capaz de controlar recurso de computación, almacenamiento y redes a través de un centro de datos.

OPEN FLOW. Es un estándar abierto que permite a los investigadores ejecuten protocolos experimentales en el campo de las redes.

OPEN VSWITCH. Es un software de código abierto, diseñado para utilizarse como un switch virtual en servidores virtualizados, se encuentra bajo licencia Apache 2.0.

ONF. (Open Networking foundation) Es un consorcio sin fines de lucro que impulsa la transformación de la infraestructura de red

QoS. (Quality of service), calidad de servicio.

SDN. Software defined network. Una nueva concepción de gestión de redes la cual se separa el plano de datos del plano de control.

RESUMEN

El objetivo del trabajo de titulación fue analizar el rendimiento que genera la asignación de políticas de QoS, entre entornos de redes convencionales y SDN, en una red empresarial. Este estudio da relevancia a la tecnología de las redes programables como alternativa para infraestructuras de red físicas, con el fin de analizar el comportamiento de esta arquitectura en escenarios reales. Open Flow permite la separación entre el plano de control y el de datos, permitiendo la flexibilización en el desarrollo de nuevas alternativas de software que se ejecuta en el plano de control. El proceso consiste en la implantación de los escenarios en dos entornos como son: SDN y convencional, para luego asignar políticas de QoS en cada uno de ellos; posterior a lo que se inyectará tráfico en la red utilizando la aplicación D-ITG, para emular el comportamiento de un nodo de red empresarial. A continuación, por medio de herramientas de medición se evaluarán los parámetros de QoS como: el ancho de banda, latencia, jitter y pérdida de paquetes. Finalmente, se efectuará el análisis de los resultados utilizando el método estadístico T-Student. Partiendo de los resultados estadísticos, en función a los parámetros de estudio; en el servidor encontramos una diferencia de: ancho de banda 9,12%, latencia 5,86%, jitter 1,02% y pérdida de paquetes 8,7%. En relación a los parámetros en el cliente hallamos una diferencia de: ancho de banda 6,44%, latencia 1,52%, jitter 6,28%, pérdida de paquetes con 6,92%. Se concluye que al aplicar políticas de QoS en los dos escenarios tanto en el escenario convencional como en el SDN, no presenta una diferencia significativa tanto en el servidor como en el cliente. Como propuesta se desarrolló una guía metodológica, en las que se incluyen las consideraciones que se debe aplicar en la práctica, producto de las pruebas realizadas.

Palabras clave: <TECNOLOGIA Y CIENCIAS DE LA INGENIERIA>, <REDES DE COMUNICACIONES>, <REDES CONVENCIONALES>, <OPENFLOW>, <QoS CALIDAD DE SERVICIO>, <REDES DEFINIDAS POR SOFTWARE>, <ANCHO DE BANDA>, < LATENCIA>, <VARIANZA DEL TIEMPO (JITTER)>, <PÉRDIDA DE PAQUETES>.

SUMMARY

The objective of the titling work was to analyze the performance generated by the allocation of QoS policies, between conventional network environments and SDN, in an enterprise network. This study gives relevance to the technology of programmable networks as an alternative for physical network infrastructures, in order to analyze the behavior of this architecture in real scenarios. Open Flow allows the separation between the control plane and the data plane, allowing flexibility in the development of new software alternatives that run in the control plane.

The process consists of the implementation of the scenarios in two environments such as: SDN and conventional, to then assign QoS policies in each of them; after what traffic will be injected into the network using the D-ITG application, to emulate the behavior of a business network node. Then, through measuring tools, the QoS parameters such as bandwidth, latency, jitter and packet loss will be evaluated. Finally, the analysis of the results will be carried out using the T-Student statistical method. Starting from the statistical results, according to the study parameters; in the server we find a difference of: bandwidth 9.12%, latency 5.86%, jitter 1.02% and packet loss 8.7%. In relation to the parameters in the client we find a difference of: width band 6.44% latency 1.52%, jitter 6.28%, packet loss with 6.92%. It is concluded that when applying Qos policies in both scenarios in both the conventional scenario and the SDN, it does not present a significant difference in both the server and the client. As a proposal, I develop a methodological guide, which includes the considerations that should be applied as a result of the test performed.

Keywords: <ENGINEERING TECHNOLOGY AND SCIENCES>, <COMMUNICATIONS NETWORKS>, <CONVENTIONAL NETWORKS>, <OPENFLOW>, <QoS QUALITY OF SERVICE>, <NETWORKS DEFINED BY SOFTWARE>, <BANDWIDTH>, <LATENCY>, <TIME VARIANCE (JITTER)>, <LOSS OF PACKAGES>

CAPÍTULO I

1 INTRODUCCIÓN

Nuestra sociedad se encuentra interconectada digitalmente, con objetos relacionados al Internet, fenómeno conocido en términos tecnológicos como el Internet of Things (**IoT**), adicionando una gran variedad y cantidad de dispositivos inteligentes, esto fomenta la necesidad de solucionar sus problemas inherentes y mejorar el rendimiento de las redes; en especial en un ambiente empresarial en donde se exige que las aplicaciones y servicios funcionen correctamente y en tiempo real, ya que se deben realizar determinados procesos tales como: enlaces cliente-servidor, transacciones electrónicas, procesamiento de datos en línea, entre otros.

Lo cual ha obligado a los investigadores crear nuevos paradigmas y plataformas tecnológicas, a través del desarrollo de redes programables, utilizando el protocolo OpenFlow; lo que ha permitido el procesamiento de paquetes en múltiples redes simultáneamente, con dispositivos heterogéneos, es decir sin intervención de fabricantes exclusivos; esta innovadora tecnología consiste en que cuando una aplicación realiza una petición a la red, los recursos de la red se asignan a un canal dedicado para esa aplicación, mediante configuración automática de flujos y prioridades del puerto para mantener la garantía prometida de la red; por lo tanto el objetivo de este proyecto será investigar las diferentes opciones para facilitar la gestión de las redes con calidad de servicio (QoS) tanto para redes SDN como para infraestructuras de redes convencionales, las cuales se prestan también para implementar políticas y mejorar el rendimiento.

Como administradores es necesario tomar en cuenta el tema de calidad de servicio, lo cual permite tener un control del tráfico que fluye en la red, pudiendo solucionar determinadas fallas, mediante la implementación de ciertas políticas de QoS. El presente trabajo analiza los problemas más comunes que se tiene dentro de una organización y propone opciones de mejora con la finalidad de aprovechar las capacidades de las redes, utilizando los recursos de una manera eficaz y eficiente; para lo cual se realizarán dos escenarios, tanto para un entorno convencional como en un entorno de SDN y se presentarán resultados los cuales aportarán con directrices para la toma de decisiones al momento de administrar una red.

El desarrollo de este proyecto, está organizado en 4 capítulos:

Capítulo I. Aspectos generales, que incluye, la descripción del problema y la formulación de objetivos generales y específicos.

Capítulo II. Marco referencial, se centra en presentar el estado del arte, en el ámbito de redes convencionales como de las redes con SDN, con sus respectivas definiciones, características, arquitectura, requerimientos actuales, herramientas y calidad de servicio.

Capítulo III. En este capítulo se realizará una descripción de la metodología, el enfoque de la investigación, la población, variable e indicador, matriz de contingencias, parámetros utilizados. Arquitectura y diseño de escenarios (convencional y con SDN). Requerimientos funcionales. Implementación de las topologías de red, direccionamiento IP, herramientas de hardware y software, lenguajes de programación, **API**, controladores, entre otros.

Capítulo IV. Con los dos escenarios implementados, en este capítulo se detalla la fase de pruebas y verificación de funcionamiento de las redes en el entorno real. Posteriormente se obtendrán resultados, los cuales serán analizados en base a indicadores establecidos, para evidenciar los beneficios de aplicar la calidad de servicio en cada una de las redes propuestas.

Capítulo V. Consecuentemente se procederá a la presentación de la propuesta, especificaciones, descripción de la metodología, comparativa costo beneficio, tabla resumen estadístico y la guía metodológica como un aporte para los administradores de red.

Finalmente se presentarán las conclusiones, recomendaciones y trabajos futuros.

1.1 Planteamiento del problema

El problema más evidente en la actualidad en la mayoría de empresas, es la utilización de tecnologías anticuadas, más aún si en su infraestructura existen equipos de distintas marcas, esto genera la necesidad de poseer un alto nivel de conocimientos; situación que no es fácil ni accesible para todos los administradores de red, los que se ven limitados al proveer soluciones o soporte técnico especializado. Una infraestructura con las características ya mencionadas presentan distintas dificultades debido a principalmente a dos razones: La primera es debido a la "Masificación del Internet", ya que su evolución amerita cada vez de más recursos para satisfacer a los usuarios y la segunda razón tiene relación con la complejidad lógica de la red a la hora de configurar de forma manual políticas y aplicar configuraciones para así tener un control de la red más personalizado, en correspondencia a requerimientos por parte de las empresas.

Por otra parte, los fabricantes de equipos de conectividad brindan soluciones propietarias, en la que cada uno de ellos posee su propio software e Interfaz de Programación de Aplicación, lo cual restringe el diseño y gestión entre equipos de diferentes marcas; ocasionando que la red se mantenga estática e inflexible. Si a esto se le agrega el problema de una arquitectura organizada de forma vertical, en donde el plano de datos como el plano de control se encuentra combinados dentro de cada uno de los dispositivos de red, se tiene como resultado una reducción en términos de innovación y evolución de la infraestructura de red. (Planas, 2016).

Ante estos problemas descritos, se introduce a las redes programables, que a pesar que existen desde hace tiempo, solo hace unos pocos años se ha empezado a implementarlas, como alternativa de solución para el fuerte crecimiento de los servicios ofrecidos vía internet y a la gran dependencia que tiene la sociedad actual con las telecomunicaciones, garantizando la demanda, la calidad del servicio y sobretodo gestionando de mejor manera los recursos de la red y optimizando su rendimiento.

1.2 Formulación del problema

¿La asignación de políticas de QoS en redes empresariales de entre entornos convencionales y SDN, cuál ofrece un mayor rendimiento?

1.3 Sistematización del problema

- ¿Cuáles son las características de las redes convencionales y las redes SDN?
- ¿Qué herramientas son las más adecuadas para el diseño de una red convencional y SDN?
- ¿Qué tipos de tráfico causan mayor congestión en una red empresarial?

- ¿Cuáles son los parámetros de evaluación y herramientas de medición de rendimiento?
- ¿Cuál es el rendimiento al implementar políticas de QoS, tanto en el entorno de redes convencionales como para redes SDN?
- ¿Cuál es el resultado del análisis del rendimiento?

1.4 Justificación de la investigación

Nuestra generación es la generación de la información ya que hacemos uso de las comunicaciones en muchas de nuestras actividades diarias; más usuarios necesitan acceder a servicios y aplicaciones, generando la necesidad de incrementar la cantidad de puntos de acceso en la red; la cual por obvias razones crece de forma exponencial, aumentando en su complejidad, así como en el esfuerzo para su eficiente gestión y control.

Con la finalidad de facilitar el control de las redes y gestionar servicios de manera eficiente se propone la aplicación de políticas de calidad de servicio tanto en un entorno de redes convencionales como en un entorno de redes SDN, realizando varias modificaciones en los equipos activos de red; permitiendo a los administradores responder de forma rápida a problemas de conexión, simplificando operaciones, separando el control de la red (plano de control) de las funciones de redireccionamiento de paquetes (plano de datos), en un ambiente empresarial.

El tener dos escenarios propuestos (convencional y SDN) aplicando políticas de calidad de servicio, se presta para evidenciar sus limitaciones, beneficios y alcances en lo referente a dar una respuesta a requerimientos de la red, lo cual para los operadores ayudará con directrices para que puedan tomar decisiones, realizar ciertas acciones tales como: configurar los equipos cuando sea necesario con un alto nivel de control, llevar un registro con información estadística en tiempo real, establecer políticas, programar diferentes configuraciones y servicios; ofreciendo un esquema totalmente dinámico sin depender de un fabricante específico, por lo que se podrá utilizar equipos más económicos, todo esto con el objetivo de disponer de un mayor control sobre el flujo de tráfico de la red mejorando su rendimiento.

De esta manera, el presente proyecto pretende ser un aporte para los administradores de red, permitiendo hacer que las infraestructuras de las redes sean escalables y flexibles; a través de un análisis que integrará conceptos de políticas de QoS, gestión centralizada y de programación, dando soluciones en un corto plazo a determinadas fallas en la red.

Por tal razón, se propone la implementación de políticas de calidad de servicio, para redes convencionales y haciendo uso de las redes definidas por software, con el fin de generar una

aplicación personalizada que provea mejor gestión en la red, en base a un estudio de la problemática actual de las redes empresariales, logrando mejorar su desempeño, de acuerdo a condiciones reales, a su nivel de carga y las restricciones de calidad de servicio de los flujos de tráfico que circulan por ella; como resultado se conseguirá ofrecer un mejor uso de los recursos de la red, generando una administración más eficiente y mejorando su rendimiento.

Es importante mencionar que este proyecto será enfocado al entorno de las redes empresariales, ya que existe una mayor complejidad al momento de gestionar el flujo del tráfico de red; las empresas tienen una enorme cantidad de datos por las prestaciones de servicios y aplicaciones que procesan día a día; como una referencia según Cisco Visual Networking Index expone que los cambios en las redes se acelerarán por el crecimiento exponencial del tráfico IP, por el uso de la nube, de la movilidad, del video y del tráfico M2M (Machine to Machine), por lo que la mayoría de organizaciones no están preparadas para soportar ésta revolución en sus infraestructuras (Cisco, 2017).

1.5 Objetivos de la investigación

1.5.1 General

- Analizar el rendimiento que genera la asignación de políticas de QoS, entre entornos de redes convencionales y SDN, dentro de una red empresarial.

1.5.2 Específicos

- Analizar las políticas de QoS aplicadas a redes SDN y redes convencionales, para determinar los requerimientos previos al desarrollo de una aplicación.
- Diseñar una aplicación basada en políticas de QoS en un ambiente empresarial partiendo del análisis previo.
- Implementar el uso de políticas de QoS, construyendo la aplicación para redes SDN y realizando la configuración respectiva en redes convencionales, aplicando en los escenarios propuestos.
- Evaluar la aplicación que implementa políticas de QoS en SDN, en contraposición a las redes convencionales; utilizando parámetros y herramientas de medición de redes, para verificar su funcionalidad.
- Desarrollar una guía de implementación de políticas de QoS como un apoyo a los administradores de red.

1.6 Planteamiento de hipótesis

A continuación, se plantea la hipótesis alternativa y la hipótesis nula:

- **Ha=** La propuesta de asignación de políticas de QoS generará mayor rendimiento en una red enterprise SDN que en un entorno CONVENCIONAL.

- **Ho=** La propuesta de asignación de políticas de QoS no generará mayor rendimiento en una red enterprise CONVENCIONAL que en un entorno SDN.

CAPÍTULO II

2 MARCO REFERENCIAL

2.1 Antecedentes del problema

El presente trabajo se centra en la asignación de políticas de calidad de servicio, en entornos convencionales y en SDN, enfocadas a las redes empresariales; las cuales se caracterizan por proveer de conectividad para que los usuarios pertenecientes a una determinada organización puedan intercambiar información, acceder a datos, beneficiarse de los servicios de procesamiento, aplicaciones y otros recursos.

Uno de los problemas inherentes a la infraestructura de red tradicional es el tráfico horizontal (este-oeste), tráfico generado por aplicaciones que hablan entre ellas: base de datos, etc.; este problema se intensifica cuando un mayor número de máquinas virtuales se ejecutan en servidores físicos, este tráfico es más frecuente por que la virtualización es aleatoria para la ubicación de los servidores. La virtualización acerca los límites del direccionamiento IP, llegando al máximo de Vlans de 4096, esto limita las opciones de administración, impidiendo que una máquina virtual pueda moverse al servidor menos cargado, si ese servidor está en algún otro clúster (Mallick, 2012).

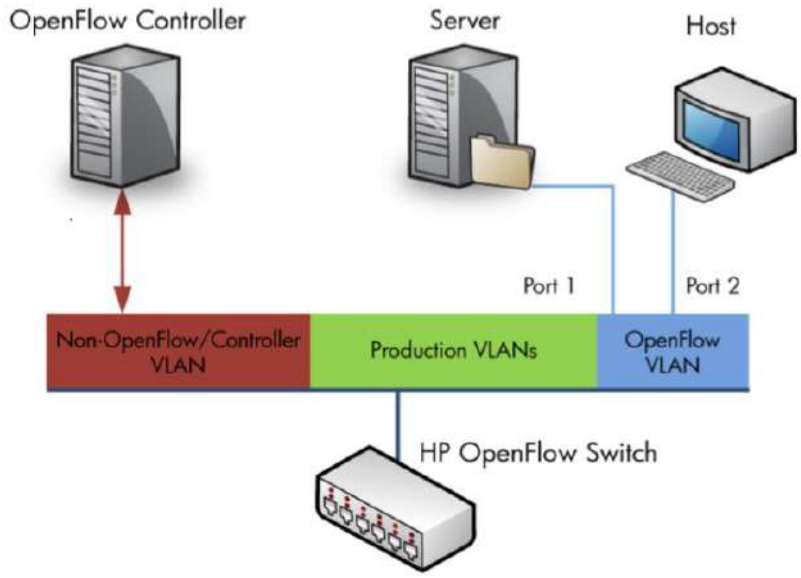


Figura 1-2: Tráfico Horizontal en infraestructura de redes de datos
Fuente: https://support.hpe.com/hpsc/doc/public/display?docId=emr_na-c05365339&docLocale=en_US

Otro inconveniente de los administradores de un centro de datos es que al querer realizar algún cambio en su arquitectura existente es difícil agregar nuevo equipamiento, lo que posiblemente implicaría la necesidad de otro conmutador de agregación o inclusive más puertos.

Por ende, para los administradores de red y para los fabricantes de equipos de networking se ha convertido en un desafío el tener la capacidad y los recursos para ofrecer en tiempo real servicios y aplicaciones sin tener interrupciones en la funcionalidad de la red y han propuesto que las empresas entren en la era de las redes programables o Evolved Programmable Network (EPN), para tener redes dinámicas, flexibles y programables; pero en cuestión de costos se complica para los administradores la adquisición de estos equipos. Por tal motivo se da solución a nivel de software para permitir que las aplicaciones se ajusten a su ingeniería de tráfico según sus requerimientos (Fanelli, 2016).

2.2 Nuevos comportamientos en las redes de datos

El Internet en sus inicios era utilizado por la mayoría de los usuarios para realizar transacciones simples, que consistían en transferencia de archivos, envío y recepción de correo electrónico y acceso remoto, pero al popularizarse el uso del Internet en la sociedad y las empresas, el número de usuarios creció exponencialmente, así como el tipo de aplicaciones que transitan hoy por la Internet el cual se ha diversificado considerablemente. En los comienzos de Internet, no existía la necesidad de manejar o priorizar el tráfico que circulaba en dicha red, por lo que no se percibía aún el concepto de QoS; esto hacía que toda la Internet funcionara bajo un sistema de "mejor esfuerzo" (best-effort). El protocolo IP trabajaba de una manera simple, ofreciendo únicamente el servicio de direccionamiento (Torre, 2017).

Actualmente, con la aparición de nuevos usos que se da a la tecnología, se añaden nuevos procedimientos para tratar el tráfico de las redes, adaptándose a las necesidades de las empresas, tales como:

- **Cambio en los patrones de tráfico.** Tiempo atrás, los datos que viajaban a través de las redes siempre fueron considerados generales, es decir, no existía clasificación en los mismos por servicios, sólo se implementaba seguridad a nivel de VLAN para agrupar las redes por usuarios y no por aplicaciones. Hoy en día, existe una mayor importancia en la clasificación de los datos que viajan por servicios o aplicación, ya que las prioridades en la disponibilidad de los datos se rigen por el tipo de servicio que proveen y su importancia para la empresa.

- **En las Aplicaciones de Red.** Cuando las aplicaciones de red eran de tipo cliente servidor, la información se obtenía de una sola fuente y sólo iba en una vía, donde el cliente consumía lo que el servidor enviaba; actualmente la información es solicitada por muchos usuarios al mismo tiempo, y así mismo, la información es proveída por muchos servidores, creando muchas vías de comunicación entre usuarios finales y servidores, lo cual complica la administración de las redes de datos, con herramientas básicas.

- **Generalización de los servicios de información.** Con el incremento del uso de teléfonos inteligentes, existen muchos más equipos que se conectan a la red, por lo que es necesario proveer servicios rápidos y efectivos a estos nuevos dispositivos, procurando no afectar a los usuarios actuales de la red.

- **Servicios en la nube.** La nube agrega nuevas variables a los servicios prestados en una red, donde la seguridad de la información es muy importante y la auditoría de la red es una medida rutinaria; este tipo de servicios requieren escalabilidad dinámica para poder crecer dependiendo de las necesidades.

2.3 Fundamentos teóricos

2.3.1 Definición de QoS

La calidad de servicio es definida por la Unión Internacional de Telecomunicaciones (UIT) como "El efecto global de la calidad de funcionamiento de un servicio que determina el grado de satisfacción de un usuario de dicho servicio". Esta definición deja en función del cliente cuáles son las características y comportamientos que lo satisfacen (minimizar el retardo, asegurar velocidad mínima, priorizar tráficos, etc., ya que cada uno puede tener unas necesidades diferentes.

Es por ello que se plantea el problema de poder ofrecer una calidad de servicio dinámica, que pueda moldearse en todo momento a los requisitos del usuario y que no se ofrezca como un servicio rígido. La calidad de servicio cobra importancia cuando la capacidad de la red es insuficiente, especialmente para aplicaciones de Streaming, multimedia, voz sobre IP, juegos en línea e IP-TV en tiempo real. Estos a menudo requieren velocidad de bits fija y son sensibles al retardo.

2.3.1.1 Calidad de servicio en las redes

Como ya se ha mencionado anteriormente en este documento, el incremento de dispositivos y conexiones a la red, han ocasionado una mayor demanda de servicios y por lo tanto, una exigencia cada vez mayor por parte de los usuarios; la implementación de la calidad de servicio prioriza el tráfico de red, obteniendo un mayor ancho de banda al tráfico más prioritario. Esto pretende resolver problemas relacionados con la pérdida de paquetes de datos, latencia, jitter, ancho de banda, garantizando un correcto funcionamiento en la red.

Entre las características de QoS está la clasificación y marcado, para diferenciar el tipo de tráfico en base al comportamiento de la red; dependiendo de cada infraestructura se podrá crear políticas para mejorar el desempeño de la red. La calidad de servicio en las redes ofrece una predicción de tiempos de respuesta para flujos de paquetes, operaciones, transacciones, etc., así como también gestiona las capacidades de las aplicaciones sensibles al jitter (audio y vídeo). Al existir congestión en la red se puede tener un control de la pérdida de paquetes y para los operadores de red, se facilita la configuración de las propiedades del tráfico (López, 2015). Si observamos las redes de telecomunicaciones tradicionales con conmutación de circuitos, la QoS está formada por varios factores, que pueden dividirse en dos grupos: factores "humanos" y "técnicos", como se muestra en la Tabla 1-2.

Tabla 1-2: Factores de calidad de servicio

Factores humanos	Factores técnicos
Estabilidad de la calidad del servicio	Confiabilidad
Disponibilidad de líneas de suscriptor	Capacidad de expansión
Tiempos de espera	Eficacia
Tiempos de eliminación de fallas	Mantenimiento del sistema
Información del suscriptor	Congestión esperando
Estabilidad del funcionamiento del sistema	Calidad de transmisión

Fuente: https://www.netlab.tkk.fi/~puhuri/htyo/Tik-110.551/iwork/iwork.html

Con la introducción de nuevas aplicaciones en tiempo real, crece el nivel de tráfico en las redes y se produce congestión, la entrega de todos los paquetes se ralentiza y más aún, si la congestión llega a ser severa, en ese caso se descartan paquetes para aliviar dicha congestión.

Como se lo observa en la Figura 2-2, el tiempo de servicio y rendimiento en función de la carga, proporcionando una perspectiva de la utilidad de la aplicación de la QoS.

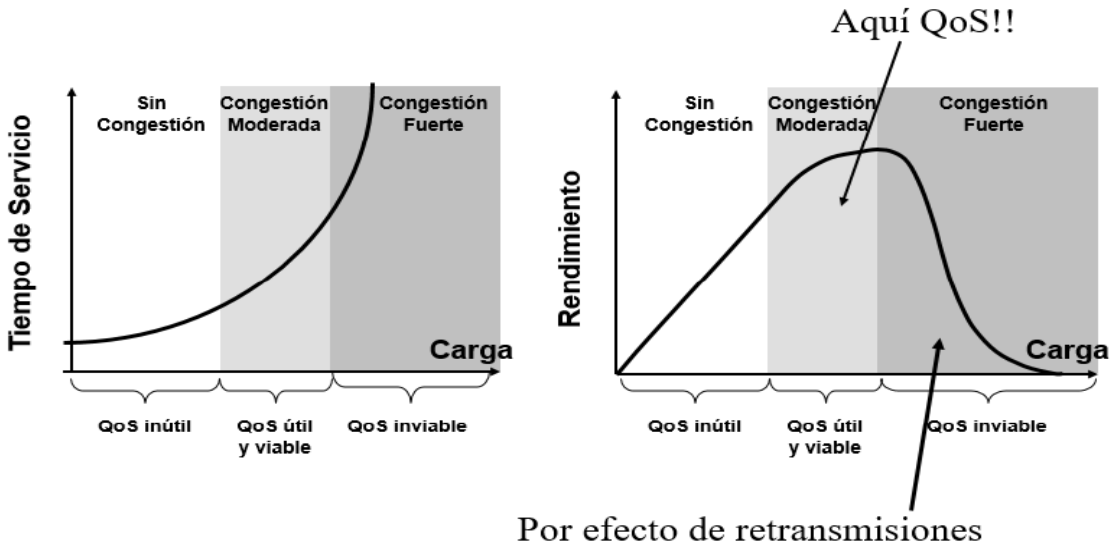


Figura 2-2: Efectos de la congestión en el tiempo, de servicio y rendimiento
Fuente: http://biblioteca.unitecnologica.edu.co/notas/tesis/0045090.pdf

Las soluciones de calidad de servicio (QoS) mejoran el rendimiento de los servicios electrónicos extremo a extremo tal como lo percibe el usuario final. Se mejoran diferentes parámetros entre ellos: el retardo, jitter y la pérdida de paquetes. Para garantizar un mejor nivel de calidad de servicio para el nivel de tráfico se debe establecer Políticas de Calidad de Servicio como, por ejemplo:

- Asignar ancho de banda en forma diferenciada
- Evitar y/o administrar la congestión en la red
- Manejar prioridades de acuerdo al tipo de tráfico

Todas las redes pueden aprovechar aspectos de QoS para una eficiencia óptima, sea par una red para una pequeña empresa, empresa o un proveedor ISP; permitiendo proporcionar soluciones de QoS de extremo a extremo a través de equipos heterogéneos donde adopten un enfoque de configuración de QoS diferente para cada tecnología. A medida que las redes sean más grandes, como es el caso de la red empresarial, llevan aplicaciones más complejas y de misión crítica y experimentan un mayor tráfico de aplicaciones multimedia web, la calidad de servicio sirve para priorizar este tráfico para asegurar que cada aplicación obtenga el servicio que requiere.

2.3.1.2 Consideraciones para definir QoS

Es importante conocer los valores aceptables, de los requerimientos de calidad de servicio de las aplicaciones que funcionan en la red. En la Tabla 2-2, se muestran los valores requeridos para el funcionamiento de las aplicaciones con calidad de servicio. Los valores fueron tomados según la

tabla del apéndice I del cuadro objetivos de calidad de funcionamiento para aplicaciones según la UIT-T Rec. G.1010 (ITU-T, 2011).

Tabla 2-2: Requerimientos de funcionamiento de aplicaciones.

Aplicación	Delay	Jitter	Pérdida de paquetes
Voz	Preferido <150 ms Límite <400ms	<1ms	<3%
Video	Preferido <150 ms Límite <400ms	N/A	<1%
Base de datos	<250ms	N/A	Nula
Internet	Preferido<2s/página Aceptable <4s/página	N/A	Nula

Fuente: UIT-T Rec. G.1010

El grupo de trabajo Metrics Performance Metrics (IPPM) de IETF está trabajando para definir métricas para el rendimiento de Internet. Existen algunas diferencias en la terminología que considera el tiempo entre las definiciones del UIT-T y las definiciones del grupo de trabajo de IPPM. Un breve resumen de las diferencias se presenta en la Tabla 3-2.

Tabla 3-2: Diferencias entre IPPM e ITU-T con respecto a las métricas de tiempo.

IPPM	ITU-T	Definición
Sincronización	Error de tiempo	Diferencia de dos relojes
Exactitud	Error de tiempo de UTC	Diferencia a tiempo real
Resolución	Periodo de muestreo	La precesión del reloj
Sesgar	Deriva de tiempo	Cambio en la sincronización o en la precisión

Fuente: https://www.netlab.tkk.fi/~puhuri/htyo/Tik-110.551/iwork/iwork.html

2.3.1.3 Parámetros de QoS en Sistemas de comunicación

Para proporcionar garantías en la transmisión de determinados flujos de los datos, la ISO (1994) introdujo el concepto QoS (Quality of Service); la cual se utiliza para medir la calidad del servicio ofrecido por una red de comunicación. El rendimiento de un servicio provisto por la red se gestiona a través de un conjunto de parámetros de QoS (por ejemplo, retraso, variación en el retraso, pérdida de paquetes, ancho de banda, tasa de bits), los mismos que se describen a continuación: (Silva, 2017).

- **Ancho de banda.** El ancho de banda está determinado por el medio de transmisión que se está utilizado, tanto por los protocolos, por la distancia y velocidad de conmutación entre los nodos intermediarios.

- **Flujo (Throughput).** La capacidad de tráfico, se denomina flujo de tráfico, la más utilizada es la capacidad de tráfico en un enlace, que se define como número máximo de bits en la capa IP (ocho veces el número de octetos en todos los paquetes IP recibidos correctamente) que se pueden transmitir correctamente entre dos hosts en un mismo intervalo de tiempo. Es importante recordar que el enlace corresponde a una conexión directa entre dos hosts. Cuando se habla de un camino completo entre el origen y el destino, la capacidad del camino es el equivalente a la capacidad del enlace de menor capacidad de este camino, que puede variar con el tiempo debido a variaciones en el nivel de congestión de los enlaces.

  El flujo en una red es el ancho de banda efectivo o la tasa de bits efectiva, en otras palabras, es la cantidad de datos transmitidos con éxito por unidad de tiempo. Puede ser definida como la diferencia entre la tasa de bits del enlace y los overheads (sobrecarga). En términos prácticos, las aplicaciones generan caudales que deben ser atendidos por la red, el caudal, en la mayoría de las redes sufren variaciones en el transcurso del tiempo; en algunas situaciones, el caudal puede cambiar rápidamente debido a las fallas en los nodos de la red o líneas o debido al ruido congestión cuando grandes flujos de datos son introducidos en la red.

- **Retardo (retraso).** Es considerado uno de los principales parámetros de rendimiento red. Denota el tiempo transcurrido para transmitir un bloque de datos de un emisor a un receptor, en la capa de aplicación, el retardo es la diferencia de tiempo (fin a fin) transcurrida entre la generación del dato en el transmisor y su presentación en el receptor. Esta diferencia incluye las parcelas procesamiento en los nodos intermedios (enrutadores, swiches y end-points), la disputa por el acceso al medio en los enlaces compartidos y el tiempo de propagación en el medio físico.

  Si la conexión entre transmisor y receptor involucra múltiples saltos, como es el común en redes conmutadas por los paquetes, la suma de todos los retrasos saltar a saltar más el retardo de procesamiento debe ser igual o inferior al retardo final-a-fin deseado; normalmente se desea limitar un cierto parámetro relativo a la curva de distribución del retardo, como un valor medio, máximo, o un porcentaje.

- **Variación del retardo (Jitter).** Corresponde a la variación del retardo a paquetes, los flujos son adicionalmente divididos en bloques de datos, y cada bloque se transmite en secuencia. Si la red es capaz de enviar todos los bloques con una secuencia uniforme, entonces cada bloque debería llegar al punto destino tras un retraso uniforme. Muchas redes no garantizan un retraso uniforme para sus usuarios.

  Variaciones en retraso son comunes, los retrasos en la transmisión son causados por muchos factores, tales como: diferencias de tiempo de procesamiento de paquetes, diferencias de tiempo de acceso a red y diferencias de tiempo de colas. Si las variaciones en los retrasos se deben a las imperfecciones del sistema en la red (software o hardware), o debido a las condiciones de tráfico dentro de la red, estas variaciones normalmente se llaman jitter. Jitter es la variación observada en el conjunto de valores de retardo de unidades de datos consecutivos.

- **Tasa de pérdidas de paquetes.** Es la razón entre la cantidad de paquetes perdidos y la cantidad de paquetes perdidos paquetes enviados. Los paquetes se pierden en la red por descarte en las filas de los nodos intermedios, o pueden ser corrompidos por colisión con otros paquetes en enlaces compartidos y aún por variación en el medio físico (teniendo, en este caso, relación con la tasa de errores de bits).

- **Acuerdos de niveles de servicio (SLA).** Contrato de servicios entre un proveedor de servicios y su cliente, el cual define las responsabilidades del proveedor en términos del nivel de funcionamiento de la red (rendimiento, tasa de perdidas, retrasos, variaciones) y la disponibilidad temporal, el método de medida, las consecuencias cuando los niveles de servicio no se consiguen o si los niveles de tráfico definidos son superados por el cliente, así como el precio de todos estos servicios.

2.3.1.4 Clases de Calidad de Servicio

Asignar calidad de servicio a determinados flujos es uno de los objetivos de este trabajo, en las redes actuales existe una tendencia de aumento del procesamiento efectuado por los switchs con el fin de descubrir qué tipo de aplicación generará un cierto flujo para asegurar que éste tenga sus necesidades aseguradas. También existe el problema de que algunos tipos de clasificadores crean problemas de intrusión de privacidad de los usuarios, esta situación se debe a que algunos clasificadores acceden a la información contenida en el paquete Deep Packet Inspection (DPI), estando así se puede analizar información privada.

Para diferenciar tipos de tráfico y para garantizar la calidad de servicio, se realiza un enfoque del tipo FCFS (first come first served), con la excepción del tráfico VoIP por tener características especiales. De este modo, ya no es necesaria una clasificación exhaustiva de la información, aliviando el procesamiento en los switchs, por medio de una arquitectura más escalable y no recurriendo a procesos que pueden considerarse menos éticos, manteniendo la garantía del tráfico existente. Con este propósito, se definieron cuatro clases de servicio, en las que cada una tiene un objetivo diferente (Cardoso, 2015) :

- **Clase 1.** Se garantiza automáticamente la calidad del servicio. Consiste en que un flujo perteneciente a la primera clase ha garantizado automáticamente su calidad de servicio, no siendo necesario verificar sus características. Esta clase protege los mensajes básicos para el funcionamiento de la red y el tráfico VoIP. La protección de los mensajes básicos de la red asegura que la red tiene sus funciones básicas a funcionar correctamente, incluso estando saturada. La garantía del tráfico VoIP, tiene como objetivo asegurar que este tipo de tráfico tenga sus garantías aseguradas. A un flujo que se ha asignado a la clase 1, tendrá como identificador una etiqueta VLan 0x, y se le asignará la garantía a lo largo del recorrido.

- **Clase 2.** Se garantiza la calidad de servicio al flujo después del análisis. A un flujo que se haya asignado a la clase 2, se garantiza una calidad de servicio durante su recorrido a lo largo de la red, la principal diferencia entre esta clase y la clase 1 es que, en lugar de ser se garantiza automáticamente, se asigna a través del switch de ingreso, su asignación es independiente del tipo de tráfico. Los flujos de esta clase se identifican mediante la etiqueta VLan 1x.

- **Clase 3.** Clase Best Effort. La clase 3 es la clase menos prioritaria entre las cuatro clases existentes. A un flujo asignado a esta clase, no se garantiza ninguna garantía. Al igual que en clase 2, un flujo se asigna a esta clase a través del switch de ingreso. Sin embargo, al contrario, a lo que sucede en la clase 2, se asigna cuando alguno de los conmutadores de la ruta de este flujo no proporciona ninguna garantía. Su identificador es una etiqueta VLan 2x.

- **Clase 4.** Clase de sondeo de características del flujo. Al llegar a un switch de ingreso, todos los flujos que no sean considerados clase 1, son colocados en esta clase. Es una clase de sondeo, y no se otorga ninguna garantía a ninguno de los sus flujos. Sin embargo, posee mayores garantías que la clase 3. Significa esto que, incluso y la red sobrecargada siempre que el tráfico con prioridad no se vea perjudicado, flujo en esta clase puede demostrar sus requisitos de ancho de banda, haciendo así la asignación de reserva para el flujo por parte del

switch de ingreso más correcto. Esta es asignada por los conmutadores controladores de ingreso. Un flujo en esta clase contiene una etiqueta VLan del estilo 3x.

2.3.1.5 Calidad asociada a las clases de servicios

Un servicio puede ser contratado con garantías de QoS, los proveedores de manera individual o colaborativa pueden proporcionar servicios con diferentes niveles de QoS, utilizando clasificaciones. Una clase de servicio (CoS - Class of Service), comprende una clasificación dada por el proveedor de aplicaciones con características de tráfico y rendimiento similares, por ejemplo, aplicaciones con requisitos de tiempo real como llamadas de voz y los requisitos de cada CoS de un proveedor, a su vez, se declaran utilizando especificaciones de QoS.

Los servicios se pueden ofrecer con diferentes niveles de QoS, de acuerdo con el valor que el consumidor está dispuesto a pagar. Los SLA (Service Level Agreement), para cada requisito de QoS, se debe definir un valor para ser utilizado como tolerable (exacto, mínimo, máximo) durante la realización de un contrato electrónico (Roberto, 2008).

2.3.1.6 Soluciones de Calidad del Servicio

Las principales soluciones de QoS son las siguientes: (Navarro, 2005)

INTSERV. Es basado en la reserva de recursos, la reserva se realiza mediante el protocolo de señalización de extremo a extremo RSVP (Protocolo de asignación de recursos) entre todos los hosts en la ruta entre origen y destino. La Arquitectura de Servicios Integrados (IntServ), definida por el IETF y desarrollada debido a la necesidad de garantizar la QoS de aplicaciones en tiempo real, como una conferencia multimedia y realidad virtual. Como el IntServ fue desarrollado para proveer QoS por flujo, es posible tener un control mayor sobre cada tráfico de la red. IntServ es caracterizado por la asignación de recursos para dos tipos de servicios: servicios garantizados para aplicaciones que requieren un retraso constante, y servicios de carga controlada para aplicaciones que requieren de seguridad y destacan el servicio de mejor esfuerzo (BE – Best Effort).

Algunas desventajas de IntServ son: todos los enrutadores deben tener noción del protocolo RSVP y ser capaces de señalar la QoS requerida; las reservas en cada dispositivo deben ser actualizadas periódicamente, ocasionando aumento del tráfico en la red y aumentando la posibilidad de que las reservas puedan expirar si los paquetes de actualización se perdido; mantener los estados en

cada enrutador, combinando con control de admisión en cada salto y aumento de los requisitos de memoria para permitir un número mayor de las reservas, haciendo que sean necesarios routers más robustos.

DIFFSERV. Es la arquitectura más difundida para proveer QoS para aplicaciones multimedia y de tiempo real, la implementación de la QoS se produce sobre la base de la definición de tipos de servicios sobre el tráfico de la red; cuyo objetivo consiste en atender el nivel de servicio deseado de cada aplicación como, por ejemplo: voz, vídeo y datos. Para proveer diferenciación los paquetes están marcados de acuerdo con las clases de servicios predeterminadas en el campo DS (Differentiated Service Field) del encabezado IP. La Arquitectura DiffServ, fue propuesta por el IETF, surgió proponiendo un enfoque más simple y eficiente que el IntServ, donde no habría necesidad de señalar y reservar los recursos, y ni mantener los estados de los flujos de la red, haciendo que el DiffServ fuera una solución ideal para Internet, ya que el IntServ no sería escalable en backbones de miles o millones de flujos.

Algunas desventajas de DiffServ incluyen, la administración y el monitoreo complejos, hay la pérdida de la granularidad, pues el QoS se aplica en la clase y existe la necesidad de clasificar el tráfico de red en clases. Con el uso del protocolo OpenFlow es posible gestionar y monitorear la red con mensajes ya predefinidos o desarrollando herramientas de acuerdo con la voluntad del usuario, no existe pérdida de granularidad, pues es posible, trabajar con flujos individualmente y con sus campos de match, y no existe la necesidad de clasificación de tráfico en clases.

MPLS. El MPLS, propuesto por el IETF, es un esquema de enrutamiento avanzado, que extiende el enrutamiento en relación con el encaminamiento de paquetes y el control de ruta. Él trabaja en una capa intermedia entre las capas 2 y 3 del modelo OSI.

GMPLS. (Generalized Multi-Protocol Label Switching) Se caracteriza por propagar errores, propagación de información de sincronización entre los conmutadores, aquí no hay una restricción del modelo de interconexión utilizado entre las redes, hay la separación entre el plan de control (señalización y enrutamiento) y el plan de datos, existe la posibilidad de establecer caminos bidireccionales, entre otros.

Uno de los mayores problemas del MPLS / GMPLS es que se limitan a los protocolos de Ingeniería de Tráfico existentes, tales como OSPF, LDP, RSVP-TE e I-BGP. más allá, muchos de estos protocolos se han ampliado para funcionar en MPLS / GMPLS. La RSVP, por ejemplo, fue desarrollado como un protocolo para la señalización de recursos en una red en la Arquitectura

IntServ, luego extendida para el uso en el MPLS y luego para el GMPLS, generando un protocolo más complejo y de codificación mayor, ocasionando desperdicio de recursos.
El uso de protocolos distribuidos eleva el tráfico de la red cuando hay cambios constantes en la misma, acarreando desperdicio de banda y aumento del ancho de banda retraso del enlace, además del gasto adicional para el recalculo de rutas y procesamiento de información en los dispositivos de red.

A partir de la versión 1.1 de OpenFlow, se han insertado algunas características del MPLS / GMPLS, como el push, pop y swap de etiquetas. Con la arquitectura SDN, el MPLS no quedará más limitado a protocolos ya existentes, pues el nuevo protocolo se puede crear y modificar de acuerdo con el deseo de cada administrador de red. Por lo tanto, es posible percibir que la SDN, y el OpenFlow, no vinieron para exterminar los protocolos tradicionales de oferta de QoS, pero sí, como una arquitectura que puede utilizarlos de una forma más eficiente, y que pueda, también, proponer futuras extensiones a la red. Esta idea se ve reforzada en el momento en que el propio protocolo OpenFlow define mensajes que permiten la inserción de encabezado MPLS en los paquetes (Rezende, 2016).

OPENFLOW. Con el uso del protocolo OpenFlow, la QoS puede aplicarse directamente a los flujos de los conmutadores sin el uso del protocolo RSVP, sin la necesidad de actualización constante y sin envío de tráfico en la red. Es importante resaltar que OpenFlow es un protocolo abierto, por lo que es posible añadir nuevas funcionalidades a los dispositivos, ya que no estaríamos más limitados a los protocolos propietarios, y cualquier grupo de investigación podría, por ejemplo, añadir una nueva funcionalidad a MPLS o DiffServ. Esta funcionalidad podría aplicarse en el futuro, a gran escala, y, por consiguiente, beneficiaría a toda la industria, las empresas y el usuario final (Rezende, 2016).

2.3.1.7 Ejemplos de aplicaciones con restricciones de QoS

Independiente de la tecnología de acceso a datos, las aplicaciones multimedia y de tiempo real requieren diferentes recursos de red y generan diferentes clases de tráfico. La QoS en las aplicaciones de audio y vídeo exigen la garantía de un caudal mínimo, bajo retardo y baja variación de retardo, pero son ligeramente tolerantes a errores y pérdidas. La pérdida de algunos cuadros en el flujo de vídeo, por ejemplo, no es suficiente para comprometer la percepción del usuario con respecto al contenido de la información audiovisual.

Por otro lado, las aplicaciones de transferencia de archivos como FTP son intolerantes a errores, pero no requieren garantías de retraso o caudal, aunque una transferencia rápida proporciona una

mejor satisfacción al usuario. La Tabla 4-2 muestra los requisitos de QoS de algunas aplicaciones multimedia y de tiempo real (Silva, 2017).

Tabla 4-2: Requisitos de QoS en diferentes aplicaciones

Requisitos de QoS	**Voz**	**FTP**	**E-mail**	**Video Broadcast**	**Video Interactivo**
Ancho de banda	Baja a Media	Baja	Baja	Alta	Alta
Paquetes Descartados	Media	Media	Media	Media	Media
Atraso	Alta	Baja	Baja	Baja	Alta
Jitter	Alta	Baja	Baja	Media	Alta

Fuente: http://btd.egc.ufsc.br/wp-content/uploads/2018/05/Madalena-Pereira-da-Silva.pdf

2.3.1.8 Monitoreo de parámetros de QoS

Para garantizar la QoS a las aplicaciones y servicios de la red, es fundamental poder medir parámetros en cada uno de los enlaces, con el fin de determinar los caminos que cumplan con las restricciones establecidas, en la actualidad son varias las herramientas desarrolladas con el fin de monitorizar el estado de los enlaces, pero en su mayoría requieren de hardware adicional, por lo que en este proyecto se busca herramientas para que no sea indispensable utilizar dispositivos externos. Con el fin de obtener estadísticas de red en un entorno de redes definidas por software, usando OpenFlow, se realizará pruebas de medición de ancho de banda disponible, bajo la siguiente fórmula.

$$UB = [(Bt2 - Bt1)/P] * [8bits]$$

Donde UB es el ancho de banda utilizado en el intervalo de tiempo.

$$P = t1 - t2$$

Y Bt1 y Bt2 son los números de bytes transmitidos en los instantes de tiempo t1 y t2, respectivamente (Ǵomez, 2016).

Es importante tomar en cuenta una estimación de Ancho de Banda de la red empresarial, partiendo de las necesidades de ese tipo de red, las cuales son:

- Servicio de Datos: navegación en Internet, Correo Electrónico, Transmisión de archivos.
- Servicio de Voz IP.
- Servicio de Videoconferencia.

2.3.1.9 Características del Tráfico

Aunque el tráfico de una red IP se realiza a través de paquetes, ya no basta con garantizar que cada uno de ellos llegue a su destino, en Internet actualmente, cada aplicación posee requisitos diferentes con consecuencias en la forma en que los paquetes circulan en la red. Para comprender los hábitos de consumo de Internet, las principales categorías identificadas fueron las siguientes: (Cardoso, 2015).

P2P - Tráfico P2P. Por ejemplo: BitTorrent.
Web - Páginas de Internet, incluyendo alojamiento de archivos, pero excluyendo streaming de vídeo y audio. Por ejemplo: www.google.es, Dropbox.
Streaming - Streaming de vídeos y de audio. Por ejemplo: Youtube, Netflix.
VoIP - Voz sobre IP. Por ejemplo: Skype.

Sin embargo, no todas las clases tienen el mismo volumen de tráfico. Mirando la Tabla 5-2, se observa que, dependiendo de la región, los hábitos de consumo son diferentes.

Tabla 5-2: Proporción de clases por región

Classe	Europa	América do Norte	América do Sul	Ásia
P2P	16,28%	5,63%	12,63%	35,63%
Web	32,88%	19,67%	34,40%	14,41%
Streaming	38,15%	63,25%	41,35%	40,23%
VoIP	4,35%	< 1%	1,74%	2,25%
Outro	8,34%	11,45%	9,88%	7,48%

Fuente: https://run.unl.pt/bitstream/10362/16557/1/Cardoso_2015.pdf

2.3.2 Redes Convencionales

En un diseño y arquitectura de las redes de datos convencionales, se presentan limitaciones para gestionar de una forma eficiente la red, en este proyecto se busca consolidar las debilidades asociadas a las arquitecturas actuales, lo cual permitirá proponer alternativas para los administradores de red, con el fin de evitar posibles problemas o solucionarlos; así como también se busca comprender las prestaciones de nuevas tecnologías de redes.
Las redes tradicionales o también llamadas redes convencionales, se caracterizan por ser distribuidas de sus componentes en el diseño y funcionamiento, en este tipo de red los componentes utilizan las direcciones de destino incluidas en las cabeceras de los datagramas para transmitir los mismos hacia su destinatario, en un proceso denominado ruteo, esto consiste en que por medio de módulos se incluyen reglas comunes para interpretar los campos de dirección y poder así tomar las decisiones de encaminamiento, entre otras.

El diseño de estas redes, especifica que en Internet la comunicación se basa en procedimientos de ruteo por cada datagrama en forma individual, por lo cual se puede apreciar que el diseño de la red Internet desde sus comienzos fue concebido con inteligencia distribuida. Es así que hoy las redes de datos están compuestas por switchs y routers, que posibilitan las comunicaciones entre clientes y servidores físicos o virtuales, pero que poco a poco se va complicando su administración.

El hecho de que estas redes son distribuidas, quiere decir que se basa en un diseño descentralizado en el cual la lógica de control y la función de distribución de paquetes (ruteo/forwarding) está embebido en cada uno de los componentes de la red, es así que cada router o switch soporta una serie de protocolos distribuidos que facilitan la toma de decisión en el direccionamiento de paquetes a lo largo de la red, como ejemplos de dichos protocolos tenemos OSPF, IS-IS, EIGRP, STP, entre otros, en la Figura 3-2 se puede observar la arquitectura de la red convencional (Osaba, 2016).

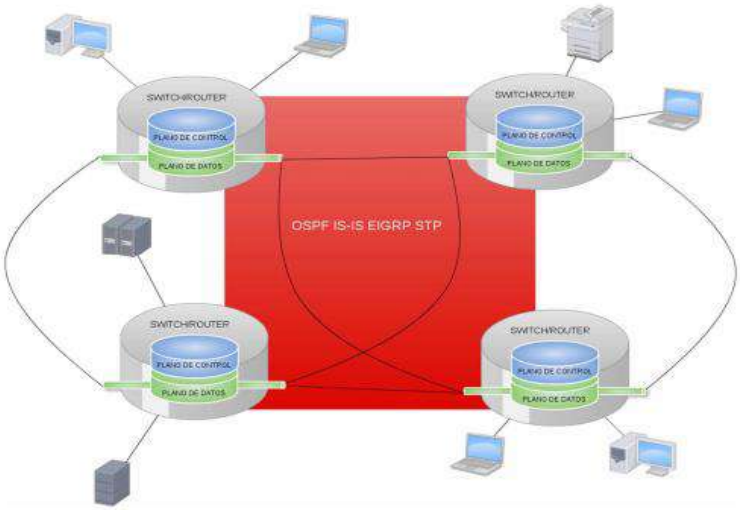


Figura 3-2: Red de datos convencional
Fuente:https://ri.itba.edu.ar/bitstream/handle/123456789/785/TELCOM.%20Osaba%20Virtualizaci%C3%B3n_de_Redes_Def_por_SW_.pdf

Las redes convencionales también se caracterizan por poseer una naturaleza compacta de sus componentes de la red, lo que implica que los equipos que posibilitan el encaminamiento de paquetes a través de las redes en la actualidad (Routers/Switches) están diseñados con una arquitectura del tipo monolítica, es decir incorporando en un mismo dispositivo físico el plano de datos, el de control y gestión, para así posibilitar el ruteo individual de los paquetes entrantes. En la Figura 4-2, se muestra la arquitectura de naturaleza monolítica de las redes convencionales.

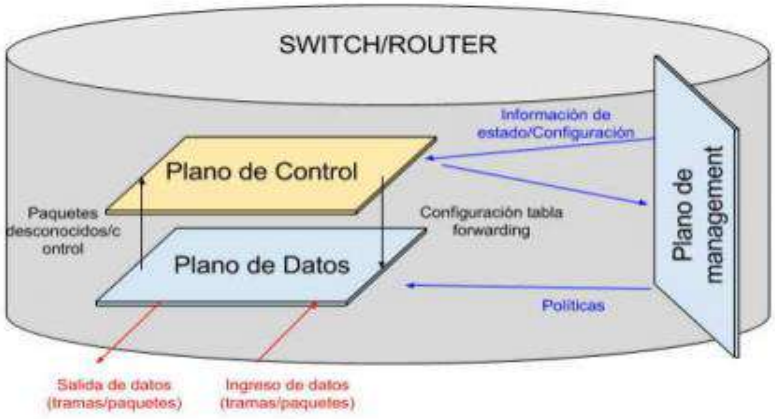


Figura 4-2: Arquitectura monolítica de switches y routers
Fuente:https://ri.itba.edu.ar/bitstream/handle/123456789/785/TELCOM.%20Osaba%20Virtualizaci%C3%B3n_de_Redes_Def_por_SW_.pdf

En el caso de un dispositivo de capa 2 (switch) del modelo OSI (Open System Interconnection), el plano de datos o forwarding, está constituido por los puertos físicos a través de los cuales se reciben y se transmiten las tramas de datos. Su principal función es la de encaminar éstas hacia los puertos de salida, utilizando para ello la información contenida en la tabla de forwarding embebida. Si la información de cabecera de una trama entrante, es encontrada en dicha tabla, esta

puede ser susceptible de modificaciones para luego ser transmitida a través del puerto correspondiente, sin la intervención de los otros planos.

Esta situación no sucede en todos los casos, el ejemplo más claro es cuando la información de la cabecera no se encuentra aún mapeada en la tabla, para este caso es necesaria la intervención del plano de control, manteniendo actualizada así la información de forwarding, de manera tal de que el plano de datos pueda retransmitir las tramas sin su intervención.

Para ello existen protocolos de control que permiten modificar dicha información y que tienen la responsabilidad de mantenerla actualizada de acuerdo a la topología actual de la red. Por otro lado, el plano de gestión tiene como función principal permitir la administración y configuración de dichos dispositivos de red. La integración vertical expuesta por parte de los planos de datos, de control y de gestión, constituye el diseño monolítico sobre el cual funcionan los componentes de una red de datos en la actualidad (Osaba, 2016).

2.3.2.1 Problemas con las redes convencionales

Alguno de los problemas por los que transitan las redes de datos en la actualidad como es el bajo nivel de innovación e equipamiento de networking. En estas dos últimas décadas los avances tecnológicos resultaron mínimos, debido a los altos costos y a los largos períodos que conllevan los nuevos desarrollos en el área; esto se da principalmente porque el diseño y la arquitectura del equipamiento de networking tradicional son propietario y cerrado. Limitando así las posibilidades de desarrollo a sólo aquellos que tienen acceso al código, interfaces e información; ya que mayoritariamente este conocimiento es accedido únicamente por los propios vendedores.

Por otro lado, es importante destacar que los altos costos de adquisición de equipamiento, constituyen en la actualidad una gran barrera para el ingreso al mercado de nueva competencia, ambos factores resultan en un limitante en la velocidad de desarrollo, que hoy en día no alcanza a satisfacer las demandas y evolución del mercado. El crecimiento en tamaño, complejidad y la aparición de nuevos servicios en las redes de datos, han motivado la generación de nuevos protocolos y/o evolución de los existentes, para posibilitar la transmisión de datos extremo a extremo.

Lo expuesto en este apartado da cuenta de la complejidad que están transitando las redes de datos convencionales, lo cual genera que las mismas sean muy estáticas con una consecuencia directa en las posibilidades de evolución, imposibilitando su adaptación a las necesidades dinámicas por

parte de los usuarios y a la provisión de nuevos servicios de valor agregado. Costos crecientes de adquisición, implementación y administración de equipamiento de networking.

La situación expuesta en el apartado anterior, genera un escenario en el cual los equipos de comunicaciones se están volviendo muy costosos, esto se debe en parte a que cada switch que conforma la red de datos debe incorporar una instancia del software asociado a la funcionalidad del plano de control que posibilita su funcionamiento autónomo. A su vez, éste se ha ido complejizando a partir de la incorporación líneas de código en respuesta a la evolución e incremento en el número de protocolos, generando además una consecuencia directa en los requerimientos de memoria y CPU (Central Processing Unit).

En una red con inteligencia distribuida, la incorporación de dicho software en cada equipo de red, es un aspecto fundamental ya que permite el encaminamiento de paquetes al destino correspondiente a lo largo de la red. Por otro lado, la complejidad tiene su consecuencia en la administración de las redes, lo cual genera un escenario de costos elevados que repercuten de manera directa en el OPEX (Operating Expenditure) de las organizaciones.
El cambio de los patrones de tráfico, los diseños convencionales de redes en general, son construidos de manera jerárquica, es decir utilizando distintos niveles de switchs ethernet que configuran una topología de tipo árbol.

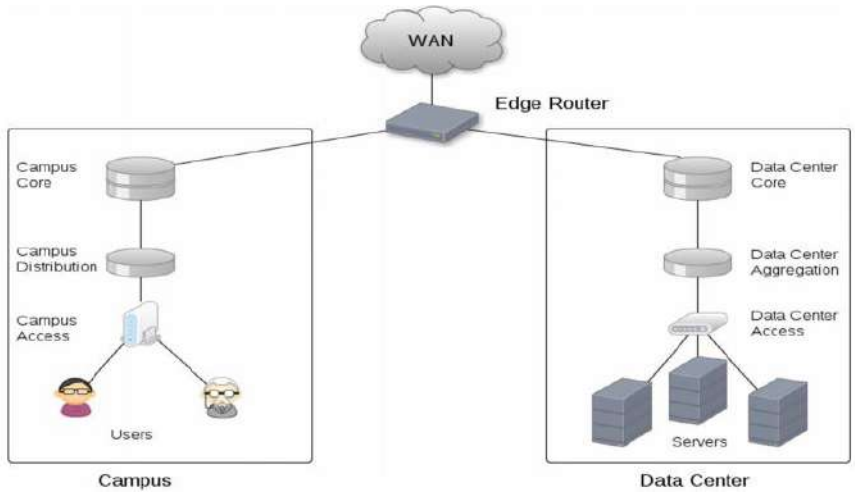


Figura 5-2: Arquitectura de una red de datos
Fuente:https://ri.itba.edu.ar/bitstream/handle/123456789/785/TELCOM.%20Osaba%20Virtualizaci%C3%B3n_de_Redes_Def_por_SW_.pdf

Este esquema que se presenta en la Figura 5-2, resulta útil para el tráfico norte-sur característico en arquitecturas del tipo cliente servidor. Sin embargo, las aplicaciones actuales en muchos casos, requieren la interacción entre distintas bases de datos y servidores para generar una respuesta de

cara al usuario. Los Nuevos servicios en la nube: infraestructura como servicio, la velocidad en la provisión de nuevos servicios ha evolucionado de manera superlativa en los últimos años.

Los servicios en la nube y la posibilidad de adquirir IaaS (Infrastructure as a Service), han sido protagonistas en este sentido. En contraposición, las arquitecturas tradicionales de Networking, por su característica de diseño poco flexible, dificultan la integración de redes, lo cual genera una consecuencia directa en la migración de servicios a la nube.

Las problemáticas enunciadas en el presente apartado son sólo algunas de las existentes en las redes actuales, las mismas se deben considerar como el punto de partida para justificar la generación de nuevos paradigmas y arquitecturas de networking. Estas deben permitir la evolución a redes que soporten el dinamismo que requieren y demandan los servicios actuales (Osaba, 2016).

2.3.3 Redes definidas por software

Gracias a SDN, el diseño y gestión de redes se ha vuelto más innovador en los últimos años; sin embargo, esta tecnología no tuvo una súbita aparición, sino que es fruto de una larga historia de innovaciones dirigidas a hacer más programables las redes.

La historia de SDN comienza hace 20 años, por comienzos de lo que hoy conocemos como Internet, del cual, debido a su gran éxito y rápida expansión surgió la necesidad de gestionar y evolucionar las infraestructuras de redes existentes, por medio de la programación. A partir de este momento la evolución de SDN se divide en tres etapas claramente diferenciadas:

Redes activas. (1995 a 2000) Con el aparecimiento del internet, provocó que paulatinamente se vaya incrementando las aplicaciones, lo que indujo a los investigadores a diseñar y probar nuevos protocolos de red, pero después de un tiempo, estos protocolos tenían que ser estandarizados por el IETF, proceso tedioso para los investigadores; así que se inclinaron por un enfoque hacia el control de la red, que consistió en que a través de una interfaz de programación (API), se permita el diseño de funcionalidades personalizadas aplicables a nodos de la red, estas redes se fueron dando a conocer, porque tuvo una reducción en el coste computacional, avanzó en lenguajes de programación y en la tecnología de máquinas virtuales; aunque no tuvieron un despliegue extendido, las redes activas ofrecieron contribuciones relacionadas con SDN como funciones programables en la red, virtualización de redes y la visión de una arquitectura unificada en distintos aparatos de red como cortafuegos, IDS, NAT, etc. (Valdivieso, A. Peral, A. Barona, L.García, 2014).

Separación del plano de control, del plano de datos (2001-2007). Los enrutadores y conmutadores convencionales tenían integrados los planos de control y datos, integración que ocasionaba inconvenientes en la configuración y depuración del comportamiento del enrutamiento, por lo que las empresas creadoras de equipos hardware comenzaron a implementar la lógica de reenvío de paquetes (plano de datos), separando el plano de control. Esto dio paso a varias innovaciones, una interfaz abierta entre ambos planos como ForCES (separación del elemento de control y reenvió) estandarizada por la IETF, la interfaz Netlink con la funcionalidad de reenvío de paquetes a nivel de Núcleo de Linux y un control lógico centralizado de la red, como con RCP (plataforma de control de enrutamiento), arquitecturas SoftRouter y el protocolo PCE (Path Computation Element) del IETF, ambos conceptos clave en diseños futuros de SDN (Nick Feamster, Jennifer Rexford, 2013).

Aparición del API de OpenFlow. (2007-2010) Con los antecedentes ya enunciados es claro que diversos grupos de investigación empezaron a investigar y experimentar redes a escala con PlanetLab y Emulab. Posteriormente GENI (Global Enviroment for Networking Innovations); a la vez un grupo de investigación de la Universidad de Stanford crearon el protocolo OpenFlow. Gracias a la adopción de este protocolo en las empresas se abrió el API, lo que permitió que los programadores controlen ciertas funcionalidades (Nick Feamster, Jennifer Rexford, 2013).

Ya en el 2014 la empresa de telecomunicaciones AVAYA hizo una demostración de redes definidas por software usando Shortest Path Bridging y OpenStack, eliminando la configuración Manual (Menon, 2014).

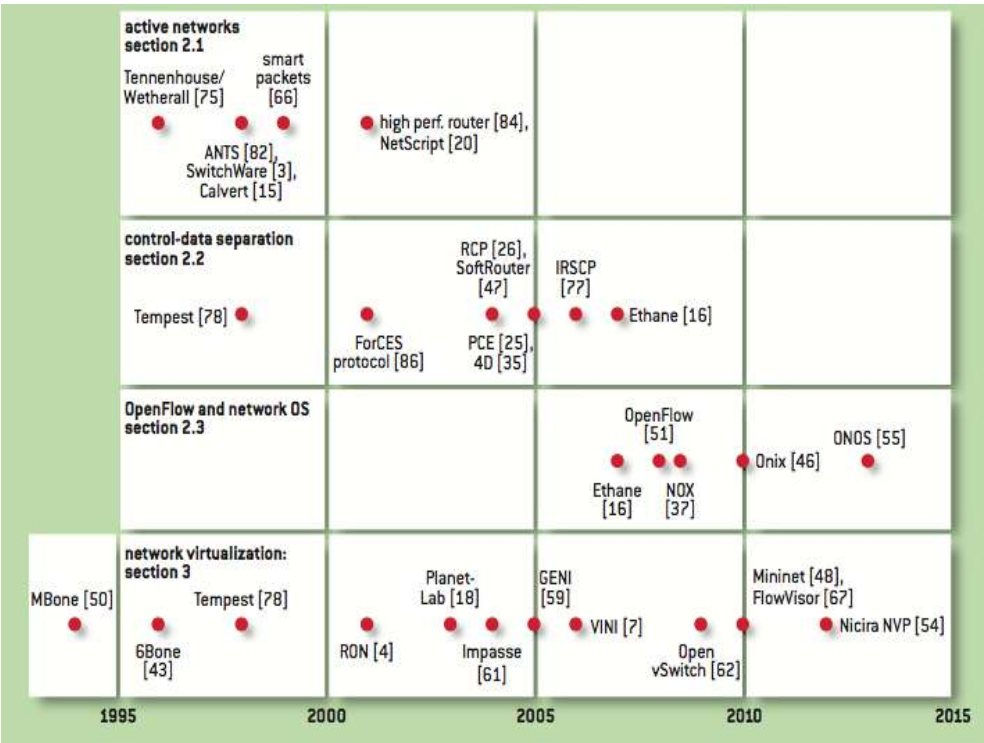


Figura 6-2: Evolución en escala de tiempo SDN
Fuente: http://queue.acm.org/detail.cfm?id=2560327

2.3.3.1 Arquitectura SDN

Según la ONF, la arquitectura SDN es (Mack-Crane, 2016).

Directamente programable. El control de red es programable directamente porque está desacoplado de las funciones de reenvío.

Ágil. Abstraer el control de expedición permite a los administradores ajustar dinámicamente todo el flujo del tráfico de la red para satisfacer las necesidades cambiantes.

Centralizada. La inteligencia de la red (lógicamente) está centralizada en controladores SDN basados en software que mantienen una visión global de la red, que aparece para las aplicaciones y las políticas de las máquinas como switch lógicos.

Configurada mediante programación. SDN permite a los administradores de red configurar, administrar, asegurar y optimizar los recursos de red rápidamente mediante programas SDN dinámicos, automatizados, que pueden escribir ellos mismos ya que no dependen de un software propietario.

Basada en estándares abiertos y neutrales. Cuando se implementa a través de estándares abiertos, SDN simplifica la operación y el diseño de la red porque las instrucciones son proporcionadas por los controladores SDN en lugar de múltiples protocolos y dispositivos específicos del proveedor.

Por otra parte, la arquitectura SDN según la Open Networking Foundation en el año 2013, consta de tres capas las cuales se describen a continuación:

Capa de aplicación. Es donde las aplicaciones realizan las demandas a la capa de control mediante un API, las cuales están diseñadas para satisfacer las necesidades de los usuarios. Algunos ejemplos de aplicación dentro de las capas SDN son las siguientes:

- **Enrutamiento adaptativo**. Conocido como balanceo de carga.
- **Itinerancia de aplicaciones.** La transferencia o HandOver al usar dispositivos móviles hace necesario proveer un servicio continuo.
- **Mantenimiento de la red.** Herramientas de configuración como traceroute o tcpdump no son suficientes, se recomienda el uso de nuevas herramientas de diagnóstico.
- **Seguridad de la red**. Permite analizar patrones de tráfico para evaluar problemas de seguridad, guiando paquetes sospechosos a un IPS.
- **Virtualización de la red.** Permite infraestructuras heterogéneas; normalmente se separa la red física en múltiples instancias virtuales.

Capa de control. Es el componente más importante de la arquitectura SDN ya que gestiona la capa de aplicación e infraestructura mediante dos interfaces, el plano de infraestructura recoge el estado de la red y según las exigencias actualiza en los dispositivos las reglas de envío. Por otra parte, se comunica con las aplicaciones SDN por medio de la traducción de sus requisitos con un lenguaje de alto nivel.

Capa de infraestructura. La capa consta de dispositivos hardware de conmutación que forman una red y realiza tareas de acuerdo a sus componentes lógicos.

- **Control**. Recoge información del estado de la red como topología o estadísticas de tráfico, comunicando al controlador y este a su vez le indica las reglas de reenvío de paquetes.
- **Datos.** El procesador de red reenvía paquetes en base a las decisiones tomadas por el plano de control.

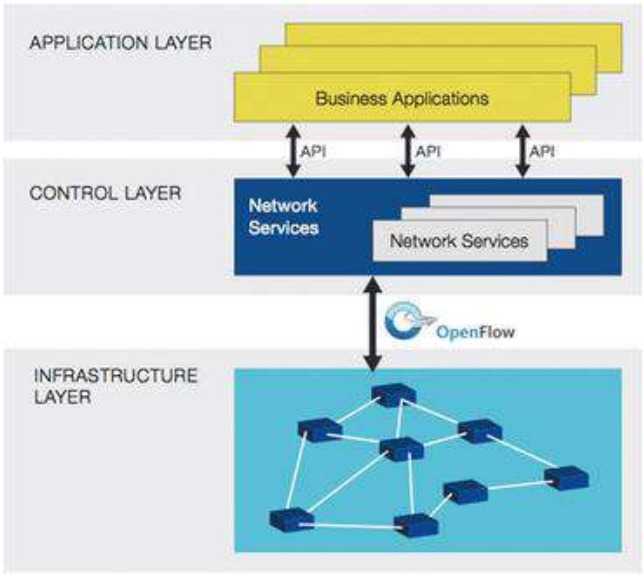


Figura 7-2: Arquitectura SDN
Fuente: https://www.opennetworking.org/sdn-resources/openflow

2.3.3.2 Integración, vertical vs. horizontal

La arquitectura SDN propone un enfoque diferente, proporcionando un ecosistema de componentes intercambiables que son más simples y económicos. Las capas de un switch tradicional se consideran integradas verticalmente ya que todo es especializado y no puede intercambiarse entre proveedores. En el caso de SDN podemos usar cualquier switch con cualquier controlador y con cualquier aplicación, siempre y cuando se respete la especificación de las interfaces abiertas entre las capas.

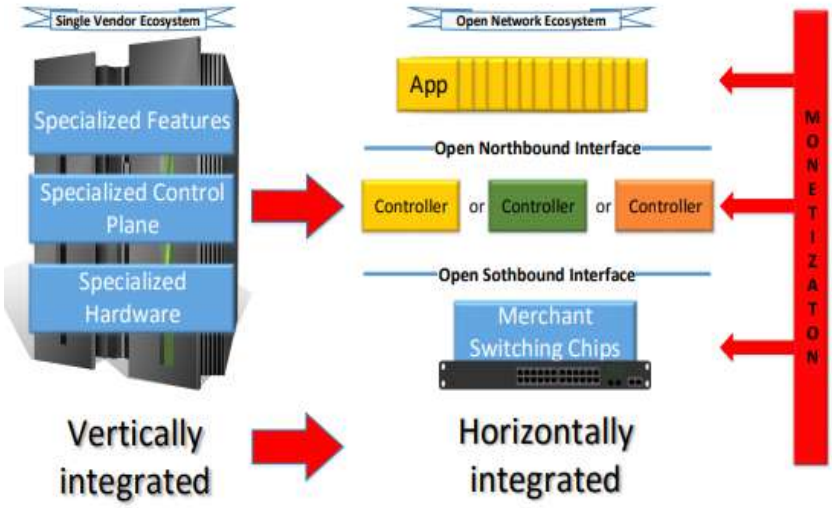


Figura 8-2: Integración Vertical vs. Integración horizontal
Fuente: http://graphics.cs.pub.ro/theses/phd/2016/ovidiu.poncea/PhD%20Thesis%20-%20Ovidiu%20Poncea.pdf

Con respecto a la monetización, cómo se gana dinero; tradicionalmente los vendedores venden principalmente hardware, el costo del desarrollo de software generalmente se incluye en el precio del hardware, a diferencia de la nueva arquitectura que propone un modelo diferente de monetización en cada capa. En la capa de datos el hardware es más económico con un software simple, en la capa del plano de control, el controlador se vende como una aplicación de software y las aplicaciones también se vende de forma independiente, muy parecido a lo que se ve en la industria de la PC, en donde el hardware, el sistema operativo y las aplicaciones se venden de forma independiente.

Específicamente en entornos de las redes programables, el enfoque horizontal habilita la comunicación entre los controladores SDN a través de un protocolo de comunicación que es similar a un protocolo de enrutamiento, de manera que el controlador de un dominio se comunique directamente con los controladores de dominios vecinos; mientras el enfoque jerárquico o vertical consiste en un controlador central que se comunica con los controladores de todos los dominios, en este enfoque, no hay intercomunicación entre los controladores de dominio.

Por otro lado, cuando un paquete llega a un conmutador en una red convencional, por medio de las reglas integradas al firmware propietario del conmutador le encaminan al paquete, el switch envía cada paquete al mismo destino por la misma trayectoria y trata a todos los paquetes de la exacta misma manera. En entornos empresariales, los conmutadores inteligentes diseñados con circuitos integrados de aplicación específica (ASICs) son sofisticados para reconocer varios tipos de paquetes y tratarlos de forma diferente, pero estos conmutadores pueden ser muy costosos.

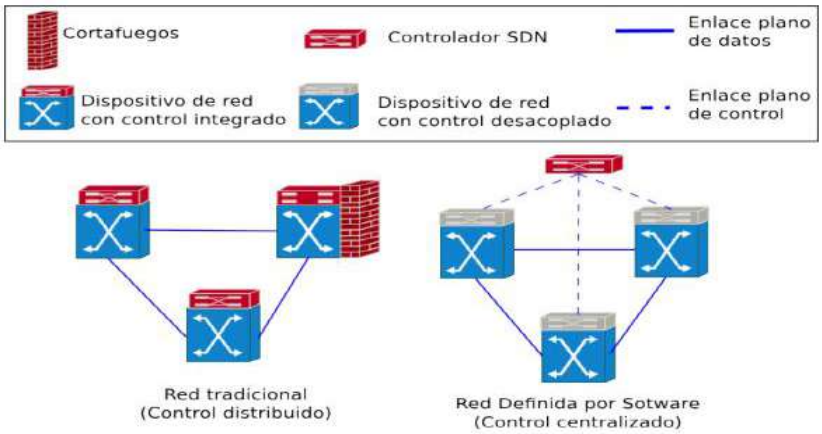


Figura 9-2: Diferencia entre una red convencional y una SDN
Fuente:http://bibliotecadigital.udea.edu.co/bitstream/10495/7416/1/MontoyaEmanuel_2017_Desarrollo%20de%20un%20esquema.pdf

En una red definida por software, un administrador de red puede darle forma al tráfico desde una consola de control centralizada sin tener que tocar conmutadores individuales, además el

administrador puede cambiar cualquier regla de los conmutadores de red cuando sea necesario dando o quitando prioridad, o hasta bloqueando tipos específicos de paquetes con un nivel de control muy detallado.

Eso es especialmente útil en una arquitectura de múltiples arrendatarios (multi-tenant architecture) de computación en la nube, porque permite al administrador manejar cargas de tráfico de manera flexible y más eficiente, permitiendo al administrado usar menos conmutadores pequeños y costosos y tener más control que nunca sobre el flujo del tráfico de red (Ǵomez, 2016).

2.3.3.3 Qué es OpenFlow

Este protocolo OpenFlow permite crear protocolos experimentales en redes de las universidades para no tener que crear la plataforma desde sus comienzos. Además, es capaz de reemplazar las funcionalidades de los protocolos de capas 2 y 3 en switchs y routers comerciales. Así como también, puede determinar por medio de controladores de red la ruta de los paquetes de red a través de una red de conmutadores.

OpenFlow permite que los switchs de diferentes proveedores, a menudo cada uno con sus propias interfaces propietarias y lenguajes de scripting, se administren de forma remota utilizando un solo protocolo abierto. OpenFlow permite la administración remota de las tablas de reenvío de paquetes de un switch de capa 3, mediante la adición, modificación y eliminación de reglas y acciones de coincidencia de paquetes. De esta forma, el controlador puede tomar decisiones de enrutamiento periódicas o ad hoc y traducirlas a reglas y acciones con una vida útil configurable, que luego se implementan en la tabla de flujo de un conmutador, dejando el reenvío real de paquetes coincidentes al conmutador a velocidad de cable para la duración de esas reglas.

Los paquetes que no se corresponden con el conmutador pueden enviarse al controlador, pudiendo este decidir modificar las reglas de tabla de flujo existentes en uno o más conmutadores o implementar nuevas reglas para evitar un flujo estructural de tráfico entre el conmutador y el controlador. Incluso podría decidir reenviar el tráfico, siempre que haya indicado al switch que reenvíe paquetes completos en lugar de solo su encabezado.

El protocolo OpenFlow está superpuesto al Protocolo de control de transmisión (TCP) y prescribe el uso de Transport Layer Security (TLS). Los controladores deben escuchar en el puerto TCP 6653 los conmutadores que desean configurar una conexión. Las versiones anteriores del protocolo OpenFlow utilizaban extraoficialmente el puerto 6633 (ONF, 2015).

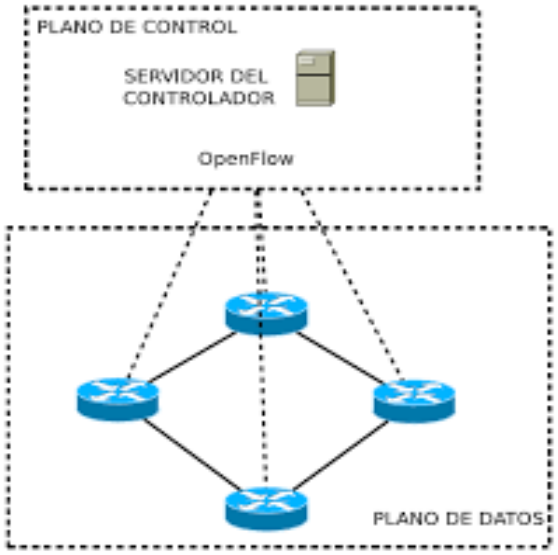


Figura 10-2: Estructura Open Flow
Fuente:http://dehesa.unex.es/bitstream/handle/10662/3540/TFGUEX_2015 Diaz_Serrano.pdf?sequence=1

OpenFlow se clasifica en:

- Protocolo de conexión, la misma que se encarga de:
  - Control de sesión.
  - Define la estructuración del mensaje y la consecuente modificación de flujos.
  - Recopilar datos estadísticos.
- Protocolo de administración y configuración:
  - Asignación de puertos físicos del dispositivo de red a un controlador específico.
  - Precisa el comportamiento del componente de red en caso de desconexión del controlador.

2.3.3.4 Características y beneficios de OpenFlow

- Con la adición de OpenFlow especificación 1.3, las siguientes características son compatibles:
  - Múltiples tablas de flujo.
  - Procesamiento de tuberías.
  - Procesamiento personalizado de tuberías.
  - Instancias de Multi-VLAN.
  - Grupos en hardware.
  - Puertos físicos, lógicos y reservados de OpenFlow.
  - Negociación de versiones.

    - Tablas de grupos.
    - Conexiones auxiliares.
    - Combinación ampliable de OpenFlow (OXM).
    - Múltiples controladores.
    - Soporte para flujos IPv6
- La configuración del lado del interruptor de OpenFlow permite al usuario:
    - Activar o desactivar OpenFlow.
    - Crear instancias de OpenFlow y configurar conexiones de controlador.
    - Mostrar configuración relacionada con OpenFlow.
    - Disponibilidad de soporte de configuración para retener la configuración de OpenFlow durante un reinicio.
- OpenFlow admite alta disponibilidad:
    - La tabla de flujo de OpenFlow se conserva a través de la conmutación por error del Módulo de administración.
    - La configuración de OpenFlow se sincroniza desde AMM (Active Management Module) a SMM (Management Module).
- OpenFlow incluye herramientas para limitar los recursos:
    - Soporte para limitar el porcentaje de motores de políticas y recursos de tabla de control de IP utilizados por OpenFlow.
    - Soporte para limitar la velocidad de la cantidad de tráfico de OpenFlow enviado al controlador.
    - Soporte para limitar la velocidad de la cantidad de tráfico OpenFlow que se reenvía por las reglas del motor de políticas programado por OpenFlow.
- Modos de operación de OpenFlow:
    - Soporte para el modo solo hardware donde solo se aceptan flujos que se pueden programar en hardware desde el controlador.
    - Soporte para el modo activo (predeterminado) donde el conmutador envía nuevos flujos al controlador.
    - El interruptor normalmente maneja el soporte para el modo pasivo donde los nuevos flujos ya no se envían al controlador.

2.3.3.5 Switch OpenFlow

El switch OpenFlow es una parte de los equipos de conmutación de paquetes, entre sus particularidades, se tiene el concepto de flujo, el cual es un conjunto de características de determinados paquetes en red. Por ejemplo, un flujo puede ser todo el tráfico HTTP, o todo el tráfico procedente de una dirección IP. Estos parámetros se definen en el momento de la creación

de la regla y cada recepción de paquetes, éste se comparará con las reglas existentes. Las especificaciones OpenFlow describen el protocolo abierto para permitir que las aplicaciones se desarrollen e inserten en las tablas de flujo diferentes dispositivos de fabricantes, desde su lanzamiento en marzo de 2008, existen disponibles diferentes especificaciones del protocolo OpenFlow, como se observa en la Tabla 6-2 la evolución y las especificaciones de este tipo de Switch.

Tabla 6-2: Comparación de especificaciones OpenFlow

Funciones	**Especificaciones OpenFow**					
	OF 1.0	**OF 1.1**	**OF 1.2**	**OF 1.3**	**OF 1.4**	**OF 1.5**
Tabla de Flujo	Única	Múltiple	Múltiple	Múltiple	Múltiple	Múltiple
MPLS	NO	SI	SI	SI	SI	SI
IPv6	NO	NO	SI	SI	SI	SI
Comunicación simultánea de múltiples controladores	NO	NO	SI	SI	SI	SI
Tabla de Grupo	NO	NO	NO	SI	SI	SI
Meter	NO	NO	NO	SI	SI	SI
Soporte a Interfaces de Fibra óptica	NO	NO	NO	NO	SI	SI

Realizado por: Marcelo Criollo, 2019

2.3.3.6 Componentes de un Switch OpenFlow

Los Switch OpenFlow poseen de tablas de flujo para su funcionamiento, las tablas de flujo ejecutan las funciones de reenvió y redireccionamiento, además tienen un canal seguro para la comunicación con el correspondiente controlador, esta conexión se realiza ya que el controlador es el encargado de administrar el dispositivo de red y lo hace mediante el protocolo OpenFlow.

Tablas de flujo. Uno de sus componentes son las tablas de flujos que contienen tres campos primordiales, el primer campo llamado "Header Fields" contiene un conjunto de entradas que son cabeceras del paquete las cuales poseen información con direcciones IP de origen y de destino, información sobre Vlan (ID y Priority), puerto de ingreso, protocolos de comunicación (TCP/UDP), cada campo del "Header Fields" a continuación se explicará, un segundo campo "Counters" contiene información de conteo, este conteo se realiza por tabla, por flujo, por puerto y por cola, el tercer campo "Actions" es donde se encuentra la información de qué se debe hacer

con el paquete de entrada, es decir, que acción de reenvío debe ejecutar y por cual puerto, con los campos anteriores se evalúa la información de paquete y con base en ello el switch toma una acción.

Todos los paquetes son procesados por el switch y comparados contra la tabla de flujo, cuando un paquete ingresa se toma la acción definida por la tabla, las acciones pueden hacer el reenvío del paquete o sacar el paquete por un puerto especifico, de la misma forma cuando ingresa un nuevo paquete el controlador es el encargado de actualizar las tablas de los switch para que conozcan e identifiquen el nuevo paquete, de esta forma los dispositivos sabrán qué hacer con él.

Tabla 7-2: Elementos de una tabla de flujo en un switch

Campos cabecera	Contadores	Acciones

Realizado por: Marcelo Criollo, 2019

Header fields (campos cabecera). El protocolo OpenFlow toma los flujos de entrada para detectar los paquetes, cada entrada contiene información específica.

Tabla 8-2: Campos de información en "Campos cabecera"

Puerto de ingreso	Ether origen	Ether Destino	Ether Tipo	Id Vlan	Prioridad Vlan	Ip origen	Ip destino	Ip Protocolo	Ip Tos Bits	Tcp/udp Puerto Origen	Tcp/Udp Puerto destino

Realizado por: Marcelo Criollo, 2019

Según el formato de la tabla de flujo, (Araú, 2013) se describe brevemente cada campo de información:

- In Port. Puerto de entrada del switch.
- Ethernet Source. Dirección MAC de origen.
- Ethernet Destination. Dirección MAC de destino.
- Ethernet Type. Tipo de marco (marco) Ethernet.
- VLAN ID. Número de identificación de la VLAN (Virtual LAN).
- VLAN PCP (Priority Code Point). Nivel de prioridad.
- IP Source. Dirección IP de origen.
- IP Destination. Dirección IP de destino.
- IP Protocol. Protocolo IP.
- IP ToS (Tipo de servicio). Tipo de servicio.
- TCP/UDP Source Port. Puerto de origen del protocolo (TCP / UDP).
- TCP/UDP Destination Port. Puerto de destino del protocolo (TCP / UDP).

Contadores. Es el campo encargado de actualizar la información, el proceso lo hace por tabla, por flujo, por puerto, además realiza un conteo detallado sobre paquetes recibidos y enviados. Los contadores son estadísticas referentes a los flujos del switch, algunos ejemplos son: el número de paquetes y bytes que pasaron por cada flujo, el tiempo desde que el flujo fue configurado en el switch y el tiempo transcurrido desde la última vez que un paquete de ese flujo fue identificado por el switch; un detalle importante es que los contadores se reinician automáticamente al alcanzar el valor máximo, sin ningún aviso, por lo tanto, es necesario que el controlador sea responsable de ese control.

Tabla 9-2: Bits usados por el campo "contador"

Contador	Bits
Por tabla	
Entradas activas	32
Búsqueda de paquetes	64
Paquetes emparejados	64
Por flujo	
Paquetes recibidos	64
Bytes recibidos	64
Duración en s.	32
Diración en ms.	32
Por puerto	
Paquetes recibidos	64
Paquetes transmitidos	64
Bytes recibidos	64
Paquetes transmitidos	64
Recibe drops	64
Paquetes transmitidos	64
Errores recibidos	64
Errores transmitidos	64
Receive Frame Alignment Errors	64
Receive Overrum Errors	64
Receive CRC Errors	64
Colisiones	64
Por cola	
Paquetes transmitidos	64
Bytes transmitidos	64
Transmit Overrun Errors	64

Fuente:http://repositorio.upct.es/bitstream/handle/10317/5254/tfg729.pdf?sequence=1

Acciones. Las entradas de flujo se enlazan mediante acciones con las cuales trata el switch a los paquetes similares. El switch puede rechazar esta entrada de flujo siempre y cuando no se procese la lista de las acciones en la orden predeterminada. También conocido como "Instructions", con los valores de este campo el Switch sabe qué hacer con el flujo de entrada, entre las acciones se encuentra principalmente el reenvío, pero esta acción posee varias naturalidades que pueden ser (Calderón, 2016).

- **All.** Reenvío por todos los puertos.
- **Controller.** Encapsula y reenvía el paquete de un flujo al controlador, se usa normalmente para el primer paquete de un flujo nuevo, esto con el fin de que el controlador determine si se agrega o no el flujo a las tablas de flujo.
- **Table.** Realiza la operación según su tabla (para paquetes de salida únicamente).
- **In_port.** Reenvía el paquete por el puerto de entrada.

Las acciones opcionales que el switch soporta para los siguientes puertos virtuales son las siguientes:

- **Normal.** Procesa el paquete utilizando la ruta de envío tradicional apoyado por el switch, es decir VLAN; este switch puede comprobar el campo Id VLAN para determinar si debe o no enviar el paquete a lo largo de la ruta de procesamiento normal.
- **Flood:** Inundación de paquetes a lo largo de spanning tree, necesarios en muchos casos para garantizar la disponibilidad de las conexiones de red, pero no incluye la interfaz de entrada.

Canal seguro de comunicación OpenFlow. El canal seguro de comunicación es el medio utilizado para la comunicación entre switch OpenFlow y el controlador OpenFlow, ya que permite el intercambio de comandos y paquetes entre estos dos elementos, para dar confiabilidad al canal, la interfaz de acceso recomendada es el protocolo SSL (Secure Socket Layer), utiliza el cifrado de datos con certificados de confianza.

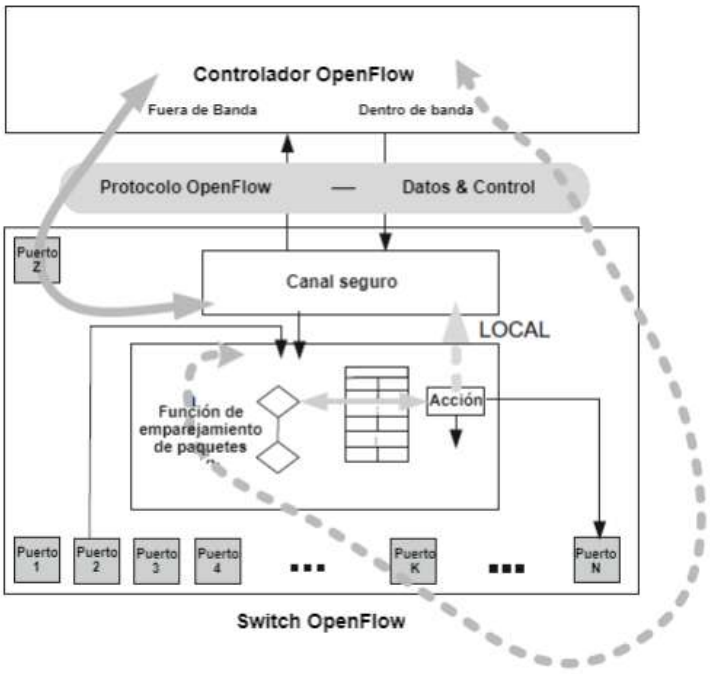


Figura 11-2: Canal seguro del switch OpenFlow
Fuente:http://tauja.ujaen.es/bitstream/10953.1/6704/1/TFG_Angela_Maria_Sanchez_Valdepeas_Lopez.pdf

El protocolo OpenFlow. Este protocolo define la comunicación entre el switch OpenFlow y el controlador. El switch establece la comunicación con el controlador en una dirección IP, usando un determinado puerto, habitualmente el 6634. Si el switch conoce la IP del controlador, iniciara una sesión TCP estándar. Cuando una conexión es establecida, cada lado envía un mensaje HELLO con la versión más alta del protocolo OpenFlow soportado. Tras la recepción de este mensaje, el receptor calcula la versión del protocolo a usar como la más pequeña entre la que se envió y la que se recibió en los mensajes HELLO. Si la versión negociada es soportada, la conexión será iniciada, de lo contrario se enviará un mensaje HELLO-FAILED y se terminará la conexión. En esta comunicación entre el controlador y el switch existen algunos tipos básicos de mensajes que pueden intercambiarse entre los elementos, los cuales se clasifican como (Villaroel, 2015).

Controlador para switch. Son mensajes iniciados por el controlador, utilizados para administrar directamente o inspeccionar el estado del conmutador, estos mensajes permiten que el controlador configure el switch, modifique estados y entradas de flujo, entre otras características. No necesitan respuesta por parte del switch necesariamente. Los mensajes son en este orden: Features (funciones de consulta), modify-state (añaden / borran / modifican entradas), read-state y packet-out.

- Simétricas. Son mensajes generados sin solicitud de los elementos, por ejemplo, son los mensajes hello y echo. El primer mensaje se intercambia entre el controlador y el switch en el inicio de la red, mientras que el segundo mensaje, se utiliza para comprobar que la conexión entre el conmutador y el controlador siga activa y para identificación de latencia y ancho de banda. El nombre de los mensajes que se envían son: hello, echo (request/reply) y experimenter.
- Asíncronas: Son mensajes enviados por el switch sin la solicitud del controlador, consisten en Informar eventos en la red, errores, cambios en el estado del switch y llegada de paquetes. Un ejemplo de este tipo de mensaje es el Packet In, que notifica la llegada de un flujo no configurado en el switch. Estos mensajes se denominan: packet-in, flow-removed y port-status.

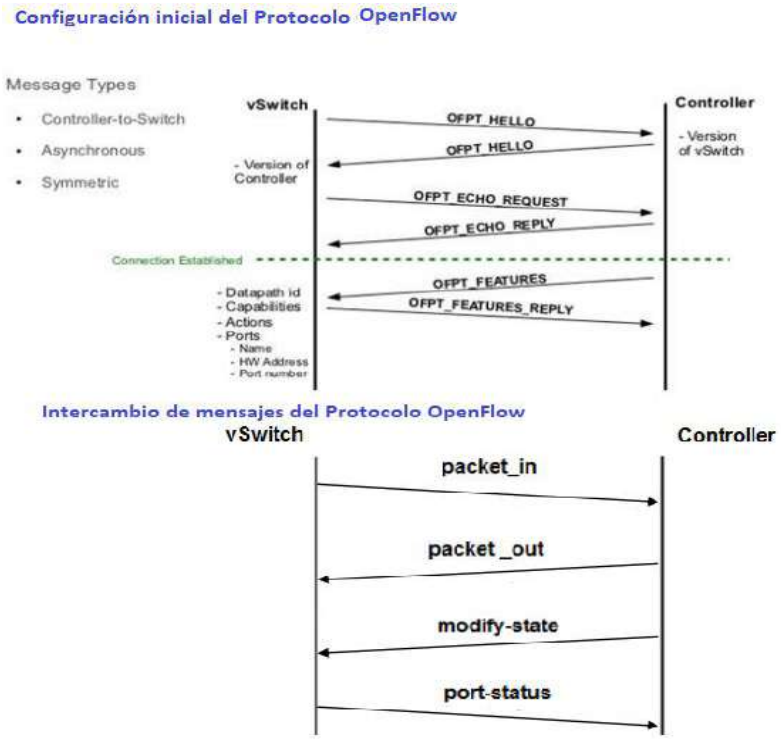


Figura 12-2: Mensajes OpeenFlow
Fuente: https://uvadoc.uva.es/bitstream/10324/15101/1/TFG-G%201628.pdf

En este proyecto se analizan los mensajes más importantes para la comprensión del trabajo elaborado, siendo las siguientes:

Hello. Mensajes del tipo simétrico, intercambiados al iniciarse una conexión entre un switch y un controlador.

Echo. Mensajes del tipo simétrico, intercambiados entre el switch y el controlador para mantener la conexión activa, pero también utilizada para medir la latencia de la conexión y su ancho de banda.

Features. Mensaje del tipo asincrónico enviado por el controlador al switch para saber cuáles son las características de éste, como por ejemplo la versión de OpenFlow que soporta. Normalmente se cambia al establecer la conexión y requiere una respuesta del switch.

Modify-State. Mensaje del tipo asincrónico enviado por el controlador para gestionar el estado del conmutador. Su principal funcionalidad es añadir, cambiar o eliminar flujo, también se puede utilizar para modificar las propiedades de los puertos del switch.

Packet-out. Mensajes de tipo asincrónico enviados por el controlador como respuesta al packet-in. El mensaje puede contener una lista de acciones a ser tomadas por el controlador; si no contiene ninguna el paquete se descartará.

Packet-in. Mensaje del tipo asincrónico. Se envía desde el conmutador al controlador cuando se recibe un paquete en el que no se ha coincidido ningún match.

Flow Removed: Mensaje del tipo asincrónico. Informa el controlador siempre que un flujo de flujo que tiene una bandera OFPFF_SEND_FLOW_REM se quita. Las entradas se quitan después de un cierto período de tiempo sin actividad, definido durante la configuración del flujo, o a través de una petición del controlador.

Read-State. Mensaje del tipo asincrónico originado en el controlador con el propósito de recoger algún tipo de información del switch. Este tipo de información puede ser, por ejemplo, el número de bytes transmitidos por un cierto flujo o la cantidad de bytes los puertos que tiene el conmutador.

2.3.3.7 Funcionamiento de OpenFlow

Teniendo en cuenta los conceptos estructurales de cada elemento, se expresa también la forma de actuación de cada componente de una red OpenFlow, mostrando paso a paso el procesamiento y reenvío de flujos. Después del montaje físico y el establecimiento de los canales seguros de comunicación entre switch y controlador, cada paquete que llega al switch OpenFlow, hace la separación de los encabezados y verifica si hay alguna entrada correspondiente en sus tablas de flujo, si existe, el conmutador aplica la acción, si no existe, envía el paquete al controlador por el canal seguro y espera instrucciones. El controlador recibe el paquete, hace el tratamiento que fue previamente establecido en su configuración para este estándar y basado en las características del

paquete, aplica una o más reglas en los switchs. El comportamiento del conmutador durante este proceso del controlador es configurable; con la premisa de ser un tratamiento rápido, es común que el switch espere la respuesta, manteniendo los paquetes originales en buffer para que sean reenviados después de la generación de la regla (Calderón, 2016) .

Las tablas de flujo en la versión 1.3 de OpenFlow, el proceso en el cual los flujos entrantes hacen Matching con las tablas es diferentes en comparación con la versión 1.0, cuando un paquete llega al switch es comparado con una tabla de flujo única y se ejecuta la acción correspondiente a la tabla, mientras que en el protocolo en su versión 1.3 el paquete realiza el proceso de Matching por todas las tablas esto es con el fin de incrementar la precisión en la acción a tomar, al pasar por las diferentes tablas de flujo se le conoce como Pipeline Processing.

El Pipeline Processing es un procedimiento en el cual los flujos de entrada pasan por múltiples tablas de flujo contenidas en los switch, de esta manera se precisa las acciones que deben tomar los paquetes en la red, se describe el funcionamiento del Pipeling Processing cuando ingresa un paquete al switch y su proceso a través de las tablas de flujo, mientras que en la Figura 13-2, se observa el proceso que sigue el paquete en cada tabla de flujo comenzando desde la tabla 0.

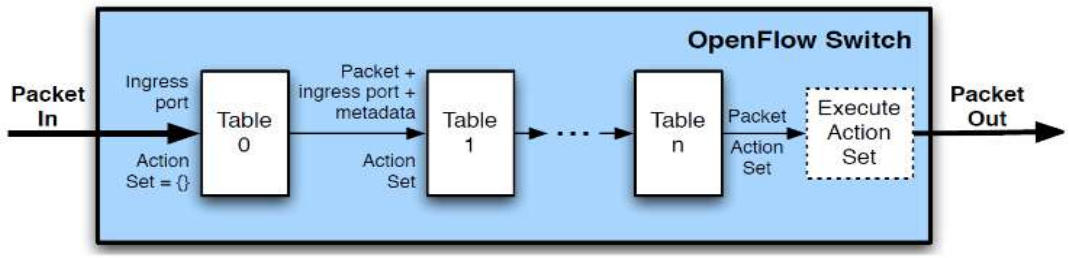


Figura 13-2: Entrada de Paquetes comparada con las tablas de flujo
Fuente:https://ri.itba.edu.ar/bitstream/handle/123456789/615/ITBA%20%20Integracion%20de%20Redes%20Ip%20%20%20Utilizando%20SDN.pdf

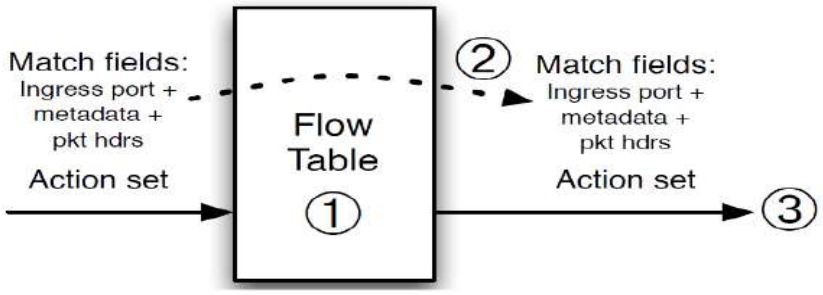


Figura 14-2: Proceso del paquete a través de una tabla de flujo
Fuente:https://ri.itba.edu.ar/bitstream/handle/123456789/615/ITBA%20%20Integracion%20de%20Redes%20Ip%20%20%20Utilizando%20SDN.pdf

En la Figura 14-2 se aprecia el orden que tiene la tabla de flujo para procesar el paquete y se describe a continuación:

- El paquete es comparado con la primera tabla de flujo, el orden se define desde la tabla de flujo 0 hasta la N.
- Se aplican las instrucciones para el paquete las cuales varían desde la salida por un puerto específico hasta la sucesión del paquete a otra tabla.
- Envío de los datos y la acción hacia la siguiente tabla.

Las tablas de flujo de la versión 1.3 del protocolo OpenFlow presentan ciertas adiciones en los flujos entrantes en comparación con la versión 1.0, en esta versión se agregan los campos Priority, Time-out y Cookie. En la Tabla 10-2 se observan los componentes de los flujos entrantes y posteriormente se explicará el funcionamiento de cada uno.

t

Tabla 10-2: Estructura de una entrada de Flujo OpenFlow 1.3

MATCH FIELDS	PRIORITY	COUNTERS	INSTRUCTIONS	TIMEOUTS	COOKIE	FLAGS

Fuente: ONF Open Networking Foundation.

Match fields: Son los encabezados del paquete, similar al protocolo OpenFlow en la versión 1.0 en la que se presentan similares encabezados.

Tabla 11-2: Cabecera de paquetes OpenFlow

Ingress Port	Ether Source	Ether Dst	Ether type	VLAN id	VLAN priority	IP source	P dst	IP Procol	P ToS bits	TCP/ UDP Src port	TCP/ UDP Dst Por

Fuente: ONF Open Networking Foundation

Prioridad. Este campo se utiliza para indicarle al sistema, a que tabla de flujo debe enviarse específicamente después de ingresar en la tabla 0.

Contadores. Este campo es destinado para la información estadística de los flujos de entrada, en la Tabla 12-2 se muestra una lista de conteos que realiza el sistema.

Tabla 12-2: Conteos realizados por el protocolo 1.3

Contador	Bits	
Por tabla		
Entradas activas	32	Opcional
Búsqueda de paquetes	64	Opcional
Paquetes emparejados 64	64	Opcional
Por flujo		
Paquetes recibidos	64	Opcional
Bytes recibidos	64	Opcional
Duración en s.	32	Opcional
Diración en ns.	32	Opcional
Por puerto		
Paquetes recibidos	64	Opcional
Paquetes transmitidos	64	Opcional
Bytes recibidos	64	Opcional
Paquetes transmitidos	64	Opcional
Recibe drops	64	Opcional
Paquetes transmitidos	64	Opcional
Errores recibidos	64	Opcional
Errores transmitidos	64	Opcional
Receive Frame Alignment Errors	64	Opcional
Receive Overrum Errors	64	Opcional
Receive CRC Errors	64	Opcional
Colisiones	64	Opcional
Duración en s.	32	Opcional
Duración en ns.	32	Opcional
Por cola		
Paquetes transmitidos	64	Opcional
Bytes transmitidos	64	Opcional
Transmit Overrun Errors	64	Opcional
Duración en s.	32	Opcional
Duración en ns.	32	Opcional
Por grupo		
Reference Count (flow entries)	32	Opcional
Packet Count	64	Opcional
Byte Count	64	Opcional
Duration (seconds)	32	Opcional
Duration (nanoseconds)	32	Opcional
Per Group Bucket		
Packet Count	64	Opcional
Byte Count	64	Opcional
Per Group Bucket		
Flow count	32	Opcional
Input packet count	64	Opcional
Input byte count	64	Opcional
Duration s.	32	Opcional
Duration ns.	32	Opcional
Per Meter Band		
In ban packet count	64	Opcional
In ban byte count	64	Opcional

Fuente: http://repositorio.upct.es/bitstream/handle/10317/5254/tfg729.pdf?sequence=1

Instructions. Son una serie de acciones y son ejecutadas en el proceso de pipeline.

Timeouts. Tiempo fuera o tiempo máximo que debe permanecer la entrada de flujo.

Cookies: Valor seleccionado por el controlador en el cual se filtra las modificaciones que se ejecuta en flujo.

Cabe mencionar que, por tratarse de un protocolo en constante desarrollo, nuevas funcionalidades que son importantes para aumentar su utilización se han añadido en la versión 1.3 del OpenFlow, pues sea añadido el soporte de flow meters, que da la posibilidad de limitar las tasas de paquetes enviados por un flujo. Con ello, es posible limitar el envío de muchos paquetes al controlador y al IDS, evitando la sobrecarga, así como limitar el uso de máquinas sospechosas.

Las Meter-tables miden la tasa de paquetes asignadas a ellas y se facilita el control de este. Consiste en entradas de medidas por flujo esto proporciona a OpenFlow implementar operaciones de QoS simples, como limitar el tráfico. En la Tabla 13-2 se describe a cada componente de un Meter.

Tabla 13-2: Principales Componentes de un Meter

COMPONENTE	DESCRIPCION
Band type	Entero sin signo de 32 bytes que identifica unívocamente.
Rate	contiene el ancho de banda mínimo asignado al flujo
Counter:	Se actualiza cuando un paquete es procesado por la meter band
Type specific arguments	Argumentos opcionales

Fuente: http://repositorio.upct.es/bitstream/handle/10317/5254/tfg729.pdf?sequence=1

2.3.3.8 Controladores SDN

El controlador es la parte medular, el alma fundamental de este tipo de redes, también es capaz de administrar y dirigir toda la topología de red. Varias de las funciones del controlador son la de centralizar todas las comunicaciones que pasa a través de los dispositivos de red, también tiene una visión global de la red.

El controlador es capaz de determinar cómo los flujos van a comportarse dentro de la red, los dispositivos que componen la red SDN se contactan con el controlador cada vez que estos lo necesiten, esto ocasiona que se ingresen nuevos flujos por la red, los dispositivos de red

desconocen o no definen qué hacer con el paquete y consulta al controlador que hacer con él; el controlador da las instrucciones a los dispositivos mediante la actualización de las tablas de flujo.

El controlador OpenFlow nos ofrece una interfaz de programación para los equipos compatibles, de manera que las aplicaciones son capaces de realizar tareas de gestión ofreciendo nuevas funcionalidades; dichas funcionalidades que nos provee son las siguientes:

- Estado de la red.
- Visualización de la topología de red.
- Cálculo de ruta.
- Una conexión TCP entre el controlador y los dispositivos.
- Una colección de APIs y servicios del controlador a las aplicaciones de gestión.

Por tratarse de un software, existen varias implementaciones de controladores OpenFlow disponibles, a continuación se describen brevemente los controladores OpenFlow que han sido más utilizados en trabajos, investigaciones y pequeñas implementaciones de redes de este tipo y se los menciona a continuación (Nikolaos, 2017) .

OpenDayLight.

OpenDayLight (ODL) es un proyecto de Código Abierto cuyo propósito es aumentar la innovación tanto en el diseño como en la implementación de SDN. Su fin es convertirse en una plataforma abierta, evitando que las aplicaciones privadas se apoderen del mercado y, a su vez la reducción de los costes de desarrollo.

La principal ventaja de OpenDayLight es que elimina las barreras de adopción, ya que algunas empresas no quieren atarse con un fabricante. Al ser una plataforma abierta, las empresas pueden optar por tecnologías de diversos fabricantes como: Hp, Big Switch Networks, Broacade, Cisco, Citrix, Ericsson, IBM, Juniper Networks, Microsoft, NEC, RedHat y VMWare, ya que son los fundadores principales del proyecto. Las principales repercusiones respecto a SDN de OpenDayLight respecto a las opciones tradicionales son:

- Una arquitectura que permita a un usuario que dentro de su infraestructura permita la instalación del controlador ODL permitiendo, por ejemplo: un protocolo BGP, un servicio AAA (Autenticación, Autorización y Contabilidad).
- Proporciona soporte para una amplia gama de protocolos y no únicamente OpenFlow.

- Soporte para el desarrollo de nuevas funcionalidades.

OpenDayLight realiza las siguientes acciones:

- Control centralizado de los dispositivos físicos y virtuales en la red.
- Control de los dispositivos con estándares y protocolos abiertos.
- Proporciona abstracción de alto nivel de sus capacidades para que los ingenieros de redes y los desarrolladores puedan crear nuevas aplicaciones para personalizar la configuración y administración de redes.

Los casos de uso para SDN son los siguientes:

- Centralizado de monitorización de red, gestión y coordinación.
- Gestión proactiva de redes e ingeniería de tráfico.
- Gestionar tanto la superposición virtual y la capa base física debajo de ella.
- Encadenamiento de paquetes a través de las diferentes máquinas virtuales.

SDN Van Controller

El software Controlador SDN de redes virtuales de aplicaciones (VAN) de HP proporciona un punto de control unificado en una red habilitada para OpenFlow, simplificando la administración, el aprovisionamiento y la orquestación. Esto permite la entrega de una nueva generación de servicios de red basados en aplicaciones.

También proporciona interfaces de programas de aplicaciones abiertas (API) para permitir a los desarrolladores de terceros entregar soluciones innovadoras para vincular dinámicamente los requisitos del negocio con la infraestructura de red a través de programas Java personalizados o interfaces de control REST de propósito general. El controlador VAN SDN está diseñado para funcionar en entornos de campus, centro de datos o proveedor de servicios.

- Plataforma de clase empresarial para la entrega de una amplia gama de innovaciones de red.
- Compatible con los protocolos OpenFlow 1.0 y 1.3.
- Soporte para más de 50 modelos de conmutadores HP habilitados con OpenFlow.
- Abrir API para habilitar el desarrollo de aplicaciones SDN de terceros.

- Arquitectura extensible, escalable y resistente del controlador.

La implementación de OpenFlow en HPE Switchs separa el tráfico de OpenFlow y el tráfico que no es de OpenFlow con las instancias de OpenFlow. El tráfico dentro de una instancia de OpenFlow no influye ni degrada el tráfico que no es de OpenFlow. Los comandos de configuración de OpenFlow se aplican por instancia.

NOX

El NOX fue uno de los primeros controladores OpenFlow, este controlador introdujo la idea de sistema operativo de red se basa en el lenguaje de programación C ++, se centra en la velocidad de procesamiento de los flujos. Su desarrollo parece haber parado a finales de 2013, desafortunadamente no fue muy utilizado debido a la escasez de su entorno de implementación y desarrollo, además, la documentación disponible no facilitaba funciones que se requiere para cumplir las condiciones impuestas por los escenarios de prueba.

POX

POX es un controlador que surgió de varias API (Application Programming Interfaz) utilizadas en el NOX, fue construido como una alternativa más amigable. Comparado con NOX, POX tiene un entorno de desarrollo más fácil para trabaja, además proporciona una GUI basada en la web y escrita en Python, que generalmente hace más cortos sus ciclos experimentales y de desarrollo.

BEACON

Beacon fue el siguiente gran paso en los controladores de código abierto. Está escrito en Java y altamente integrado en el Eclipse IDE, sin embargo, no era lo suficientemente flexible, ya que era limitado para destacar topologías (sin bucles).

FLOODLIGHT.

El controlador Floodlight, está escrito en JAVA, es compatible con OpenFlow 1.0 y 1.3 y su desarrollo continúa activo de acuerdo con su repositorio de código. Se construyó usando Apache Ant, una herramienta muy popular de compilación de software, que hace que el desarrollo de Floodlight sea más fácil y ágil. Posee una comunidad muy activa y una gran cantidad de características que se pueden agregar; crea un sistema que cumple impecablemente los requisitos

de una organización específica. Además de una interfaz Web y una GUI basada en Java están disponibles y la mayor parte de la funcionalidad de Floodlight es expuesto a través de una REST API.

RYU.

El Ryu es un controlador de SDN, implementado en el lenguaje de programación Python, es un programa de código abierto; Ryu se utiliza para la administración de redes y aplicaciones de control, uno de sus puntos fuertes es el soporte a varios protocolos tales como OpenFlow, Network Configuration Protocol (NETCONF).

Su desarrollo sigue activo, soporta las versiones 1.0 a 1.5 de OpenFlow completamente y utiliza pruebas para comprobar su compatibilidad con switchs, switchs además de realizar pruebas unitarias en su código, dando mayor confiabilidad en compatibilidad con las versiones de OpenFlow y de los switchs utilizados. Debido a su compatibilidad con los protocolos OpenFlow, posee un desarrollo activo, documentación actualizada y de fácil acceso.

Ryu en japonés significa "flujo", es un controlador abierto, facilita la administración para manejar el tráfico, el código fuente del controlador Ryu se puede encontrar fácilmente en GitHub y está disponible bajo la licencia Apache 2.0, adicionalmente el controlador Ryu admite la administración de red NETCONF y OF-config.

2.3.3.9 Características de los principales controladores

En la Tabla 14-2, se puede observar las características principales de los controladores más populares en la actualidad.

Tabla 14-2: Características de los principales controladores

	Beacon	Floodlight	NOX	POX	Trema	Ryu	ODL
Soporte OpenFlow	OF v1.0	OF v1.0, 1.3, 1.4, 1.5	OF v1.0	OF v1.0, 1.3, 1.4, 1.5	OF v1.3	OF v1.0, v1.2, v1.3, v1.4, 1.5 y extensiones Nicira	OF v1.0
Virtualización	MiniNet y Open vSwitch	MiniNet y Open vSwitch	MiniNet y Open vSwitch	MiniNet y Open vSwitch	MiniNet y Open vSwitch	Construcción de una herramienta virtual de simulación	MiniNet y Open vSwitch
lenguaje de Desarrollo	Java	Java	C++	Python	Rudy/C	Python	Java
Provee REST API	NO	SI	NO	NO	Si (Básica)	Si (Básica)	SI
Interfaz Grafica	Web	Web	Python+, QT4	Python+, QT4 , WEB	NO	WEB	WEB
Soporte de plataformas	Linux, Mac OS, Windows y Android para móviles	Linux, Mac OS, Windows	Linux	Linux, Mac OS, Windows	Linux	Linux	Linux, Mac OS, Windows
Soporte de OpenStack	NO	SI	NO	NO	SI	SI	SI
Multiprocesos	SI	SI	SI	NO	SI	NO	SI
Tiempo en el mercado	5 años	4 años	8 años	3 años	4 años	4 años	3 años
Documentación	Buena	Buena	Media	Pobre	Pobre	Media	Media

Fuente:http://dspace.uclv.edu.cu/bitstream/handle/123456789/7955/Ramon%20Zadiel%20Estrada%20de%20la%20Torre.pdf

2.4 Herramientas de emulación y simulación

En este entorno, con el controlador OpenFlow y el conmutador seleccionado, sería posible crear una red física, o valerse de una herramienta de emulación o simulación de red. Con el fin de familiarizarse con la línea de la elección de los switchs virtuales, se optó por un ambiente emulado para mayor flexibilidad y agilidad durante la ejecución de experimentos; con una diversidad de escenarios. Un levantamiento de información de herramientas compatibles con SDN / OpenFlow fue hecho y se describe a continuación.

2.4.1 NS-3

El ns-3 es un simulador de redes de código abierto, el objetivo de este simulador es ser un método flexible y popular entre investigadores en diferentes ambientes simulados o emulados de red. Aunque muy utilizado para las simulaciones de red, se encontró poca documentación sobre el uso de OpenFlow junto al ns-3, especialmente cuando este flujo es relacionado con los controladores externos. Se han encontrado documentos y modelos para ns-3 utilizando un controlador interno simulado, debido a la dificultad de integración con un controlador externo, este simulador no se mostró opción viable para ese trabajo.

2.4.2 EstiNet

EstiNet es un simulador y emulador de redes SDN con la capacidad de utilización de controladores externos. Se implementan las funcionalidades de forma completa, pero es un software propietario y una licencia es necesaria para su uso. Por ser un programa propietario y necesitar el uso de una licencia, el mismo no se utilizó para ese trabajo, EstiNet puede trabajar en modo simulador y emulador y es un emulador confiable incluso utilizando grandes cargas de datos.

2.4.3 Mininet

Mininet es un emulador de redes, permite creación de redes de máquinas virtuales, switchs, controladores y enlaces, esto hace posible el control sobre el entorno de pruebas de forma general. Se trata de un programa de código abierto, existe una gran documentación de su uso e integración con los switchs OpenFlow y controladores externos y su desarrollo sigue activo. La creación de ambientes es muy rápida, lo que hace que el tiempo de los experimentos se aproveche, permitiendo una mayor variedad de ambientes.
Su mayor limitación es que actualmente no puede exceder el uso de la CPU y el uso de red de un solo servidor. Debido a la gran documentación disponible y su buena integración con controladores y switchs OpenFlow es elegido para utilizarse en este trabajo de investigación.

2.5 Herramientas de análisis de red

A lo largo del proyecto se ha utilizado diversas herramientas de análisis de red, con el fin de verificar su funcionalidad y rendimiento en cada escenario propuesto. Entre las principales herramientas utilizadas se mencionan las siguientes:

2.5.1 Iperf

Esta es una herramienta utilizada en redes informáticas para la realización de pruebas, permite crear flujos de datos TCP y UDP con la que se puede medir el rendimiento de una red. También nos permite ajustar varios parámetros para realizar pruebas y ajustes en una red; este puede funcionar como cliente o servidor midiendo el rendimiento en los puntos extremos, siendo un software de código abierto y múltiple plataforma.

2.5.2 WireShark

Esta es una herramienta utilizada en redes informáticas para la captura y posterior análisis de tráfico de red. Admite una gran cantidad de diversos protocolos permitiendo de ser necesario filtrarlos. Su gran potencia radica en sus continuas actualizaciones siendo un software de código abierto y múltiple plataforma.

2.5.3 Mgen

Mgen es una herramienta utilizada para la medición de capacidades de enlaces, por medio de scripts con el fin de realizar procesos de flujos de tráfico, que pueden ser generados de maneras diferentes.

2.5.4 TcpDump

TcpDump es una herramienta la que, a través de un terminal, permite la captura de tráfico, y aunque no es muy potente en sus características, se puede realizar en tiempo real la captura de tráfico por medio de ssh.

2.6 Generadores de tráfico

Para crear tráfico existen variedad de herramientas, las más conocidas son: ping, wget, curl, netperf, netcat, entre otros. A continuación, se detallan otros generadores de tráfico.

2.6.1 Hping

Hping permite crear paquetes personalizados de ICMP, UDP, TCP y Raw IP. Puede ser útil para pruebas de cortafuegos, escaneo de puertos avanzado, prueba de red usando diferentes protocolos, le permiten enviar archivos, también es posible probar diferentes casos ataques. Esta herramienta no tiene interfaz de usuario, por lo que solo se puede ejecutar a través de la línea de comando.

2.6.2 Curl

Curl es una biblioteca y una herramienta que se ejecuta desde la línea de comandos para transferir datos hacia o desde un servidor a través de enlaces (URL). La aplicación es compatible con un número muy grande de protocolos (DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMB, de SMBS, SMTP,

SMTPS, TELNET, TFTP) y características como conexión SSL, autenticación de usuarios y cookies. Se utilizará Curl para interactuar con la API REST del controlador FloodLight.

2.6.3 SIPp

Simulador de llamadas SIP, utilizado para volver a crear llamadas VoIP, para el uso en la simulación, problemas en el lanzamiento de múltiples llamadas desde un host en momentos diferentes han hecho que este no podía utilizarse en la simulación final, pero sólo para pruebas.

2.6.4 Ping

Ping es un comando que viene instalado por defecto en el Mininet, teniendo como el objetivo de probar el RTT de un host a otro, utilizando el protocolo ICMP. Este programa, debido a su simplicidad y rapidez, fue muy utilizado en el desarrollo con el fin de probar las conexiones.

2.6.5 D-TIG

El generador de tráfico de Internet distribuido (D-ITG) es una plataforma forma capaz de producir tráfico que se adhiere con precisión a patrones definidos por el tiempo de salida entre paquetes (IDT) y los procesos estocásticos de tamaño de paquete (PS). Tales procesos se implementan como una secuencia aleatoria variables, hay disponible una gran variedad de distribuciones de probabilidad: constante, uniforme, exponencial, Pareto, Cauchy, normal, Poisson y gamma. Además, D-ITG incorpora algunos modelos propuso emular fuentes de varios protocolos: TCP, UDP, ICMP, DNS, Telnet y VoIP (G.711, G.723, G.729, detección de actividad de voz, RTP comprimido). Esto significa que el usuario simplemente elige uno de los protocolos soportados y la distribución de IDT y PS será automáticamente configurada.

CAPÍTULO III

3 DISEÑO DE LA INVESTIGACIÓN

3.1 Tipo y diseño de investigación

Este proyecto es una investigación cuasi-experimental (Kirk, 1995) donde se afirma que los diseños cuasi-experimentales se utilizan cuando la asignación de sujetos o grupos experimentales aleatorios no es posible. Por ende, las políticas de QoS a ser implementadas en este proyecto, dependerán exclusivamente del ámbito en la que se desarrolle, según sus propias necesidades; debiendo destacar que éstas políticas son flexibles y adaptables en un entorno empresarial.

3.2 Métodos de investigación

Para el presente proyecto se utilizará el método hipotético-deductivo, ya que parte de una hipótesis para llegar a casos particulares, realizando un análisis en diferentes ambientes de prueba los cuales serán evaluados mediante la experimentación y medición de las variables de estudio propuesto para este proyecto.

Se utilizará la distribución de probabilidad denominada T-Student, para examinar las diferencias entre dos muestras independientes y pequeñas que tengan distribución normal y homogeneidad de sus varianzas, con las que se realizará múltiples comparaciones de cada arquitectura con sus respectivos indicadores de rendimiento (Turcios, 2015).

Es importante mencionar que para el desarrollo de la implementación de políticas de QoS en escenarios SDN, se escogerá al controlador más adecuado en base a la escala de medición Likert.

3.3 Enfoque de la investigación

La investigación tendrá un enfoque cualitativo-cuantitativo: cualitativa por lo que se buscará la mejor alternativa de solución para mejorar el rendimiento de una red, facilitando la gestión a los administradores y cuantitativa por las mediciones recopiladas en los ambientes de pruebas a realizarse.

3.4 Alcance investigativo

El alcance de este proyecto, pretende establecer Políticas de QoS para la gestión de tráfico tanto en un ambiente tradicional como bajo la plataforma SDN.
En un inicio se parte de un análisis de las políticas de QoS en entornos SDN y en convencionales, con el objetivo de determinar los requerimientos previos al desarrollo de escenarios propuestos. Posteriormente se categorizará el tráfico con mayor prevalencia en una red empresarial de ámbito académico, por tener acceso a contenidos relacionados con estudios de flujos de tráfico.

Una vez seleccionadas las políticas de calidad de servicio y definido el tipo de tráfico, se procederá con la elección de herramientas de hardware y software a ser utilizadas en los escenarios, los cuales serán implementados en una infraestructura física equivalente y constan de un conmutador conectado con dos terminales (servidor y cliente).

El primero de estos, será basado en una infraestructura real convencional, el cual tiene como conmutador un equipo HPE 3800 Series. El segundo escenario será implementado con tecnología SDN, utilizando como conmutadores dos módulos Zodiac Fx y como escenario final, se implementará con tecnología SDN, utilizando el equipo HPE 3800 Series.

Cabe indicar que el servicio propuesto será implementado en un escenario bajo el sistema operativo Linux, y se enfocará en la principal característica de SDN, que es la separación del plano de control (software) del plano de datos (hardware); respetando las jerarquías de las tres capas de ésta arquitectura; es decir en la primera capa se encontrará la infraestructura de red compuesta por un conjunto de hosts, switchs, etc.

Mientras que en la segunda capa, existirá un controlador SDN que soporte el protocolo OpenFlow; con este componente se podrá priorizar paquetes, administrar flujos, entre otras funciones de red; por medio de API's que se configurarán para gestionar la red y controlar el flujo de la capa de datos; y como tercera capa estará formada por todos aquellos servicios y aplicaciones de usuario cuya misión será la de comunicar al controlador sus necesidades, para tomar decisiones y dar solución a problemáticas actuales que día a día enfrentan los administradores de red. En las siguientes figuras, se muestra los escenarios a ser implementados.

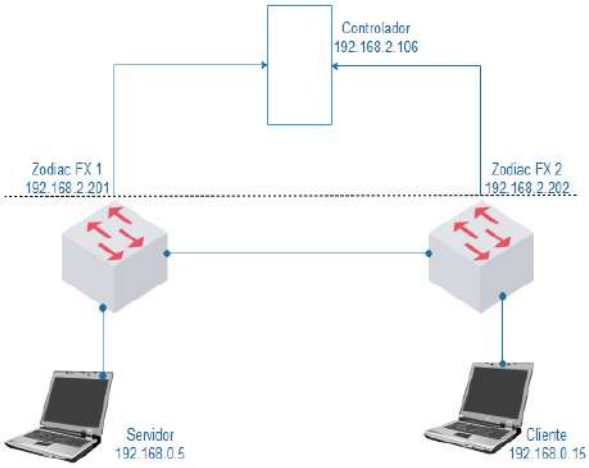


Figura 1-3: Diseño de la Red SDN con Zodiac FX
Realizado por: Marcelo Criollo, 2019

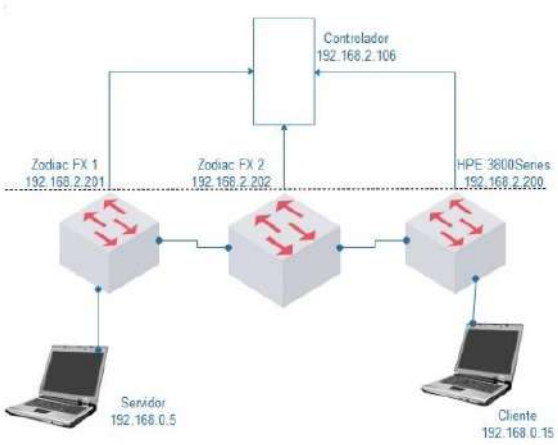


Figura 2-3: Diseño de la red SDN con Hp
Realizado por: Marcelo Criollo, 2019

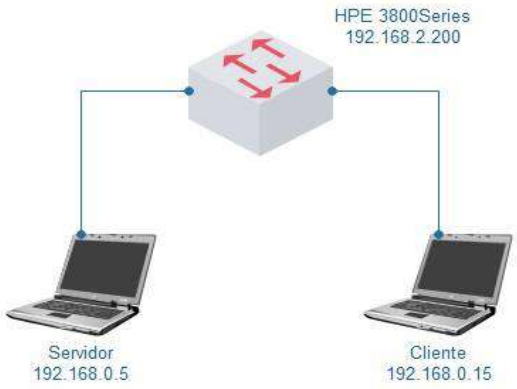


Figura 3-3: Diseño de la Red Tradicional con Hp

Realizado por: Marcelo Criollo, 2019

Se debe mencionar que si bien es cierto este proyecto se enfoca en entornos empresariales, estos escenarios propuestos fueron diseñados con una topología básica, debido a que el análisis de tráfico se centra en un nodo de la red, el mismo que a pesar de ser pequeño, usa altas cargas de tráfico.

Adicionalmente se ha considerado que por medio de ambientes de pruebas se realizará mediciones de las variables de estudio, tales como latencia, jitter, ancho de banda y pérdida de paquetes; provocando en la red situaciones extremas con el fin de poner a prueba la funcionalidad de las políticas de QoS; en otras palabras, se considerará que por la red circule tráfico, de esta manera se simulará condiciones de saturación.

Las variables de estudio fueron elegidas, desde una perspectiva de red empresarial, según lo define Cisco para maximizar la disponibilidad y el rendimiento en sus equipos de tele-presencia, éstas demandas de red están bien documentadas en la Guía de diseño de Cisco Tele-Presence Network Systems 1.1, lo cual sirve como una referencia de QoS, más aún si los servicios de Tele-Presencia requieren de parámetros puntuales para garantizar una calidad de servicio (Cisco, 2008). Los requisitos de nivel de servicio de tiempo real propuesto por Cisco en aplicaciones como: Tele-Presencia y videoconferencia, se definen mediante los siguientes parámetros:

- Bandwidth.
- Latency (delay).
- Jitter (variations in delay).
- Paquetes perdidos.

En la guía mencionada, se presenta una tabla comparativa que resume los requisitos del nivel de servicio de aplicaciones genéricas de Video-Conferencia y Tele-Presencia, la misma que se muestra a continuación:

Tabla 1-3: Requisitos de nivel de servicio, video conferencia y tele-presencia

Service Level Parameter (Target Values)	(Generic) Videoconferencing/Video Telephony	Cisco TelePresence
Bandwidth	384 kbps or 768 kbps + network overhead	1.5 Mbps to 12.6 Mbps + network overhead
Latency	400-450 ms latency	150 ms latency
Jitter	30-50 ms peak-to-peak jitter	10 ms peak-to-peak jitter
Loss	1% random packet loss	0.05% random packet loss

Fuente: https://www.cisco.com/c/dam/en/us/td/docs/solutions/TelePresence_Network_Systems_1-1_DG.pdf

3.5 Población de estudio

Para determinar la población de estudio se debe tomar en consideración que se utilizarán tres escenarios en los que aplicarán políticas de QoS, el primero en un entorno convencional y los otros dos escenarios con tecnología SDN, pero con diferente hardware. Cabe indicar que estos escenarios propuestos no son un estándar para todo tipo de infraestructura de red, sin embargo, se intenta englobar los principales componentes inherentes a una red empresarial.

Específicamente para la población de estudio, se definirán tres ambientes empresariales de prueba, uno de tipo convencional y dos basados en SDN; debido a esto se considera a la muestra como el total de la población existente, la que será determinada por el número de pruebas realizadas para el análisis, pues se aplicará el método de análisis estadístico T-Student, de distribución normal, con n < = 30, por ende, la muestra será de 30 interacciones de prueba.

3.6 Unidad de análisis

La delimitación de la unidad de análisis escogido para el proyecto correspondería al número de pruebas o experimentos que serán de 30 pruebas por cada índice y por cada escenario propuesto.

3.7 Selección de la muestra

Se utilizará el método de muestreo estratificado, aplicando estimadores puntuales como son la proporción, de tal forma que tengan la misma probabilidad de ser seleccionado, para ser parte de la muestra. La muestra será de treinta interacciones de prueba, con sus respectivas mediciones, con la finalidad de obtener datos numéricos en base a indicadores tales como: latencia, jitter, ancho de banda y paquetes perdidos; de esta manera se verificaría el funcionamiento de las políticas de QoS evaluando el rendimiento de la red y comparando los tres escenarios.

3.8 Tamaño de la muestra

Considerando que en el presente proyecto no se tendrá la certeza que las muestras recabadas tengan una tendencia de distribución normal, que, según el teorema central de límite, se considera que según (Triola, 2009) debe cumplir:

- La muestra debe ser del tipo aleatoria simple.
- La desviación estándar debe ser conocida.

- La población debe estar normalmente distribuida (n>30).

Por ámbito de la investigación se optará por la realización de estudio de tamaño de muestra sin distribución normal con n<=30, por lo que no se utilizaran fórmulas específicas sino se usarían pruebas no paramétricas.

3.9 Técnica de recolección de datos primarios y secundarios

Como técnicas, se basará este proyecto en la técnica de investigación que consistirá en la recopilación de datos para tener un fundamento teórico y poder realizar el respectivo análisis que se ha planteado como uno de los objetivos específicos en este documento. Así como también se utilizará la técnica de la observación para el momento de la implementación y las pruebas, se obtendrán resultados los cuales permitirán realizar cambios y corregir posibles errores.

Los instrumentos utilizados serán simuladores, apuntes de las observaciones realizadas, registros de eventos e instrumentos de medición como son los analizadores de red. Estos instrumentos permitirán obtener la información adecuada para este trabajo de investigación.

Tabla 2-3: Recolección de la información

PREGUNTAS	EXPLICACIÓN
1. ¿Para qué?	Recolectar información primaria para comprobar y contrastar con la hipótesis.
2. ¿A qué personas o sujetos?	La población se tomará realizando un total de 30 pruebas en los escenarios de prueba uno tradicional y dos basados en SDN.
3. ¿Sobre qué aspectos?	VI. Asignación de políticas de calidad de servicio en entornos SDN y convencional. VD. Rendimiento de la red.
4. ¿Quién?	Investigador.
5. ¿Cuándo?	De acuerdo al cronograma establecido.
6. ¿Lugar de recolección de la información?	Escenarios de prueba.
7. ¿Cuántas veces?	1 sola vez.
8. ¿Qué técnica de recolección?	Búsqueda de información, Pruebas, Observación y Análisis.
9. ¿Con qué?	Papers, analizadores de red e inyector de tráfico.
10. ¿En qué situación?	Situación normal y cotidiana.

Realizado por: Marcelo Criollo, 2019

3.10 Instrumentos para procesar datos recopilados

Para el procesamiento de datos y análisis estadísticos se utilizará un software especializado SPSS versión 23, apoyado en un análisis técnico que será realizado al momento de evaluar las políticas de QoS implementadas; abarcando aspectos de diseño de topología, interfaz, y parámetros: ancho de banda, latencia, jitter y pérdida de paquetes, en los tres escenarios implementados.

3.11 Variables e indicadores

Variable independiente.

- Políticas de QoS,

Variable dependiente.

- Rendimiento de la red.

3.12 Operacionalización de variables

Tabla 3-3: Operacionalización de variables

Hipótesis general	Variables	Tipo	Indicadores
La implementación de políticas de QoS en redes empresariales SDN, mejorará el rendimiento frente a redes convencionales.	Políticas de QoS.	Independiente	Implementar políticas de QoS en redes empresariales SDN y convencionales.
	El rendimiento de la red.	Dependiente	Evaluación de rendimiento en redes empresariales SDN y convencionales a través de parámetros y herramientas de medición.

Realizado por: Marcelo Criollo, 2019

3.12.1 Matriz de consistencia

Tabla 4-3: Matriz de consistencia

Formulación de problema	Objetivo General	Hipótesis General	Variables	Indicadores	Índices	Técnicas	Instrumentos
¿La asignación de políticas de QoS en redes empresariales de entre entornos convencionales y SDN, cuál ofrece un mayor rendimiento?	Analizar el rendimiento que genera la asignación de políticas de QoS, entre entornos de redes convencionales y SDN, dentro de una red empresarial.	La implementación de políticas de QoS en redes empresariales SDN, mejorará el rendimiento frente a redes convencionales.	**Independiente:** Políticas de QoS.	Tráfico.	-Tipo de tráfico. -Prioridades.	-Investigación. Análisis. Observación.	-Escenario lógico de la red. -Simulación de topologías. -Equipos de red. -Controladores SDN. -Herramientas de desarrollo de red.
				Equipamiento	-Cantidad de conmutadores y equipos terminales.		
				Herramientas SW	-Cantidad de máquinas virtuales -S.O. Ubuntu -Porcentaje de Efectividad de protocolos y versiones -Popularidad de controladores -Analizadores de red Generador de Tráfico		
				Lenguaje de Programación	-Nivel de complejidad de programación -Tiempo de compilación y ejecución -Compatibilidad de plataformas y controladores		
				Normas y estándares de QoS	-Confiabilidad -Seguridad -Eficiencia -Desempeño -Disponibilidad		
				Políticas de QoS	-Bloqueo de tráfico y puertos -Demanda de servicios Tiempo de respuesta		
			Dependiente. El rendimiento de la red.	Pruebas de rendimiento	-Latencia. -Jitter. -Ancho de Banda -Pérdida de paquetes	-Observación. -Experimentación -Pruebas.	-Analizadores de red. -Procesamiento de resultados. -Registro de eventos.

Realizado por: Marcelo Criollo, 2

3.13 Propuesta tecnológica

3.13.1 Desarrollo del proyecto

Los escenarios han sido implementados usando equipamiento SDN con una topología básica debido a que el análisis de tráfico está enfocado en un nodo de la red, adicionalmente se ha considerado que en los ambientes de prueba se evaluará el rendimiento de la red bajo los siguientes parámetros de calidad de servicio: latencia, jitter, ancho de banda y paquetes perdidos; simulando condiciones de saturación con el generador de tráfico DIT-G para medir el rendimiento de la red.

3.13.2 Políticas de QoS

Basado en el análisis de las políticas de QoS detallados en la sección anterior, estas políticas son implantadas en 3 escenarios reales para evaluar el rendimiento de la red, por lo cual se propone un algoritmo con la finalidad de diseñar las políticas de QoS en redes convencionales y SDN, como se muestra en la siguiente figura.

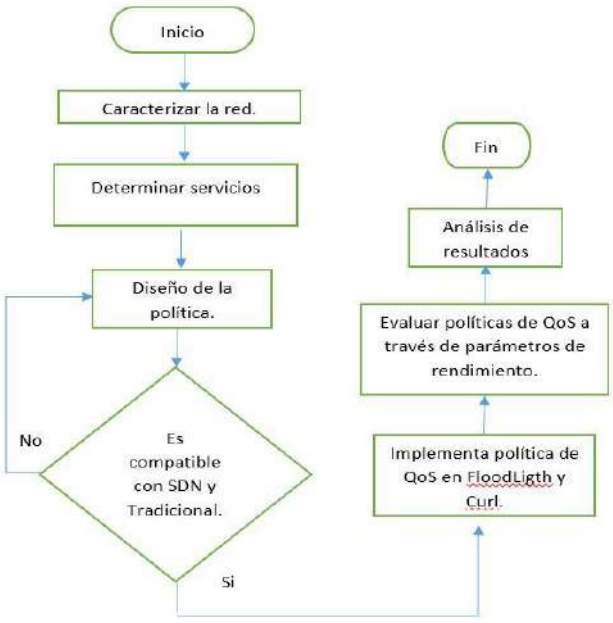


Figura 4-3: Algoritmo para el diseño de políticas de QoS
Realizado por: Marcelo Criollo, 2019

Para la aplicación de las políticas de QoS, se propuso que estas sean compatibles tanto para escenarios convencionales como SDN, seleccionando una ACL numerada, la que corresponde a un escenario tradicional y con funcionalidades similares en el escenario SDN, considerando que las pruebas a realizar deben ser ejecutadas en ambas tecnologías con parámetros similares.

3.13.3 Implementación de políticas

Teniendo en cuenta lo anterior, se procede a elegir las políticas de calidad de servicio que serán implementadas en los escenarios. Antes de aplicar los procedimientos y técnicas de la calidad de servicio, se deber realizar un análisis de la red actual, ya que se debe tener en cuenta aspectos como: escalabilidad, comportamiento y los medios físicos que están asociados con la red y fundamentalmente realizar un análisis de las características de los tipos de tráfico a cruzar por la red de comunicación.

Para realizar este análisis de calidad de servicio en un entorno empresarial, se seleccionó una institución de educación superior, caso puntual la Pontificia Universidad Católica del Perú, debido a que en este tipo de ambientes académicos se genera congestión de tráfico en la red y se presentan diversos problemas al momento de gestionar a la infraestructura de red, ya que en la actualidad por el gran número de conexiones simultaneas por parte del personal administrativo, docente y de los estudiantes, existen circunstancias específicas en donde el aumento en el ancho de banda no es garantía suficiente para poder soportar las crecientes aplicaciones, caracterizadas en su mayoría por tráficos multimedia.

Es en este aspecto donde las técnicas de Calidad de Servicio QoS se hacen esenciales para brindar un mejor rendimiento en la red, englobando una serie de mecanismos y estrategias que posibilitan la priorización del tráfico y la asignación de recursos a la red, de tal manera que la información alcance su destino de forma veraz y predecible, haciendo a su vez las redes más eficaces y confiables para todo tipo de aplicaciones (Kang, Rottenstreich, Rao, & Rexford, 2015) .

Como parte de la investigación para el análisis de las políticas de Calidad de Servicio, en el presente trabajo, se consultaron diferentes fuentes de información, enfocadas con las problemáticas y las soluciones más comunes que se tiene en una infraestructura de red, para ello se interactuó con administradores de red y con los usuarios, para tener una visión real de esta temática, pero además con el fin de fundamentar el estudio, se llegó a la plataforma académica de SCOPUS, en donde se generaron 12 publicaciones sobre QoS en entornos empresariales ver (ANEXO1), de las cuales hay un artículo llamado: “Alpaca: Compact Network Policies With Attribute-Encoded Addresses”, en el que consta un estudio realizado a 22 universidades sobre el tema de análisis de políticas de calidad de servicio y se propone un algoritmo personalizado, denominado ALPACA, para ofrecer QoS (Kang et al., 2015).

Para llegar a desarrollar este algoritmo, se presenta un estudio realizado a través de una encuesta enfocada en QoS y control de acceso a las universidades ver (ANEXO 2), con el fin de analizar

las técnicas que cada una aplican para mejoramiento de la infraestructura de red, por tal motivo se hace mención en este documento del estudio realizado en dicho artículo.

El análisis indica que, en las universidades encuestadas, se encuentran comúnmente hosts limitados por una determinada cuota de ancho de banda, en seguridades limitan a los usuarios externos que puedan acceder a un host determinado o acceso a un conjunto limitado de servicios en el campus, otras opciones de seguridad corresponden a diferentes restricciones sobre qué puerto están permitidos para acceder a las aplicaciones y servicios (HTTP, POP3, FTP). También, algunas entidades ofrecen una mayor calidad de servicio para los anfitriones asignados para uso educativo (por ejemplo, para transmitir medios multimedia de alta calidad en un aula); por otro lado, uno de los trabajos como administradores es, realizar el equilibrio de carga del servidor para la Web interna, es decir permitir acceso servicios basados según el rol, para evitar la carga pesada de un grupo de usuarios que comprometan el rendimiento de otros usuarios.

Otra opción de políticas de QoS aplicada en la mayoría de estas entidades analizadas, es la asignación de hosts a VLAN basadas en el rol (por ejemplo, facultad, personal y estudiante), para el aislamiento del tráfico, evitando el rastreo de paquetes y el tráfico excesivo de difusión.
Luego de haber identificado las posibles mejoras realizadas en universidades, según el artículo mencionado anteriormente en este apartado, se deben definir las políticas de QoS a ser implementadas en este proyecto propuesto; en este caso en particular, se decidió dar prioridad a un cierto tráfico que sale y entra de una determinada dirección IP, así como también dar acceso o restringir servicios, protocolos, puertos, interfaces o equipos físicos, utilizando ACLs, Firewall, reglas y configuraciones. Se realiza la implementación de las políticas en un ambiente de pruebas tanto con tecnología SDN como en un ambiente convencional.

3.13.4 Categorización del tráfico en redes empresariales

La importancia de categorizar el tráfico de Internet en entornos empresariales, radica en determinar políticas en una red; por su naturaleza el Internet está conformado en mayor porcentaje por el protocolo IP. Según el proyecto de titulación denominado "DISEÑO E IMPLEMENTACIÓN DE UN BALANCEADOR DE CARGA PARA LA OPTIMIZACIÓN DE LOS RECURSOS DE PROTECCIÓN EN UNA RED ENTERPRISE MEDIANTE UN BANCO DE FIREWALLS N:1 CONTROLADO VÍA SDN", desarrollado en la Pontificia Universidad Católica del Perú, en el que se realizó un estudio de los flujos de tráfico, concluye que se considera a los flujos que tienen más tiempo de duración, cuando son superior a los 10 minutos y estos representan el 20% del volumen de tráfico total, dentro de este tipo de flujos están las sesiones

P2P, es decir, se generan flujos de mayor duración, cuando todos los funcionan sin clientes ni servidores fijos, sino que tienen un compartimiento como iguales entre sí.

Para realizar esta categorización de tráfico, el autor del proyecto de titulación mencionado (Quisphe, 2017), desarrolla un script, utilizando la librería pypcapfile de Python (GitHub, 2017a) con algunas modificaciones, la cual fue extraída del repositorio de GitHub (GitHub, 2017b); que al ser ejecutado por un tiempo de duración de 20 minutos y procesando 23 millones de paquetes, se puede observar que el resultado obtenido, demuestra que el protocolo TCP se encuentra en una mayor proporcionalidad en relación a los otros protocolos analizados en ese estudio, como son: UDP, ICMP, GRE y ESP.

Tabla 5-3: Estadísticas en una traza de 20 min

Estadísticas\ Protocolo	ICMP	TCP	UDP	GRE	ESP
% PAQUETES	0.27	80.3	19.2	-	-
% VOLUMEN DE TRÁFICO (BYTES)	0.04	90.5	9.3	0.13	-
% FLUJOS	2.4	52.1	45.5	-	-

Fuente: (Quisphe, 2017)

Otra prueba adicional realizada en ese artículo, en donde se analiza por 6 horas y media, 484 700 000 paquetes, se obtiene los resultados mostrados en la siguiente tabla:

Tabla 6-3: Estadísticas en traza de 6 horas.

Estadísticas\ Protocolo	ICMP	TCP	UDP	GRE	ESP
% PAQUETES	0.31	77.5	21.9	0.2	-
% VOLUMEN DE TRÁFICO (BYTES)	0.05	89.4	10.3	0.17	-
% FLUJOS	2.5	60.4	36.9	-	-

Fuente: (Quisphe, 2017)

Con este análisis, se procederá a realizar las pruebas para la implementación de políticas de calidad, capturando tráfico con los protocolos TCP y UDP, ya que como se puede apreciar en las trazas analizadas hay mayor concentración de tráfico en estos dos protocolos.

3.13.5 Recursos de topología

3.13.5.1 Hardware

- Un conmutador HP 3800 Series 24G-25FP + J9575, para la implementación de los escenarios propuestos, el equipo HP funciona con el protocolo OpenFlow, lo que nos permite probar redes SDN, así como también al ambiente tradicional. Tiene un

procesador ASIC / ARM @ 350 MHz; Freescale P2020 @ 1200 MHz, 4 Gb de flash y 2 Gb SDRAM.

- Dos Zodiac FX: es un switch OpenFlow desarrollado por Northbound Networks, que permite funcionalidades SDN para ser probado fácilmente en hardware. Sin embargo este equipo está diseñado para pruebas y no capacitado para entornos de específicamente para los servicios SDN y no se puede usar en entornos de producción. Se basa en el Atmel Microcontrolador ARM® Cortex®-M4 y tiene cuatro 10/100Mb puertos Ethernet. Compatible con Openflow 1.0 y 1.3.
- 3 computadoras (cliente, servidor, controlador).

3.13.5.2 Software

- Virtual Box 5.2.12: Herramienta de virtualización de código abierto, multiplataforma para Linux, Windows y MacOSX.
- Ubuntu 16.04 Desktop: sistema operativo base para el controlador.
- Java JDK 10.02: complemento para D-ITG.
- Controladores: OpenDayLight Nitrogen, Floodlight, RYU, Hp Van Controller, controladores evaluados para seleccionar el más apropiado.
- Wireshark 2.6.0: analizador de tráfico de red.
- D-ITG 2.61, GUI 0.92: Generado de tráfico.

3.13.6 Selección del controlador

Cumpliendo los objetivos de este proyecto se realizaron pruebas y experimentos analizando las variables Ancho de Banda, Latencia y Jitter en diferentes escenarios de red, divididas en pruebas físicas y simuladas con la misma topología encontradas en el capítulo III.

En donde se decidió escoger como controlador a utilizar a Floodlight, por ser el más completo en sus funcionalidades en comparación a los demás controladores probados, está decisión de controlador se fundamente en la evaluación mediante Escala de LIKERT, como se observa a continuación (Seaman, 2007).

Tabla 7-3: Evaluación de controladores con la escala de Likert

Controlador / Características	OpenDayLight	CALF.	Ryu	CALF.	Floodlight	CALF	SDN VAN Controller	CALF
PROGRAMACIÓN	Java	3	Python	5	Java	3	Java	3
SOPORTE – PROTOCOLO OF	OpenFlow V1.O –V1.3	2	OpenFlow V1.O – V1.3	2	OpenFlow V1.O – V1.3	2	OpenFlow V1.O – V1.3	2
HERRAMIENTAS DE SIMULACIÓN	MININET	5	MININET	5	MININET	5	MININET	5
PLATAFORMAS S.O. OpenStack	Windows * Linux * OS X	**5**	Linux	**2**	Windows * Linux * OS X	**5**	Linux	**2**
PROVEE REST API	Si	**5**	Si (Básica)	**5**	Si	**5**	Si (Básica)	**5**
TIPO DE SOFTWARE	Código Abierto	**5**	Código Abierto	**5**	Código Abierto	**5**	Código Abierto	**5**
DIFICULTAD DE INSTALACIÓN	Complejo	**3**	Fácil	**5**	Complejo	**3**	Fácil	**5**
COMPATIBILIDAD TOPOLÓGICA DE EQUIPOS	No	**3**	No	**3**	Si	**5**	No	**3**
TIEMPO EN EL MERCADO	2 años 5 meses	**3**	3 años	**4**	4 años	**5**	3 años	**4**
DOCUMENTACIÓN	Buena	4	Buena	4	Muy Buena	5	Media	**3**
TOTALES		38		40		43		37

Realizado por: Marcelo Criollo, 2019

3.13.7 Selección de simulador

Por otro lado, según lo mencionado en el marco teórico existen variedad de simuladores, esto fue útil como aprendizaje previo para incorporarnos en el mundo de SDN; para el desarrollo de este proyecto de grado se decidió optar por el simulador Mininet ya que tiene ciertas ventajas que a continuación serán mencionadas.

Tabla 8-3: Comparación de los principales simuladores de red.

General	Mininet	Packet Tracer	GNS3	EstiNet
Free software	Yes	Yes	Yes	No
Open source	Yes	No	Yes	Yes
Publicly downloadable	Yes	No	Yes	No
Windows support	No	Yes	Yes	No
Linux support	Yes	Yes	Yes	Yes
Fully functional IOS	No	No	Yes	No
Simulation mode	No	Yes	Yes	Yes
Emulation mode	Yes	No	Yes	Yes
Compatible with real world controllers	Yes	No	No	Yes
Result repeatable	No	No	No	No
Scalability	Middle by multiple processes	No	No	High by single process
Performance result correctness	Depend of resources	No	Yes	Yes
Wifi	Yes	Yes	No	Yes
Gui support	Yes	Yes	No	No

Fuente:http://www.artigos.com/index.php?option=com_mtree&task=att_download&link_id=22407&cf_id=24

3.13.8 Selección de generador de tráfico

Con la finalidad de realizar las pruebas, se utilizó al generador de tráfico D-ITG, el cual permite inyectar diferentes tipos de tráfico personalizado, recreando así un ambiente real. Fue elegido este software en base a un estudio realizado en un artículo llamado: D-ITG Distributed Internet Traffic Generator, en el cual, se presenta un análisis comparativo entre generadores de tráfico, como se observa en la tabla siguiente:

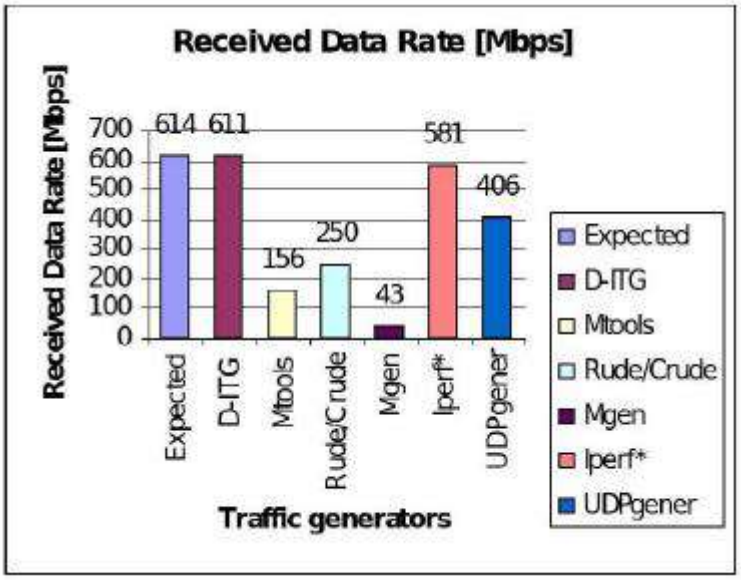


Figura 5-3: Análisis comparativo de generadores de tráfico
Fuente: http://ijeee.iust.ac.ir/article-1-692-en.pdf

Adicionalmente fue escogido D-ITG, porque permite implementar algunos protocolos en la capa de aplicación como: VoIP, Telnet, DNS, Quake III, entre otros y en general, permite implementar protocolos hasta en la capa de transporte, por lo que se logra emular la pila de protocolos TCP/IP completa, en consecuencia, la medida que se obtiene usando el D-ITG es aceptable porque usa los protocolos de las capas de transporte, red e interfaz (R. Mosavi, F. Farabi, 2015).

3.13.9 Selección de hardware

Zodiac FX

Es un conmutador OpenFlow desarrollado por Northbound Networks, que permite que la funcionalidad de la red SDN sea probada fácilmente en hardware. Sin embargo, el conmutador está diseñado específicamente para servicios SDN y no se puede usar en entornos de producción. Se basa en el microcontrolador Atmel ARM® Cortex®-M4 y tiene cuatro puertos Ethernet 10 / 100Mbs.

En la práctica, el conmutador se basa en una placa de circuito y contiene cuatro interfaces, tres de ellas para OpenFlow y una para tráfico Ethernet estándar. La siguiente imagen, muestra un conmutador en funcionamiento, en la esquina superior izquierda hay un conector Micro-USB para

administración de energía y fuente de alimentación, en la parte inferior cuatro conectores RJ-45 para tráfico OpenFlow y Ethernet. Este conmutador admite las versiones 1.0 y 1.3 de OpenFlow e incluye para la gestión del dispositivo tanto la línea de comando como una interfaz web.

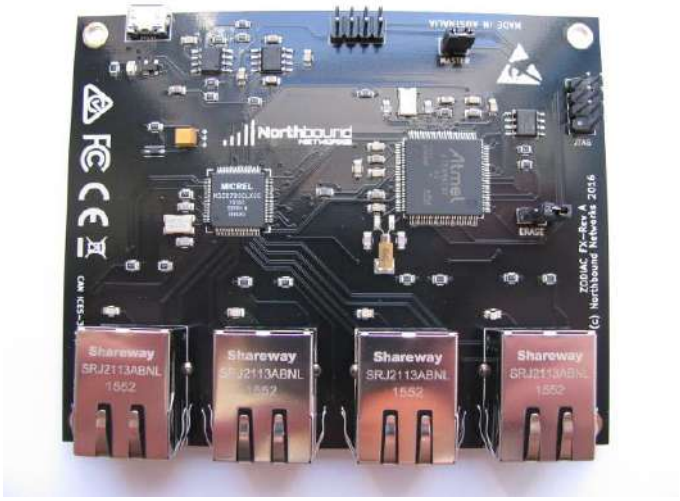


Figura 6-3: Conmutador Zodiac Fx
Fuente: http://petanode.com/blog/posts/zodiac-fx-review.html

Switch HPE 3800 Series

Para la implementación de los escenarios propuestos, se adquirió un Switch HP que trabaja con protocolo OpenFlow, esto permitió realizar las pruebas para las redes SDN, así como también ayudó para el ambiente tradicional.

Figura 7-3: Conmutador Hp 3800
Fuente: https://www.nautilusnet.com/products/j9584a-hpe-3800-24sfp-2sfp-switch-managed-l3-flexcampus-l3-new.html

3.14 Implementación

3.14.1 Pasos previos

Como medio de adiestramiento práctico con la tecnología SDN se hizo uso de Mininet, herramienta orientada al aprendizaje en un ambiente simulado. Esta fue el medio mediante el cual se realizó la construcción de diversas topologías de prueba y posteriormente implementar los

escenarios reales. Ya que para para mitigar el impacto que generaba el tiempo de adquisición de los equipos físico de prueba.

Una de las primeras pruebas realizadas fue la creación de topología básica en Mininet como se observa en la Figura 3-8.

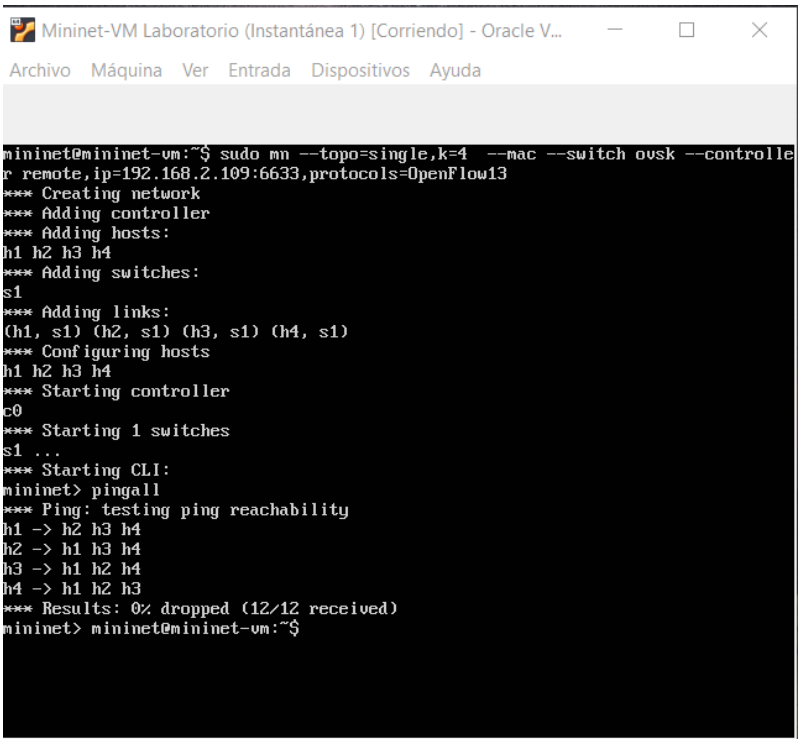


Figura 8-3: Creación de escenario inicial en Mininet
Realizado por: Marcelo Criollo, 2019

A partir de la creación de los escenarios de prueba se consigue conectar este escenario en Mininet con varios controladores.

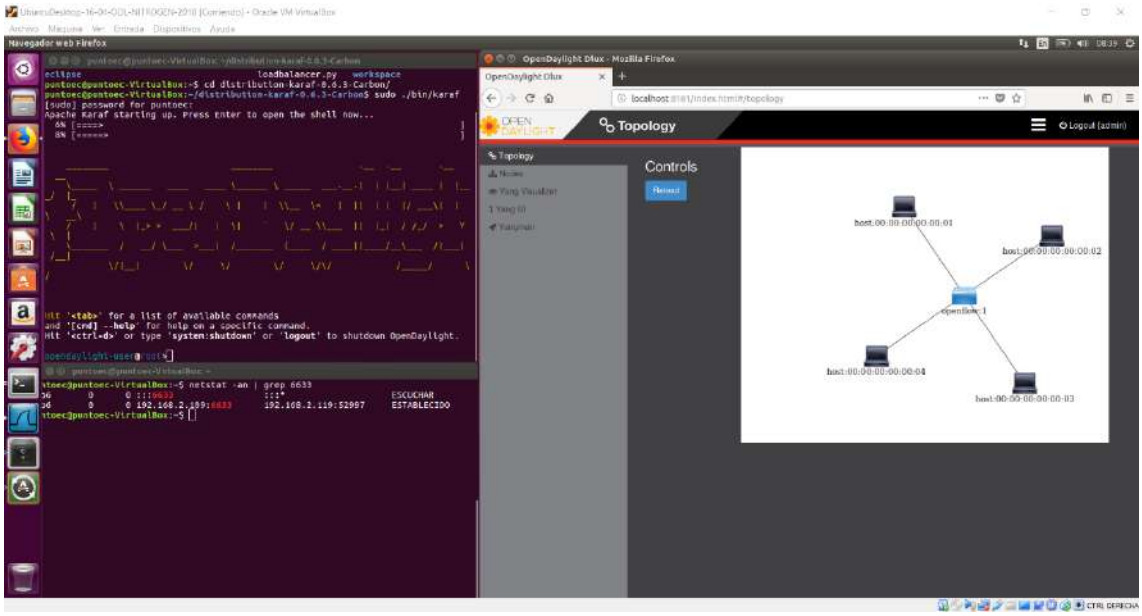


Figura 9-3: Ejecución escenario Mininet y OpenDayLigth
Realizado por: Marcelo Criollo, 2019

Como se observa en la Figura 9-3 ya se realiza la conexión entre Mininet y el controlador OpenDayLigth. En consecución se diversifico las subsecuentes pruebas con otros controladores siendo el Aruba VAN SDN Controller.

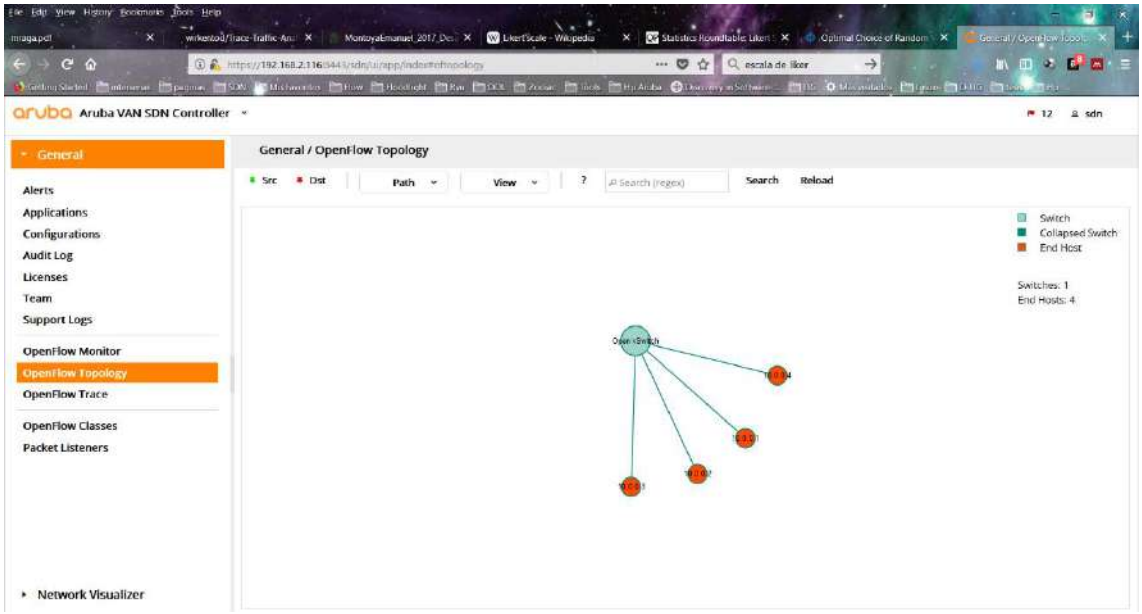


Figura 10-3: Ejecución escenario Mininet y Aruba VAN SDN Controller
Realizado por: Marcelo Criollo, 2019

De la manera como se ilustra en la Figura 10-3 se realiza la conexión de Mininet y el controlador Aruba VAN SDN Controller. Todas estas actividades nos ayudaron a tener un mayor y mejor concepción de lo que es la tecnología SDN y el protocolo OpenFlow

3.14.2 Diseño de escenario de pruebas SDN con Zodiac FX

En la realización de las pruebas respectivas como se muestra en la Figura 4-6 se visualiza la separación del plano de datos del plano de control. Donde el plano de control está compuesto por: el controlador (FloodLigth) con la dirección ip 192.168.2.106/24 y los switchs SDN Zodiac con el direccionamiento respectivo de 192.162.2.201, 192.168.2.202.

El plano de datos está conformado por los equipos servidor y cliente de pruebas a los que se les asigna las direcciones 192.168.0.5, 192.168.0.15 respectivamente.

Se debe enfatizar que los planos de datos y de control están totalmente separados de una manera lógica.

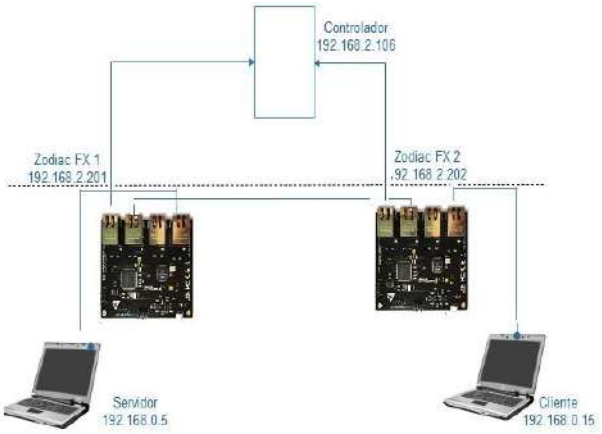


Figura 11-3: Escenario de pruebas SDN con Zodiac.
Realizado por: Marcelo Criollo, 2019

Una vez que la topología respectiva está instalada es posible realizar pruebas estándar como: ping entre los equipos de pruebas (cliente, servidor), además verificar los flujos instalados en cada una de los switchs; en las Figuras 13-3, 14-3 se observa los flujos instalados en los switchs Zodiac. Adicionalmente se observa la topología que genera el controlado FloodLigth.

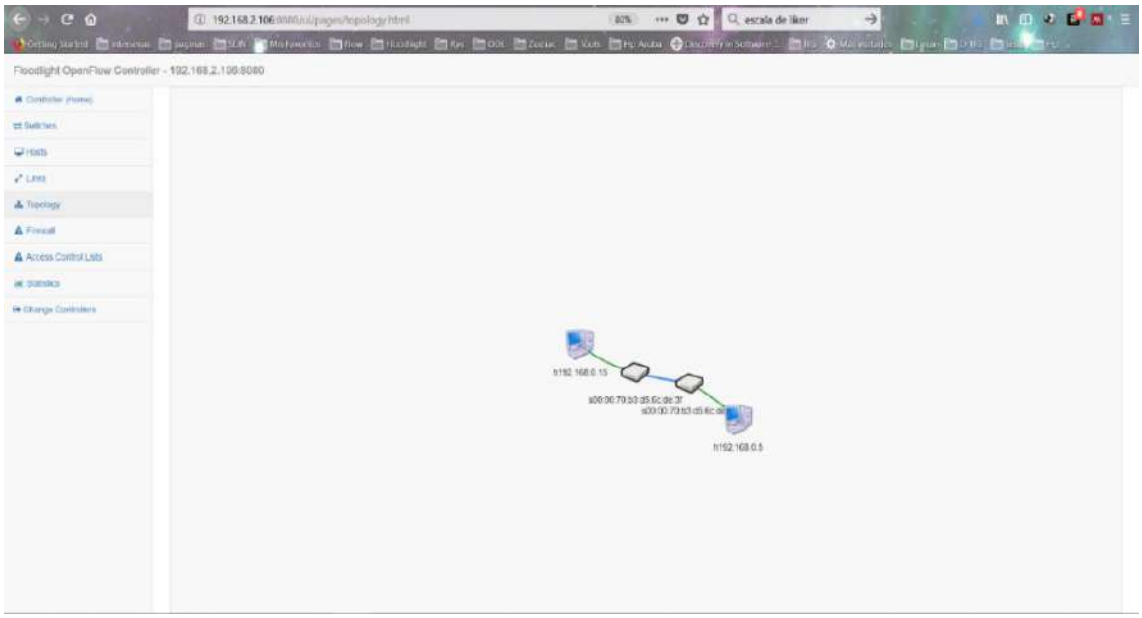


Figura 12-3: Escenario generado por FloodLigth SDN con Zodiac.
Realizado por: Marcelo Criollo, 2019

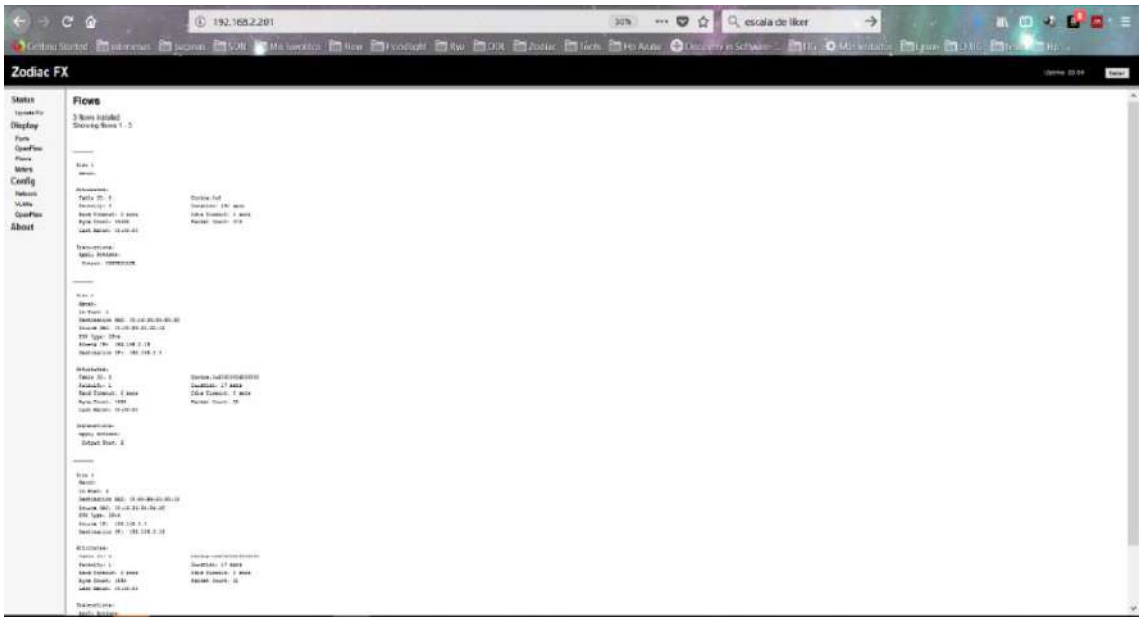


Figura 13-3: Flujos instalados en switch Zodiac - SDN con Zodiac
Realizado por: Marcelo Criollo, 2019

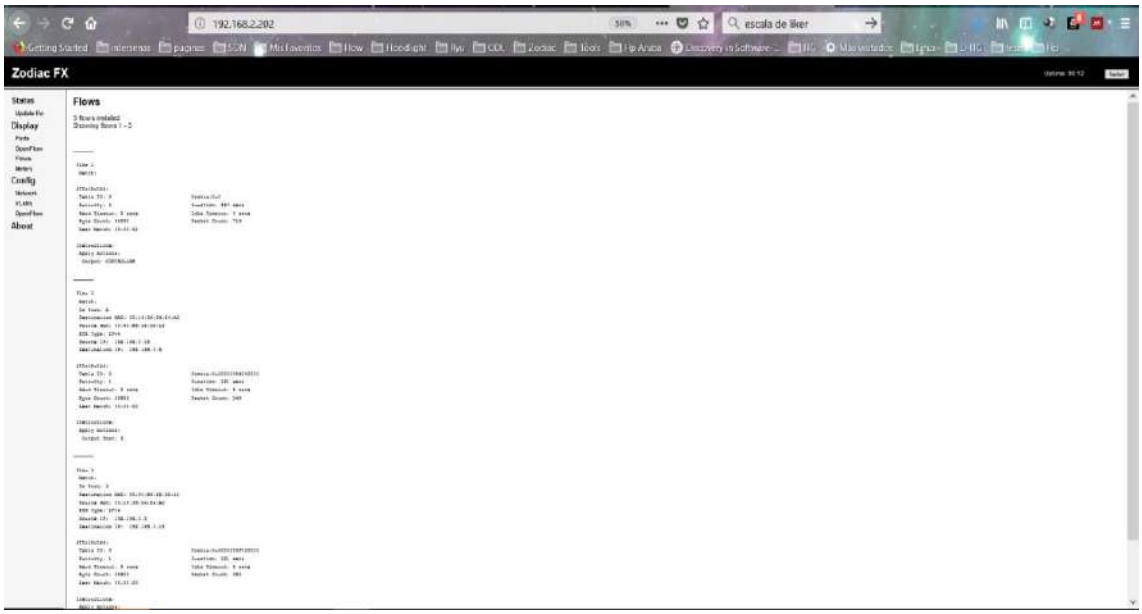


Figura 14-3: Flujos instalados en switch Zodiac - SDN con Zodiac
Realizado por: Marcelo Criollo, 2019

Verificado la conexión exitosa entre los equipos se procede a ingresar las reglas de QoS seleccionadas mediante Curl. Las reglas de firewall que se ingresaron fueron las siguientes.

```
Reglas de firewall

curl -X POST -d '{"switchid": "00:00:70:b3:d5:6c:de:48"}' http://192.168.2.106:8080/wm/firewall/rules/json
curl -X POST -d '{"src-ip":"192.168.0.5","dst-ip":"192.168.0.15","priority":"99"}' http://192.168.2.106:8080/wm/firewall/rules/json

Reglas de acl

curl -X POST -d '{"src-ip":"192.168.0.0/24","dst-ip":"192.168.0.0/24","action":"allow"}'  http://192.168.2.106:8080/wm/acl/rules/json
```

En las Figuras 15-3, 16-3 se observan tanto las reglas de firewall como las ACLs ingresadas en el controlador FloodLigth.

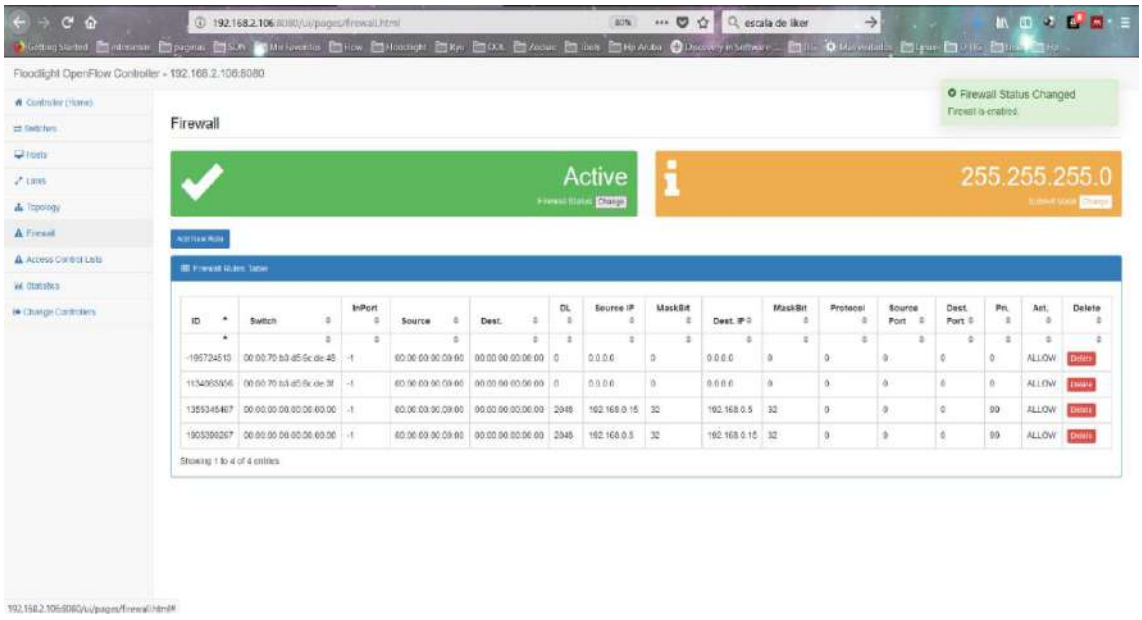


Figura 15-3: Reglas Firewall ingresadas en FloodLight SDN con Zodiac.
Realizado por: Marcelo Criollo, 2019

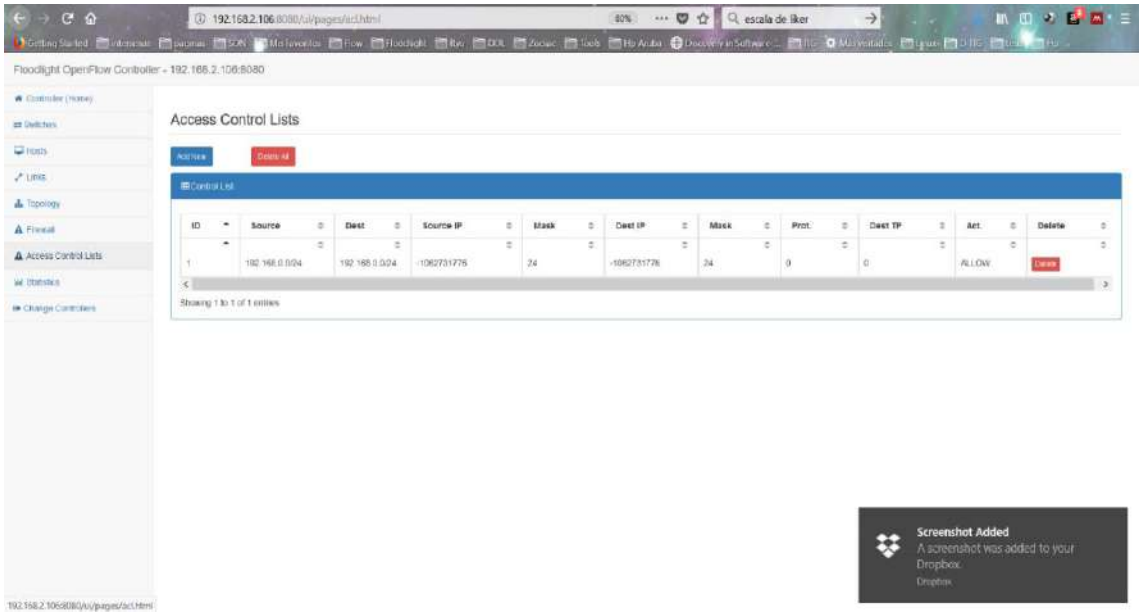


Figura 16-3: ACL ingresada en FloodLigth SDN con Zodiac
Realizado por: Marcelo Criollo, 2019

Una vez ingresadas las políticas de QoS, mediante las reglas firewall y Acl's se procede con la inyección de tráfico Tcp/Udp mediante D-ITG, la misma que se ejecutan tanto como en el servidor como en el cliente según las cargas necesarias. En el lado del servidor se crean los flujos a inyectar, ingresándose la dirección ip del cliente, finamente ejecutando los comandos Loger y Sender como se lo puede visualizar en la Figura 17-3 y 18-3.

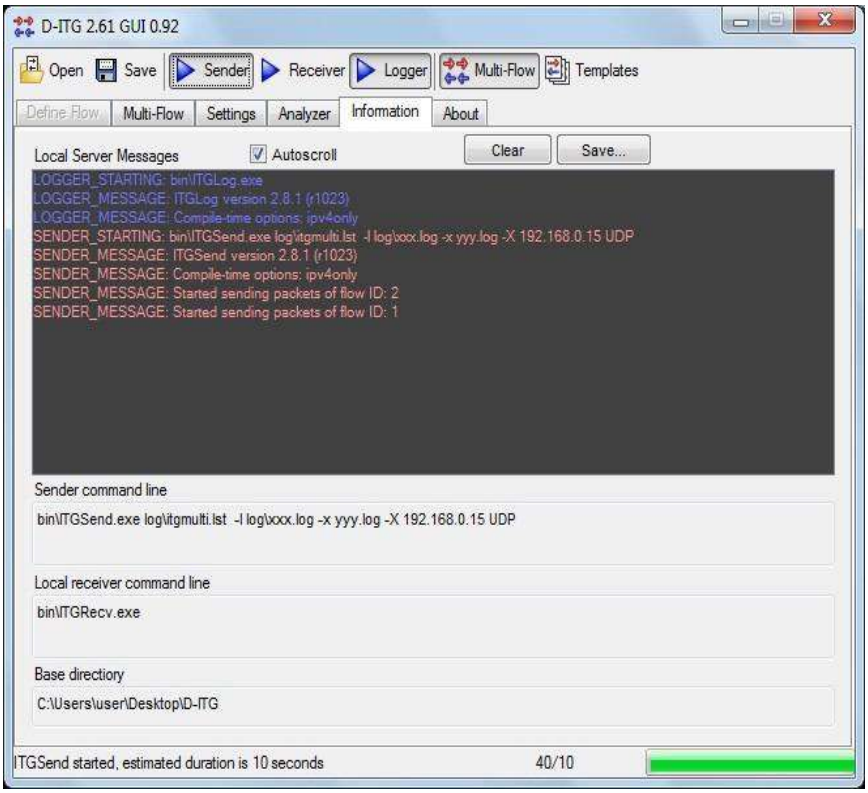


Figura 17-3: Inyección de tráfico desde el servidor SDN con Zodiac.
Realizado por: Marcelo Criollo, 2019

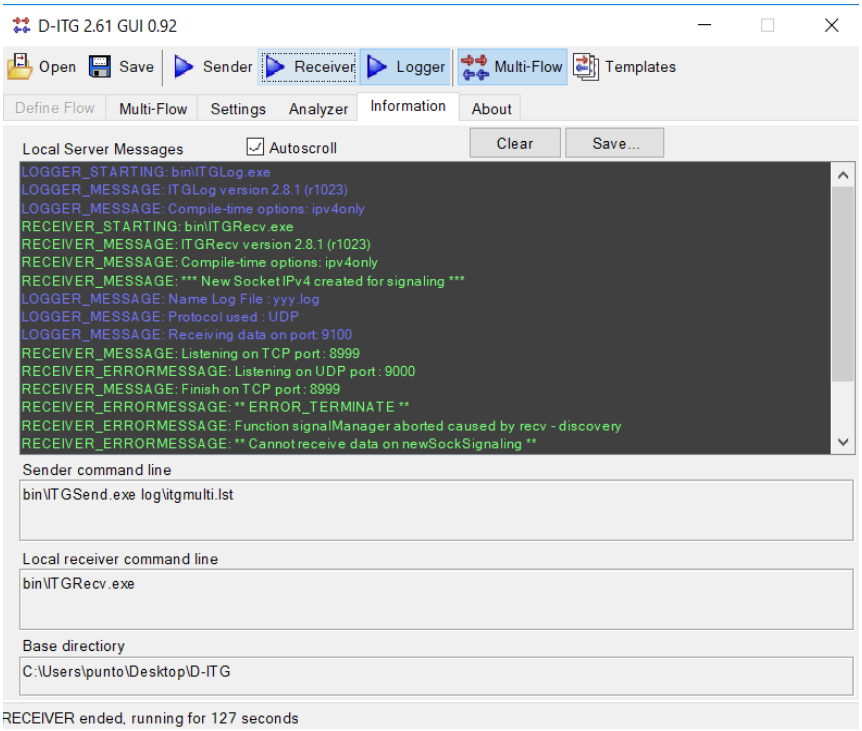


Figura 18-3: Recepción de tráfico desde el cliente SDN con Zodiac.
Realizado por: Marcelo Criollo, 2019

Terminada la inyección de datos se procede con la recuperación de los datos generados.

```
ITGDec version 2.8.1 (r1023)
Compile-time options: ipv4only
------------------------------------------------------------
Flow number: 1
From 192.168.0.5:56571
To    192.168.0.15:8999
------------------------------------------------------------
Total time                =      8.808272 s
Total packets             =         35598
Minimum delay             =      0.000037 s
Maximum delay             =      1.000752 s
Average delay             =      0.003565 s
Average jitter            =      0.000574 s
Delay standard deviation  =      0.058441 s
Bytes received            =      18226176
Average bitrate           =  16553.690440 Kbit/s
Average packet rate       =   4041.428330 pkt/s
Packets dropped           =         70095 (66.32 %)
Average loss-burst size   =      3.856671 pkt
------------------------------------------------------------
------------------------------------------------------------
Flow number: 2
From 192.168.0.5:49282
To    192.168.0.15:9002
------------------------------------------------------------
Total time                =     12.076609 s
Total packets             =            66
Minimum delay             =      0.000024 s
Maximum delay             =     12.063046 s
Average delay             =      8.261827 s
Average jitter            =      0.231501 s
Delay standard deviation  =      4.193556 s
Bytes received            =         33792
Average bitrate           =     22.385092 Kbit/s
Average packet rate       =      5.465110 pkt/s
Packets dropped           =             0 (0.00 %)
Average loss-burst size   =      0.000000 pkt
------------------------------------------------------------
```

Figura 19-3: Captura datos con D-ITG SDN con Zodiac.
Realizado por: Marcelo Criollo, 2019

3.14.3 Diseño de escenario de pruebas SDN con HP

En la realización de las pruebas respectivas como se muestra en la Figura 4-14 se visualiza la separación del plano de datos del plano de control. Donde el plano de control está compuesto por: el controlador (FloodLigth) con la dirección ip 192.168.2.106/24 y los switchs SDN Zodiac con el direccionamiento respectivo de 192.162.2.201, 192.168.2.202, además del switch Hp 3800 con la dirección 192.168.2.200.

El plano de datos está conformado por los equipos servidor y cliente de pruebas a los que se les asigna las direcciones 192.168.0.5, 192.168.0.15 respectivamente. Se debe enfatizar que los planos de datos y de control están totalmente separados de una manera lógica.

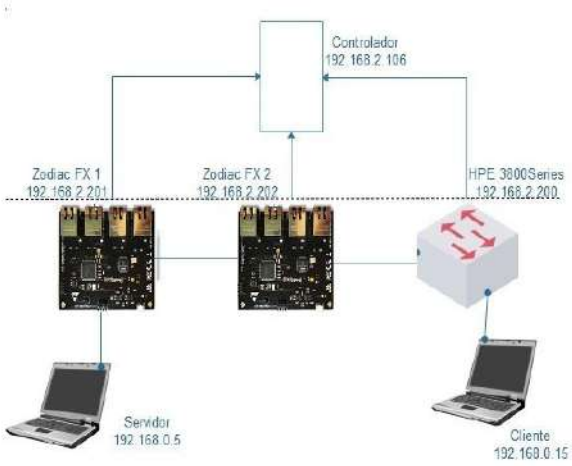


Figura 20-3: Escenario de pruebas SDN con Hp.
Realizado por: Marcelo Criollo, 2019

Una vez que la topología respectiva está instalada es posible realizar las pruebas estándar como: ping entre los equipos de pruebas (explicado anteriormente). Se observa la topología que genera el controlado FloodLigth.

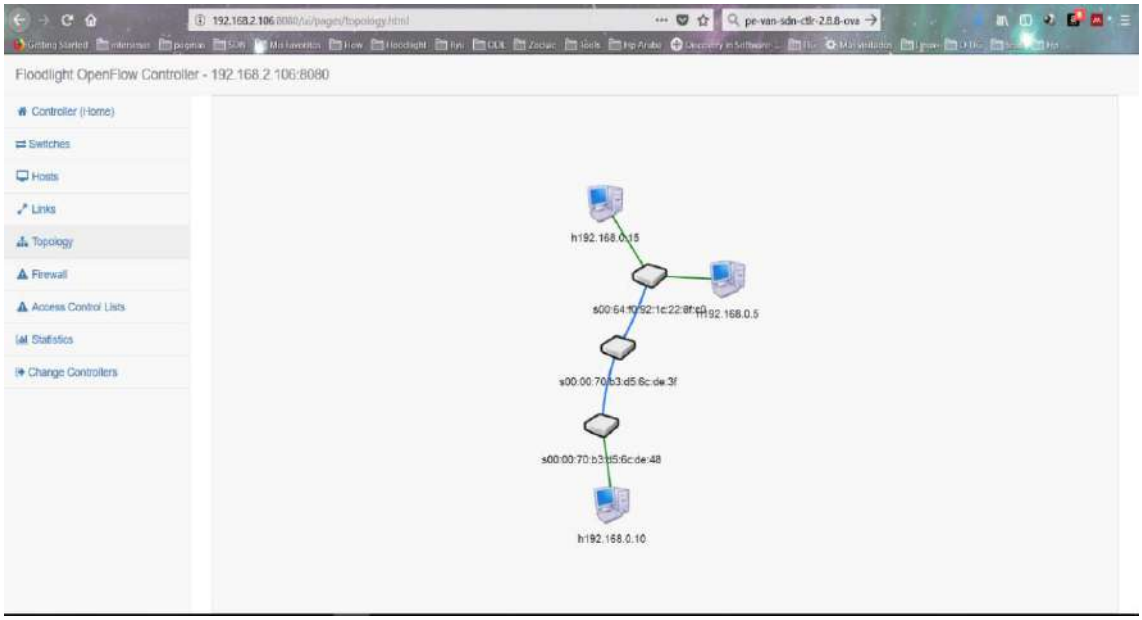


Figura 21-3: Escenario generado por FloodLigth SDN con Hp
Realizado por: Marcelo Criollo, 2019

Verificado la conexión exitosa entre los equipos se procede a ingresar las reglas de QoS seleccionadas mediante Curl.

```
Reglas de firewall switch Hp

curl -X POST -d '{"switchid": "00:64:f0:92:1c:22:8f:c0"}'http://192.168.2.106:8080/wm/firewall/rules/json
curl -X POST -d '{"src-ip":"192.168.0.5","dst-ip":"192.168.0.15","priority":"99"}' http://192.168.2.106:8080/wm/firewall/rules/json
curl -X POST -d '{"src-ip":"192.168.0.15","dst-ip":"192.168.0.5","priority":"99"}' http://192.168.2.106:8080/wm/firewall/rules/json

Reglas de ACL switch Hp.

curl -X POST -d '{"src-ip":"192.168.0.0/24","dst-ip":"192.168.0.0/24","action":"allow"}'  http://192.168.2.106:8080/wm/acl/rules/json
```

En la Figura 22-3 se observan tanto las reglas de firewall como las ACLs en ingresadas en el controlador FloodLigth.

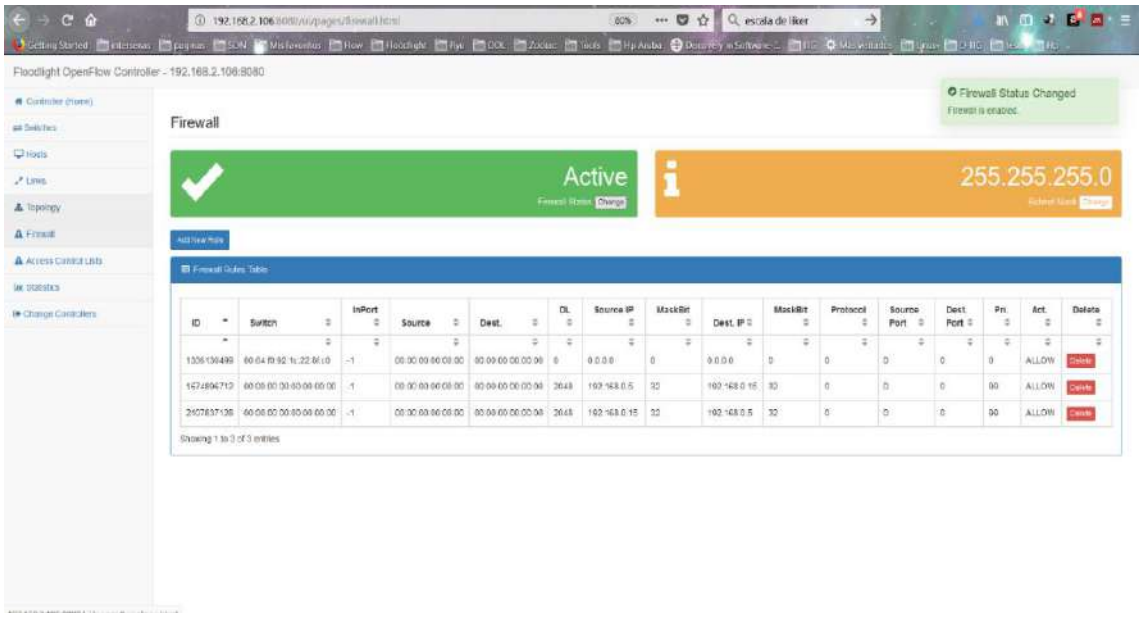


Figura 22-3: Reglas Firewall ingresadas en FloodLight SDN con Hp
Realizado por: Marcelo Criollo, 2019

Ingresadas las reglas anteriormente mencionadas se verifica que las mismas han sido ingresadas al switch Hp, en la cual se puede evidenciar que estas sean ACL o reglas de firewall son correspondientes a la categoría de QoS como se lo visualiza en la figura.

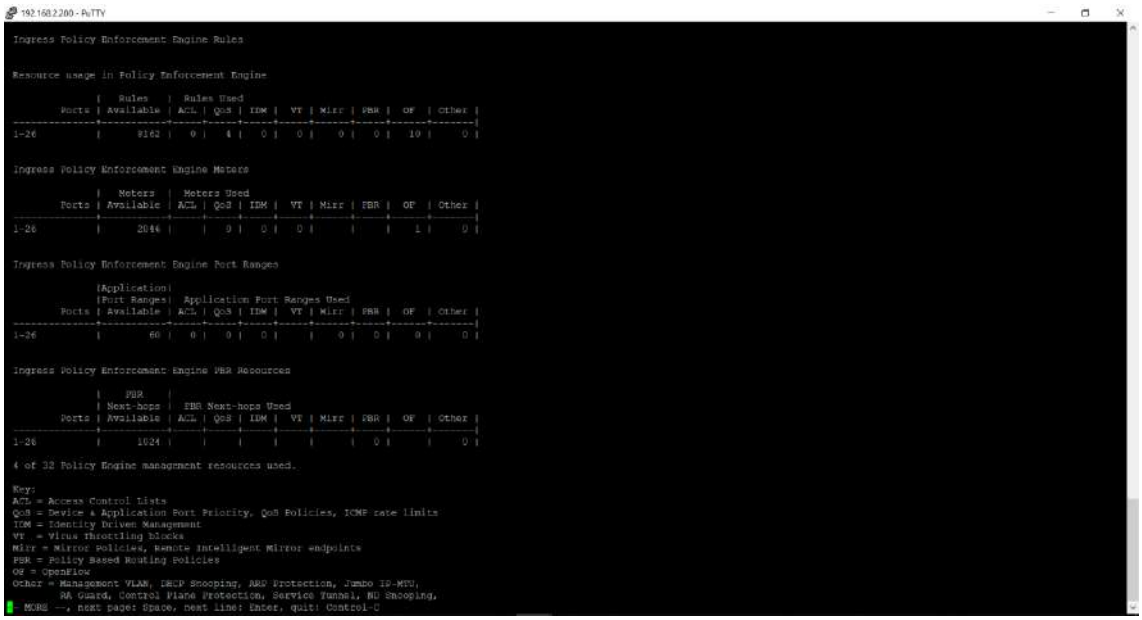


Figura 23-3: QoS instalado en Switch Hp con SDN
Realizado por: Marcelo Criollo, 2019

Una vez ingresadas las políticas de QoS, mediante las reglas firewall y Acls se procede con la inyección de tráfico TPC/UDP mediante D-ITG, la misma que se ejecutan tanto como en el servidor como en el cliente según las cargas necesarias. En el lado del servidor se crean los flujos a inyectar, ingresándose la dirección ip del cliente, finamente ejecutando los comandos Logger y Sender.

3.14.4 Diseño de escenario de pruebas convencional con HP

En la realización de las pruebas respectivas como se muestra en la Figura 4-19 se observa que según el paradigma tradicional se encuentra integrado en el hardware lo que hemos descrito como plano de datos y plano de control.

El plano de datos está conformado por los equipos servidor y cliente de pruebas a los que se les asigna las direcciones 192.168.0.5, 192.168.0.15 respectivamente. Se debe enfatizar que los planos de datos y de control están totalmente integrados de manera física.

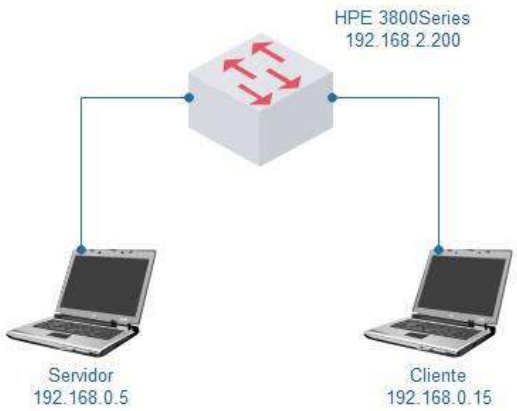


Figura 24-3: Escenario de pruebas convencional con Hp.
Realizado por: Marcelo Criollo, 2019

Se realizará el ingreso de las reglas de firewall y ACL de similar manera como en el l escenarios anteriores de SDN con Zodiac y SDN con HP.

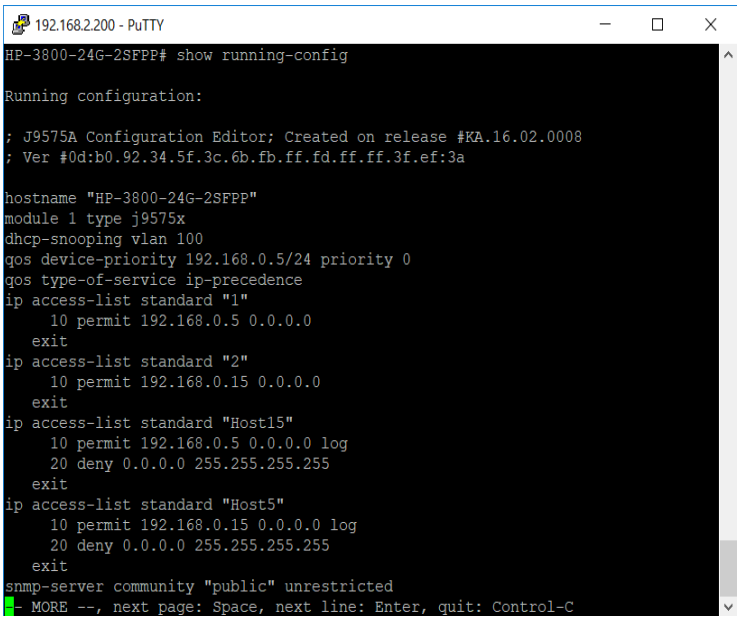


Figura 25-3: Reglas ingresadas a Switch Hp.
Realizado por: Marcelo Criollo, 2019

Posterior a lo cual se realizó la inyección de tráfico como de similar forma que en los apartados anteriores.

3.15 Costos de implementación

Para la realización del presente proyecto junto con los escenarios propuestos, se necesitó adquirir equipos físicos, los que facilitaron el desarrollo del mismo; por tal motivo se presenta en la siguiente tabla los costos requeridos:

Tabla 9-3: Costos de implementación

CANTIDAD	DESCRIPCIÓN	Costo Unitario	Costo Total
1	SWITCH HPE 3800 Series	2.700,00	2.700,00
2	Módulos Ethernet Zodiac FX	120,00	240,00
2	Equipos terminales (Portátiles)	650,00	1.300,00
1	Equipo para virtualización (Controlador SDN)	875,00	875,00
1	Switch para el Plano de Control HP Procurve 1810G	280,00	280,00
2	Adaptadores USB LAN	10,00	20,00
8	Patch Cord cat 5e de 3 metros certificados	2.40	19,20
	TOTAL:		**$ 5.434,20**

Realizado por: Marcelo Criollo, 2019

CAPÍTULO IV

4 RESULTADOS Y DISCUCIÓN

Esta investigación está enfocada en estudiar las políticas de QoS en redes empresariales para el análisis de rendimiento en entornos tradicionales y SDN, en el cual para su implantación se utilizó un controlador SDN con software libre y con equipos reales los cuales soportan el protocolo OpenFlow. Se plantearon dos escenarios idénticos el uno con el uso de SDN y el otro con enfoque de redes tradicionales, realizándose las pruebas respectivas en cada uno de estos.

El rendimiento será analizado en los diferentes escenarios mediante la toma de 30 muestras, tomadas del flujo de datos en tiempo real de cada uno de los indicadores planteados, los mismos que serán ponderados, analizados y comparados a través de datos, cuantificando su rendimiento lo que permitirá demostrar la hipótesis planteada. Midiendo los efectos que la variable independiente produce sobre la variable dependiente, se procederá a verificar el rendimiento bajo los siguientes parámetros:

I1.- Latencia: Tiempo promedio de retardos en la red.
I2.- Jitter: Diferencia del tiempo en la transmisión de paquetes.
I3.- Ancho de banda: Cantidad de información que se puede enviar.
I4.- Pérdida de paquetes: Cantidad de paquetes descartados.

4.1 Muestras, análisis escenarios

Para realizar el análisis se utiliza el método estadístico T-Student; aplicado en primera instancia en dos escenarios con tecnología SDN en diferente hardware, con el fin de determinar cual tiene mejor respuesta en rendimiento; para luego compararlo con el escenario tradicional. El análisis de los escenarios será entre clientes y servidores.

Para verificar si existe diferencia significativa entre las dos medias se debe analizar el p-valor (Sig.) de la prueba T; en donde se asume que el valor de significancia es 0.05.

Para corroborar estos análisis se utilizó además el proceso de cinco pasos descrito por (Lind, 2012)

1. Establecer las hipótesis nula y alternativa
2. Seleccionar el nivel de significancia

3. Calcular el estadístico de prueba (en este caso el valor T)
4. Establecer una regla de decisión; en este caso dependiendo del valor crítico se rechazará o aceptará la hipótesis nula
5. Tomar una decisión.

Con base en esto, las hipótesis nula y alternativa planteadas son:

$$Hipótesis\ Nula\ H0: \overline{X_1} = \overline{X_1}$$
$$Hipótesis\ Alternativa\ H1: \overline{X_1} \neq \overline{X_1}$$

Dónde:

- $\overline{X_1}$ es la media de la primera muestra
- $\overline{X_2}$ es la media de la segunda muestra

Este tipo de hipótesis se debe comprobar a través de un aprueba de dos colas. El nivel de significancia seleccionado fue de 0.05 que corresponde a un nivel de confiabilidad del 95%. Por otro lado para el cálculo del estadístico de prueba se utilizó la fórmula planteada por (Lind, 2012) para muestras con varianzas desiguales.

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Dónde:

- $\overline{X_1}$ es la media de la primera muestra
- $\overline{X_2}$ es la media de la segunda muestra
- s_1^2 es la varianza de la primera muestra
- s_1^2 es la varianza de la segunda muestra
- n_1 tamaño de la primera muestra
- n_2 tamaño de la segunda muestra

Si las varainzas son iguales se utilizará la siguiente fórmula:

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{S_p^2(\frac{1}{n_1} + \frac{2}{n_2})}}$$

Para plantear la regla de decisión es necesario establecer un valor crítico que generalmente se obtiene de la Tabla de Probabilidad T-Student. Para hallar este valor crítico es necesario obtener los grados de libertad los cuales se obtienen aplicando la siguiente fórmula para muestras con varianzas desiguales.

$$gl = \frac{\left[\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right]^2}{\frac{(\frac{s_1^2}{n_1})^2}{n_1 - 1} + \frac{(\frac{s_2^2}{n_2})^2}{n_2 - 1}}$$

En el caso de que las varianzas sean iguales los grados de libertad son iguales al número total de elementos muestreados menos el número de muestras. La regla de decisión para aceptar o rechazar la hipótesis nula es: si el estadístico de prueba t, es mayor o menor que el valor crítico se rechazará o se aceptará la hipótesis nula H0.

4.1.1 Indicador Latencia

Para el análisis de latencia se tomaron un total de 30 muestras.

Tabla 1-4: Indicador Latencia

Indicador de latencia						
Número de muestras	Con SDN Zodiac		Con SDN Hp 3800		Convencional Hp 3800	
	Servidor	Cliente	Servidor	Cliente	Servidor	Cliente
1	0,080839	5,004491	0,006534	4,076261	0,004113	4,115534
2	0,199508	5,037720	0,012435	4,100288	0,003302	4,114228
3	0,106652	5,020532	0,011870	4,111754	0,003182	4,110148
4	0,113927	5,022136	0,008999	4,108514	0,002975	4,104375
5	0,029118	5,005005	0,014894	4,120615	0,003351	4,103788
6	0,044949	5,031259	0,007613	4,120742	0,003209	4,108287
7	0,025093	5,034275	0,006705	4,121475	0,003102	4,090407
8	0,018356	5,039506	0,002897	4,118607	0,003113	4,087929
9	0,004002	5,060113	0,002904	4,128180	0,003051	4,091584
10	0,050808	5,071101	0,005951	4,134828	0,003158	4,081353
11	0,086113	5,122211	0,002907	4,137911	0,003201	4,081041
12	0,008404	5,095412	0,005528	4,147832	0,003315	4,084085
13	0,041264	5,148783	0,043728	4,144106	0,003817	4,074153
14	0,059031	5,924350	0,006279	4,146190	0,003440	4,077020
15	0,051645	5,126685	0,036428	4,151445	0,003050	4,058541
16	0,030581	5,133660	0,006162	4,162932	0,003584	4,075821
17	0,003351	5,138961	0,003826	4,162773	0,003823	4,070612
18	0.038158	5,152561	0,030933	4,175040	0,002901	4,061531
19	0,007201	5,159100	0,003886	4,172756	0,003716	4,058323
20	0,009439	5,158988	0,004877	4,170450	0,003921	4,068801
21	0,040978	5,175790	0,005672	4,174430	0,003706	4,035477
22	0,07138	5,182616	0,004609	4,183196	0,003971	4,029128
23	0,028926	4,964495	0,008090	4,193870	0,003705	4,017333
24	0,017921	4,941627	0,006678	4,181672	0,003896	4,030524
25	0,016218	4,937142	0,182435	4,195710	0,004239	4,025313
26	0,044726	4,935418	0,004303	4,193695	0,003614	4,024901
27	0,005033	4,927545	0,004641	4,192158	0,003805	4,015602
28	0,006938	4,912559	0,005038	4,191285	0,003670	4,011203
29	0,018795	4,915941	0,004323	4,201961	0,003564	4,014516
30	0,003205	3,915807	0,005431	4,200346	0,003938	4,002269

Realizado por: Marcelo Criollo, 2019

4.1.1.1 Análisis de Latencia en Servidor, SDN con Hp vs. SDN con Zodiac

Tabla 2-4: Estadísticas de grupo, SDN con Hp y SDN con Zodiac.

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Latencia - Servidor	SDN con Hp	30	,01521920	,033100157	,006043234
	SDN con Zodiac	30	,04208620	,042550589	,007768639

Realizado por: Marcelo Criollo, 2019

Tabla 3-4: Prueba de muestras independientes, SDN con Hp y SDN con Zodiac

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Latencia Servidor	Se asumen varianzas iguales	3,508	,066	-2,730	58	,008	-,026867000	,009842379
	No se asumen varianzas iguales			-2,730	54,690	,009	-,026867000	,009842379

Realizado por: Marcelo Criollo, 2019

La prueba F permite concluir que las varianzas de las dos muestras no son significativamente diferentes (p-valor es mayor que 0.05) con un valor de 0,066. Al analizar los resultados de la primera fila (Prueba T) se observa que el p-valor (0.008) es menor que 0.05, lo que lleva a rechazar la hipótesis de igualdad de medias permitiendo concluir que existe diferencia significativa entre las medias o promedios de SDN con Hp y SDN con Zodiac para la variable latencia en el servidor.

Dado que las varianzas son iguales, el valor t obtenido con la fórmula descrita en el apartado anterior es igual a -2.72 el cual es mayor al valor crítico de 2.002 que se obtuvo de la tabla de distribución t de dos colas con 58 grados de libertad. Estos resultados permiten rechazar la hipótesis nula de igualdad de medias corroborando el análisis realizado. La Figura 1-4 muestra la regla de decisión aplicada.

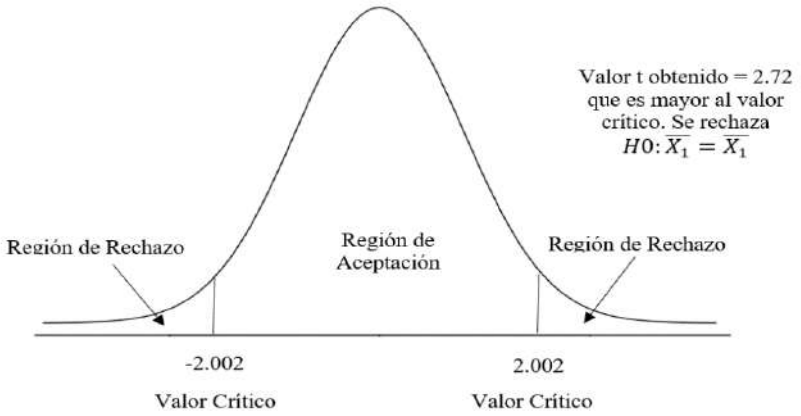


Figura 1-4: Regla decisión aplicada, SDN con Hp y SND con Zodiac
Realizado por: Marcelo Criollo, 2019

4.1.1.2 Análisis de Latencia en Servidor, SDN con Hp vs. Tradicional con Hp

Tabla 4-4: Estadísticas de grupo, SDN con Hp y Tradicional con Hp

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Latencia - Servidor	SDN con Hp	30	,01521920	,033100157	,006043234
	Tradicional con Hp	30	,00351440	,000370625	,000067667

Realizado por: Marcelo Criollo, 2019

Tabla 4-5: Prueba muestras independientes, SDN con Hp y Tradicional con Hp

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Latencia Servidor	Se asumen varianzas iguales	8,170	,006	1,937	58	,058	,011704800	,006043613
	No se asumen varianzas iguales			1,937	29,007	,063	,011704800	,006043613

Realizado por: Marcelo Criollo, 2019

La prueba F permite concluir que las varianzas de las dos muestras son significativamente diferentes (p-valor es menor que 0.05). Al analizar los resultados de la segunda fila (Prueba T) se observa que el p-valor (0.063) es mayor que 0.05 lo que lleva a aceptar la hipótesis de igualdad de medias permitiendo concluir que no existe diferencia significativa entre las medias o promedios de SDN Hp y Tradicional Hp para la variable latencia en el servidor.

Dado que en este caso las varianzas son desiguales, el valor t obtenido con la fórmula descrita en el apartado anterior es igual a 1.93 el cual es menor al valor crítico de 2.045 que se obtuvo de la

tabla de distribución t de dos colas con 29 grados de libertad. Estos resultados permiten aceptar la hipótesis nula de igualdad de medias corroborando el análisis realizado. La Figura 2-4 muestra la regla de decisión aplicada:

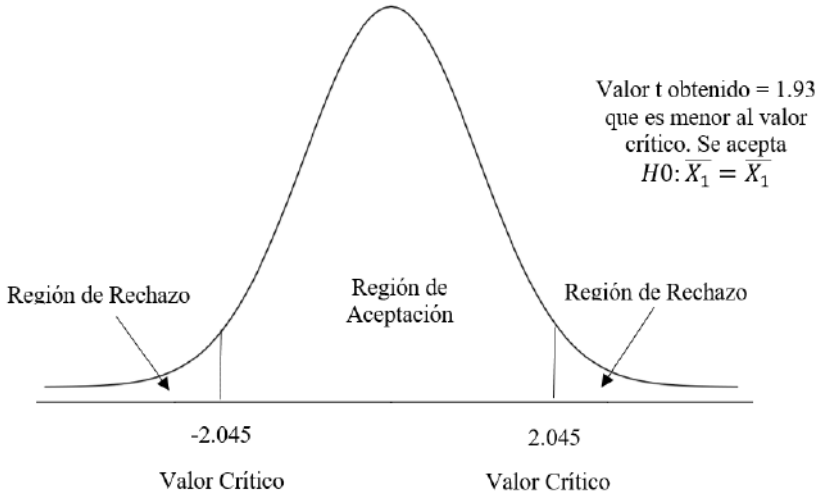


Figura 2-4: Regla decisión aplicada, SDN con Hp y Tradicional con Hp.
Realizado por: Marcelo Criollo, 2019

4.1.1.3 Resumen Latencia en Servidor.

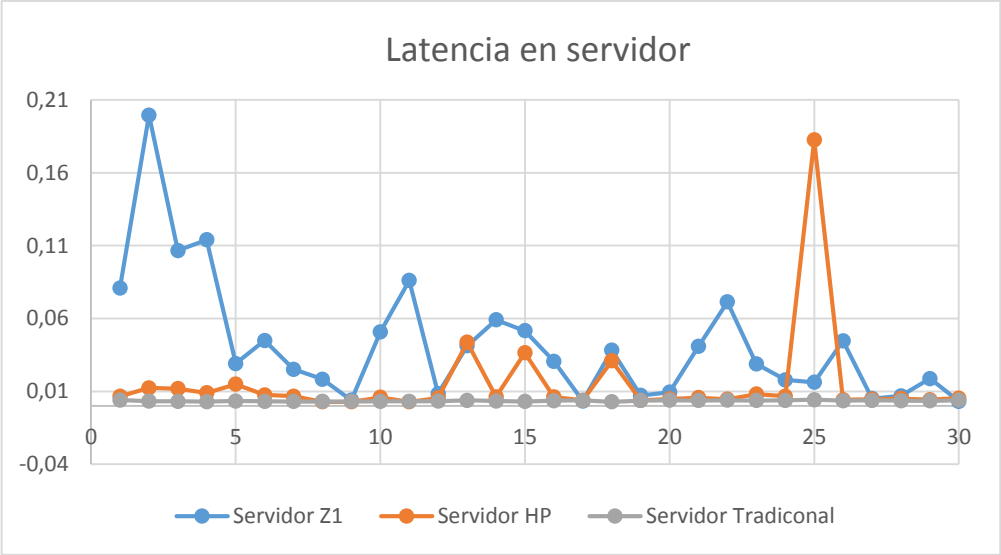


Figura 3-4: Latencia en el servidor**.**
Realizado por: Marcelo Criollo, 2019

En la Figura 3-4 se detallan los valores obtenidos de las pruebas de Latencia en el Servidor de entre los equipos: SDN con Zodiac, SDN con Hp y Tradicional con Hp. En la que se observan las fluctuaciones que presentan las mediciones realizadas en los escenarios propuestos, llegándose a la conclusión que el mejor rendimiento de Latencia en el Servidor se obtiene del equipo

Tradicional con Hp. Es importante destacar que en Latencia el valor entre más pequeño sea, es indicativo de mejor rendimiento.

4.1.1.4 Análisis de Latencia en Cliente, SDN con Hp vs. SDN con Zodiac

Tabla 6-4: Estadísticas de grupos, SDN con Hp y SDN con Zodiac

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Latencia – Cliente	SDN con Hp	30	4,15403407	,034374479	,006275893
	SDN con Zodiac.	30	5,04319297	,279096836	,050955878

Realizado por: Marcelo Criollo, 2019

Tabla 7-4: Prueba de muestras independientes, SDN con Hp y SDN con Zodiac.

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Latencia Cliente	Se asumen varianzas iguales	5,819	,019	-17,319	58	,000	-,889158900	,051340903
	No se asumen varianzas iguales			-17,319	29,880	,000	-,889158900	,051340903

Realizado por: Marcelo Criollo, 2019

En la Tabla 6-4 se muestra el promedio de latencia entre los dos equipos (SDN con HP y SDN con Zodiac). En la Tabla 7-4 la prueba estadística revela que las varianzas de las dos muestras no son iguales ya que el p-valor (sig. = 0.019) es menor que el valor de significancia que se asume igual a 0.05 lo cual permite rechazar la hipótesis nula de igualdad de varianzas entre las dos muestras.

En este caso vemos que el p-valor es igual a 0.000 menor que 0.05 por lo que se rechaza la hipótesis nula y se puede concluir que si existe diferencia significancia entre las dos medias. Al analizar los valores de los promedios se puede corroborar que la latencia del equipo SDN con Hp es menor que la latencia del equipo SDN con Zodiac por lo que la latencia es menor en SDN con Hp al comparar ambas tecnologías.

En este análisis se comprobó que las varianzas son desiguales, por lo que el valor t obtenido con la fórmula descrita en el apartado anterior es igual a -17.32 el cual es menor al valor crítico de -2.042 que se obtuvo de la tabla de distribución t de dos colas con 30 grados de libertad. Estos

resultados permiten rechazar la hipótesis nula de igualdad de medias corroborando el análisis realizado. La figura 4-5 muestra la regla de decisión aplicada.

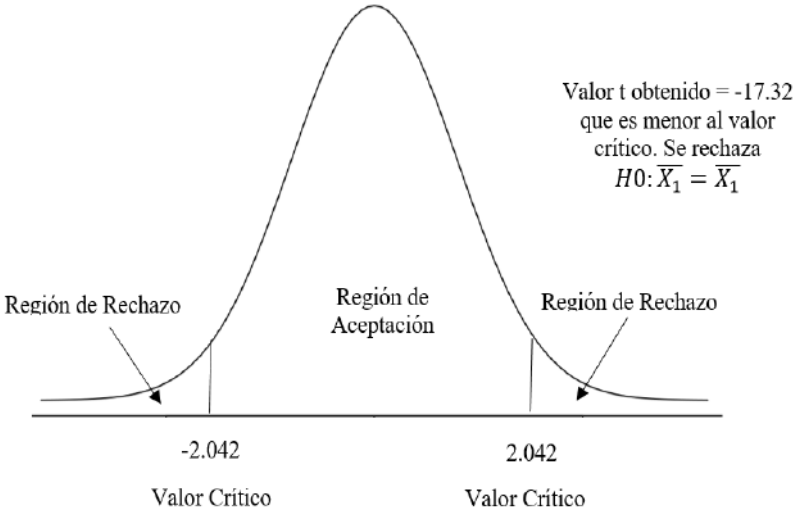


Figura 4-4: Regla decisión aplicada, SDN con Hp y SDN con Zodiac.
Realizado por: Marcelo Criollo, 2019

4.1.1.5 Análisis de Latencia en Cliente, SDN con Hp vs. Tradicional con Hp

Tabla 8-4: Estadística de grupo, SDN con HP y Tradicional con Hp

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Latencia - Clientes	SDN con Hp	30	4,15403407	,034374479	,006275893
	Tradicional con Hp	30	4,06412757	,035134246	,006414606

Realizado por: Marcelo Criollo, 2019

Tabla 9-4: Prueba muestras independientes, SDN con Hp y Tradicional con Hp.

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Latencia Cliente	Se asumen varianzas iguales	,012	,914	10,018	58	,000	,089906500	,008974074
	No se asumen varianzas iguales			10,018	57,972	,000	,089906500	,008974074

Realizado por: Marcelo Criollo, 2019

De forma similar al caso anterior, se asumen varianzas desiguales ya que el p-valor de la prueba F (0.012) es menor que 0.05 permitiendo rechazar la hipótesis de igualdad de varianzas. Al verificar la prueba T (primera fila) se observa que el p-valor (0.000) es menor que el valor de

significancia 0.05 lo que permite rechazar la hipótesis nula de igualdad de medias y permite concluir que la latencia del método Tradicional con Hp es significativamente diferente que la latencia de SDN con HP.

Dado que en este caso las varianzas son iguales, el valor t obtenido con la fórmula descrita en el apartado anterior es igual a 10.02 el cual es mayor al valor crítico de 2.002 que se obtuvo de la tabla de distribución t de dos colas con 58 grados de libertad. Estos resultados permiten rechazar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. La Figura 5-4 muestra la regla de decisión aplicada.

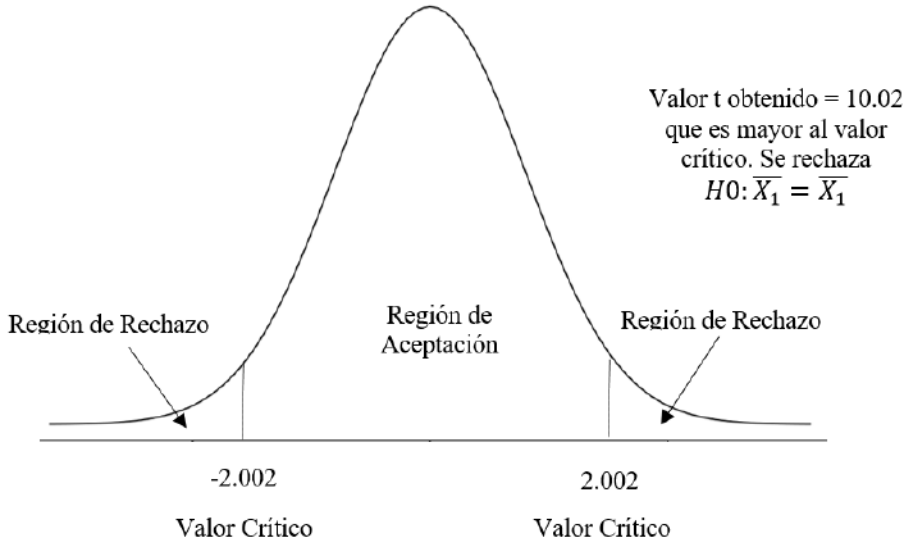


Figura 5-4: Regla decisión aplicada, SDN con Hp y Tradicional con Hp
Realizado por: Marcelo Criollo, 2019

4.1.1.6 Resumen Latencia en cliente

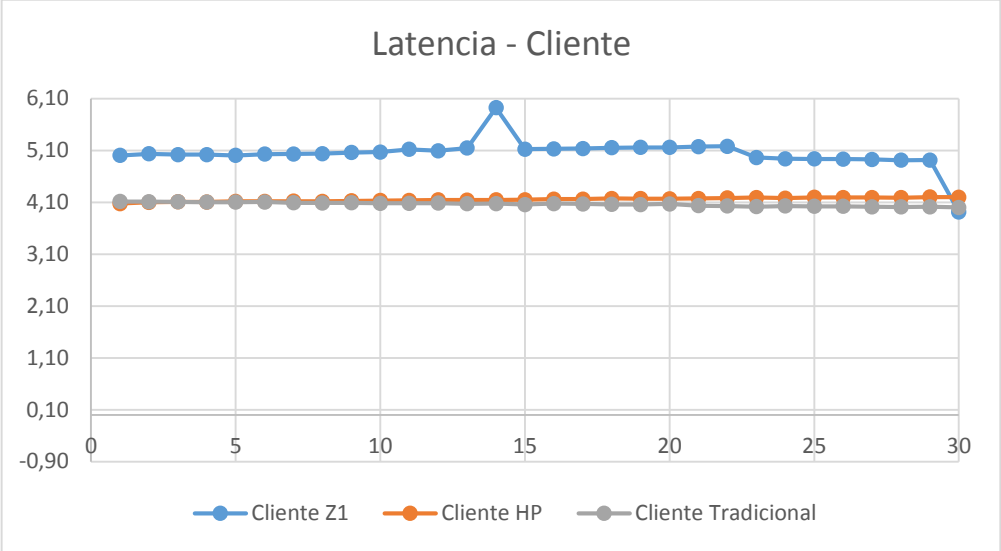


Figura 6-4: Latencia en cliente
Realizado por: Marcelo Criollo, 2019

En la Figura 6-4 se detallan los valores obtenidos de las pruebas de Latencia en el Cliente de entre los equipos: SDN con Zodiac, SDN con Hp y Tradicional con Hp. En la que se observan las fluctuaciones que presentan las mediciones realizadas en los escenarios propuestos, llegándose a la conclusión que el mejor rendimiento de Latencia en el Cliente se obtiene del equipo Tradicional con Hp. Es importante destacar que en Latencia el valor entre más pequeño sea, es indicativo de mejor rendimiento.

4.1.2 Indicador Jitter

Para el análisis de jitter se tomaron un total de 30 muestras.

Tabla 10-4: Indicadores de Jitter.

Indicador de jitter						
Número de muestras	Con SDN Zodiac		Con SDN H 3800		Convencional Hp 3800	
	Servidor	Cliente	Servidor	Cliente	Servidor	Cliente
1	0,005048	0,001519	0,000984	0,000773	0,003730	0,000909
2	0,006828	0,001980	0,003942	0,002328	0,002514	0,000651
3	0,005353	0,001604	0,002957	0,001684	0,002036	0,000543
4	0,006111	0,001870	0,002833	0,001878	0,001637	0,000558
5	0,003933	0,001161	0,002895	0,001374	0,001479	0,000836
6	0,002460	0,000782	0,001639	0,000677	0,001166	0,000562
7	0,002286	0,001015	0,001670	0,001168	0,001103	0,000520
8	0,002657	0,001222	0,000973	0,000349	0,000922	0,000409
9	0,001016	0,000266	0,000855	0,000254	0,000900	0,000539
10	0,003022	0,000985	0,001157	0,000924	0,000790	0,000674
11	0,002519	0,001619	0,000665	0,000408	0,000722	0,002963
12	0,001487	0,000652	0,001237	0,000683	0,000639	0,000388
13	0,001915	0,000638	0,001211	0,000644	0,000661	0,000335
14	0,00275	0,000826	0,000994	0,000687	0,000582	0,000297
15	0,002928	0,000818	0,001794	0,000251	0,000577	0,000313
16	0,001292	0,000797	0,000836	0,000547	0,000566	0,000327
17	0,000664	0,000408	0,000523	0,000281	0,000569	0,000281
18	0,038185	0,000755	0,001932	0,000840	0,000517	0,000437
19	0,000858	0,000477	0,000566	0,000388	0,000554	0,000239
20	0,001219	0,000592	0,000508	0,000228	0,000514	0,000309
21	0,001584	0,000895	0,000541	0,000288	0,000502	0,000348
22	0,002661	0,001523	0,000505	0,000231	0,000481	0,000242
23	0,001386	0,000808	0,000959	0,000747	0,000462	0,000258
24	0,001442	0,000518	0,000795	0,000662	0,000447	0,000365
25	0,001511	0,000745	0,002984	0,000998	0,000480	0,000338
26	0,001439	0,000679	0,000482	0,000264	0,000425	0,000205
27	0,005033	0,000244	0,000413	0,000203	0,000393	0,000302
28	0,001057	0,000427	0,000402	0,000197	0,000395	0,000213
29	0,001804	0,000798	0,000508	0,000382	0,000388	0,000305
30	0,000692	0,000259	0,000481	0,000216	0,000397	0,000311

Realizado por: Marcelo Criollo, 2019

4.1.2.1 Análisis de Jitter en Servidor, SDN con Hp vs. SDN con Zodiac

Tabla 11-4: Estadísticas de grupo, SDN con Hp y SDN con Zodiac

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Jitter – Servidor	SDN con HP.	30	,00127470	,000953951	,000174167
	SDN con Zodiac.	30	,00370467	,006719114	,001226737

Realizado por: Marcelo Criollo, 2019

Tabla 12-4: Prueba de muestras independientes, SDN con Hp y SDN con Zodiac

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Jitter Servidor	Se asumen varianzas iguales	4,088	,048	-1,961	58	,055	-,002429967	,001239039
	No se asumen varianzas iguales			-1,961	30,169	,059	-,002429967	,001239039

Realizado por: Marcelo Criollo, 2019

La Tabla 11-4 muestra el promedio de Jitter entre los dos equipos (SDN con HP y SDN con Zodiac). En la Tabla 12-4 la prueba estadística revela que las varianzas de las dos muestras no son iguales ya que el p-valor (sig. = 0.048) es menor que el valor de significancia que se asume igual a 0.05 lo cual permite rechazar la hipótesis nula de igualdad de varianzas entre las dos muestras.

Con base en esto se seleccionan únicamente los resultados correspondientes a la segunda fila. Para verificar si existe diferencia significativa entre las dos medias se debe analizar el p-valor (Sig.) de la prueba T. Si es que este valor es menor que el valor de significancia que se asume (0.05) se rechaza la hipótesis nula de igualdad entre las medias.

En este caso vemos que el p-valor es igual a 0.059 mayor que 0.05 por lo que se acepta la hipótesis nula y se puede concluir que no existe diferencia significativa entre las dos medias. Al analizar los valores de los promedios se puede corroborar que el Jitter del equipo SDN Hp es apenas menor que el Jitter del equipo SDN Zodiac por lo que no existe diferencia significativa en el Jitter al comparar ambos equipos SDN.

Las varianzas son desiguales, el valor t obtenido con la fórmula descrita anteriormente es igual a -1.96 el cual es mayor al valor crítico de -2.042, que se obtuvo de la tabla de distribución t de dos colas con 30 grados de libertad. Este valor T se ubica en la región de aceptación llevando a aceptar

la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. La Figura 7-4 muestra la regla de decisión aplicada.

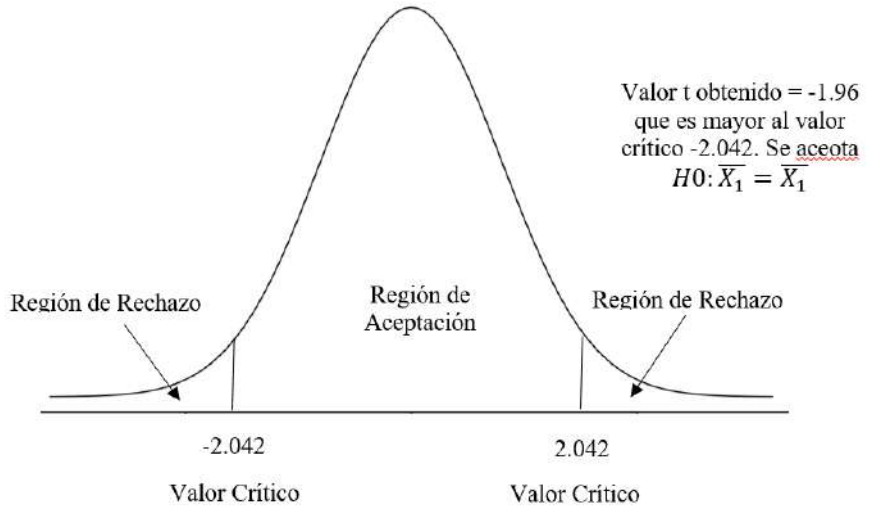


Figura 7-4: Regla de decisión aplicada, SDN con HP y SDN con Zodiac.
Realizado por: Marcelo Criollo, 2019

4.1.2.2 Análisis de Jitter en Servidor, SDN con Hp vs. Tradicional con Hp

Tabla 13-4: Estadísticas de grupo, SDN con HP y Tradicional con Hp

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Jitter – Servidor	SDN con HP	30	,00127470	,000953951	,000174167
	Tradicional con HP	30	,00088493	,000744166	,000135866

Realizado por: Marcelo Criollo, 2019

Tabla 14-4: Prueba muestras independientes, SDN con HP y Tradicional con HP

		Prueba de Levene de igualdad de varianzas		prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Jitter Servidor	Se asumen varianzas iguales	2,823	,098	1,765	58	,083	,000389767	,000220893
	No se asumen varianzas iguales			1,765	54,757	,083	,000389767	,000220893

Realizado por: Marcelo Criollo, 2019

La prueba F permite concluir que las varianzas de las dos muestras no son significativamente diferentes (p-valor es mayor que 0.05) con un valor de 0,98. Al analizar los resultados de la primera fila (Prueba T) se observa que el p-valor (0.083) es mayor que 0.05 lo que lleva a aceptar la hipótesis de igualdad de medias permitiendo concluir que no existe diferencia significativa entre las medias o promedios de SDN con HP y Tradicional con HP para la variable Jitter en el servidor.

Ya que se asume varianzas iguales, el valor t obtenido con la fórmula descrita anteriormente es igual a 1.76 el cual es menor al valor crítico de 2.002 es el que se obtuvo de la tabla de distribución t de dos colas con 58 grados de libertad. Este valor T se ubica en la región de aceptación llevando a aceptar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. La siguiente Figura 8-4 muestra la regla de decisión aplicada.

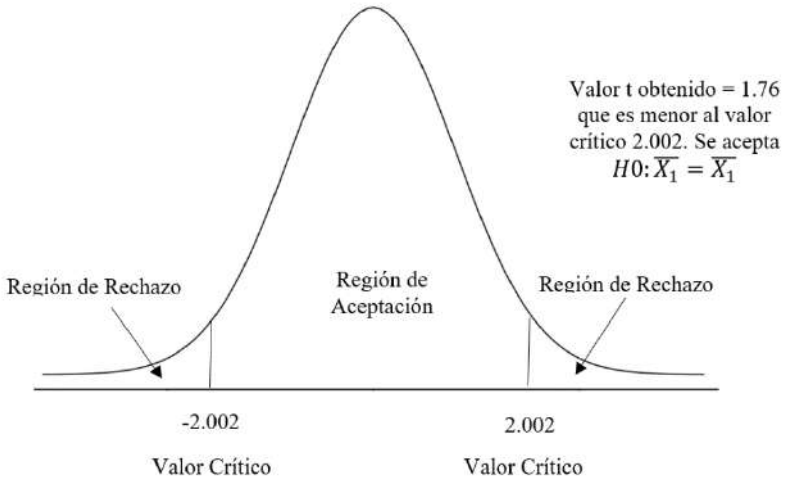


Figura 8-4: Regla de decisión aplicada, SDN con HP y Tradicional con HP.
Realizado por: Marcelo Criollo, 2019

4.1.2.3 Resumen Jitter en Servidor

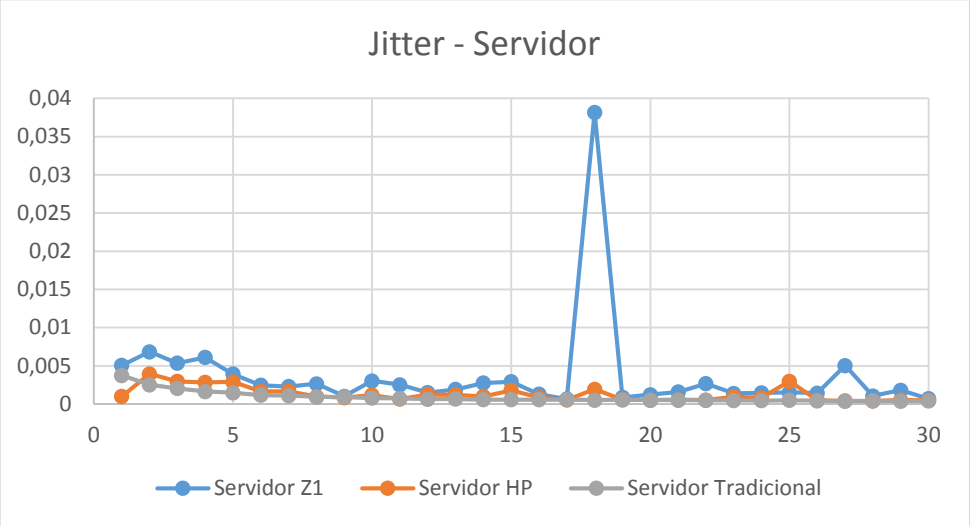


Figura 9-4: Jitter en Servidor
Realizado por: Marcelo Criollo, 2019

En la Figura 9-4 se detallan los valores obtenidos de las pruebas de Jitter en el Servidor de entre los equipos: SDN con Zodiac, SDN con Hp y Tradicional con Hp. En la que se observan las fluctuaciones que presentan las mediciones realizadas en los escenarios propuestos, llegándose a la conclusión que el mejor rendimiento de Jitter en el Servidor se obtiene del equipo Tradicional con Hp. Es importante destacar que en Jitter el valor entre más pequeño sea, es indicativo de mejor rendimiento.

4.1.2.4 Análisis de Jitter en Cliente, SDN con Hp vs. SDN con Zodiac

Tabla 15-4: Estadísticas de grupo, SDN con HP y SDN con Zodiac

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Jitter - Cliente	SDN con HP	30	,00068513	,000537810	,000098190
	SDN con Zodiac.	30	,00089607	,000473115	,000086379

Realizado por: Marcelo Criollo, 2019

Tabla 16-4: Prueba de muestras independientes, SDN con HP y SDN con Zodiac

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Jitter Cliente	Se asumen varianzas iguales	,067	,797	-1,613	58	,112	-,000210933	,000130777
	No se asumen varianzas iguales			-1,613	57,073	,112	-,000210933	,000130777

Realizado por: Marcelo Criollo, 2019

Con respecto a la variable Jitter, la hipótesis de igualdad de varianzas debe aceptarse ya que el p-valor de esta prueba (Sig. = 0.797) es mayor que 0.05 lo que lleva a aceptar la hipótesis nula de igualdad de varianzas. Con base en esto se deben analizar los resultados de la prueba t de la primera fila de la tabla anterior para verificar la diferencia de medias de esta variable. Se evidencia que el p-valor es igual a 0.112 que es mayor que el valor de significancia 0.05 lo que lleva a aceptar la hipótesis nula de igualdad de medias entre SDN con HP y SDN con Zodiac en la variable Jitter, es decir no hay diferencia significativa entre estas tecnologías.

Al igual que en el caso anterior se asumen varianzas iguales por lo que el valor t obtenido con la fórmula descrita anteriormente es igual a -1.61 el cual es mayor al valor crítico de - 2.002 que se obtuvo de la tabla de distribución t de dos colas con 58 grados de libertad. Este valor T se ubica en la región de aceptación llevando a aceptar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. Figura 10-4 muestra la regla de decisión aplicada.

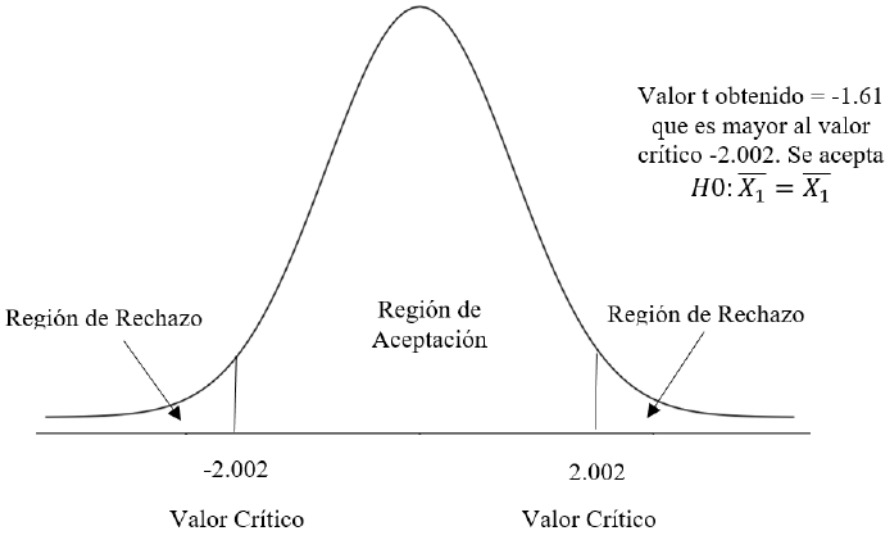


Figura 10-4: Regla de decisión aplicada, SDN con HP y SDN con Zodiac
Realizado por: Marcelo Criollo, 2019

4.1.2.5 Análisis de Jitter en Cliente, SDN con Hp vs. Tradicional con Hp

Tabla 17-4: Estadística de grupo, SDN con HP y Tradicional con HP

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Jitter - Cliente	SDN con HP	30	,00068513	,000537810	,000098190
	Tradicional con HP	30	,00049923	,000498145	,000090948

Realizado por: Marcelo Criollo.

Tabla 18-4: Prueba muestras independientes, SDN con HP y Tradicional con HP

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Jitter Cliente	Se asumen varianzas iguales	1,875	,176	1,389	58	,170	,000185900	,000133839
	No se asumen varianzas iguales			1,389	57,663	,170	-,000185900	,000133839

Realizado por: Marcelo Criollo, 2019

Se observa que la prueba F permite concluir que las varianzas de las dos muestras son iguales (p-valor es mayor que 0.05). Por otro lado, los resultados de la prueba T muestran un p-valor igual a 0.170 que es mayor a 0.05 lo que permite aceptar la hipótesis nula de igualdad de medias entre las dos tecnologías lo que permite concluir que no existe diferencia significativa en el Average Jitter entre SDN con HP y Tradicional con Hp.

Según la prueba de varianzas desarrollada en el software, se asumen que las varianzas son iguales, por lo tanto, el valor t obtenido con la fórmula descrita anteriormente es igual a 1.39 el cual es menor al valor crítico de 2.002 que se obtuvo de la tabla de distribución t de dos colas con 58 grados de libertad. Este valor t se ubica en la región de aceptación llevando a aceptar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. La siguiente Figura 11-4 muestra la regla de decisión aplicada:

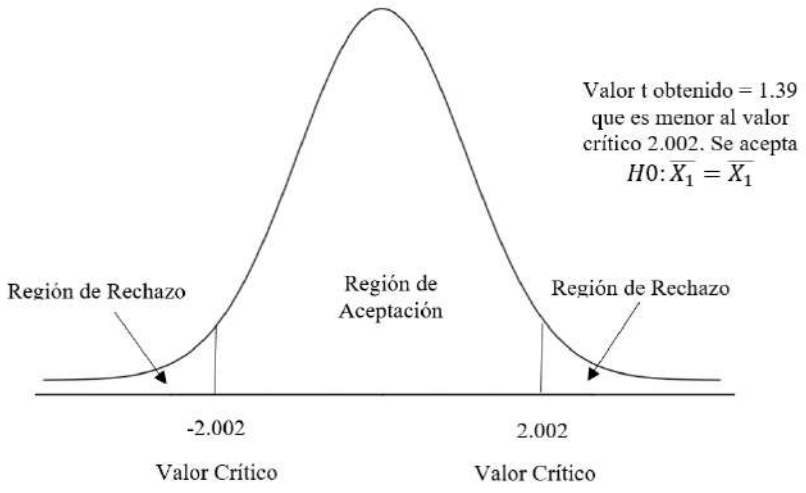


Figura 11-4: Regla de decisión aplicada.
Realizado por: Marcelo Criollo, 2019

4.1.2.6 Resumen Jitter en Cliente

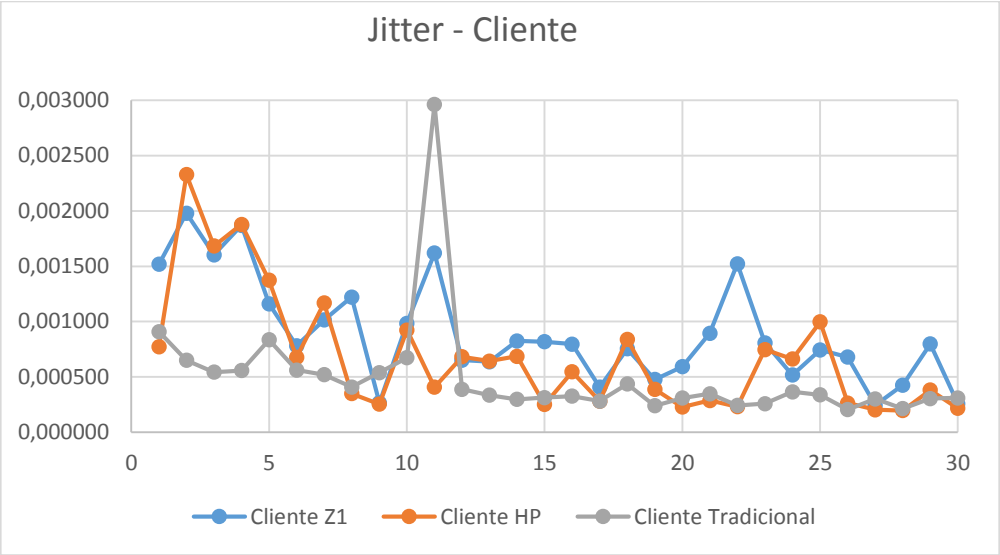


Figura 12-4: Jitter en Cliente
Realizado por: Marcelo Criollo, 2019

En la Figura 12-4 se detallan los valores obtenidos de las pruebas de Jitter en el Cliente de entre los equipos: SDN con Zodiac, SDN con Hp y Tradicional con Hp. En la que se observan las fluctuaciones que presentan las mediciones realizadas en los escenarios propuestos, llegándose a la conclusión que el mejor rendimiento de Jitter en el Cliente se obtiene del equipo Tradicional con Hp. Es importante destacar que en Jitter el valor entre más pequeño sea, es indicativo de mejor rendimiento.

4.1.3 Indicador Ancho de Banda

Para el análisis de ancho de banda se tomaron un total de 30 muestras.

Tabla 19-4: Indicadores de Ancho de Banda

Indicador de ancho de banda (Kbps/seg)						
Número de muestras	Con SDN Zodiac		Con SDN H 3800		Convencional Hp 3800	
	Servidor	Cliente	Servidor	Cliente	Servidor	Cliente
1	6593,80391	6.645,719845	56945,211720	61.049,963229	8.179,828017	8167,494250
2	8242,882873	8.337,431716	10713,564392	10.715,123277	11.586,658863	11.588,953768
3	9087,689723	9.205,482119	13209,045425	13.210,293239	14.964,157669	14.958,984431
4	9446,936009	9.559,573745	17459,246572	17.484,564574	18.392,122545	18.371,850786
5	9318,856372	9.415,941792	13032,391899	17.460,406092	21.751,177321	21.773,432774
6	11023,398218	11.294,507373	22262,678033	22.712,248572	25.151,143792	25.184,543891
7	9922,985717	10.362,638439	26529,086716	27.097,021868	28.540,617966	28.515,105835
8	12055,899250	12.612,838409	31899,818977	31.901,133670	31.961,125856	31.971,240301
9	14360,235604	16.212,729444	35303,691768	35.326,207821	34.561,872505	35.267,735201
10	14819,965635	17.069,656363	36006,881799	36.756,749151	38.697,528787	38.777,077277
11	15583,210972	17.699,393173	41182,222452	42.085,387152	41.216,033529	42.161,868695
12	14845,627990	15.845,269628	43165,990030	43.252,297645	45.508,990371	45.555,821707
13	11473,865280	14.407,312948	12196,089786	32.905,970383	47.399,275157	48.445,117577
14	16397,779742	18.915,714573	45440,646231	45.524,051782	50.106,268534	50.212,136359
15	16781,473699	19.300,654536	43867,269818	53.979,811259	55.515,544217	55.749,649765
16	16136,028505	18.628,741952	46600,897456	46.899,015698	53.779,199951	53.943,757999
17	18586,971966	22.468,702676	60194,714221	60.727,035066	55.885,604948	56.079,281672
18	16374,800641	20.355,659725	43643,165238	43.973,370238	65.498,260265	65.857,521037
19	18300,643179	24.167,219200	54058,695236	55.543,091396	57.487,840277	58.934,383892
20	19100,563671	25.627,931807	59225,774007	59.607,859200	60.717,409014	61.036,779828
21	4945,486091	7.345,525924	61788,022664	64.719,659298	66.482,227901	68.682,168393
22	2315,321707	3.406,994170	62043,556490	62.737,898884	66.307,076170	69.282,830106
23	4973,304026	7.217,395887	40410,394028	40.804,048489	64.966,980687	66.829,122180
24	13972,58717	21.591,089012	47997,512076	50.739,634982	67.799,382913	68.688,242634
25	10402,57364	17.086,218811	11227,375307	19.327,363184	66.357,340258	69.311,852874
26	11279,41026	19.237,154139	61224,156968	66.700,313573	70.444,742357	71.684,301230
27	16335,17801	27.995,107512	68356,435667	69.725,786615	71.756,848405	72.805,901598
28	12886,68123	24.496,850896	69970,667729	71.581,084333	73.074,993308	74.022,875110
29	7516,741233	13.797,422135	61935,238536	67.414,264979	74.200,311465	75.292,901988
30	13640,79372	26.607,681275	61322,020658	66.490,165552	74.306,710329	75.966,644826

Realizado por: Marcelo Criollo, 2019

4.1.3.1 Análisis Ancho de Banda en Servidor, SDN HP vs. SDN Zodiac

Tabla 20-4: Estadísticas de grupo, SDN con HP y SDN con Zodiac.

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Ancho de banda - Servidor	SDN con HP	30	41973,74872997	18921,742286925	3454,621692616
	Tradicional con HP	30	12224,05653477	4387,494233427	801,043187524

Realizado por: Marcelo Criollo, 2019

Tabla 21-4: Prueba de muestras independientes, SDN con HP y SDN con Zodiac

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Ancho de banda Servidor	Se asumen varianzas iguales	38,599	,000	8,389	58	,000	29749,692195200	3546,277093992
	No se asumen varianzas iguales			8,389	32,109	,000	29749,692195200	3546,277093992

Realizado por: Marcelo Criollo, 2019

La prueba F permite concluir que las varianzas de las dos muestras son significativamente diferentes (p-valor es menor que 0.05). Al analizar los resultados de la segunda fila (Prueba T) se observa que el p-valor (0.000) es menor que 0.05 lo que lleva a rechazar la hipótesis de igualdad de medias permitiendo y concluir que existe diferencia significativa entre las medias o promedios de SDN con HP y SDN con Zodiac para la variable ancho de banda en el servidor.

Dado que se asume que las varianzas son diferentes el valor t obtenido con la fórmula descrita anteriormente es igual a 8.4 el cual es menor al valor crítico de 2.037 que se obtuvo de la tabla de distribución t de dos colas con 32 grados de libertad. Este valor T se ubica en la región de rechazo llevando a rechazar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. La Figura 13-4 muestra la regla de decisión aplicada.

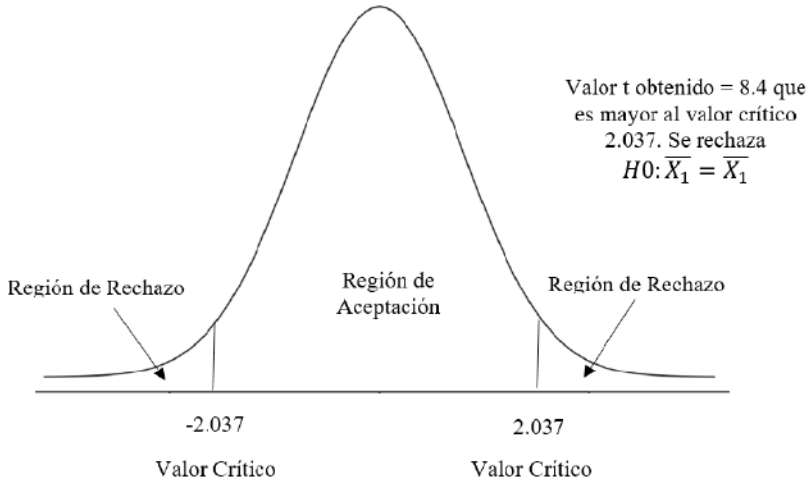


Figura 13-4: Regla de decisión aplicada
Realizado por: Marcelo Criollo, 2019

4.1.3.2 Análisis Ancho de Banda en Servidor, SDN con Hp vs. Tradicional Hp

Tabla 22-4: Estadísticas de grupo, SDN con Hp y Tradicional con Hp

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Ancho de banda - Servidor	SDN con HP	30	41973,74872997	18921,742286925	3454,621692616
	Tradicional con HP	30	48753,24244590	20771,930681206	3792,418332343

Realizado por: Marcelo Criollo, 2019

Tabla 23-4: Prueba muestras independientes, SDN con HP y Tradicional con HP

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Ancho de banda Servidor	Se asumen varianzas iguales	,630	,431	-1,322	58	,192	-6779,493715933	5129,994916819
	No se asumen varianzas iguales			-1,322	57,502	,192	-6779,493715933	5129,994916819

Realizado por: Marcelo Criollo, 2019

La prueba F permite concluir que las varianzas de las dos muestras son significativamente diferentes (p-valor es mayor que 0.05) con un valor de 0,431. Al analizar los resultados de la primera fila (Prueba T) se observa que el p-valor (0.192) es mayor que 0.05 lo que lleva a aceptar la hipótesis de igualdad de medias permitiendo y concluir que no existe diferencia significativa entre las medias o promedios de SDN con Hp y Tradicional con HP para la variable ancho de banda en el servidor.

Las varianzas de las dos muestras son iguales y el valor t obtenido con la fórmula descrita anteriormente es igual a 1.33 el cual es menor al valor crítico de 2.002 que se obtuvo de la tabla

de distribución t de dos colas con 58 grados de libertad. Este valor T se ubica en la región de aceptación llevando a aceptar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. La Figura 14-4 muestra la regla de decisión aplicada.

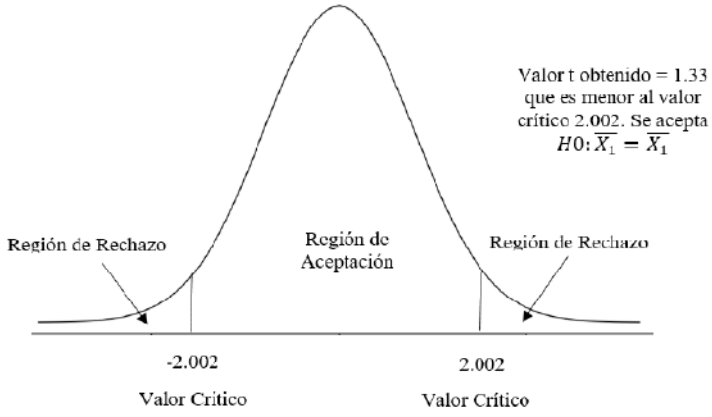


Figura 14-4: Regla de decisión aplicada
Realizado por: Marcelo Criollo, 2019

4.1.3.3 Resumen Ancho de Banda en Servidor

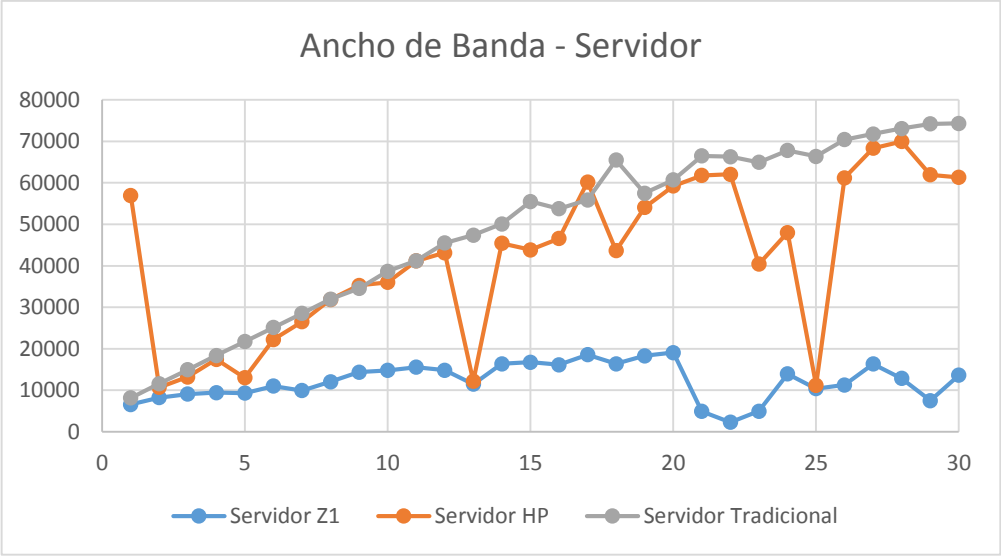


Figura 15-4: Ancho de Banda en Servidor
Realizado por: Marcelo Criollo, 2019

En la Figura 15-4 se detallan los valores obtenidos de las pruebas de Ancho de Banda en el Servidor de entre los equipos: SDN con Zodiac, SDN con Hp y Tradicional con Hp. En la que se observan las fluctuaciones que presentan las mediciones realizadas en los escenarios propuestos, llegándose a la conclusión que el mejor rendimiento de Ancho de Banda en el Servidor se obtiene del equipo Tradicional con Hp. Es importante destacar que en Ancho de Banda el valor entre más grande sea, es indicativo de mejor rendimiento.

4.1.3.4 Análisis de Ancho de Banda en Cliente, SDN con Hp vs. SDN con Zodiac

Tabla 24-4: Estadísticas de grupo, SDN con HP y SDN con Zodiac

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Ancho de banda - Cliente	SDN con HP	30	44615,06070670	18649,720388613	3404,957516003
	Tradicional con HP	30	15897,15197413	6649,967082081	1214,112325841

Realizado por: Marcelo Criollo, 2019

Tabla 25-4: Prueba de muestras independientes, SDN con HP y SDN con Zodiac

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Ancho de banda Cliente	Se asumen varianzas iguales	28,172	,000	7,944	58	,000	28717,908732567	3614,941828791
	No se asumen varianzas iguales			7,944	36,257	,000	28717,908732567	3614,941828791

Realizado por: Marcelo Criollo, 2019

La prueba F permite concluir que las varianzas de las dos muestras son significativamente diferentes (p-valor es menor que 0.05). Al analizar los resultados de la segunda fila (Prueba T) se observa que el p-valor (0.000) es menor que 0.05 lo que lleva a rechazar la hipótesis de igualdad de medias permitiendo concluir que si existe diferencia significativa entre las medias o promedios de SDN con Hp y SDN con Zodiac para la variable ancho de banda.

Según la prueba de varianzas desarrollada en el software, se asumen que las varianzas son diferentes, por lo tanto, el valor t obtenido con la fórmula descrita anteriormente es igual a 7.93 el cual es mayor al valor crítico de 2.028 que se obtuvo de la tabla de distribución t de dos colas con 36 grados de libertad. Este valor T se ubica en la región de rechazo llevando a rechazar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. La Figura 16-4 muestra la regla de decisión aplicada.

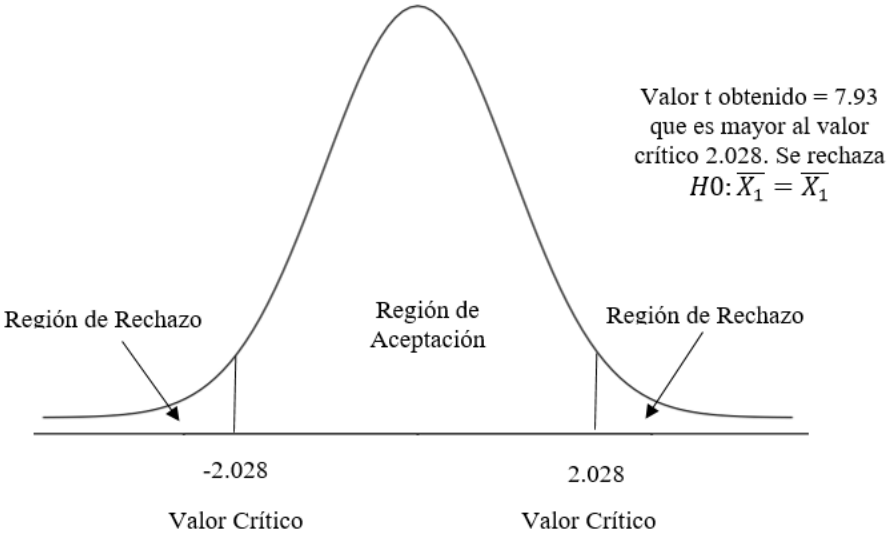


Figura 16-4: Regla de decisión aplicada
Realizado por: Marcelo Criollo, 2019

4.1.3.5 Análisis de Ancho de Banda en Cliente, SDN con Hp vs. Tradicional Hp

Tabla 26-4: Estadística de grupo, SDN con HP y Tradicional con HP

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de erro estándar
Ancho de banda - Cliente	SDN con HP	30	44615,06070670	18649,720388613	3404,957516003
	Tradicional con HP	30	49503,98593280	21349,611478148	3897,887933518

Realizado por: Marcelo Criollo, 2019

Tabla 27-4: Prueba muestras independientes, SDN con HP y Tradicional con HP

		Prueba de Levene de igualdad de varianzas		prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Ancho de banda Cliente	Se asumen varianzas iguales	,950	,334	-,945	58	,349	-4888,925226100	5175,641605448
	No se asumen varianzas iguales			-,945	56,971	,349	-4888,925226100	5175,641605448

Realizado por: Marcelo Criollo, 2019

En esta prueba se asume igualdad entre las varianzas ya que el p-valor de la prueba F (0.334) es mayor a 0.05. Los resultados de la prueba T por otro lado permiten concluir que no existe diferencia entre la tecnología SDN con HP y Tradicional con HP ya que el p-valor es mayor que 0.05 llevando a aceptar la hipótesis nula de igualdad de medias.

Ya que las varianzas de las muestras son iguales el valor t obtenido con la fórmula descrita anteriormente es igual a -0.945 el cual es mayor al valor crítico de -2.002 que se obtuvo de la

tabla de distribución t de dos colas con 58 grados de libertad. Este valor T se ubica en la región de aceptación llevando a aceptar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. La Figura 17-4 muestra la regla de decisión aplicada

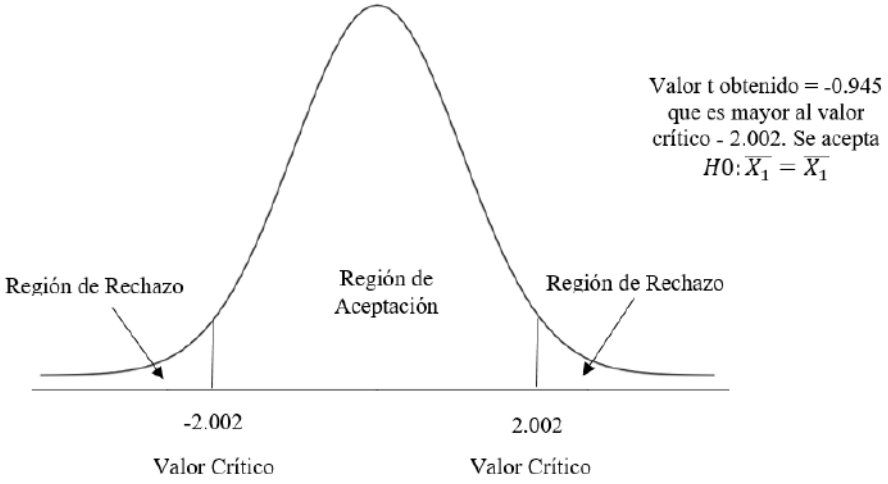


Figura 17-4: Regla de decisión aplicada
Realizado por: Marcelo Criollo, 2019

4.1.3.6 Resumen Ancho de Banda Cliente

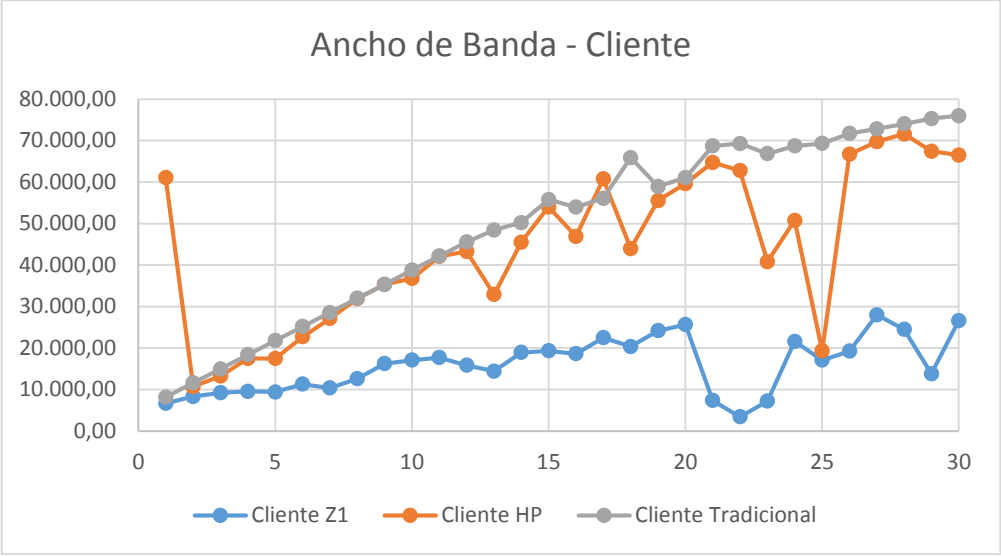


Figura 18-4: Ancho de Banda Cliente
Realizado por: Marcelo Criollo, 2019

En la Figura 18-4 se detallan los valores obtenidos de las pruebas de Ancho de Banda en el Cliente de entre los equipos: SDN con Zodiac, SDN con Hp y Tradicional con Hp. En la que se observan las fluctuaciones que presentan las mediciones realizadas en los escenarios propuestos, llegándose a la conclusión que el mejor rendimiento de Ancho de Banda en el Cliente se obtiene del equipo Tradicional con Hp. Es importante destacar que en Ancho de Banda el valor entre más grande sea, es indicativo de mejor rendimiento.

4.1.4 Indicador Pérdida de Paquetes

Para el análisis de pérdida de paquetes se toman un total de 30 muestras.

Tabla 28-4: Indicador de pérdida de paquetes

Indicador de pérdida de paquetes						
Número de0 muestras	Con SDN Zodiac		Con SDN HP 3800		Convencional	
	Servidor	Cliente	Servidor	Cliente	Servidor	Cliente
1	2,310000	1,640000	13,280000	7,080000	0,000000	0,000000
2	3,120000	2,100000	4,760000	4,760000	0,000000	0,000000
3	3,510000	2,300000	4,980000	4,970000	0,000000	0,000000
4	4,090000	2,940000	4,890000	4,890000	0,000000	0,000000
5	3,560000	2,580000	9,910000	5,020000	0,000000	0,000000
6	8,350000	6,240000	9,810000	9,790000	0,020000	0,000000
7	9,350000	5,360000	5,170000	5,120000	0,070000	0,000000
8	23,810000	20,300000	0,270000	0,240000	0,070000	0,000000
9	18,920000	8,630000	0,240000	0,170000	0,050000	0,000000
10	23,750000	12,260000	5,240000	5,180000	0,080000	0,000000
11	23,810000	13,540000	0,260000	0,100000	0,190000	0,000000
12	35,120000	31,020000	5,300000	5,110000	0,160000	0,000000
13	34,090000	27,750000	25,670000	6,540000	0,200000	0,000000
14	37,380000	27,790000	5,720000	5,540000	0,220000	0,000000
15	36,620000	27,170000	0,900000	0,020000	0,470000	0,000000
16	36,200000	26,450000	7,120000	6,540000	0,300000	0,000000
17	40,650000	28,300000	1,070000	0,170000	0,350000	0,000000
18	44,030000	30,480000	7,950000	7,300000	0,650000	0,000000
19	47,280000	30,460000	0,760000	0,160000	0,500000	0,000000
20	45,000000	26,220000	0,870000	0,260000	0,590000	0,000000
21	53,550000	31,000000	4,750000	0,200000	3,180000	0,000000
22	53,730000	31,910000	1,240000	0,120000	4,330000	0,000000
23	53,340000	39,550000	9,360000	8,490000	0,920000	0,000000
24	58,910000	36,510000	14,020000	9,070000	1,160000	0,000000
25	70,560000	51,650000	47,020000	8,790000	4,180000	0,000000
26	65,780000	41,680000	8,630000	0,430000	1,710000	0,000000
27	66,180000	42,040000	2,060000	0,100000	1,500000	0,000000
28	74,200000	50,830000	2,250000	0,100000	1,280000	0,000000
29	70,990000	46,790000	8,390000	0,400000	1,550000	0,000000
30	74,380000	50,030000	8,230000	0,500000	2,280000	0,000000

Realizado por: Marcelo Criollo, 2019

4.1.4.1 Análisis de Pérdida Paquetes en Servidor, SDN con HP vs. SDN con Zodiac

Tabla 29-4: Estadísticas de grupo, SDN con HP y SDN con Zodiac

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de error estándar
Pérdida de paquetes – Servidor.	SDN con HP.	30	7,33733333	9,209188790	1,681360145
	SDN con Zodiac.	30	37,419000	23,7757212	4,34083294

Realizado por: Marcelo Criollo, 2019

Tabla 30-4: Prueba de muestras independientes, SDN con HP y SDN con Zodiac

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Pérdida de paquetes Servidor	Se asumen varianzas iguales	27,515	,00	6,462	58	,000	30,08167	4,65508
	No se asumen varianzas iguales			6,462	37,510	,000	30,08167	4,65508

Realizado por: Marcelo Criollo, 2019

Para este caso, el comportamiento del indicador de paquetes perdidos se lo realizar observado la prueba F que permite concluir que las varianzas de las dos muestras son significativamente diferentes (p-valor es menor que 0.05) con un valor de 0,00. Al analizar los resultados de la segunda fila (Prueba T) se observa que el p-valor (0.00) es menor que 0.05 lo que lleva a rechazar la hipótesis de igualdad de medias, concluyendo que existe diferencias significativas entre las medias o promedios de SDN con HP y SDN con Zodiac.

Según la prueba de varianzas desarrollada en el software, se asumen que las varianzas son diferentes, por lo tanto, el valor t obtenido con la fórmula descrita anteriormente es igual a 6,462 el cual es menor al valor crítico de - 2.013 que se obtuvo de la tabla de distribución t de dos colas con 46 grados de libertad. Este valor T se ubica en la región de rechazo llevando a rechazar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. La Figura 19-4 muestra la regla de decisión aplicada.

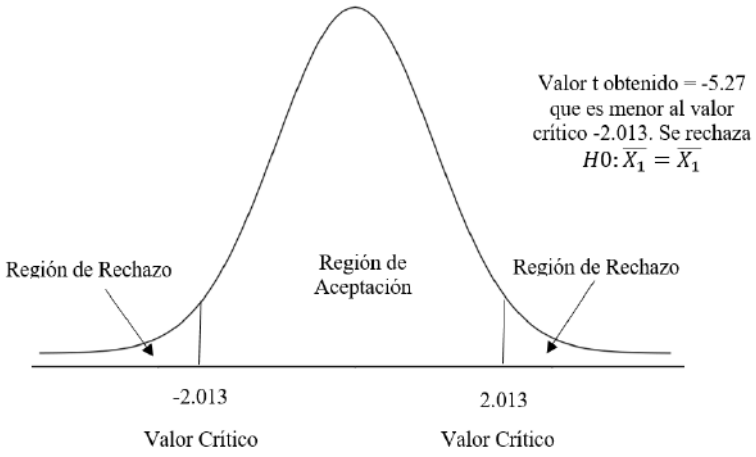


Figura 19-4: Regla de decisión aplicada
Realizado por: Marcelo Criollo, 2019

4.1.4.2 Análisis de Pérdida de Paquetes en Servidor, SDN con Hp vs. Tradicional Hp

Tabla 31-4: Estadísticas de grupo, SDN con HP y Tradicional con Hp

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de error estándar
Pérdida de paquetes - Servidor	SDN con HP	30	7,33733333	9,209188790	1,681360145
	Tradicional con HP	30	,86700000	1,203609944	,219748106

Realizado por: Marcelo Criollo, 2019

Tabla 32-4: Prueba muestras independientes, SDN con HP y Tradicional con Hp

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Pérdida de paquetes - Servidor	Se asumen varianzas iguales	11,287	,001	3,816	58	,000	6,470333333	1,695659508
	No se asumen varianzas iguales			3,816	29,990	,001	6,470333333	1,695659508

Realizado por: Marcelo Criollo, 2019

De igual manera ocurre al analizar el indicador de paquetes perdidos comparando el escenario SDN con Hp y el escenario Tradicional con Hp, en donde se la prueba F que permite concluir que las varianzas de las dos muestras son significativamente diferentes (p-valor es menor que 0.05) con un valor de 0,01. Al analizar los resultados de la segunda fila (Prueba T) se observa que el p-valor (0.01) es menor que 0.05 lo que lleva a rechazar la hipótesis de igualdad de medias, concluyendo que existe diferencias significativas entre las medias o promedios.

En este análisis se asumen varianzas desiguales, por lo tanto, el valor t obtenido con la fórmula descrita anteriormente es igual a 3.81 que es mayor al valor crítico de 2.042 que se obtuvo de la tabla de distribución t de dos colas con 30 grados de libertad. Este valor T se ubica en la región de rechazo llevando a rechazar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. La Figura 20-4 muestra la regla de decisión aplicada.

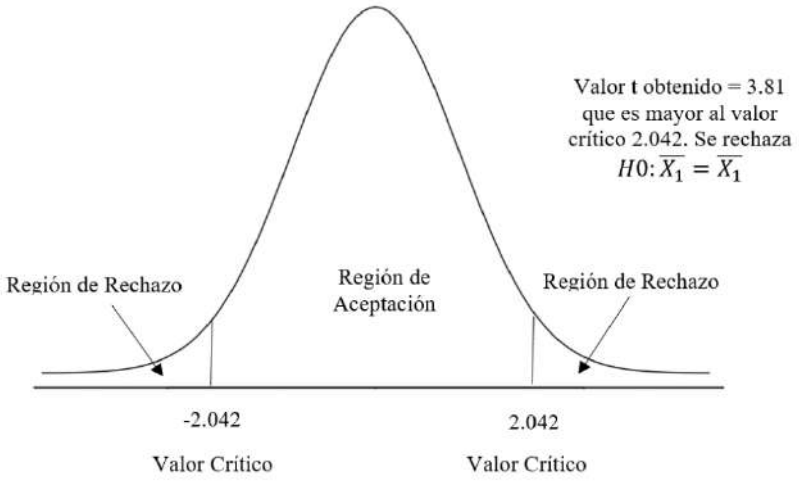


Figura 20-4: Regla de decisión aplicada
Realizado por: Marcelo Criollo, 2019

4.1.4.3 Resumen Pérdida de Paquetes Servidor

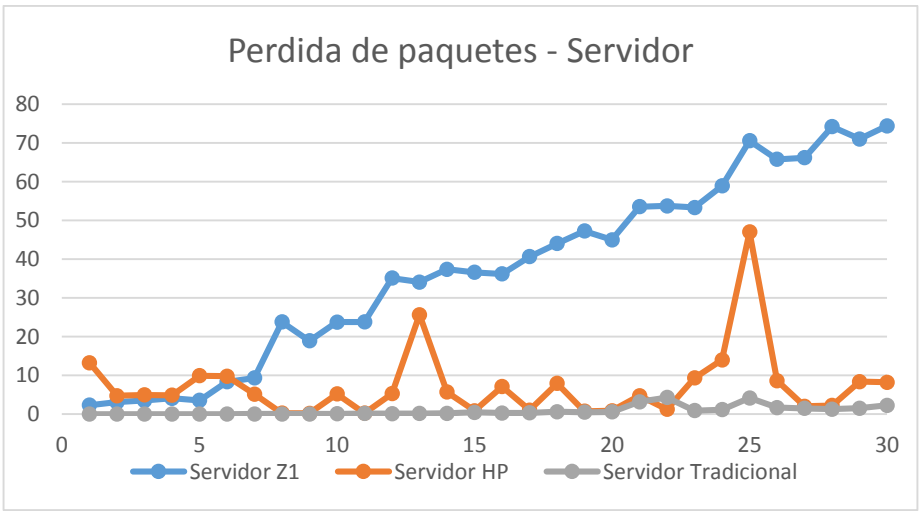


Figura 21-4: Paquetes perdidos en servidor
Realizado por: Marcelo Criollo, 2019

En la Figura 21-4 se detallan los valores obtenidos de las pruebas de Paquetes perdidos en el Servidor de entre los equipos: SDN con Zodiac, SDN con Hp y Tradicional con Hp. En la que se observan las fluctuaciones que presentan las mediciones realizadas en los escenarios propuestos, llegándose a la conclusión que el mejor rendimiento de Pérdida de Paquetes en el Servidor se

obtiene del equipo Tradicional con Hp. Es importante destacar que en Pérdida de paquetes el valor entre más pequeño sea, es indicativo de mejor rendimiento.

4.1.4.4 Análisis de Pérdida de Paquetes en Cliente, SDN con Hp vs. SDN con Zodiac

Tabla 33-4: Estadísticas de grupo, SDN con HP y SDN con Zodiac

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de error estándar
Pérdida de paquetes - Clientes	SDN con HP	30	3,57200000	3,428315561	,625921922
	SDN con Zodiac.	30	25,18400000	16,082649807	2,936276695

Realizado por: Marcelo Criollo, 2019

Tabla 34-4: Prueba de muestras independientes, SDN con HP y SDN con Zodiac

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Pérdida de paquetes –Cliente.	Se asumen varianzas iguales	39,588	,000	-7,199	58	,000	-21,612000000	3,002249004
	No se asumen varianzas iguales			-7,199	31,630	,000	-21,612000000	3,002249004

Realizado por: Marcelo Criollo, 2019

Dado que la prueba F de igualdad de varianzas arroja un p-valor (Sig) valor menor que el nivel de significancia de 0.05 se rechaza la hipótesis de igualdad de varianzas por lo que se deben verificar únicamente los datos de la segunda fila. La prueba T (cuando se asumen varianzas desiguales) muestra un p-valor menor a 0.05 lo que lleva a rechazar la hipótesis nula de igualdad de medias permitiendo concluir que existen diferencias significativas en los paquetes perdidos entre las dos tecnologías analizadas.

Al analizar las medias del primer cuadro se puede evidenciar que existen más paquetes perdidos en la tecnología SDN con Zodiac en comparación con SDN con HP. Según la prueba de varianzas desarrollada en el software, se asumen que las varianzas son diferentes, por lo tanto, el valor t obtenido con la fórmula descrita anteriormente es igual a 7.2 el cual es mayor al valor crítico de 2.037 que se obtuvo de la tabla de distribución t de dos colas con 32 grados de libertad. Este valor T se ubica en la región de rechazo llevando a rechazar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. La Figura 22-4 amuestra la regla de decisión aplicada.

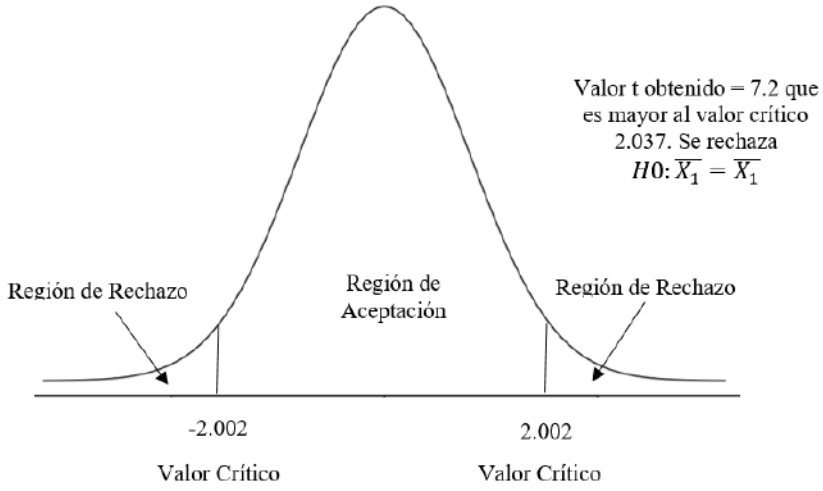


Figura 22-4: Regla de decisión aplicada
Realizado por: Marcelo Criollo, 2019

4.1.4.5 Análisis de Pérdida de Paquetes en Cliente, SDN con Hp vs. Tradicional Hp

Tabla 35-4: Estadística de grupo, SDN con HP y Tradicional con HP

Variable para clasificar los Grupos		N	Media	Desviación estándar	Medida de error estándar
Pérdida de paquetes - Clientes	SDN con HP	30	3,57200000	3,428315561	,625921922
	Tradicional con HP.	30	,00000000	,000000000	,000000000

Realizado por: Marcelo Criollo, 2019

Tabla 36-4: Prueba de muestras independientes, SDN con HP y Tradicional con HP

		Prueba de Levene de igualdad de varianzas		Prueba t para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar
Pérdida de paquetes – Cliente.	Se asumen varianzas iguales	186,645	,000	5,707	58	,000	3,572000000	,625921922
	No se asumen varianzas iguales			5,707	29,000	,000	3,572000000	,625921922

Realizado por: Marcelo Criollo, 2019

La prueba F permite concluir que las varianzas de las dos muestras son significativamente diferentes (p-valor es menor que 0.05) con un valor de 0,00. Al analizar los resultados de la segunda fila (Prueba T) se observa que el p-valor (0.00) es menor que 0.05, lo que lleva a rechazar la hipótesis de igualdad de medias permitiendo concluir que existen diferencias significativas entre las medias o promedios de SDN con HP y Tradicional con HP para la variable Pérdida de paquetes en el cliente. Según la prueba de varianzas desarrollada en el software, se asumen que

las varianzas son diferentes por lo tanto el valor t obtenido con la fórmula descrita anteriormente es igual a 5.7 el cual es mayor al valor crítico de 2.045 que se obtuvo de la tabla de distribución t de dos colas con 29 grados de libertad. Este valor T se ubica en la región de rechazo llevando a rechazar la hipótesis nula de igualdad de medias corroborando el análisis realizado con el software SPSS a través del p-valor. En la Figura 23-4 muestra la regla de decisión aplicada.

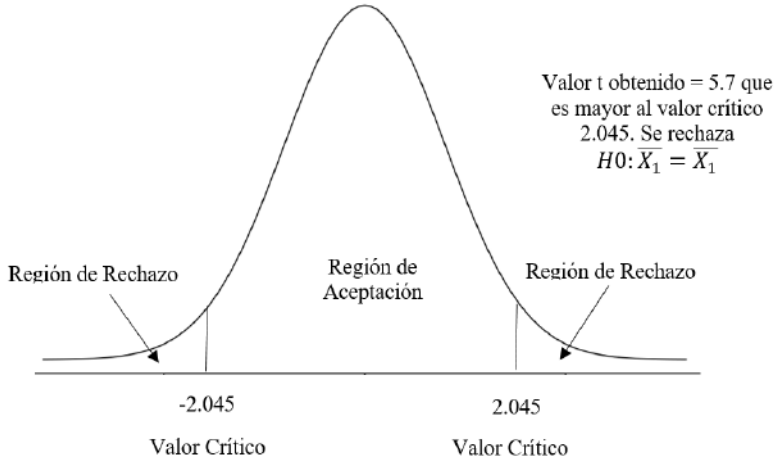


Figura 23-4: Regla de decisión aplicada
Realizado por: Marcelo Criollo, 2019

4.1.4.6 Resumen Pérdida de Paquetes Cliente

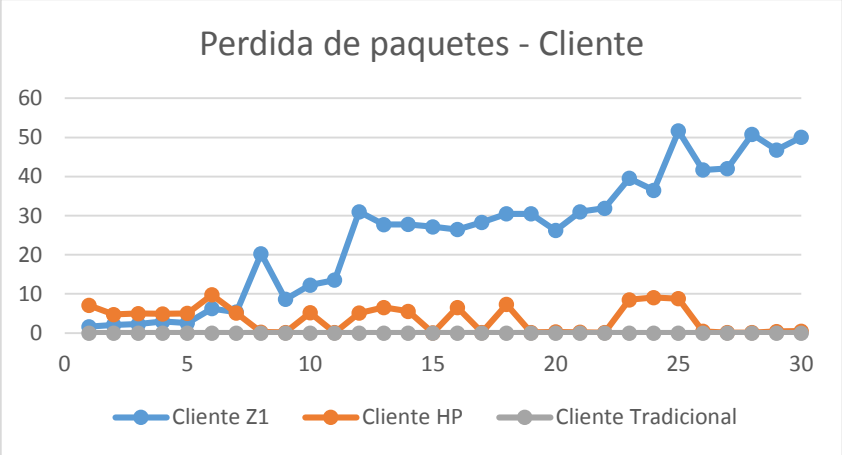


Figura 24-4: Pérdida de paquetes Cliente
Realizado por: Marcelo Criollo, 2019

En la Figura 24-4 se detallan los valores obtenidos de las pruebas de Paquetes perdidos en el Cliente de entre los equipos: SDN con Zodiac, SDN con Hp y Tradicional con Hp. En la que se observan las fluctuaciones que presentan las mediciones realizadas en los escenarios propuestos, llegándose a la conclusión que el mejor rendimiento de Pérdida de Paquetes en el Cliente se obtiene del equipo Tradicional con Hp. Es importante destacar que en Pérdida de paquetes el valor entre más pequeño sea, es indicativo de mejor rendimiento.

CAPÍTULO V

5 PROPUESTA

Como propuesta del presente trabajo de investigación se realiza una guía de implementación con especificaciones técnicas de hardware, software y recomendaciones en las configuraciones de los equipos.

5.1 Especificación de hardware

- Un conmutador HP 3800 Series 24G-25FP + J9575, para la implementación de los escenarios propuestos, el equipo HP funciona con el protocolo OpenFlow, lo que nos permite probar redes SDN, así como también al ambiente tradicional. Tiene un procesador ASIC / ARM @ 350 MHz; Freescale P2020 @ 1200 MHz, 4 Gb de flash y 2 Gb SDRAM.
- Dos Zodiac FX: es un switch OpenFlow desarrollado por Northbound Networks, que permite funcionalidades SDN para ser probado fácilmente en hardware. Sin embargo este equipo está diseñado para pruebas y no capacitado para entornos de específicamente para los servicios SDN y no se puede usar en entornos de producción. Se basa en el Atmel Microcontrolador ARM® Cortex®-M4 y tiene cuatro 10/100Mb puertos Ethernet. Compatible con Openflow 1.0 y 1.3.
- 3 computadoras (cliente, servidor, controlador).

5.2 Especificación de software

- Virtual Box 5.2.12: Herramienta de virtualización de código abierto, multiplataforma para Linux, Windows y MacOSX.
- Ubuntu 16.04 Desktop: sistema operativo base para el controlador.
- Java JDK 10.02: complemento para D-ITG.
- Controladores: OpenDayLight Nitrogen, Floodlight, RYU, Hp Van Controller, controladores evaluados para seleccionar el más apropiado.
- Wireshark 2.6.0: analizador de tráfico de red.
- D-ITG 2.61, GUI 0.92: Generado de tráfico.

5.3 Descripción de la metodología

Los escenarios han sido implementados utilizando equipos SDN con topología básica debido a que el análisis de tráfico se centra en un nodo de la red, el mismo que a pesar de ser pequeño usa cargas altas de tráfico. Adicionalmente, se ha considerado que por medio de ambientes de pruebas se realizará mediciones de las variables de estudio: latencia, jitter, ancho de banda y pérdida de paquetes; evaluando la funcionalidad de las políticas de QoS incluyendo condiciones de saturación, con el generador de tráfico DIT-G para medir el rendimiento de la red en los escenarios de estudio.

En base al análisis de políticas de QoS, se diseña las políticas a ser implementadas en tres escenarios reales para la evaluación de rendimiento en el presente estudio, para ello se aplica el modelo propuesto para diseño de políticas de QoS en redes Convencionales y SDN detallado en la Figura 1.5.

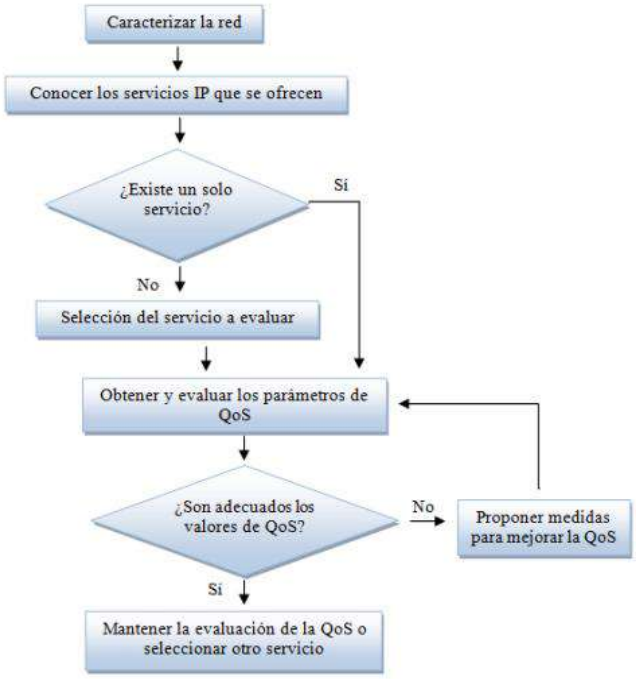


Figura 1-5: Modelo de evaluación de rendimiento
Fuente: http://revistatelematica.cujae.edu.cu/index.php/tele/article/download/259/241

5.4 Comparativa costo beneficio

SDN se encuentra implementado en hardware bajo el protocolo OpenFlow, en esta sección no se pretende dar una descripción detallada de fabricantes, sino más bien dar algunas opciones que

están disponible en el mercado actualmente. A continuación, se enumera los conmutadores comerciales con sus respectivos fabricantes, y la versión de OpenFlow compatible.

Tabla 1-5: Comparativa costo beneficio

Switch Company	Series	Version O.F.	Costo
Juniper	MX, EX	v1.0 y v1.3.1	$2,368.00
Cisco	Cisco cat6k, catalyst 3750, 6500 series	v1.0	$4689.09
Brocade	Brocade MLX, Brocade CER, Brocade CES, Brocade ADX Series	v1.3	$3287.25
Arista Networks	Arista 7124FX, Arista 7050	v1.0	$2195.351
IBM	RackSwitch G8264 y G8264T	v1.3.1	N/D
HP	HP 2920, 3500, 3500, 3800, 5130, 5400, 5930, 6200, 6600, 8200, 12500, 10500, 11900, and 12900 Switch Series	v1.0 y v1.3	$2,143.89
Dell	S4810-ON, S6000-ON, Dell Force10 Z9000, N3000, N4000 Series	v1.3	$2,999.99
NEC	PF1000, NEC IP8800, NEC PF5240, NEC PF5820, PF5248, PF5820, IP8800/S3640	v1.0 y v1.3.1	N/D
Pica8	P-3297, P-3930, P-3920, P-3922, P-5401, P-3780, P-3295	v1.2 y v1.4	$12,500.00
Northbound Networks	Zodiac FX	v1.0 – v1.3	$74,10
Northbound Networks	Zodiac GX	v1.0 – v1.5	$140.00

Realizado por: Marcelo Criollo, 2019

En la Tabla 1-5 se presentan costos de conmutadores que soportan el protocolo OpenFlow como una referencia para los administradores; en este proyecto se propone 3 escenarios, los cuales fueron implementados con diferente infraestructura con el fin de dar opciones a los gestores de red.

El primer escenario fue diseñado con el Switch Hp 3800 series el cual es compatible con el protocolo OpenFlow como se puede observar en la tabla anterior se encuentra disponible en el mercado, así como también existen otros switchs con características similares, pero a un mayor costo.

El segundo escenario propuesto se lo implementó con otra opción de hardware como lo es el módulo Zodiac FX pudiendo ser una alternativa más económica para trabajar con el protocolo OpenFlow.

Para el tercer escenario se reutilizó el switch hp 3800, con la diferencia que fue configurado sin la activación del protocolo OpenFlow; realizando de esta manera una comparativa a nivel de hardware, con el objetivo de proporcionar soluciones de QoS.

5.5 Tabla resumen estadística

Tabla 2-5: Tabla resumen estadística

Parámetros	Comparativa	Servidor		Cliente	
		Prueba F	**Prueba T**	**Prueba F**	**Prueba T**
Latencia	SDN con Hp y SDN con Zodiac	Sig=0,066>0,05	Sig=0,008<0,05	Sig=0,019<0,05	Sig=0,00<0,05
		No existe diferencia significativa, entre las dos tecnologías.	Rechaza la hipótesis de igualdad. Basado en las medias, la mejor respuesta es con el escenario SDN HP.	Existe diferencia significativa, entre las dos tecnologías.	Rechaza la hipótesis de igualdad. La mejor respuesta se obtiene con el escenario SDN HP.
	SDN con Hp y Convencional con HP	Sig=0,006<0,05	sig=0,063>0,05	Sig=0,012<0,05	Sig=0,00<0,05
		Existe una diferencia significativa, entre las dos tecnologías.	Se acepta la hipótesis de igualdad. Basado en las medias la mejor respuesta es con el escenario convencional con HP.	Existe una diferencia significativa, entre las dos tecnologías.	Rechaza la hipótesis de igualdad. Basado en las medias la mejor respuesta es con el escenario convencional con HP.
Jitter	SDN con Hp y SDN con Zodiac	Sig=0,048<0,05	Sig=0,059>0,05	Sig=0,0797>0,05	Sig=0,112>0,05
		Existe una diferencia significativa, entre las dos tecnologías.	Se acepta la hipótesis de igualdad. Basado en las medias la mejor respuesta es con el escenario SDN HP.	No existe diferencia significativa, entre las dos tecnologías.	Acepta la hipótesis nula de igualdad. Basado en las medias, la mejor respuesta es SDN HP.
	SDN con Hp y Convencional con HP	Sig=0,098>0,05	Sig=0,083>0,05	Sig=0,176>0,05	Sig=0,170>0,05
		No existe diferencia significativa, entre las dos tecnologías.	Acepta la hipótesis de igualdad. Basado en las medias la mejor respuesta es con el escenario convencional con HP.	No existe diferencia significativa, entre las dos tecnologías.	Acepta la hipótesis nula de igualdad. Basado en las medias, la mejor respuesta es SDN HP.
Ancho de banda	SDN con Hp y SDN con Zodiac	Sig=0,00<0,05	Sig=0,00<0,05	Sig=0,00<0,05	Sig=0,00<0,05
		Existe una diferencia significativa, entre las dos tecnologías.	Rechaza la hipótesis de igualdad. Basado en las medias, la mejor respuesta es con el escenario SDN HP.	Existe una diferencia significativa, entre las dos tecnologías.	Rechaza la hipótesis de igualdad. Basado en las medias, la mejor respuesta es con el escenario SDN HP.
	SDN con Hp y Convencional con HP	Sig=0,431>0,05	Sig=0,192>0,05	Sig=0,334>0,05	Sig=0,349>0,05
		No existe diferencia significativa, entre las dos tecnologías.	Acepta la hipótesis de igualdad. Basado en las medias la mejor respuesta es con el escenario convencional con HP.	No existe diferencia significativa, entre las dos tecnologías.	Acepta la hipótesis de igualdad. Basado en las medias la mejor respuesta es con el escenario convencional con HP.
Pérdida de paquetes	SDN con Hp y SDN con Zodiac	Sig=0,00<0,05	Sig=0,00<0,05	Sig=0,00<0,05	Sig=0,00<0,05
		Existe una diferencia significativa, entre las dos tecnologías.	Rechaza la hipótesis de igualdad. Basado en las medias, la mejor respuesta es con el escenario SDN HP.	Existe una diferencia significativa, entre las dos tecnologías.	Rechaza la hipótesis de igualdad. Basado en las medias, la mejor respuesta es con el escenario SDN HP.
	SDN con Hp y Convencional con HP	Sig=0,01<0,05	Sig=0,01<0,05	Sig=0,00<0,05	Sig=0,00<0,05
		Existe una diferencia significativa, entre las dos tecnologías.	Rechaza la hipótesis de igualdad. Basado en las medias, la mejor respuesta es con el escenario convencional HP.	Existe una diferencia significativa, entre las dos tecnologías	Rechaza la hipótesis de igualdad. Basado en las medias, la mejor respuesta es con el escenario convencional HP.

Realizado por: Marcelo Criollo, 2019

5.6 Tabla de resultados de hipótesis

De acuerdo al planteamiento de las hipótesis alternativa y nula tenemos:

Ha= La propuesta de asignación de políticas de QoS generará mayor rendimiento en una red enterprise SDN que en un entorno CONVENCIONAL.

Ho= La propuesta de asignación de políticas de QoS generará mayor rendimiento en una red enterprise CONVENCIONAL que en un entorno SDN.

Teniendo en consideración los parámetros de estudio y su respectivo análisis estadístico podemos concluir que su relación con la hipótesis de forma resumida es la siguiente:

Tabla 3-5: Tabla de resultados de hipótesis.

	Servidor		Cliente	
	Ha	Ho	Ha	Ho
Latencia	0	**1**	0	**1**
Jitter	0	**1**	0	**0**
Ancho de banda	0	**1**	0	**1**
Pérdida de paquetes	0	**1**	0	**1**

Realizado por: Marcelo Criollo, 2019

En donde se indica que: 0= No hay mejora y 1= Hay mejora. Según los resultados estadísticos obtenidos no se tiene mejora en el servidor al evaluar la tecnología SDN con HP con los parámetros: latencia, jitter, ancho de banda y pérdida de paquetes.
En el parámetro jitter con el cliente se observa que no hay diferencia significativa al evaluar la tecnología SDN con HP y tradicional con HP, por lo que se concluye que estas tecnologías tienen estadísticamente un funcionamiento similar.

Estadísticamente este estudio se enfoca en evaluar el rendimiento de la red en función de 4 parámetros que son: jitter, latencia, ancho de banda y paquetes al aplicar políticas de QoS; en el que refleja resultados muy similares independientemente de la tecnología que se utilizó, es decir tanto en un escenario tradicional como en un escenario bajo la plataforma de SDN, se obtuvo valores equivalentes, los cuales son visualizados en las gráficas presentadas en este apartado. Por ende, no se puede generalizar cuál de estas tecnologías es superior a la otra, ya que al realizar el análisis estadístico con la prueba T-Student, se determina que no se tiene mejora en el servidor, mientras que en el cliente no hay diferencia significativa al evaluar la tecnología SDN con HP y tradicional con HP, concluyendo que la implementación de políticas de QoS tienen igual relevancia. El equipo HP se presta para trabajar aplicando dos tecnologías de concepción distinta tanto en entorno tradicional como SDN.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

GUÍA METODOLÓGICA

POLÍTICAS DE QoS EN REDES EMPRESARIALES PARA EL ANÁLISIS DE RENDIMIENTO, EN ENTORNOS CONVENCIONALES Y SDN.

MARCELO DANIEL CRIOLLO BUSTAMANTE

Riobamba - Ecuador

Marzo, 2020

Contenido

ÍNDICE DE FIGURE

INTRODUCCIÓN.

Por medio del desarrollo de este proyecto se presenta un material práctico que permitirá a los administradores interesados en el área de las telecomunicaciones, profundizar en conceptos de tecnologías de nueva generación emergentes como SDN y aprovechar de mejor manera los recursos, dando una óptima calidad de servicio en su infraestructura de red. Además, este documento pretende ser un instructivo para ganar habilidades técnicas en el área de redes de telecomunicación y busca acelerar el despliegue, uso y desarrollo de la tecnología SDN en las empresas.

Cabe mencionar que, a más de los pasos para la instalación de herramientas, esta guía le ofrece configuraciones para la asignación de políticas de Calidad de Servicio, tanto para entornos SDN, como para redes convencionales.

IMPLEMENTACIÓN

A. Pasos para implementación de redes definidas por software, con el fin de brindar las bases prácticas para diferentes escenarios.

1. Instalar la máquina virtual

Notas: Esto fue probado en una computadora portátil con 16GB de RAM.

- Obtener el software de VirtualBox de su página web, preferiblemente la última versión disponible, https://www.virtualbox.org/wiki/Downloads.
- Instalar el software, ejecutando el archivo *.exe descargado y seguir los pasos indicados por el instalador

2. Instalar el servidor Ubuntu 16.04 en VirtualBox

- Descarga la imagen .iso para la versión Ubuntu-16.04-desktop.
- Abra Oracle Virtual Box y haga clic en "Nuevo".
- Ingrese el nombre de su máquina virtual. Luego, seleccione Tipo como Linux y Versión como Ubuntu (64 bits) en el menú desplegable
- Allocate RAM as per your usage. 2048 MB is the recommended memory size.
- Select “Create a virtual hard disk now” as we are installing Ubuntu on Virtual Box for first time.
- Seleccione "VDI (imagen de disco de Virtual Box)" como el tipo para su archivo de disco duro virtual.
- Seleccione "Asignación dinámica" ya que no queremos mantener la restricción en el tamaño del archivo de disco duro virtual.

- Ingrese el nombre del archivo de Disco Duro Virtual. (también podemos dejarlo como nombre predeterminado igual que el nombre de VM).
- Su máquina virtual ahora se creará con la configuración anterior. (Visible en el panel izquierdo de Virtual Box).
- Seleccione la máquina virtual y haga clic en "Iniciar".

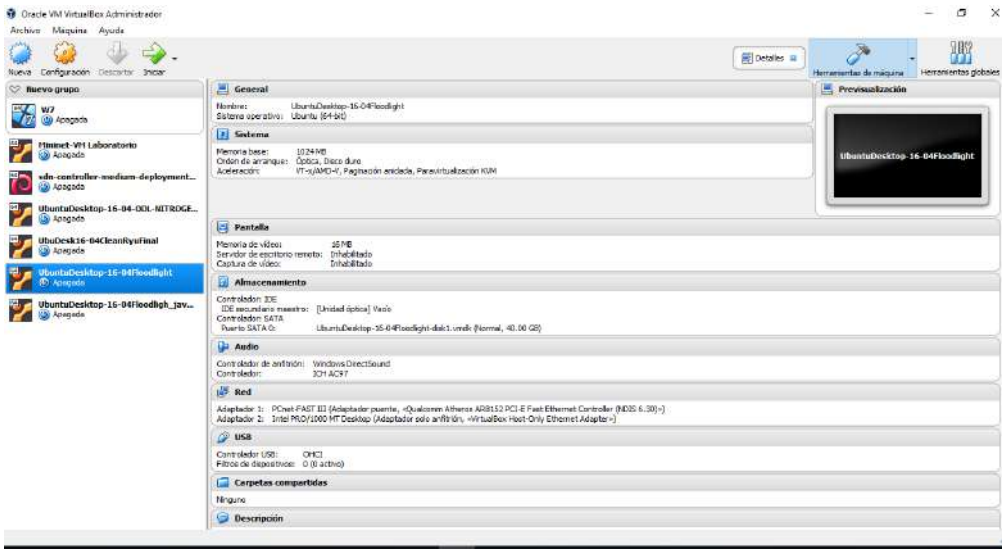


Figure 1 Ubuntu 16.04 en VirtualBox

- Cuando la instalación esté completa, agregue el adaptador puente como una interfaz de red, para ello, seleccione su máquina virtual y vaya a la pestaña Configuración. Vaya a Red-> Adaptador 1.
- Seleccione el cuadro "Habilitar adaptador" y conéctelo a "adaptador puente", como se visualiza en la siguiente pantalla.

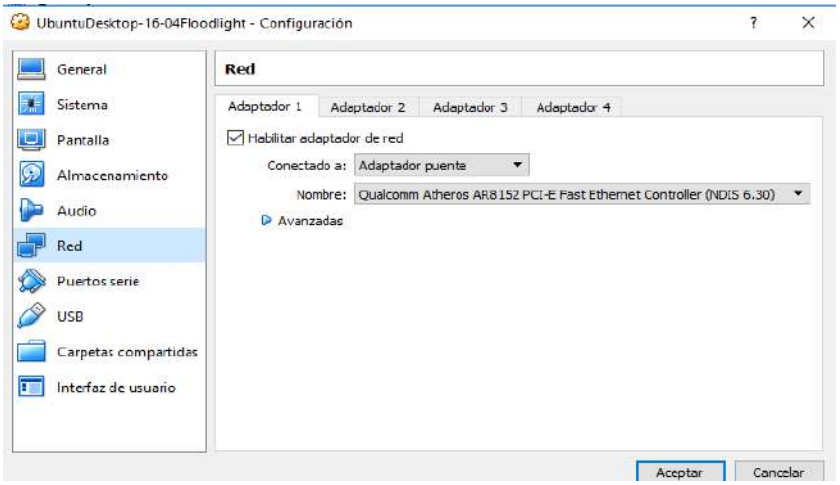


Figure 2 Configuración de la Tarjeta de Red en Ubuntu 16.04

- Configure la interfaz de red como estática (edite / etc / network / interfaces)

Interfaz de red

auto eth1

iface eth1 inet estático

dirección 192.168.2.106

máscara de red 255.255.255.0

red 192.168.2.0

3. Instalar JAVA jre o jdk

- sudo apt-get install openjdk-7-jre
- Establecer JAVA_HOME. Agregar al final de / etc / profile
- exportar JAVA_HOME = / usr / lib / jvm / java-7-openjdk-amd64 / jre

4. Instalación de Mininet

- Descargar la imagen de Mininet desde /github.com/mininet/mininet/wiki/Mininet-VMImages 3.- Doble click en archivo. ovf, esto importará y creará la imagen con las configuraciones necesarias (asignará RAM, configuraciones por defecto, etc.).
- Correr Mininet desde la VM. Se le solicitarán los siguientes user y pass: *User: mininet *Pass: mininet

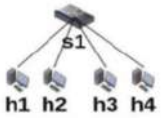
Comandos básicos en Mininet

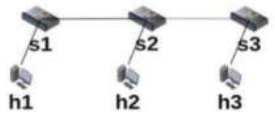
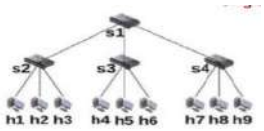
La siguiente guía incluye comandos básicos, los cuales fueron tomados de la página mininet.org/walkthrough/.

1.- Línea de comandos: El signo '$': Precede comandos de Linux (como cd, ls, etc) mininet>. Para comenzar, abra la terminal y escriba lo siguiente:

$ sudo mn -h

Esto entregará una lista con los comandos más típicos de Mininet, generalmente se ejecutan con el comando mn. Para diseñar una red básica, puede simplemente introducir el comando sin ningún parámetro. Esto creará una red simple con un switch s1 y dos hosts h1 y h2. Puede crear otras topologías usando – topo=TOPOLOGIA. Algunas topologías admitidas son:

--topo simple, 	La topología simple consiste en un switch y n hosts conectados a dicho switch

–-topo linear, 	Linear crea x switchs con y hosts conectados a cada switch. En este caso, todos los switchs están conectados entre sí.
--topo tree, 	Crea una topología de árbol, con una profundidad d y número de ramas f. Puede revisar los demás comandos viendo la ayuda (mn –h).

En la terminal de Mininet, puede probar los siguientes comandos:

```
mininet> help
mininet> nodes
mininet> dump
```

Estos comandos muestran la ayuda, además de la información de los nodos, la red. Además, puede usar el comando ifconfig para obtener información de la interfaz de red de un host, de la misma forma que un equipo real:

```
mininet> h1 ifconfig -a
```

Para el caso de especificar el controlador, Mininet utiliza el controlador que tiene incorporado por defecto. El parámetro que permite seleccionar un controlador distinto es –controller, usando la siguiente sintaxis:

```
$ sudo mn --controller remote
```

La sintaxis general del parámetro --controller es:

```
--controller remote, ip=[controller IP], port=[controller port]
```

Algunos de los comandos que podemos ejecutar desde la consola de Mininet son:

- mininet> net // muestra información sobre la red
- mininet> h1 ping –c1 h2 // manda un ping desde el Host 1 (h1) al Host 2 (h2)
- mininet> h1 ifconfig // muestra información sobre los interfaces de h1
- mininet> exit // cierra la consola

Cabe recalcar que para que se pueda ejecutar comandos con permisos de root se debe teclear dichos comandos anteponiendo sudo.

Una vez que se cierre Mininet se recomienda limpiar la topología de la red previa, ejecutando:

```
$ sudo mn –c
```

5. Wireshark en Mininet

- Después que esté instalado Mininet, entonces podemos comenzar Wireshark primero con usuario root, escribir el siguiente comando para la instalación:

```
sudo apt install wireshark -qt
```

- Para ejecutar Wireshark, utilice el comando:

```
sudo wireshark &
```

- Para trabajar con Wireshark desde Windows, se debe ejecutarse con la aplicación X terminal, con eso desde Windows se ejecuta Wireshark y desde VM se ejecuta Mininet.

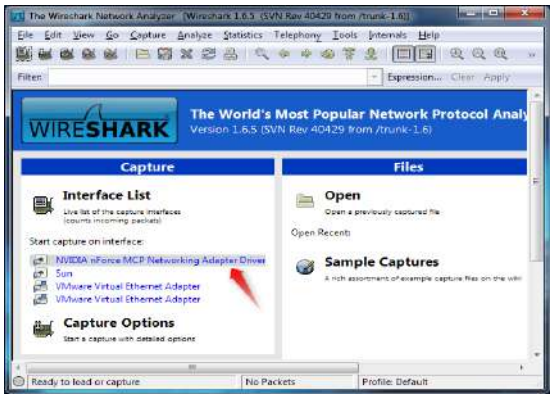


Figure 3 Interfaz Wireshark
Fuente: Autor

El hecho de que los hosts creados sean emulados implica que los paquetes que son enviados y recibidos son reales. Esto permite, por ejemplo, la capacidad de visualizar dichos paquetes usando Wireshark.

- Usando la misma red creada, pruebe a ver el nombre de la interfaz del switch s1. Luego, abra Wireshark y comience monitorear el tráfico de alguna de las interfaces asociadas a los hosts (s1-eth) y ejecute el siguiente comando:

```
mininet> h1 ping -c 4 h2
```

El comando enviará cuatro pings desde el host h1 al host h2. Podrá visualizar en Wireshark todo el tráfico asociado al ping (requerimientos ARP y paquetes de request y reply) como si se tratara de una red real.

5.1 API de Python en Mininet

Para diseñar topologías más complejas y personalizadas, es recomendable usar la API de Python. Un ejemplo de esto se puede ver en la carpeta /mininet/custom/topo-2sw-2host.py el cual muestra cómo definir una topología de 2 switches y 2 hosts. Mininet define cada topología como una clase que se puede instanciar como objeto usando Python. Cada clase crea sus nodos y enlaces usando métodos, para los cuales se pueden definir distintos parámetros. Los métodos más importantes para la creación de topología son addHost, addLink, y addSwitch. addHost(): Crea un host con un nombre determinado por addSwitch(): Crea un switch con un nombre determinado por Nota: addHost y addSwitch no aceptan nombres que no sean de la forma h1,h2,h3… o s1,s2,s3… etc. addLink(, , bw=, delay=,loss=): Crea un enlace entre los nodos n1 y n2, con los parámetros opcionales BW(Ancho de banda), delay, y porcentaje de pérdidas L. También admite otros parámetros como jitter, latencia, etc.

```
self.addLink( host, switch, bw=10, delay='5ms',loss=1)
```

Existen varias formas de implementar una topología a partir de un script. Una de las formas más simples se ejemplifica en el archivo adjunto test_net.py, ya que basta con modificar los métodos que crean los nodos y enlaces según se desea. Luego, para correr la red en mininet basta con ejecutar el script. También puede intentar implementarlo usando el siguiente ejemplo de mininet como base:

```
$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo
```

5.2 Monte un servidor http en el host h1

Luego, haga un requerimiento GET al servidor desde el host h2. Para ello, use los siguientes comandos.

```
mininet> h1 python -m SimpleHTTPServer 80 &
mininet> h2 wget -O - h1
```

5.3 Mininet y conexión con el protocolo NAT

Ingrese a la carpeta /mininet/examples/ y ejecute el script nat.py. Este archivo crea una red con topología tree con conexión a internet usando el protocolo NAT. Compruebe que los hosts efectivamente tienen conexión a internet haciendo un ping a alguna página conocida. Use el DNS (e.g. www.unapagina.cl) en vez de la IP.

5.4 Mininet y protocolo FTP

- Cree un archivo de 10 Mb usando

```
fallocate –l 10MiB un-archivo.png
```

- Instale un servidor FTP en Mininet usando

```
sudo apt-get install ftpd
```

- Inicie simultáneamente las terminales xterm para los host h1 y h2.
- Inicie un servidor FTP en el host h2 usando inetd.
- Inicie una captura tcpdump (a un archivo) para paquetes entre h1 y h2 en el puerto 21. e.g:

```
tcpdump –w ftp-transfer host 10.0.0.1 and host 10.0.0.2 and port 21
```

- 6.- En el host h1, cambie el directorio a /tmp, use "cd /tmp". Luego, inicie una sesión ftp usando el comando ftp 10.0.0.2 en una terminal. Ingrese usando Name: mininet y Password: mininet.
-
- 7.- Obtenga el archivo creado usando get (e.g. get un-archivo.png)

6. FLOODLIGHT

6.1 Instalación

Descargar el proyecto del controlador Floodlight de la página web oficial http://www.projectfloodlight.org/download/

o con el comando:

```
$ wget https://github.com/floodlight/floodlight/archive/v1.2.zip
```

Figure 4 Descarga controlador Floodlight.

Fuente: Autor

- Descomprimir el proyecto con el comando:

```
$ unzip v1.2.zip
```

Figure 5 Descomprimir controlador

- Una vez que se ha descomprimido el proyecto se genera una carpeta floodlight-1.2. Al ingresar a ésta carpeta se encuentra el archivo para ejecutar el controlador.

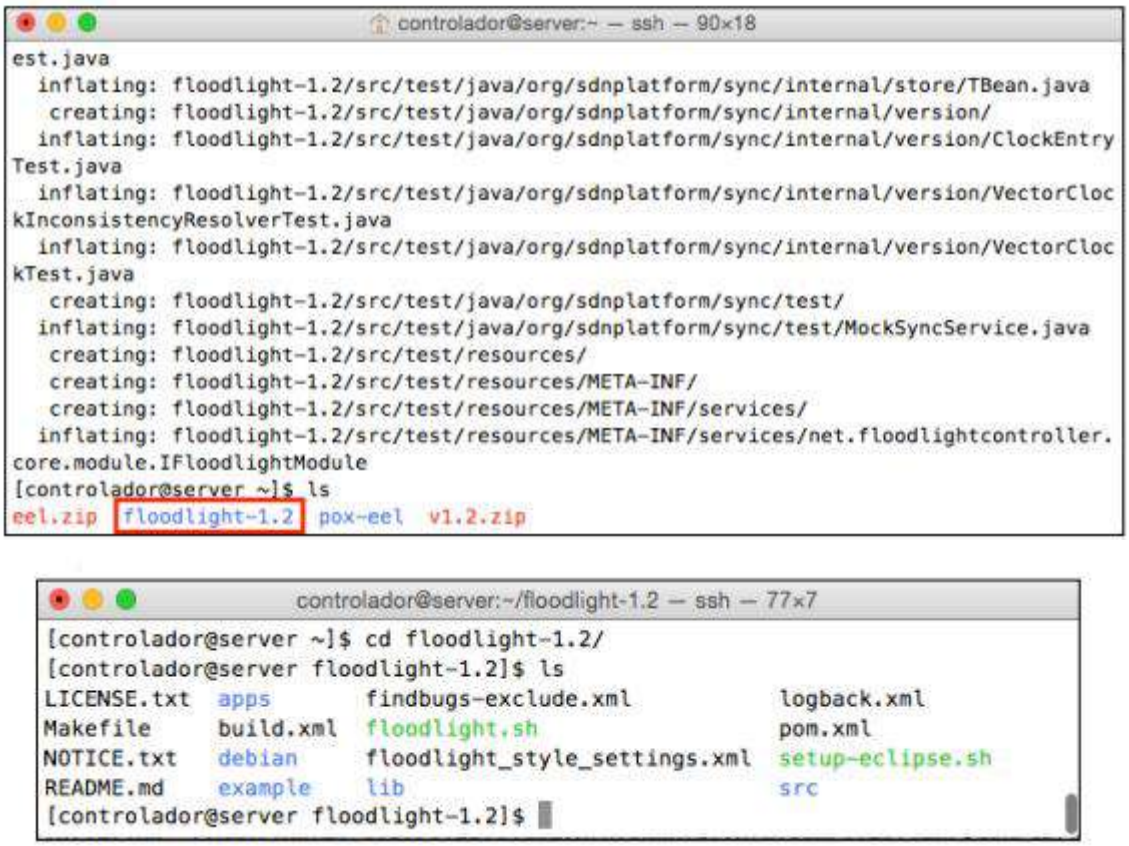
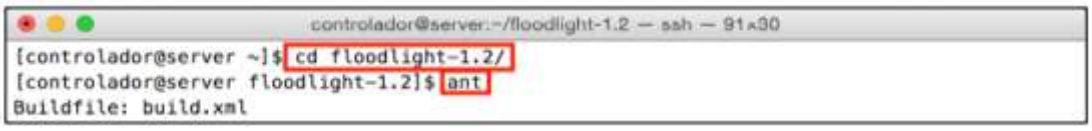


Figure 6 Carpeta floodlight-1.2 del controlador instalada

Figura A6.

- Para compilar el proyecto java de Floodligth se debe usar el programa ant, se lo descarga previamente y basta con ingresar a la carpeta del proyecto y ejecutar el comando:

```
$ ant
```



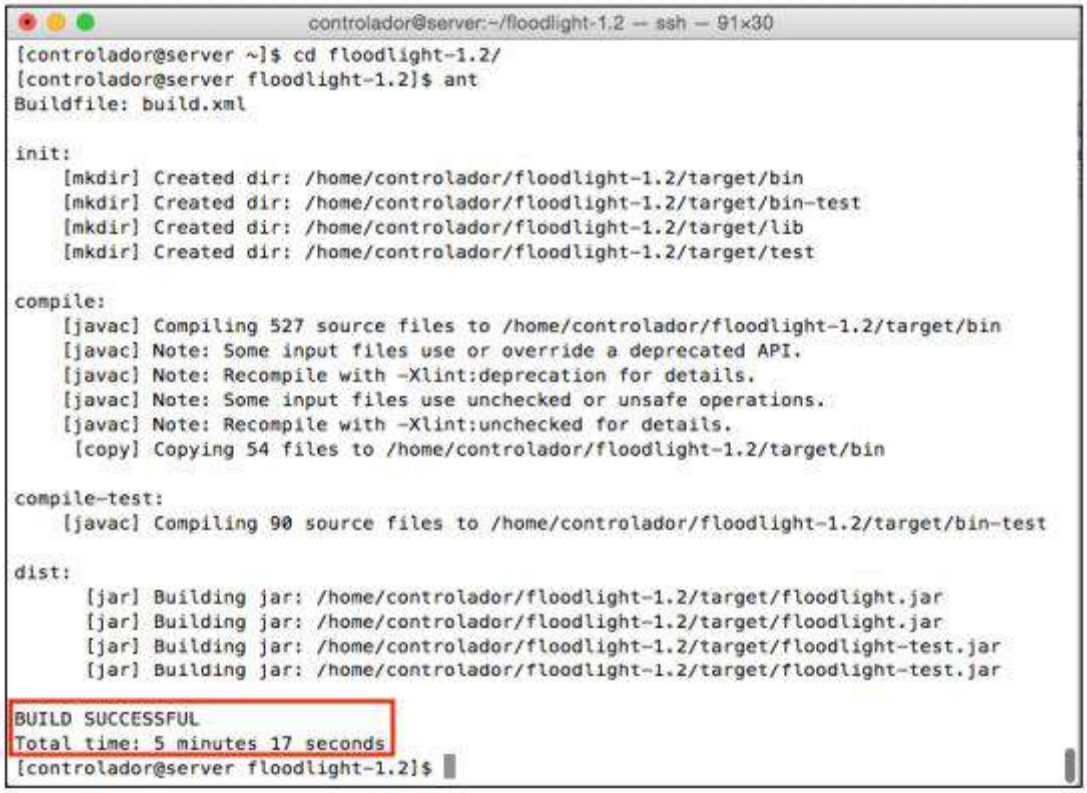


Figure 7 Compilar controlador Floodlight

- Al completar el proceso de compilación se crea la carpeta target.

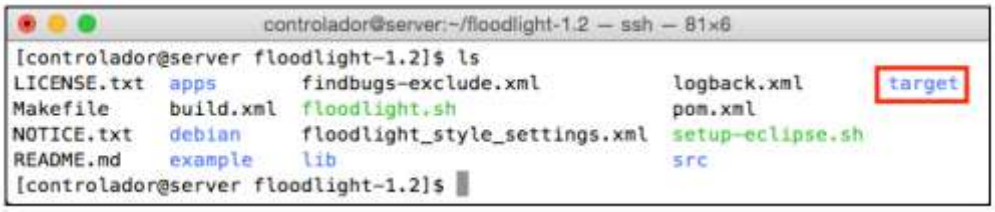
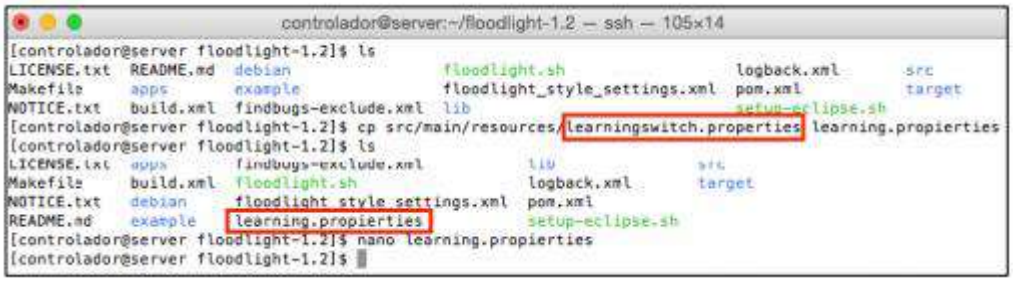


Figure 8 Carpeta target instalada

6.2 Propiedades y ejecución de los controladores

Floodlight: Editar el archivo de propiedades "learning.properties" del controlador Floodlight y asignar puertos diferentes a los que están por defecto.



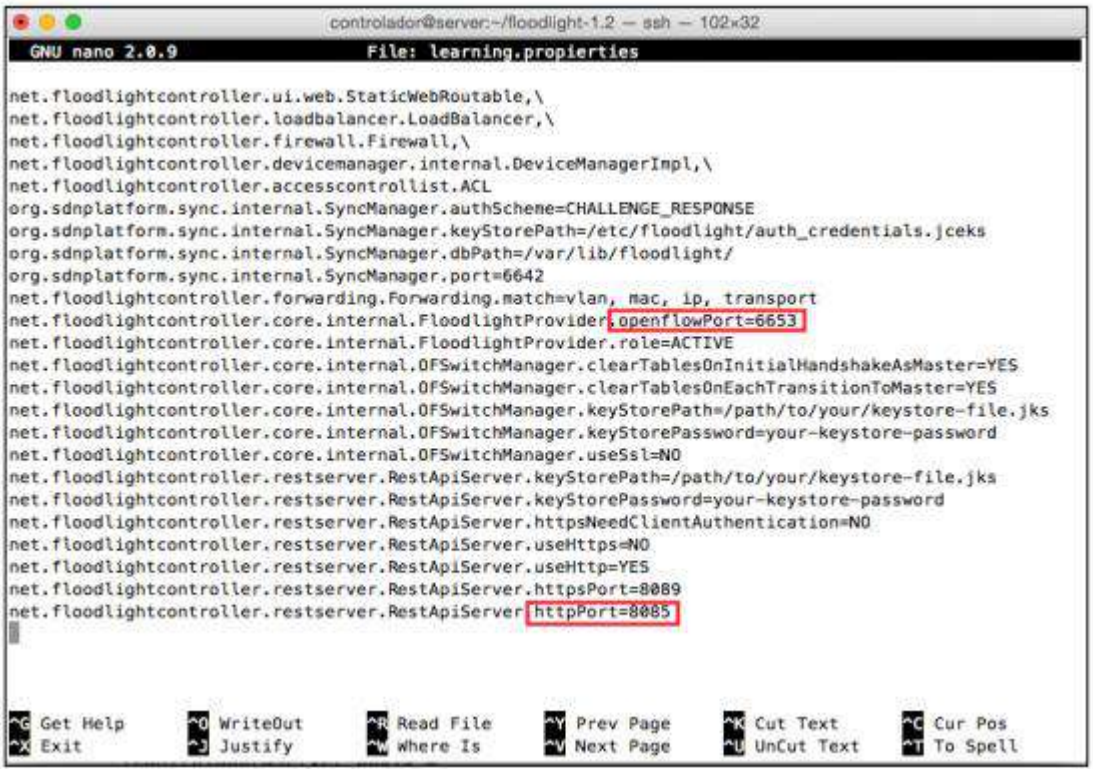


Figure 9 Edición y configuración de las propiedades del controlador

- Al finalizar los cambios, ya se puede ejecutar el controlador como se muestra en la siguiente Figura.

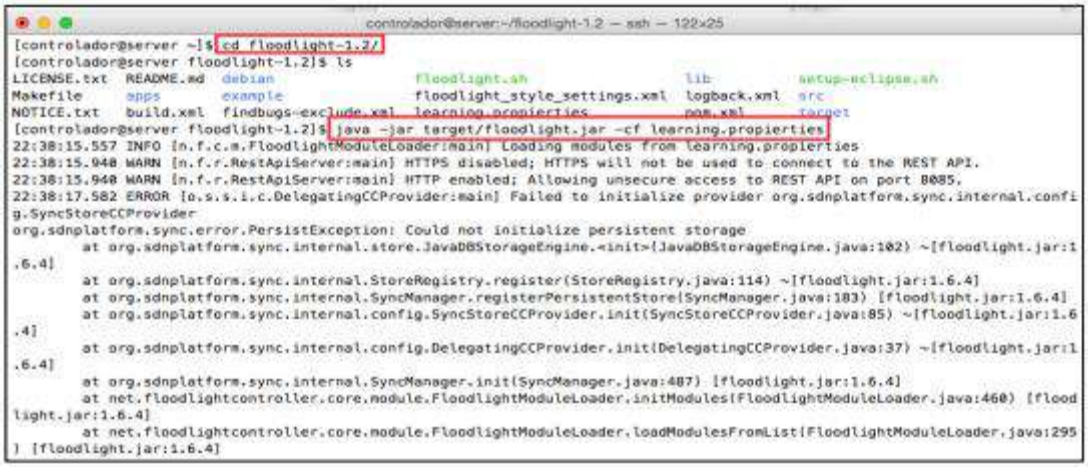


Figure 10 Ejecución del controlador Floodlight

6.3 Instalación CURL

CURL es una herramienta que permite hacer peticiones HTTP. CURL nos ayudará a hacer uso del API REST de las aplicaciones Ryu. A continuación, tienen un pequeño manual de instalación y uso:

Instalación:

```
$ sudo apt-get install curl
```

Uso de la opción GET:

```
$ curl http://localhost:8080
```

Uso de la opción PUT:

```
$ curl -X PUT http://localhost:8080/resource
```

Uso de la opción POST:

```
$ curl -X POST -d '<json>' http://localhost:8080/resource
```

7. Configuraciones ZodiacFX

- Conecte el cable USB entre el ZODIAC FX Y el computador. Verifique la conexión con: dmesg

```
sudo dmesg

[3010068.043396] usb 1-4: new full-speed USB device number 10 using xhci_hcd
[3010068.185334] usb 1-4: New USB device found, idVendor=03eb, idProduct=2404
[3010068.185345] usb 1-4: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[3010068.185352] usb 1-4: Product: Zodiac
[3010068.185357] usb 1-4: Manufacturer: Northbound Networks
[3010068.210327] cdc_acm 1-4:1.0: ttyACM0: USB ACM device
[3010068.212786] usbcore: registered new interface driver cdc_acm
[3010068.212796] cdc_acm: USB Abstract Control Model driver for USB modems and ISDN adapters
```

Como puede ver al conectar el cable usb que se creó la interfaz ttyACM0, la conexión se la realiza a través de minicom

- Instalación y configuración minicom

```
patrick@stretch:~/.ssh$ sudo apt-get install minicom
patrick@stretch:~/.ssh$ sudo usermod --append --groups dialout $USER
patrick@stretch:~/.ssh$ sudo minicom --device /dev/ttyACM0
Welcome to minicom 2.7
```

- Esta es la interfaz y configuración por defecto:

```
OPTIONS: I18n
Compiled on Apr 22 2017, 09:14:19.
```

```
Port /dev/ttyACM0, 13:24:57

Press CTRL-A Z for help on special keys
Zodiac_FX#
 _____          ___          ________  __
/__  /  ____  ____/ (_)___ _______   / ____/ |/ /
  / /  / __ \/ __  / / __ `/ ___/  / /_   |   /
 / /__/ /_/ / /_/ / / /_/ / /__   / __/  /   |
/_____/\____/\__,_/_/\__,_/\___/  /_/    /_/|_|
          by Northbound Networks
```

```
Zodiac_FX(config)# show config

-------------------------------------------------------------------------
Configuration
 Name: Zodiac_FX
 MAC Address: 70:B3:D5:XX:XX:XX
 IP Address: 10.0.1.99
 Netmask: 255.255.255.0
 Gateway: 10.0.1.1
 OpenFlow Controller: 10.0.1.8
 OpenFlow Port: 6633
 Openflow Status: Enabled
 Failstate: Secure
 Force OpenFlow version: Disabled
 EtherType Filtering: Disabled

-------------------------------------------------------------------------
```

- Configuraciones Básicas: Configuración de la dirección IP, máscara de subred y puerta de enlace.

```
Zodiac_FX# show ports
Zodiac_FX#config
Zodiac_FX(config)# set ip-address 192.168.2.204
 IP Address set to 192.168.2.204
# Zodiac_FX(config)# set netmask 255.255.255.0
Zodiac_FX(config)# set gateway 192.168.2.1
 Gateway set to 192.168.2.1

Zodiac_FX(config)# show config

-------------------------------------------------------------------------
Configuration
 Name: Zodiac_FX
 MAC Address: 70:B3:D5:XX:XX:XX
 IP Address: 192.168.2.2
 Netmask: 255.255.255.0
 Gateway: 192.168.2.1
 OpenFlow Controller: 10.0.1.8
```

```
OpenFlow Port: 6633
Openflow Status: Enabled
Failstate: Secure
Force OpenFlow version: Disabled
EtherType Filtering: Disabled
-------------------------------------------------------------------------
Zodiac_FX(config)# save
Writing Configuration to EEPROM (197 bytes)
```

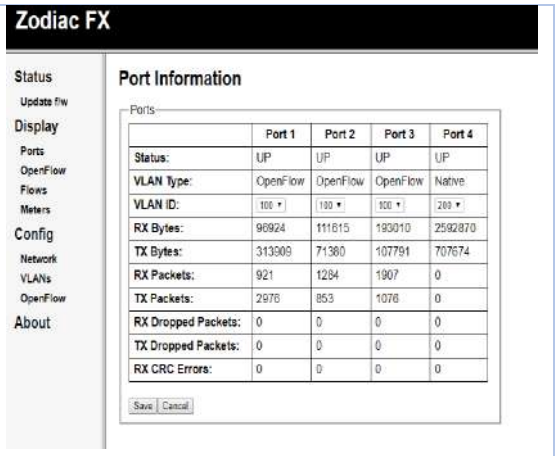
- Listado de los comandos disponibles para ZODIAC FX:

```
Zodiac_FX#
```

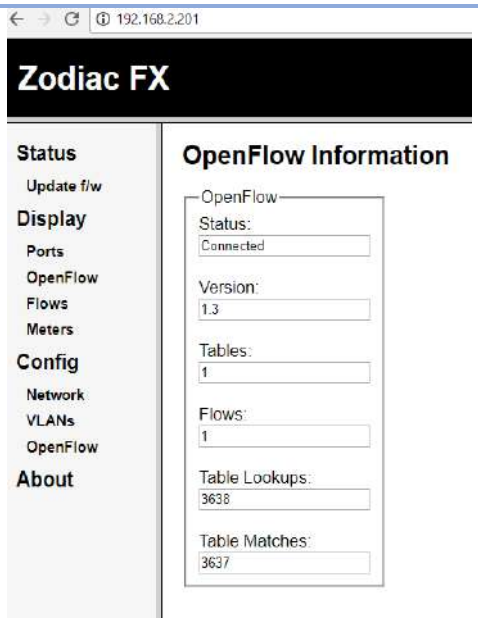
Base:	Config:	OpenFlow:	Debug:
config openflow debug update show status show version show ports restart help	save restart show config show vlans set name <name> set mac-address <mac address> set ip-address <ip address> set netmask <netmasks> set gateway <gateway ip address> set of-controller <openflow controller ip address> set of-port <openflow controller tcp port> set failstate <secure|safe> add vlan <vlan id> <vlan name> delete vlan <vlan id> set vlan-type <vlan id> <openflow|native> add vlan-port <vlan id> <port> delete vlan-port <port> set of-version <version(0|1|4)> set ethertype-filter <enable|disable> factory reset	show status show tables show flows show meters enable disable clear flows exit	read <register> write <register> <value> mem trace exit

- Capturas de las Configuraciones del Zodiac FX

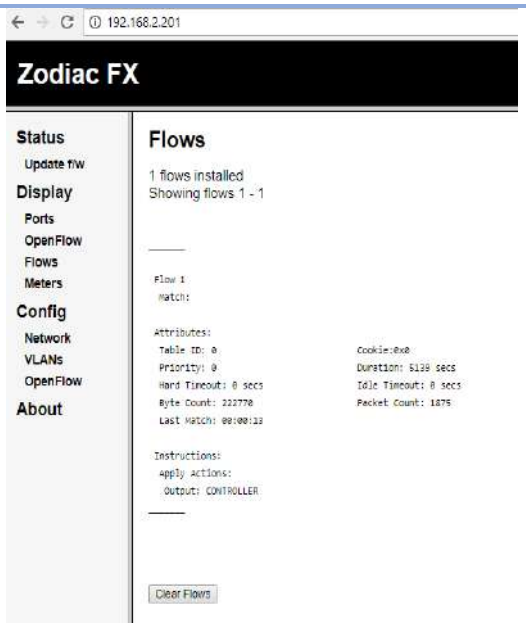
Interrfaz Web del Zodiac FX



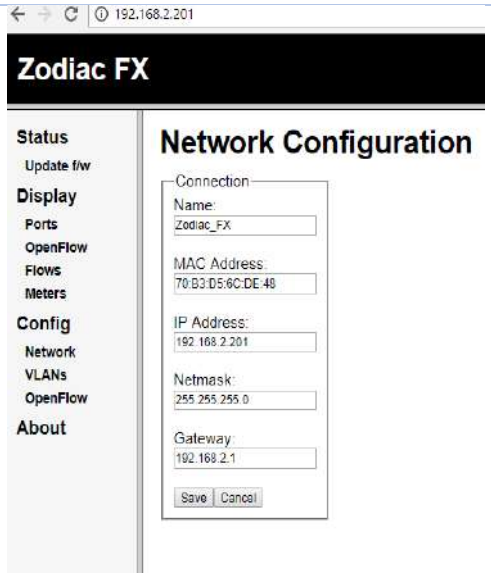
Información de los Puertos del Zodiac



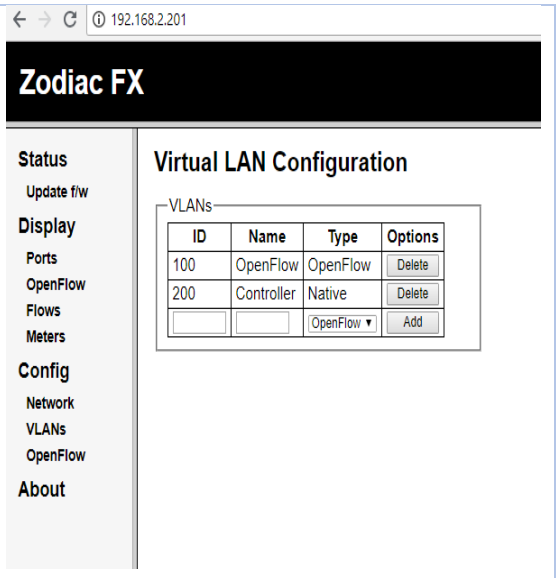
Información del Protocolo OpenFlow



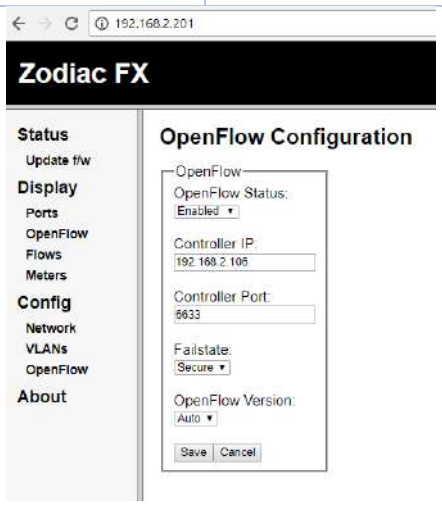
Pantalla de Flujos del Zodiac FX



Configuración de la Red



Configuración de VLANs



Configuración del Protocolo OpenFlow

8. HPE 3800 Series switch OpenFlow setup

- Conéctese al equipo por medio del cable de consola o mediante Puerto SSH
- Activación del Protocolo OpenFlow en equipo HP

```
# config

(config) # openflow
(openflow) # controller-id 1 ip [controller IP #1] controller-interface v
lan 1
(openflow) # controller-id 2 ip [controller IP #2] controller-interface v
lan 1
```

(openflow) # controller-id 3 ip [controller IP #3] controller-interface v lan 1

In the case of HP, it only allows users to add three controller IPs
(openflow) # instance oflow
(of-inst-oflow) # member vlan 10
(of-inst-oflow) # controller-id 1
(of-inst-oflow) # controller-id 2
(of-inst-oflow) # controller-id 3
(of-inst-oflow) # version 1.0 (or 1.3) -> OpenFlow 1.3 is not working well
(of-inst-oflow) # enable
(of-inst-oflow) # exit
(openflow) # enable

Check the configuration of OpenFlow

(openflow) # show openflow controllers
(openflow) # show openflow instance oflow

- **Para configuración de Políticas de QoS:**

  Pasos generales para implementar ACL

  Configure una o más ACL.
  Esto crea y almacena la (s) ACL (s) en la configuración del Switch HP3800.

- Asignar una ACL: Este paso utiliza una de las siguientes aplicaciones para asignar la ACL a una interfaz:

  - VACL: cualquier tráfico IPv4 que ingresa al conmutador en una VLAN determinada

  - Static Port ACL: cualquier tráfico IPv4 que ingrese al switch en un puerto, lista de puertos o troncal estática

- Opciones para las políticas de permiso / denegación

  - Dirección Origen IPv4
  - Dirección IPv4 de destino
  - Opciones de protocolo IPv4:
  - Cualquier tráfico IPv4
  - Cualquier tráfico de un tipo de protocolo IPv4 específico (0-255)

- Cualquier tráfico TCP (solo) para un puerto TCP específico o rango de puertos, incluido el uso opcional de bits de control TCP o el control del tráfico de conexión (establecido) en función de si la solicitud inicial debe permitirse.

- Cualquier tráfico UDP (solo) o tráfico UDP para un puerto UDP específico

- Cualquier tráfico ICMP (solo) o tráfico ICMP de un tipo y código específico

- Cualquier tráfico IGMP (solo) o tráfico IGMP de un tipo específico

- Cualquiera de los anteriores con precedencia específica y / o configuración de ToS (se aplica solo a las series HP Switch 2620 y 2920)

- Para una ID de ACL extendida, use un número único en el rango de 100-199 o una cadena de nombre único de hasta 64 caracteres alfanuméricos.

- Planifique con cuidado las aplicaciones de ACL antes de configurar ACL específicas.

- Estructura de configuración ACL

  Después de ingresar un comando ACL, es posible que desee inspeccionar la configuración resultante. Esto es especialmente cierto cuando ingresa múltiples ACE en una ACL. Además, es útil comprender la estructura de configuración cuando se utiliza la siguiente información.

  La estructura básica de ACL incluye cuatro elementos:

  Identidad y tipo de ACL: identifica la ACL como estándar o extendida y muestra el nombre o número de la ACL.

  Entradas de comentarios opcionales.

  Una o más entradas de lista de denegación / permiso (ACE): una entrada por línea.

Elemento	Notas
Tipo	Estándar o Extendido
Identificador	• Alfanumérico; Hasta 64 caracteres, incluidos espacios • Numérico: 1-99 (Estándar) o 100-199 (Extendido)
Observación	Permite hasta 100 caracteres alfanuméricos, incluidos espacios en blanco. (Si se utilizan espacios, la observación debe estar encerrada en un par de comillas simples o dobles.) Una observación se asocia con un ACE particular y tendrá el mismo número de secuencia que el ACE. (Se permite un comentario por ACE.)

Máximos ACE por conmutador	El límite superior de las ACE admitidas por el conmutador depende del uso simultáneo de recursos por ACL, QoS, IDM, Mirroring y otras características configuradas.

Denegación implícita: cuando una ACL está en uso, niega cualquier paquete que no tenga una coincidencia con las ACE configuradas explícitamente en la lista. La denegación implícita no aparece en los listados de configuración de ACL, pero siempre funciona cuando el switch usa una ACL para filtrar paquetes. (No puede eliminar la denegación implícita, pero no puede eliminarla con un permiso ni permitir ninguna declaración).

Estructura ACL estándar

Las ACE individuales en una ACL estándar incluyen solo una declaración de permiso / denegación, el direccionamiento de origen y un comando de registro opcional (disponible con declaraciones de "denegar" o "permitir").

Estructura de configuración de ACL extendida

ACE individuales en una ACL extendida incluyen:

- Una declaración de permiso / denegación
- Dirección IPv4 de origen y destino
- Elección de los criterios de IPv4, incluida la precedencia opcional y ToS
- Comando de registro ACL opcional (para denegar o permitir entradas)
- Declaraciones de comentarios opcionales

Comandos CLI para crear una ACL

Puede usar la CLI del conmutador o un editor de texto sin conexión para crear una ACL. Esta sección describe el método CLI, que se recomienda para crear ACL cortas.

Insertar o agregar un ACE a una ACL
Estas reglas se aplican a todas las ACE de IPv4 que crea o edita utilizando la CLI:

Nombrado ACLs de IPv4: Agregue una ACE al final de una ACE nombrada usando el comando ip access-list para ingresar el contexto de ACL con nombre (nacl) e ingresar la ACE sin el número de secuencia.

Por ejemplo, si desea agregar una ACL de "permiso" al final de una lista llamada "Lista-1" para permitir el tráfico desde el dispositivo a las 10.10.10.100:

```
HP Switch(config)# ip access-list standard List-1
HP Switch(config-std-nacl)# permit host 10.10.10.100
```

Inserte un ACE en cualquier parte de una ACL nombrada especificando un número de secuencia. Por ejemplo, si desea insertar un nuevo ACE como línea 15 entre las líneas 10 y 20 en una ACL existente llamada "Lista-2" para denegar el tráfico de IPv4 desde el dispositivo en 10.10.10.77:

```
HP Switch(config)# ip access-list standard List-2
HP Switch(config-std-nacl)# 15 deny host 10.10.10.77
```

ACL de IPv4 numeradas: agregue una ACE al final de una ACL numerada mediante el comando access-list <1-99 | 100-199>. Por ejemplo, si desea agregar un ACE "permiso" al final de una lista identificada con el número "11" para permitir el tráfico IPv4 desde el dispositivo a las 10.10.10.100:

```
HP Switch(config)# access-list 11 permit host 10.10.10.100
```

Para insertar un ACE en cualquier parte de una ACL numerada, utilice el mismo proceso descrito anteriormente para insertar un ACE en cualquier parte de una ACL nombrada. Por ejemplo, para insertar un tráfico de rechazo de ACE IPv4 desde el host en 10.10.10.77 como línea 52 en una ACL existente identificada (nombrada) con el número 11:

```
HP Switch(config)# ip access-list standard 99
HP Switch(config-std-nacl)# 52 deny host 10.10.10.77
```

CONCLUSIONES

- El estudio de la aplicación de políticas de calidad en redes Enterprise basado en la experiencia y el "estado del arte", incluyendo las recomendaciones de organismos internacionales como ITU-T, ETSI y TIA, son de gran importancia para implementar soluciones que permitan mejorar el rendimiento de la red, sin embargo los administradores de redes son los que constantemente gestionan, monitorean y detectan fallas en su trabajo, siendo capaces de aportar con alternativas para establecer políticas de QoS, por lo que se ha considerado sus observaciones en el presente trabajo.

- Para diseñar una red SDN se realizó un análisis de equipos físicos disponibles, el costo que representa adquirirlos, las características, librerías, licencias, limitaciones de fabricantes y la posibilidad de realizar pruebas para comparar escenarios virtuales como reales; cabe mencionar que la implementación de los escenarios propuestos con sus respectivas configuraciones para la calidad de servicio, implicó limitaciones en el funcionamiento de los controladores, relacionados con las versiones existentes del protocolo OpenFlow, esto es importante al interactuar con equipos reales, por problema de compatibilidad; se utilizó OpenFlow en la versión 1.3 para los escenarios de estudio.

- A pesar que los resultados no corroboran alguna mejora al aplicar SDN. Esta tecnología presenta un gran potencial debido a la flexibilidad y personalización en sus funcionalidades al gestionar la red; en base al análisis estadístico se concluye que el comportamiento del rendimiento en una red enterprise con políticas de QoS en entornos de redes convencionales y SDN presentan una diferencia significativa tanto en el servidor como en el cliente en los parámetros latencia y pérdida de paquetes; mientras que en los parámetros jitter y ancho de banda no existe una diferencia significativa.

- En los escenarios propuestos se observó un mejor rendimiento de la red al asignar políticas de QoS en un ambiente convencional a nivel de servidor y cliente en los parámetros: latencia, ancho de banda y pérdida de paquetes; en lo que respecta al parámetro jitter en el cliente se detectó una mejora de rendimiento en un entorno SDN con HP y en base a los resultados en pruebas se determinó que el hardware Hp SDN es superior al Zodiac SDN, esto influyó en el rendimiento al aplicar políticas de calidad de servicio (QoS).

RECOMENDACIONES.

- Como recomendación al momento de realizar el análisis de las Políticas de calidad, es importante enfocarse con el entorno real que se va a trabajar, y realizar un estudio previo sobre el tráfico de la red, en diferentes situaciones, por ejemplo, en días laborables y en los días de vacaciones con el fin de detectar posibles conflictos existentes y a partir de ellos, investigar cuales serían los parámetros de QoS a ser tomados en cuenta.

- Es importante conocer las características de los diferentes tipos de controladores, compatibilidad y las funciones que éstos ofrecen para seleccionar adecuadamente el que se va emplear en una implementación de red.

- A razón gque SDN con el protocolo OpenFlow es un campo relativamente nuevo, para entender de mejor manera se recomienda hacer pruebas cambiando de controladores y experimentando diferentes flujos de paquetes, inclusive programando reglas para determinar su comportamiento, con la finalidad de familiarizarse con esta tecnología.

- Por medio de este proyecto sería interesante propagar la tecnología SDN para que los operadores de red puedan dar soluciones en menor tiempo a problemas relacionados con la conmutación de paquetes, enrutamiento, entre otros; utilizando las herramientas adecuadas, con criterios más claros, fundamentados en el análisis realizado en este documento para tomar decisiones certeras.

BIBLIOGRAFÍA

Araú, M. R. A. G. de. (2013). *Uma abordagem para aprovisionamento de qos em redes definidas por software baseadas em openflow*. Universidade Federal de Pernambuco. Retrieved from http://www.cin.ufpe.br/~tg/2013-1/mraga.pdf

Calderón, F. M. M. (2016). *Desarrollo de un prototipo de red definida por software sdn para la gestión median te recursos de estándar abierto*. Pontificia Universidad Católica del Ecuador.Retrieved from http://repositorio.pucesa.edu.ec/bitstream/ 123456789/1638/1 /76160.pdf

Cardoso, R. F. F. (2015). *Uma abordagem sdn para o controlo e admissão de tráfego*. Universidade Nova de Lisboa. Retrieved from https://run.unl.pt/bitstream/10362/16 557/1/Cardoso_2015 .pdf

Cisco. (2008). C*isco telepresence network systems 1.1 design guide*. Retrieved from https://www.cisco.com/c/dam/en/us/td/docs/solutions/TelePresence_Network_Systems_1-1_DG.pdf

Cisco. (2017). *Cisco visual networking index: global mobile data traffic forecast update, 2011–2016; [Visual Networking Index (VNI)]. cisco, 2016–2021*. https://doi.org/10.1109/ SURV.2008.080403

Fanelli, M. (2016). *Iot y la evolución en las redes empresariales*. Retrieved from http://www.itendencias.net/noticias/709/iot-y-la-evolucion-en-las-redes-empresariales. html

GitHub. (2017a). *Pure python library for handling libpcap savefiles.* Retrieved from https://github.com/wirkentod/pypcapfile

GitHub. (2017b). *Trace traffic analizer.* Retrieved from https://github.com/wirkentod/Trace-Traffic-Analizer

Ǵomez, E. F. M. G. (2016). *Desarrollo de un esquema de enrutamiento din ́amico basado en matrices de tr ́afico para garantizar calidad de servicio en redes definidas por software usando el protocolo openflow*. Universidad de Antoquia. Retrieved from Emanuel Fernando Montoya Gómez

ITU-T. (2011). End-user multimedia qos categories. Retrieved from https://www.itu.int/rec/ dologin_pub.asp?lang=s&id=T-REC-G.1010-200111-I!!PDF-E&type=items

Kang, N., Rottenstreich, O., Rao, S., & Rexford, J. (2015). *Alpaca: compact network policies with attribute-carrying addresses*. Retrieved from https://www.cs.princeton.edu /~jrex/papers/alpaca. pdf

Kirk, R. E. (1995). *Experimental design: procedures for the behavioral sciences*. Retrieved from

http://tesis-investigacion-cientifica.blogspot.com/2013/08/.html

Lind, M. y W. (2012). *Estadistica aplicada a los negocios y la economia*. McGraw-Hill/Interamericana Editores, s.a. de c.v.

López, M. S. (2015). *Análisis de redes sdn utilizando mininet e implementación de un deep packet inspector*. Universidad de nueva granada. Retrieved from http://dtstc.ugr.es/it/pfc/proyectos_realizados/downloads/Memoria2015_ManuelSanchez.pdf

Mack-Crane, B. (2016). Onf data path. Retrieved from https://www.opennetworking.org/technical-communities/areas/specification/open-datapath/

Mallick, A. (2012). *Traditional network infrastructure model and problems associated with it.* Retrieved from https://www.pluribusnetworks.com/blog/traditional-network-infrastructure-model-and-problems-associated-with-it/

Menon, S. (2014). *Avaya software defined data center*. tech field day. Retrieved from http://techfieldday.com/video/avaya-software-defined-data-center/

Navarro, D. F. (2005). *Controlador de ancho de banda*. Universidad de Mendoza. Retrieved from http://www.um.edu.ar/es/imagenes-contenido/UM-MTI-NavarroD.pdf

Nick Feamster, Jennifer Rexford, E. Z. (2013). *The road to sdn.*

Nikolaos, K. (2017). *Software defined networks reactive flow programming and load balance switchin*. university of piraeus. Retrieved from http://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/10680/Kallianiotis_Nikolaos.pdf?sequence=1&isAllowed=y

ONF. (2015). *Openflow switch specification.* Retrieved from https://3vf60mmveq1g8vzn48q2o71a-wpengine.netdna-ssl.com/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

Osaba, M. N. (2016). *Virtualización en redes definidas por software*. Instituto Tecnológico de Buenos Aires. Retrieved from https://ri.itba.edu.ar/bitstream/handle/123456789/785/TELCO-M. Osaba Virtualización_de_Redes_Def_por_SW_.pdf?sequence=1&isAllowed=y

Planas, A. L. (2016). *Configuración de un e ntorno de emulación que permit a el diseño , desarrollo y evaluación de software defined networks con calidad de servicio, (2003)*. Retrieved from https://upcommons.upc.edu/bitstream/handle/2117/81460/104342.pdf?sequence=1&isAllowed=y

Quisphe, C. (2017). *Diseño e implementación de un balanceador de carga para la optimización de los recursos de protección en una red enterprise mediante un banco de firewalls n:1 controlado vía Sdn*. Pontífica Universidad Católica del Perú.

R. Mosavi, F. Farabi, S. K. (2015). O*ptimal choice of random variables in d-itg traffic generating tool using evolutionary algorithms*. Retrieved from http://ijeee.iust.ac.ir/article-1-692-en.pdf

Rezende, P. H. A. (2016). *Extensões na arquitetura sdn para o provisionamento de qos através do monitoramento e uso de múltiplos caminhos*. Universidade Federal de Uberlândia. Retrieved from https://repositorio.ufu.br/bitstream/123456789/17550/1/Extensoes ArquiteturaSDN.pdf

Roberto, M. (2008). *Diseño e implementación de un modelo de calidad de servicio en la red del ipn*. Instituto Politécnico Nacional. Retrieved from https://tesis.ipn.mx/jspui/bitstream/ 123456789/17256/1/DiseÃ±o e implementaciÃ³n de un modelo de calidad de servicio en la red del ipn.pdf

Seaman, I. E. A. and C. A. (2007). *Likert scales and data analyses. Retrieved from http://asq.org/quality-progress/2007/07/statistics/likert-scales-and-data-analyses.html*

Silva, M. P. da. (2017). *Um modelo de gerenciamento da qualidade de experiência para a provisão de serviços cientes de contexto*. Universidade Federal de Santa Catarina. Retrieved from http://btd.egc.ufsc.br/wp-content/uploads/2018/05/Madalena-Pereira-da-Silva.pdf

Torre, R. Z. E. de la. (2017). *Aplicaciones de sdn/nfv en redes inalámbricas de área local*. Universidad Central "Marta Abreu" de Las Villas.

Triola, M. F. (2009). *Estadística* (10ma Ed.).

Turcios, R. A. S. (2015). *T-Student. Usos y abusos.* Retrieved from http://www.scielo.org.mx /scielo.php?script=sci_arttext&pid=S0188-21982015000100009

Valdivieso, A. Peral, A. Barona, L.García, L. (2014). Evolution and opportunities in the development iot applications. *Http://Journals.Sagepub.Com/Doi/Full/10.1155/ 2014/735142*.

Villaroel, R. de P. (2015). A*plicación de sdn en redes ópticas: análisis preliminar*. Universidad de Valladolid. Retrieved from https://uvadoc.uva.es/bitstream/10324/15101/1/TFG-G 1628.pdf

ANEXOS

ANEXO A: Búsqueda en SCOPUS – Políticas de QoS

Documents

Export Date: 26 Jul 2018

Search: ("QoS" OR "Quality of service") AND (POLICIES) AND ("SDN"

1) Fahmin, A., Lai, Y.-C., Hossain, M.S., Lin, Y.-D.

Performance modeling and comparison of NFV integrated with SDN: Under or aside?

(2018) Journal of Network and Computer Applications, 113, pp. 119-129.

DOI: 10.1016/j.jnca.2018.04.003

Document Type: Article

Source: Scopus

2) Xavier, G.P., Kantarci, B.

A survey on the communication and network enablers for cloud-based services: state of the art, challenges, and opportunities.

(2018) Annales des Telecommunications/Annals of Telecommunications, 73 (3-4), pp. 169-192.

DOI: 10.1007/s12243-018-0629-4

Document Type: Review

Source: Scopus

3) Yi, B., Wang, X., Li, K., Das, S.K., Huang, M.

A comprehensive survey of Network Function Virtualization.

(2018) Computer Networks, 133, pp. 212-262. Cited 3 times.

DOI: 10.1016/j.comnet.2018.01.021

Document Type: Review

Source: Scopus

4) Kang, N., Rottenstreich, O., Rao, S.G., Rexford, J.

Alpaca: Compact network policies with attribute-encoded addresses.

(2017) IEEE/ACM Transactions on Networking, 25 (3), art. no. 7855776, pp. 1846-1860.

DOI: 10.1109/TNET.2017.2657123

Document Type: Article

Source: Scopus

5) Alsmadi, I.M., Alazzam, I., Akour, M.

A systematic literature review on software-defined networking.

(2017) Studies in Computational Intelligence, 691, pp. 333-369. Cited 3 times.

DOI: 10.1007/978-3-319-44257-0_14

Document Type: Book Chapter

Source: Scopus

6) Alsmadi, I.M., Zarour, M.

Empirical evidences in software-defined network security: A systematic literature review

(2017) Studies in Computational Intelligence, 691, pp. 253-295.

DOI: 10.1007/978-3-319-44257-0_11

Document Type: Book Chapter

Source: Scopus

7) Thyagaturu, A.S., Mercian, A., McGarry, M.P., Reisslein, M., Kellerer, W.

Software Defined Optical Networks (SDONs): A Comprehensive Survey

(2016) IEEE Communications Surveys and Tutorials, 18 (4), art. no. 7503119, pp. 2738-2786. Cited
65 times.

DOI: 10.1109/COMST.2016.2586999

Document Type: Review

Source: Scopus

8) Glisic, S.

Advanced Wireless Networks: Technology and Business Models: Third Edition

(2016) Advanced Wireless Networks: Technology and Business Models: Third Edition, pp. 1-832.

Cited 3 times.

DOI: 10.1002/9781119096863

Document Type: Book

Source: Scopus

9) Masoudi, R., Ghaffari, A.

Software defined networks: A survey

(2016) Journal of Network and Computer Applications, 67, pp. 1-25. Cited 35 times.

DOI: 10.1016/j.jnca.2016.03.016

Document Type: Review

Source: Scopus

10) Duan, Q., Vasilakos, A.V.

Federated selection of network and cloud services for high-performance software-defined cloud computing

(2016) International Journal of High Performance Computing and Networking, 9 (4), pp. 316-327.

DOI: 10.1504/IJHPCN.2016.077824

Document Type: Article

Source: Scopus

11) Kang, N., Rottenstreich, O., Rao, S., Rexford, J.

Alpaca: Compact network policies with attribute-carrying addresses

(2015) Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT 2015, art. no. 2836092, . Cited 5 times.

DOI: 10.1145/2716281.2836092

Document Type: Conference Paper

Source: Scopus

12) Kreutz, D., Ramos, F.M.V., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.

Software-defined networking: A comprehensive survey

(2015) Proceedings of the IEEE, 103 (1), art. no. 6994333, pp. 14-76. Cited 958 times.

DOI: 10.1109/JPROC.2014.2371999

Document Type: Article

Source: Scopus

Search: ("QoS" OR "Quality of service") AND (POLICIES) AND ("SDN" OR "SOFTWARE DEFINED
NETWORK") AND ("PERFORMANCE ANALYSIS") AND "ENTERPRISES NETWORK"

ANEXO B: Resultados de encuesta según el estudio de la PUCP

Fuente:
https://docs.google.com/document/d/1zkxMV1II_g5-VyPK7Ra4vsdSEUC9LRylEfA_UoYXNSE/edit

Summary of the survey

QoS

1. Rate limit based on

- Location. UC Santa Cruz, Internet connections of different bandwidth.
- Application type. e.g., Rose-Hulman, gives very low priority or rate limits P2P
- Past history. e.g., UIUC, sets different rates based on last 24 hour total traffic volume
- Role. UI Chicago, UConnecticut, Cornell, UTAustin, sets different rates for students, paid students, faculty, visitors, affiliates etc
- Usage. e.g., CMU, handles differently for devices registered for teaching and researching purpose.
- Time. e.g., Rose-Hulman, gives more bandwidth to students during off hours (2am - 6am)

2. Max total traffic volume is regulated based on above factors too.
3. Rate limit/traffic volume is performed at device-level or user-level.

ACL

1. Block service ports at personal workstation, e.g., ftp, http, smtp. (Almost all universities)
2. Provide preset 10 ACL types to IP block owner for Internet traffic. Must define non-overlap IP prefix for different ACL types (UIUC)
3. Allow students to perform special ACL setup, but the device will be given different IP (UIUC)
4. Guests cannot print (UI Chicago)
5. ACL change when public/registered devices are compromised or stolen (Duke)
6. Allow IP block owners (e.g., departments) to write ACL rules for inter-department and Internet traffic (Cornell), especially to block traffic from other departments (identified by subnet).

Detailed version

We surveyed 22 universities in total, with a focu on QoS and Access Control.

- University of Illinois - Urbana Champaign
  Fuente: https://housing.illinois.edu/resources/technology/help/policies

  Rate Limiting
  - Rate limits based on usage in 24 hours (4GB: 25Mb/s, 6GB: 1 Mb/2)
  - Rate limit is unique to "your computer".

  Dorm firewall
  - By default, firewalls are "fully closed": where certain connections to applications (ports) are enabled or disabled.
  - Residents can request change to firewall settings to "mostly closed" firewall.

  Campus firewall:
  Fuente: http://www.cites.illinois.edu/firewall/index.html

  - Control Internet traffic, but not inter-department traffic
  - Ten static firewall groups that users (in fact those who own IPs) can request to join: http://www.cites.illinois.edu/firewall/plandetails.html
  - A contiguous IP range (IP prefix precisely) must be assigned to the same group : http://www.cites.illinois.edu/firewall/participation.html

- University of Illinois - Chicago
  Fuente: https://accc.uic.edu/policy/wireless
  Firewall
  - You may not run servers on your Res-Net or UIC wireless-connected computer with the intention of having other people accessing the servers. This includes, but is not limited to FTP, telnet, peer-to-peer, and mail servers as well as all others. Servers such as HTTP, IRC, DNS, and others are specifically prohibited.
  - Guests cannot run any file sharing/downloading program.

  Rate limit
  - 4GB in 24 hours for all your devices. If threshold is reached, network connections are suspended.
  - Rate limits on guest as well.

- Carnegie Mellon University

  Fuente: http://www.cmu.edu/computing/network/guidelines/bandwidth.html

  QoS

  - QoS support for special machines for teaching research applications.

  Access Control

  – Any machine which provides public commercial services (e.g., websites) is explicitly prohibited from the campus network
  – Banned OS for security concerns: "There are some operating systems which are known to cause problems in Carnegie Mellon's network environment. These operating systems are banned from being used in residence halls or via dedicated remote access services. At this time, the only operating systems explicitly banned are NT Server and Netware. If other operating systems become restricted, an announcement will be made on official.computing-news."

- University of Connecticut

  Fuente:

  http://huskytech.uconn.edu/violation_bandwidth.html

  http://policy.uconn.edu/wp-content/uploads/2012/05/Information-Security-Policy-Manual.pdf

  http://huskytech.uconn.edu/services/

  QoS

  - 16GB per student every 7 days.

  Firewall

  - Federation: departments and other University organizations may set up their own server and maintain their own information. They may also point to information located on servers outside of UConn.
  - Access Control – Users are given access based on their role.

- Duke University

  Fuente:

  http://oit.duke.edu/net-security/network/resnet-policy.php

  http://security.duke.edu/

  http://oit.duke.edu/enterprise/facilities-design/Communications_Facilities_Standards_April_2010.pdf

Firewall

- Report compromised personal computers, public computers, lost or stolen devices
- Cornell
- QoSStudents pay Internet bill (say over 100G/month)

Edge ACL

Fuente: http://www2.cit.cornell.edu/security/edgeacls/

- Owner of subnets can request ACL set up
- e.g., Restrict access between specific on-campus subnets, Block specific IP addresses, ports, and protocols to protect specific systems or applications from unintended remote accessl Block all Windows Networking (NetBIOS) from the Internet (non-Cornell network) or from everywhere, and Block all inbound TCP connections not established by systems on your subnet.
- As of November 2006, over 100 departments are using Edge ACLs. Out of the 700 VLANs on campus, over 350 have Edge ACLs applied to them.

- Georgia Tech Y University of Texas - Austin

  Fuente: http://www.utexas.edu/its/help/network/1772

  QoS

  - Student: 1G/week free. Can purchase higher rate.
  - Faculty: 500GB/week! Even part-time staff is 10GB/week.

- University of California - Santa Cruz

  Fuente: http://its.ucsc.edu/security/bandwidth.html

  QoS

- Total rate limits for different locations: (housing, town center and other residential areas)

- North Dakota State University

  Fuente: http://www.ndsu.edu/resnet/bandwidth.php

  QoS

  - 5GB per day. The quota is reset at 6am.
  - Campus traffic, such as Email and Blackboard are not counted.
  - Students that exceed the quota will be put in a restricted pool (300kbps). After 6am, they are moved to a slightly better pool (1Mbps).

- Penn. State University - Altoona

  Fuente: http://www.altoona.psu.edu/oit/restech_Bandwidth.php

  QoS and ACL

- 4GB per week to resources outside psu.edu. First or second violation results in moving to 56kbps shared pool for the remainder of the week. The third violation results in the pool for the whole semester. More violation revoke the access.

- Columbia University

  Fuente: http://policylibrary.columbia.edu/network-protection-policy

  QoS

  - Quotas are 2000 Megabyte/hr download and 700 Megabyte/hr upload.

- Stanford University

  Fuente: http://acomp.stanford.edu/about/policy/aup

  ACL

  - Student Computing clusters may be used only by members of the residence in which they are located, unless the local residence community decides otherwise. In any case, residential computing clusters are for the use of **on-campus residents only**. (Brief, incidental, low priority use may also be permitted by the local residence community to academic advisors and other faculty or staff members invited to the dorm.)

- Purdue University (Student conduct and regulations -> ResNet)

  Fuente: http://www.housing.purdue.edu/ResidentialLife/yourcomputer.html

  QoS

  - For example, uses that take up an unusually high portion of the bandwidth for extended periods of time may cause us to filter your use of the Internet to control use and allow others fair access. Use of academic resources on Purdue computers should not be affected by these restrictions.

  ACL

  Fuente: http://www.itap.purdue.edu/about/security.html

  - Identity assignment and role-based access

- University of California - Berkeley y Rutgers University

  Fuente: http://ruwireless.rutgers.edu/index.php?page=bandwidth

  QoS

  - 3Mbps per host
  - Internet traffic is capped to 2Gbps

- Rose-Hulman Institute of Technology

  Fuente: https://web.rose-hulman.edu/eit/Policies/Pages/Bandwidth-Utilization.aspx

QoS

- usage measured in 36-hour sliding windows, max rates depend on usage
- P2P application is separately capped at 30Mbps for the entire campus
- Different rate for off-hours, i.e., 2am - 6am

Firewall

- Block ports (services)

- Missouri State University

  Fuente: http://resnet.missouristate.edu/info/bandwidth.php

  QoS

  - 200G per week
  - Traffic shaping prioritizes traffic, e.g., P2P is given very low priority.

  Firewall

  - No student run servers, e.g., SMTP or web

- Worcester Polytechnic Institute

  Fuente: https://www.wpi.edu/Admin/IT/About/networkusagestandard.html

  QoS

  - 25 GB per day or 75 GB per week

- Liberty University

  Fuente: http://www.liberty.edu/informationservices/development/index.cfm?PID=26076

  QoS

  - Devices owned by the same user share the same bandwidth limit.
  - Can purchase more bandwidth

- Northern Illinois University y University of North Carolina - Chapel Hill

  Fuente: http://help.unc.edu/help/unc-chapel-hill-network-acceptable-use-policy/

  ACL:

  - Positions of users

    Fuente: http://doit.niu.edu/doit/policies_root/sasp.shtml