



Ohbrayan Fabrizio Ávila Basáez

Estudio de Factibilidad Técnica Económica para la Implementación de Laboratorio de SDN

Informe Proyecto de Título de Ingeniero Civil Electrónico



Valparaíso, 01 de junio de 2018



Estudio de Factibilidad Técnica Económica Para la Implementación de Laboratorio de SDN

Ohbrayan Fabrizio Ávila Basáez

Informe Final para optar al título deIngeniero Civil Electrónico, aprobada por la comisión de la Escuela de Ingeniería Eléctrica de la Facultad de Ingeniería de la Pontificia Universidad Católica de Valparaíso conformada por

Sr. Francisco Javier Alonso Villalobos Profesor Guía

Sr. Felipe Esteban Leighton Gonzalez Segundo Revisor

Sr. Sebastián Fingerhuth Massmann Secretario Académico

Valparaíso, 01 de junio de 2018

Resumen

El presente proyecto desea establecer la importancia del uso de las redes definidas por software y virtualización de funciones de red en el área de redes de computadoras, sus protocolos de comunicación utilizados y a su vez generar profesionales con el conocimiento de esta tecnología para el sector laboral. Por consiguiente, este informe propone la implementación de un laboratorio de redes con dichos tópicosy equipado con esta tecnología de vanguardia para realizar experiencias prácticas para el estudio de la misma. Todo esto en conjunto de un análisis económico al respecto para su creación tanto al breve como al largo plazo.

Palabras claves: redes de computadores, programación, laboratorio de red, experiencias técnicas.

Abstract

The present project wishes to establish the importance of the use of software-defined networks and the virtualization of network functions in the area of computer networks, their communication protocols used and, in turn, generate professionals with knowledge of this technology for the labor sector , therefore this report proposes the implementation of a network laboratory with such topics and equipped with this cutting-edge technology to carry out practical experiences for the study of it. All this in conjunction with an economic analysis in this sense for its creation both in the short and long term.

Keywords: computer networks, programming, network laboratory, technical experiences.

Índice general

Introducción	1
1 Antecedentes Generales	2
1.1 Problemática	
1.2 Objetivos	
1.2.1 Objetivo General	
1.2.2 Objetivos Específicos	
2 Estado del Arte	4
2.1 Organizaciones que trabajan y apoyan el desarrollo de SDN y NFV	4
2.1.1 ONF (Open Networking Foundation) [1]	4
2.1.2 ODL (Open Daylight) [2]	5
2.2 Casas de estudios Capacitadas en SDN y NFV	5
2.2.1 Internacionales	5
2.2.2 Nacionales	6
2.3 Debilidades y Fortalezas de su enseñanza	6
3 Conceptos Técnicos de SDN y NFV	8
3.1 Arquitectura actual de las redes y Arquitectura propuesta por SDN	8
3.1.1 Capas SDN	9
3.1.2 Controladores SDN	10
3.1.3 Controlador SDN para Comportamiento	10
3.1.4 Controlador SDN para Emplazamiento	10
3.1.5 Controlador SDN para Tipo de Flujo	11
3.1.6 Modelo OSI y su relación con el Modelo SDN	12
3.2 Conceptos Técnicos de NFV	13
3.3 Arquitectura y Topología NFV	14
3.4 Capa de Infraestructura fisica	15
3.4.1 Hardware de Cómputo	15
3.4.2 Hardware de Almacenamiento	17
3.4.3 Hardware de interconexión de redes	17
3.5 Cana de Virtualización	18

3.6 Capa de infraestructura virtual	19
3.6.1 Máquinas virtuales	19
3.6.2 Almacenamiento Virtual	19
3.6.3 Redes Virtuales	19
3.7 Resumen NFVI	20
3.8 Soluciones Actuales de SDN	21
3.8.1 Controlador SDN Brocade	21
3.8.2 Open SDN Ecosystem de HP (Hewlett-Packard)	22
3.8.3 Open MANO	23
3.9 Protocolo de Comunicación Existente para SDN y NFV	24
3.10 Protocolo OpenFlow	
3.10.1 Componentes Switch OpenFlow	24
3.10.2 Funcionamiento de un Switch OpenFlow	26
3.10.3 Canal de Control (OpenFlow Channel)	30
3.10.4 Secuencia de Mensajes Protocolo OpenFlow	32
3.10.5 Observaciones	37
4 Distraforme de vi Entermo de Tuebeia	20
4 Plataformas y Entorno de Trabajo	
4.1 Mininet	
4.2 ODL (OpenDayLight)	
4.3 GNS3 (Version 2.0)	
4.4 WireShark	
4.5 Instalación Plataformas de Trabajo	
4.5.1 GNS3 (Version 2.0)	
4.5.2 WireShark	
4.6 Experiencias de Laboratorio	
4.7 Diseño de las experiencias de laboratorio	
4.8 Temas seleccionados para las experiencias	
4.8.1 Pre-informe y Experiencia N°1 (Apéndice D.1 y D.2)	
4.8.2 Pre-informe y Experiencia N°2 (Apéndice D.3 y D.4)	
4.8.3 Pre-informe y Experiencia N°3(Apéndice D.5 y D.6)	
4.9 Observaciones	43
5 Estudio Económico para la Implementacion de Laboratorio SDN	44
5.1 Factibilidad Operacional	
5.2 Factibilidad Técnica	
5.3 Factibilidad Económica	
5.3.1 Propuesta de bajo costo y facil implementación	
5.3.2 Propuesta robusta y de gran escalabilidad	
5.3.3 Valorización Productiva	
5.3.4 Herramienta de Evaluación CUNE	
5.4 Análisis final de Factibilidad	
5.5 Beneficios y Conclusiones de su respectiva instalación	

muice genera	Índice	genera
--------------	--------	--------

Discusión y conclusiones	54
Bibliografía	56

Introducción

El constante avance de la tecnología ha permitido solucionar diversos problemas, con un enfoque más práctico e intuitivo previniendo errores y mejorando sistemas actuales. Es en este ámbito en donde las redes definidas por software o SDN (software defined networking) en conjunto con la virtualización de funciones de red o NFV (Network Function Virtualization) se han instaurado como una solución para las redes computacionales del futuro en el área de las telecomunicaciones. Dado que en la actualidad son estas redes las que se ven en la necesidad de progresar a un sistema más robusto, con una fácil gestión de los equipos conectados entre sí para impedir interrupciones al transferir y/o recibir información.

Las redes definidas por software proponen un enfoque centralizado, otorgando la capacidad de administrar diversas características en una red de computadoras, modificando su protocolo de comunicación por medio de un lenguaje de programación estándar denominado OpenFlow, permitiendo edificar y manipular la totalidad de la red IP desde un solo controlador maestro en base a un protocolo abierto.

OpenFlow se encuentra diseñado específicamente para la tecnología SDN, este protocolo de comunicación implica un cambio radical para lógica actual, en vista que permite la interconexión de equipos que no posean el mismo fabricante evitando la dependencia de un solo proveedor y mejora el crecimiento de las redes para ciertas aplicaciones o servicios y reduce costos para dichas soluciones debido a su protocolo de comunicación particular, a diferencia de lo que sería un entorno en base a un lenguaje común, abierto y capaz de adaptarse a los diferentes equipos que soporten esta tecnología. En consecuencia, SDN propone reducir los costos en el diseño e implementación de redes de computadoras e invita a los diferentes fabricantes a innovar ante la competencia. Es por esto que diversas industrias apuestan a la implementación de esta tecnología. Empresas como Google, Microsoft e Intel son miembros de 'Open Networking Foundation (ONF)' Fundación que promueve el uso de SDN, la estandarización del protocolo OpenFlow y tecnologías relacionadas.

Por ende, en el desarrollo del presente informe se pretende abordar los fundamentos de las redes definidas por software con el fin de implementar experiencias para un laboratorio de redes considerando SDN como una herramienta próxima tanto en el ambiente académico como laboral y a su vez realizar un análisis económico para su respectiva implementación y avance.

Antecedentes Generales

1.1 Problemática

Actualmente las redes de comunicación son bastante complejas, no solo a nivel de su topología, sino que también de su implementación, estas redes en su mayor parte poseen problemáticas de escalabilidad y dependencia del fabricante, puesto que cada equipo posee un gran grado de autonomía coordinándose con sus equipos vecinos y tomando sus propias decisiones al momento de trabajar la información. Todo esto implica dificultades al momento de administrar la topología de la red.

Conjuntamente en el área de redes de computadoras, la interconectividad entre diversos equipos y aplicaciones computacionales es una necesidad cada vez más ineludible, y es que las industrias en general, buscan soluciones adaptables, de bajo costo y con un soporte de calidad, ante cualquier dificultad. Por lo que, para cumplir con estas soluciones, es necesario generar un perfil profesional adecuado al respecto, con un gran manejo de tecnologías vanguardistas que cumplan con estos requisitos como lo son tanto SDN y NFV.

Por consiguiente, también es necesario el estudio de esta tecnología de manera apropiada para la formación de dichos profesionales, con un enfoque esencial en trabajo de laboratorio por sobre la noción básica generando experiencias prácticas, otorgando así herramientas para sus participantes al momento de enfrentarse al mundo laboral y adaptarse a los cambios futuros de esta tecnología aún en desarrollo.

Actualmente, en Chile se presenta una falta de profesionales con los conocimientos deseados en el área de redes computadores capaces de implementar y configurar esta tecnología a pesar que ya un grupo importante de diversas industrias poseen interés en las redes definidas por software y su implementación en sus respectivas empresas.

Finalmente, se debe seleccionar debidamente el equipamiento necesario para su estudio en el aula, que cumpla con la capacidad de reflejar todas o la gran mayoría de funciones de esta tecnología, además de ser adquiribles por la casa de estudio competente interesada en su capacitación.

En el presente proyecto, se presenta la temática de SDN y NFV como un área de formación académica con un enfoque a un sector laboral emergente aún no cubierto, introduciendo los conceptos necesarios para su noción general, administración y soluciones actuales para una futura implementación en el aula.

1.2 Objetivos

Los siguientes objetivos representan las metas deseadas de esta propuesta, fundamentales para su correcta ejecución.

1.2.1 Objetivo General

Diseñar un laboratorio de redes definidas por software y virtualización de funciones de red adecuado, que posea una factibilidad económica, para su implementación en la casa de estudios correspondiente y que permita también una escalabilidad de sus componentes en el futuro.

1.2.2 Objetivos Específicos

- Revisar el estado del arte dela tecnología SDN y su perspectiva a futuro
- Comprender el protocolo de comunicación Openflow.
- Comprender Plataformas de simulación NFV y SDN.
- Creación de guías que permitan entender el entorno SDN y tecnologías similares.
- Análisis técnico económico sobre esta tecnología.
- Factibilidad Técnica para su implementación.

2 Estado del Arte

SDN es un concepto versátil para la implementación en las redes de computadoras teniendo en cuenta su capacidad de adaptabilidad y los diferentes beneficios asociados a costo y administración de los recursos disponibles. Es esencial poder comprender que ofrece importantes soluciones a problemáticas en el entorno profesional y que por ende su estudio y aplicación apuntan a masificar su tecnología como estructura para todas las redes.

Actualmente existen diferentes compañías en el área de Networking¹ y virtualización de redes (NFV) que se dedican a desarrollar soluciones SDN para sus productos adhiriéndose a la norma estándar de OpenFlow, o generando soluciones en base a sus protocolos.

A continuación, se presenta organizaciones, establecimientos educativos y soluciones en el ámbito de SDN.

2.1 Organizaciones que trabajan y apoyan el desarrollo de SDN y NFV

Dentro de las diversas organizaciones que estudian y trabajan las redes definidas por software, sin duda la fundación pionera y más conocida es ONF, esta corresponde a una fundación sin fines de lucro orientada al usuario y dedicada a acelerar la adopción de SDN y NFV. Ha realizado con éxito la inclusión de dispositivos de red y software de control, fomentando la Aparición de una amplia gama de plataformas de código abierto y estableciendo OpenFlow como el primer standard para SDN. Actualmente en 2017 consta de 110 miembros, destacando *AT&T*, *Broadcom*, *Brocade*, *Cisco*, *Google*, *Huawei*, *Intel*, *Microsoft* y *Yahoo!*Entre otras.

2.1.1 ONF (Open Networking Foundation) [1]

Este proyecto es promovido por *The Linux Foundation* con el cual se impulsa al desarrollo, distribución y adopción global de la plataforma SDN Open Daylight (ODL) de código abierto y de índole neutral. Entre sus miembros destacan Brocade, Cisco, Ericsson, IBM, Microsoft, Red Hat, VMware, PLUMgrid.

¹Networking: redes de cómputo que vinculan dos o más dispositivos informáticos con el propósito de compartir datos a nivel software y hardware.

2.1.2 ODL (Open Daylight) [2]

Este proyecto es promovido por *The Linux Foundation* con el cual se impulsa al desarrollo, distribución y adopción global de la plataforma SDN Open Daylight (ODL) de código abierto y de índole neutral. Entre sus miembros destacan Brocade, Cisco, Ericsson, IBM, Microsoft, Red Hat, VMware, PLUMgrid.

2.2 Casas de estudios Capacitadas en SDN y NFV

Ante la necesidad de implementar esta nueva tecnología, se hace necesaria la ejecución de experiencias relacionadas en el ámbito académico usando SDN y NFV en laboratorios de redes. Por tanto, variadas casa de estudios presentan cursos o laboratorios de redes definidas por software, ya sea para una noción sobre esta temática o una formación profesional.

2.2.1 Internacionales

Las siguientes casas de estudios internacionales presentan nociones básicas o un perfil profesional en el uso SDN y NFV:

Universidad de Stanford:

Stanford ha sido un líder en SDN y continúa teniendo un interesante programa de investigación que incluye la verificación de red como un gran tema. En sus dependencias nace y se crea el movimiento de redes definidas por software (SDN) y el protocolo OpenFlow que se creó como una interfaz para programar los dispositivos de comunicación de estas redes, desarrolló varias plataformas y herramientas de código abierto. Además, al involucrar a académicos, proveedores de dispositivos, operadores de telecomunicaciones y proveedores de servicios en la investigación en una etapa temprana, han podido cultivar y operar ecosistemas que han llevado a la aplicación práctica generalizada de los conceptos SDN. También promovieron el desarrollo y la adopción de SDN liderando el movimiento de estandarización y alentando a la comunidad de código abierto. Su logro como institución innovadora en infraestructura de TIC es muy notable.

Universidad Carlos III de Madrid:

La Universidad Carlos III de Madrid ya cuenta en su departamento de Telemática con el Magister especialista de SDN y NFV, donde la malla curricular incluye laboratorios de estos tópicos, además de abordar conceptos como el big data, la administración y orquestación de NF.

Universidad de Buenos Aires:

Esta universidad ya cuenta con líneas de investigación en áreas como internet de las cosas (IoT), redes definidas por software y áreas temáticas en lo que respecta a comunicación de datos, todo esto a cargo de su departamento de computación que forma parte de la facultad de ingeniería de dicha casa de estudios.

Universidad Federal de Rio de Janeiro:

Universidad dedicada a la investigación de esta tecnología,se ha interesado en las diversas áreas de aplicación de la misma, es decir todo lo referente a control de acceso, gestión de una red, centros de datos, incluso en redes domésticas. Generando diversos proyectos al respecto.

Universidad de Princeton:

Esta institución forma parte de la fundación SDX central, de la cual es colaborativa, posee El departamento de informática de Princeton donde tiene investigadores trabajando activamente en varios aspectos de SDN, incluida la colaboración con Cornell en Frenetic, un lenguaje de abstracción para abstracciones de redes de alto nivel. Actualmente se encuentra trabajando en Princeton University Frenetic y Princeton University Pyretic como proyectos sobre SDN.

Universidad Tecnológica de Georgia:

Universidad que ya posee asignaturas enfocadas solamente en SDN, su concepción, uso, funciones y aplicaciones con programas como virtual box, mininet y el lenguaje OpenFlow.

Universidad de Yales:

El Grupo Yale Haskell ha desempeñado un papel integral en la concepción, evolución, implementación y aplicación del lenguaje de programación Haskell. Lo más notable es que han utilizado Haskell para construir una variedad de lenguajes incrustados específicos de dominio (DSEL): lenguajes que usan Haskell como un marco dentro del cual construir abstracciones específicas de un dominio subyacente. Uno de sus proyectos clave de DSEL es Nettle, un lenguaje para configurar enrutadores de red OpenFlow de una manera declarativa y de alto nivel.

Entre otras universidades, ya que constantemente se van incorporando más universidades interesadas en comprender y aplicar estos conocimientos

2.2.2 Nacionales

Actualmente en nuestro país SDN y NFV son temas de estudio en el área de investigación y no representan académicamente asignaturas o laboratorios concretos

2.3 Debilidades y Fortalezas de su enseñanza

La Pontificia Universidad Católica de Valparaíso al igual que las demás universidades nacionales se encuentra sin cursos o laboratorios en el tema, esto es debido en gran parte a los siguientes factores:

- Ausencia del equipamiento necesario.
- Falta de experiencias para laboratorios prácticos.
- Falta de asignaturas obligatorias dedicadas al área.

No obstante, estas debilidades se pueden transformar en la oportunidad para mejorar el perfil de egreso de Ingenieros Electrónicos y Civil Electrónicos, además de generar interés en el alumnado con el fin de obtener fortalezas como:

- Estudiantes que desean obtener estos conocimientos y aplicarlos a futuro.
- Docentes preparados con conocimiento en el área.
- Actualización de Laboratorio con Equipamiento acorde.

3 Conceptos Técnicos de SDN y NFV

Se establece las redes definidas por software como la realización de la división del plano de datos de la red y el plano de control de la red, generando que este nuevo planteamiento en su arquitectura permita un dinamismo, una adaptación mucho mayor y un ancho de banda amplio para la ejecución de aplicaciones. En las siguientes secciones se presentarán conceptos técnicos sobre su topología y arquitectura.

3.1 Arquitectura actual de las redes y Arquitectura propuesta por SDN

La arquitectura actual de la red de computadoras establece conexiones entre diferentes equipos por medio de elementos de conmutación como routers, switch, data centers, entre otros y estos dispositivos se encuentra conectados entre sí por diferentes medios (Par de cobre, aire, fibra óptica, etc.), no obstante la característica principal de cada elemento enrutador es que se define como autónomo, es decir cada protocolo de comunicación se encuentra en la memoria del enrutador y bien puede ser óptimo para el elemento no significa que la totalidad de la red se encuentre óptimamente definida, en la figura 3-1 se presenta una topología estándar para un red actual.

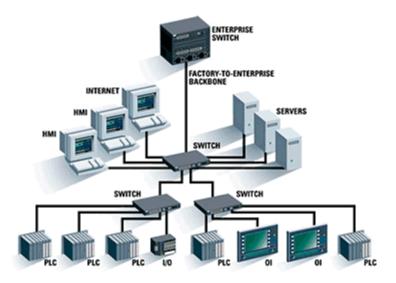


Figura 3-1: Representación de una arquitectura de red en el ámbito empresarial [3]

La arquitectura propuesta por SDN establece un plano de control desacoplado de los elementos de conmutación y compartido con el plano de datos o forwarding, todo esto en base a un control programático de SDN de manera estándar Openflow y OF-Config lo cual se representa a través de paquetes de datos que ingresan al dispositivo switch, pero que son controlados por los servidores por medio del software de control y gestión en los mismos, como se observa en la figura 3-2.

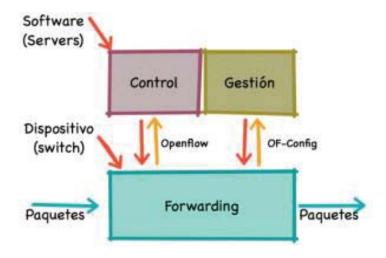


Figura 3-2: Representación de un elemento de conmutación con lógica SDN [4]

3.1.1 Capas SDN

La arquitectura mencionada para SDN permite asignar diferentes tipos de capas lógicas las cuales corresponden a:

- Capa de Aplicación
- Capa de Control
- Capa de Infraestructura

Donde la Capa de Aplicación es en la cual se encuentran las aplicaciones desarrolladas por usuarios o empresas y las cuales se conectan con la capa de control por medio de las denominadas Northbound APIs. Las Northbound APIs mayormente permiten el tratamiento de la información entre ambas capas, pues contienen la codificación adecuada para la capa de control, la cual reenvía la información basada en la codificación recibida.

La capa de Infraestructura, contiene los diversos equipos físicos de enrutamiento los cuales se conectan a la capa de control por medio de las denominadas Southbound APIs. Las SouthBound APIs procesan los paquetes recibidos desde el controlador SDN para reenviar la codificación adecuada a los elementos de conmutación en la capa de infraestructura, Openflow y otros protocolos de comunicación se encuentran en estas APIs.

Finalmente, en la Capa de control como su nombre lo indica, es donde se centraliza toda la información y paquetes, es decir gestiona las instrucciones y las distribuye a los elementos físicos en la red.

A continuación, en la figura 3-3 se presenta la arquitectura de red SDN con sus respectivas capas y aplicaciones.

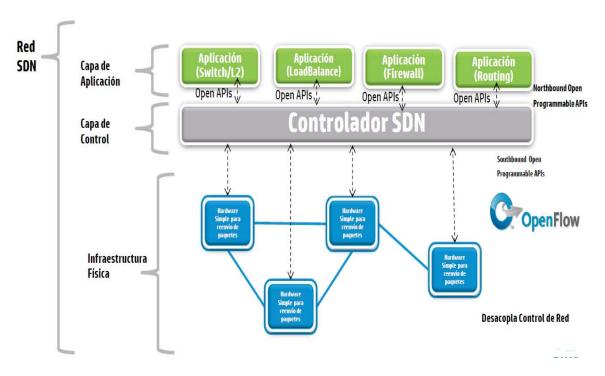


Figura 3-3: Arquitectura de Red SDN, sus respectivas capas y elementos [5]

3.1.2 Controladores SDN

Existen diferentes controladores SDN cada uno posee una plataforma de trabajo específica en la que podrá escribir el software de control de la red y que tendrá diferentes características, siendo estas las que llevan a definir los tipos de controladores:

- Controlador SDN por Comportamiento.
- Controlador SDN para Emplazamiento.
- Controlador SDN para Tipo de Flujo.

3.1.3 Controlador SDN para Comportamiento

En este tipo de controlador existen dos tipos, el proactivo y el reactivo. En el caso del proactivo, es el que completa las tablas de flujos de manera anticipada, mientras que el reactivo obtiene el primer paquete de flujo y designa las entradas en las tablas de flujo por sí mismo.

3.1.4 Controlador SDN para Emplazamiento

Esta característica hace referencia a la topología de la red SDN y su implementación. Por una parte, existe un controlador SDN para un conjunto de Switches, y por otra donde existe un controlador maestro y centralizado de toda la red.

3.1.5 Controlador SDN para Tipo de Flujo

Al igual que en las características anteriores, existen dos opciones para el tipo de flujo, el primero sería el enrutamiento por flujo determinado o por defecto, y la segunda opción correspondería al enrutamiento por flujo agregado, donde cada tabla se le asigna una categoría de flujos.

A continuación, se presenta la tabla 3-1 con diversos Controladores SDN en el mercado o en desarrollo y sus características destacables.

Tabla 3-1: Principales controladores SDN, su plataforma, desarrollador y características [6]

Nombre	Plataforma	Organización	Características
NOX/POX	C++/Python	Stanford	El primer Controlador Openflow
Beacon	Java	Stanford	Admite tanto la operación basada en eventos como la operación de procesos en hilos, proporciona ui web.
Floodlight	Java	Big Switch	Amigable para el desarrollador con muchos documentos.
Ryu	Python	NTT laboratorios OSRG group	Con soporte para OpenStack.
Maestro	Java	Rice University	Soporte multi-procesos en hilo
OMNI	Python/Java	Universidade Federal do Rio de Janeiro	Interfaz web en NOX.
McNettle	Haskell	Yale University	Lenguaje expresivo declarativo de alto nivel, optimización multinúcleo.
Trema	Ruby/C		Red virtual DSL.
OpenDayLight Controller	Java	Linux Foundation	Proporcionar API REST y GUI web

3.1.6 Modelo OSI y su relación con el Modelo SDN

A continuación, se definirán y comparan los modelos OSI y SDN, sus estándares y protocolos, los que definen el envío y/o recepción de información en una red de computadoras

El modelo OSI (modelo de interconexión de sistemas abiertos) corresponde al modelo actual y posee siete capas las cuales se definen a continuación:

- Capa de Aplicación: Tiene como misión controlar las funciones a realizar por los programas de usuario de manera que les permita el acceso al entorno OSI.
- Capa de Presentación: se encarga de formatear los datos para que sean entendibles y legibles para la capa de aplicación.
- Capa de Sesión: Se encarga de poder tener varias sesiones independientes para cada aplicación, como ejemplo se puede hablar de una sesión ftp.
- Capa de Transporte: Transporta los paquetes de extremo a extremo. Trabaja con dos protocolos TCP/IP.
- Capa de Red: Direccionamiento Lógico a través de la IP, se encarga de enrutar los paquetes.
- Capa de enlace de datos: Direccionamiento físico a través de la MAC address
- Capa física: Se encarga de la transmisión binaria, es decir ceros y unos.

Como se habia comentado en el apartado anterior, las capas que consituyen el modelo SDN son tres, la capa de aplicación, de control y de infraestructura. si quisieramos comparar estos modelos de una manera sencilla. Se puede indicar una relacion directa entre sus capas de la siguiente forma:

- ❖ Para la Capa de aplicación en SDN se puede relacionar con:
 - > Capa de Aplicación
 - > Capa de Presentacion
 - Capa de sesion

Todo esto debido a que en el modelo OSI, Estas tres capas se encargan de definir la navegación y su velocidad, mientras que en constraste SDN corresponde ha aplicaciones virtuales operando en dichos niveles y dependerá de la aplicación el cómo se generará la navegación.

- Para la capa de Control SDN, se puede relacionar con:
 - > Capa de Transporte
 - Capa de Red

Tal constraste se debe a que el controlador SDN posee la visíon completa de toda la red y administra las direcciones IP y rutas para los dispositivos físicos.

- ❖ Para la capa de Infraestructura se puede relacionar con:
 - > Capa de enlace de datos
 - Capa Fisica

Finalmente, esta comparación se debe a que ahora seran los elementos físicos enrutadores los que realizaran el direccionamiento físico, mejorando la respuesta y disminuyendo los costos, puesto que se permitirá la heterogeneidad de dispositivos. Como se presenta en la figura 3-4:

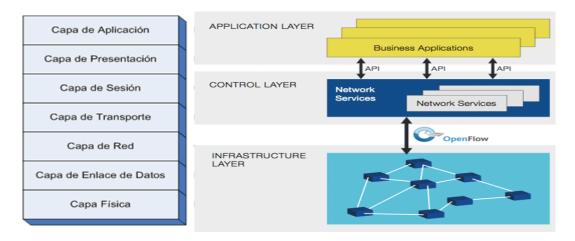


Figura 3-4: Representación Modelo OSI y SDN [7] [8]

3.2 Conceptos Técnicos de NFV

NFV corresponde a la virtualización de las funciones de la red,teniendo como objetivo apresurar el uso de nuevos servicios, tales como, NAT, firewall, IPS, DNS y el crecimiento del almacenamiento caché para ampliar sus ingresos, lo que permite incrementar su ritmo yrenovar los servicios de red y su distribución.

NFV logra consolidar múltiples equipos de red en servidores estándar de la industria; ahora los switches y almacenamiento pueden estar ubicados en centros de datos, nodos de red o en la casa del usuario final reduciendo así la cantidad de hardware propietario muchas veces necesarios para implementar servicios de red, además de disminuir los costos CAPEX y OPEX para la ejecución de los servicios. Dichos costos se refieren a los costos asociados a la producción y a los costos de mantenimiento de equipos respectivamente. En la siguiente figura 3-5 se presenta la interrelación entre SDN, NFV y sus aplicaciones futuras.



Figura 3-5: Interrelación entre las tecnologías SDN, NFV y sus aplicaciones [9]

Cabe destacar si bien ambas funciones SDN y NFV conforman una solución potente, estas no dependen una de la otra, sin embargo, su complemento es idóneo debido a las características similares de administración de los recursos, es decir un uso centralizado de los componentes que conforman la red. A continuación, se presentarán conceptos técnicos sobre NFV, su topología y arquitectura.

3.3 Arquitectura y Topología NFV

El entorno de trabajo óptimo para la tecnología NFV se encuentra determinado por las funciones de red virtuales VNF (Virtualized Network Functions), la infrastructura NFV (NFVI), El administrador de la infraestructura virtual VIM (Virtualized infrastructure Manager) y finalmente puede contemplar también controladores SDN, presentando su relación en la figura 3-6.

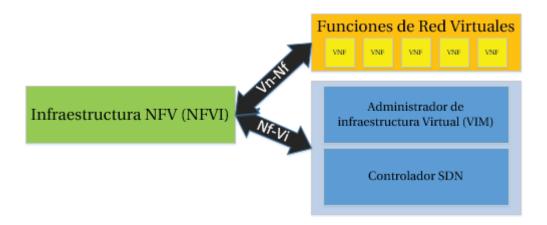


Figura 3-6: Entorno de trabajo NFV

Donde el VIM es el encargado de distribuir y administrar la infraestructura virtual, mientras que las funciones de redes virtuales es un software que opera exclusivamente con NFVI, por lo cual las conexiones de "Vn-Nf" corresponden a la vía de datos por el cual pasa el tráfico de red, mientras que la conexión "Nf-Vi" corresponde a la ruta de control que es de uso exclusivo para la gestión y además también limita el tipo de trafico de red a través de ella.

En detalle la infraestructura de NFV (NFVI) se compone de tres capas:

- Capa de infraestructura física
- Capa de virtualización
- Capa de infraestructura virtual

A continuación, se procede a detallar cada una de ellas.

3.4 Capa de Infraestructura fisica

En la industria, las soluciones más comunes para la infraestructura física es la utilización de 1U o 2U para varias aplicaciones NFV desarrolladas en la actualidad, este término 1U y 2U hace referencia a la cantidad de racks dentro de los servidores y por ende a su tamaño.

3.4.1 Hardware de Cómputo

Las características técnicas estándar que deben tener estos elementos hoy en día en una infraestructura física, en cuanto a lo que tiene relación a realizar los cómputos y procesos en una red NFV se especificaran en la siguiente Tabla 3-2

Si bien las características anteriores indicadas en la Tabla 3-2 corresponden a la infraestructura estándar también existen los 'Blade Chassis' para procesos de mayor densidad, pero con menos requerimientos de capacidad de almacenamiento, a continuación, se detallan sus características

Blade Chassis

Estos servidores denominados Blade se encargan de empaquetar múltiples nodos de computación en un solo chasis. Se ocupan de la red, la energía, la refrigeración y la gestión de hardware para todo el conjunto de nodos. El tamaño de estos servidores puede ser tan pequeño como de dos a cuatro nodos de cálculo en un 2U a dieciséis servidores en un 16U según los requerimientos deseados.

Adicionalmente existe una solución de infraestructura equilibrada llamada infraestructura híper convergente (HCI) 'Hypercoverged Infraestructure' de sus siglas en inglés, es generalmente usado por los proveedores de servicios de nube (CSP) que desean implementar servidores que posean una mayor sencillez en comparación con los servidores blade Chassis además de establecer características y mejoras para realizar en las funciones del almacenamiento, el conjunto de nodos de cálculo y la interconexión. A continuación la tabla 3-2 presenta el estándar de especificaciones:

Tabla 3-2: Descripción para especificaciones estándar para tecnologías NFV

Componentes	Ejemplo de la especificación	Razón Fundamental
Procesador	Dual-socket Intel® Xeon® E5-2600 v4/v3	Este componente posee la mejor características precio/ funcionamiento.
Memoria	192-256 GB DDR4 Memory	La memoria es sorprendentemente cara y debe ser cuidadosamente dimensionada para que coincida con su utilización. Por ejemplo, si su VM es de 8 GB y tiene previsto ejecutar 48 VM / servidor, entonces cualquier memoria de más de 384 GB es un desperdicio.
Flash	2 x 400 GB NVMe	De ser o no necesaria y cuánto debe utilizarse depende totalmente de los requerimientos de la Máquina virtual.
Disco Duro	6 x 4TB 7.5K RPM SATA Drives	Dependiendo de la necesidad de permanencia, puede o no necesitar almacenamiento interno (además del almacenamiento de inicio). Si lo hace, tiene que ser del tamaño sobre la base de su VM.
Network interface Card (NIC)	2 x 10GE	El rendimiento E/S va a estar limitado por la arquitectura general del servidor. El ancho de banda de la NIC debe dimensionarse para que coincida con el rendimiento de E/S. Hay consideraciones adicionales como Caminos redundantes dobles vs. caminos sencillos, etc.
Ventiladores y suministro de energía	Un solo ventilador y suministro de energía	La idea principal del uso de la nube es mover la resiliencia en software, y obtener 99.999% de disponibilidad de servicio, mientras que utiliza la disponibilidad de hardware 99.9%. Sin embargo, dependiendo de la escala, la estrategia de ahorro, el coste del soporte del proveedor del servidor y si los VNF son verdaderamente propios de la nube, podría considerar los ventiladores duales redundantes.
Interfaz de gestión	Controlador de administración de la placa base (BMC)	Se trata de un procesador distinto de la CPU principal y es responsable de gestionar la integridad del servidor. Este componente vale la pena el incremento desde un de precio menor.

HCI

Esta solución combina el almacenamiento, el cómputo y la configuración de la red en un solo equipo de excelente escalabilidad, siendo así más flexible que los servidores Blade. Las soluciones híper-convergentes pueden incluir funcionalidades adicionales de valor agregado como alta disponibilidad, respaldo, recuperación de desastres, seguridad, etc.

La conveniencia de esta solución tiene que ver con una flexibilidad reducida en cuanto a implementar, configurar, escalar y actualizar independientemente las distintas capas. Pues, esto podría conducir a la subutilización de hardware y a problemas de rendimiento.

Para las cargas de trabajo de NFV, hay proveedores de CSP importantes que están comenzando a proporcionar cargas de trabajo convergentes e híper-convergentes y comenzando a sintonizarlos para obtener más servicios de red.

3.4.2 Hardware de Almacenamiento

SAN, NAS O SDS

Mientras que la mayoría de las cargas de trabajo de NFV se enfocan en las necesidades de computación y de red, algunas funciones como almacenamiento en caché (vídeo y otros) o analítica y almacenamiento de datos grande requieren grandes y rápidas agrupaciones de almacenamiento.

El hardware de almacenamiento viene en forma de almacenamiento conectado directamente (es decir, almacenamiento en un nodo de cálculo), almacenamiento SAN tradicional (red de área de almacenamiento) o NAS (almacenamiento conectado a la red), que puede ser almacenado como almacenamiento virtual para VMs. De forma alternativa, se puede usar el almacenamiento definido por software (SDS) donde el hardware es un servidor estándar de la industria optimizado para el almacenamiento.

Un servidor de almacenamiento típico podría ser una caja de 2U o 4U con un gran número discos de estado sólido (SSD de 36, 48, 60) de o unidades de disco duro (HDD), y mucho menos poder de cálculo y memoria comparada con los servidores de computación (Por ejemplo, un único socket con 64 GB de memoria.

3.4.3 Hardware de interconexión de redes

Open Switches o Hardware Propietario

El hardware de red se presenta en forma de switch L2 / L3 patentados o switch de 'bare-metal'. Los switch de 'bare-metal' son dispositivos para establecer en red lo que el servidor estándar de la industria debe calcular o almacenar. Estos conmutadores usan ASIC "Circuito Integrado para Aplicaciones Específicas" (application-specific integrated circuit – ASIC, por sus siglas en inglés). Estos switch desagregan el hardware del software, por tanto, ejecutan el software en base a una Open Network Linux u otros paquetes de software de conmutador de red de terceros.

3.5 Capa de Virtualización

La capa de virtualización para NFVI se encuentra en la parte superior del hardware y es una plataforma de software que normalmente implica un hipervisor. Los hipervisores se encargan de administrar los recursos de la máquina física y ofrecen el equivalente de una máquina física a la aplicación que pueda estar ejecutando, De hecho, las tres funciones principales del hipervisor son:

- Dividir los recursos de la máquina física
- Proporcionar aislamiento entre diferentes VM (esto se hace con la ayuda de la CPU)
- Emular todos los periféricos necesarios, p. Tarjetas NIC.

Los principales hipervisores utilizados hoy en día para NFV son VMware® vSphere, Xen Project, KVM y KVM en tiempo real. Los cuales se abordarán a continuación:

VSphere

VSphere es un hipervisor propietario de VMware. consolidado y extremadamente estable. Para cargas de trabajo que no son nativas de la nube, vSphere también tiene varias características enriquecidas que permiten la migración de VM entre hipervisores y alta disponibilidad, es lo que se llama un hipervisor de tipo 1 o 'bare-metal', ya que se ejecuta directamente en el hardware.

Xen Project

Este proyecto de virtualización y software para servicios de nube posee variadas y poderosas características, que lo establecen como una excelente elección para muchas organizaciones. Entre las más destacables se encuentran:

- Soporta múltiples sistemas operativos invitados: Linux, Windows, NetBSD, FreeBSD
- Soporta múltiples plataformas Cloud: CloudStack, OpenStack
- Escalabilidad, Seguridad, Flexibilidad, Migración de Máquinas virtuales

Además del soporte de múltiples proveedores, el proyecto cuenta con el apoyo de una serie de distribuidores de software y servicios importantes. Esto ofrece a los usuarios finales numerosos lugares para encontrar soporte, así como numerosos proveedores de servicios con los que trabajar.

KVM

KVM (Kernal-based Virtual Machine) Se trata de un hipervisor de tipo abierto, es decir un hipervisor también conocido como alojado, donde se ejecuta en la parte superior de un sistema operativo Linux (denominado sistema operativo host para diferenciarlo del sistema operativo que se ejecuta en la VM). Por esta razón, al usar KVM, también se debe elegir el sistema operativo anfitrión. Los sistemas operativos (OS) anfitrión del sistema más utilizados son RHEL, SUSE y Ubuntu.

KVM en Tiempo Real

Este hipervisor de código abierto es una variante de KVM, el cual proporciona un comportamiento determinista mejorando el rendimiento de pequeños paquetes en comparación a KVM.

3.6 Capa de infraestructura virtual

Esta capa claramente corresponde a todo lo que serían las máquinas virtuales, almacenamiento virtual y a la interconexión de redes virtuales, en la siguiente sección se detalla sobre sus características

3.6.1 Máquinas virtuales

Las máquinas virtuales (VMs) son creadas por el hipervisor como se mencionó anteriormente. Los hipervisores presentan APIs para crear, eliminar, migrar y administrar máquinas virtuales. En el caso de KVM, esto se hace a través de una librería llamada libvirt, y para vSphere, se hace a través de un administrador de máquina virtual llamado vCenter. Estas máquinas virtuales son lo que VNFs están alojados dentro y que actúan como el caballo de batalla que alimenta el núcleo de servicios de red para el concepto NFV.

3.6.2 Almacenamiento Virtual

La responsabilidad de virtualizar el almacenamiento de bloques o archivos reside en el software SAN, NAS o SDS, y la VM se presenta simplemente con un LUN (número de unidad lógica) o un recurso compartido de archivos. La capa de virtualización puede agregar características adicionales como instantáneas, copias de seguridad, compartición de VM, replicación, etc.

3.6.3 Redes Virtuales

Un hipervisor contiene un switch virtual o enrutador que tiene cuatro funciones:

- Cambiar entre VMs en el mismo nodo físico sin tener que ir a un switch externo más utilizado para crear cadenas de funciones de servicio.
- Proporcionar servicios de red donde las redes expuestas a las máquinas virtuales están completamente abstraídas con su propio espacio de direcciones y topologías diferentes de la física o la red subyacente.
- Proporcionar servicios de seguridad como la aplicación de listas de control de acceso.
- Habilitar la conectividad con Internet.

Hay muchas tecnologías de redes virtuales utilizadas en NFV. Algunas de las tecnologías abiertas comunes son:

OpenVswitch (OVS)

OVS es un proyecto de switch virtual estable que se inició en 2009, Es un switch de múltiples capas que puede abarcar hipervisores (es decir, a través de nodos de computación) e incluye seguridad,

supervisión, encapsulación de red de superposición, enlace NIC etc. OpenFlow es el método principal de programación de OVS, Junto con OVSDB para la configuración, aunque también se admiten protocolos adicionales.

vRouter

OpenContrail es un controlador SDN de código abierto que no admite OVS. En su lugar, requiere su propio software de red virtual denominado vRouter. VRouter, como su nombre podría sugerir, también admite el enrutamiento y se puede programar a través de BGP.

FD.io

El Fast Data Project (FD.io) es un sustituto de alto rendimiento para OVS. Utiliza un motor de procesamiento de paquetes vectoriales para procesar paquetes en paralelo para obtener una mejora del rendimiento de 5x-39x sobre OVS, aunque hay una versión OVS con DPDK que tiene un desempeño significativamente mejor que OVS de stock.

DPDK

DPDK es un conjunto de bibliotecas de software que pasan por alto el kernel, evitando así el costoso modo kernel a modo de usuario. DPDK también utiliza el modo de sondeo en lugar de las interrupciones para evitar una gran cantidad de interruptores de contexto innecesarios y que consumen mucho tiempo.

3.7 Resumen NFVI

En consecuencia, la información anterior se representa en el siguiente esquema resumen de la figura 3-7 sobre la infraestructura para NFVI

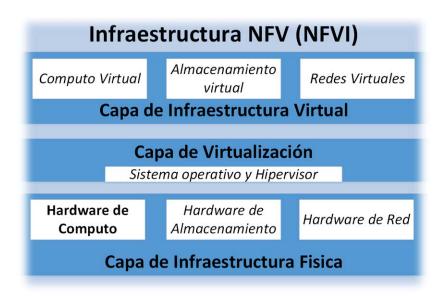


Figura 3-7: Arquitectura NFV

3.8 Soluciones Actuales de SDN

En la siguiente sección se presentan soluciones reales establecidas en el mercado por diferentes compañías del área para la aplicación de SDN y NFV.

3.8.1 Controlador SDN Brocade

Actualmente Brocade cuenta con cuatro soluciones en controladores y aplicaciones para tecnología SDN: Controlador SDN Brocade, Brocade Flow Optimizer, Brocade Flow Manager, Brocade VNF Manager

No obstante, es de interés el controlador SDN Brocade que ofrece por así decirlo la solución modelo. Y es que Brocade al ser miembro de ODL y ONF posee todas las herramientas necesarias para establecer un controlador de implementación general, en consecuencia su Controlador SDN posee una plataforma independiente, así como del sistema operativo anfitrión e hipervisor²-agnóstico, por consiguiente estas características lo establecen como la solución idónea. Cualquier equipo de red (físico o virtual) de cualquier proveedor puede ser operado dentro del dominio del controlador siempre y cuando sea compatible con cualquiera de las interfaces estándar de OpenDaylight southbound o tenga un complemento OpenDaylight. Con el controlador Brocade SDN, las organizaciones de TI pueden introducir gradualmente partes cada vez mayores de su red existente en el dominio del controlador sin interrupciones. [10]

En la figura 3-8 se presenta la topología y el entorno de trabajo para el controlador SDN Brocade.

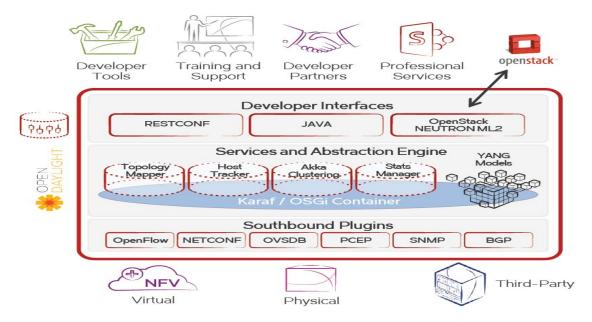


Figura 3-8: Representación del entorno de Trabajo abierto para controlador SDN Brocade [10]

²Hipervisor: es una plataforma que permite aplicar diversas técnicas de control de virtualización para utilizar, al mismo tiempo, diferentes sistemas operativos en una misma computadora.

Detalles técnicos

Los protocolos de comunicación soportados por este controlador, corresponden a OpenFlow principalmente, a BGP, OVSDB y Netconf. Como primer lenguaje de comunicación corresponde a Java y Python, además actualmente es soportado por un sistema operativo Ubuntu o RedHat

3.8.2 Open SDN Ecosystem de HP (Hewlett-Packard)

HP ha logrado establecer servicios de soporte de diseño e implementación para SDN de manera adecuada y escalable, contando ya con el uso del standard OpenFlow desde el año 2011. Al igual que Brocade, Hewlett-Packard mantiene la arquitectura propuesta por ONF como se puede observar en la Figura 3-9, además HP se enfoca en la creación y uso de aplicaciones dando su sello personal

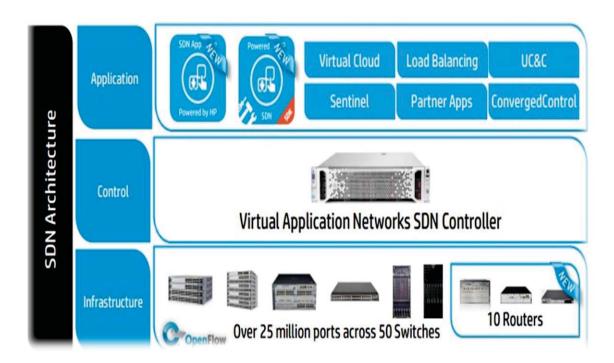


Figura 3-9: Arquitectura SDN propuesta por HP [11]

HP dispone de SDN developer Kit, una herramienta 'open' de desarrollo de aplicaciones SDN, que permite simular y verificar su correcta ejecución, además de contar con SDN app store una tienda online que permite descargar aplicaciones SDN de HP o de terceros. Ideal para el uso en aula. En la figura 3-10 se puede ver la arquitectura de la solución Open SDN ecosystem de HP.

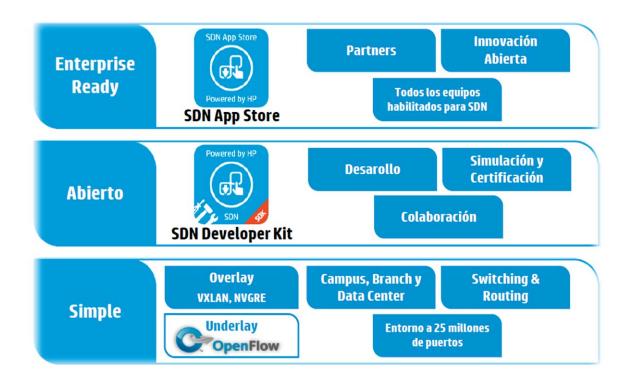


Figura 3-10: Open SDN Eco System [12]

3.8.3 Open MANO

Openmano es la implementación de referencia de un NFV-O (Network Functions Virtualization Orchestrator), que permite la creación de escenarios complejos de red virtual. Interactúa con un VIM de NFV a través de su API y ofrece una interfaz hacia el northbound, basada en REST, donde se ofrecen servicios de NFV, incluida la creación y eliminación de servicios de red o VNF.

También incluye la implementación de referencia OpenVIM que es un NFV VIM (Virtualised Infrastructure Manager), con soporte para un rendimiento alto y predecible. Interactúa con los nodos de computación en la infraestructura de NFV y un controlador de flujo abierto para proporcionar capacidades informáticas y de red y desplegar máquinas virtuales. Ofrece una interfaz orientada al norte similar a OpenStack (API openvim), donde se ofrecen servicios mejorados en la nube, incluida la creación, eliminación y gestión de imágenes, instancias y redes.

Además, Openmano posee una GUI(Graphical User Interface) web para interactuar con la API de openmano de una manera gráfica y fácil de usar. También se proporciona una interfaz de línea de comandos para los usuarios más avanzados. En la Figura 3-11 se puede observar la respectiva interfaz. Cabe destacar que esta solución aún está en desarrollo y es objeto de futuros análisis y desarrollo, no obstante, es una muy buena opción como una solución libre de no pago.

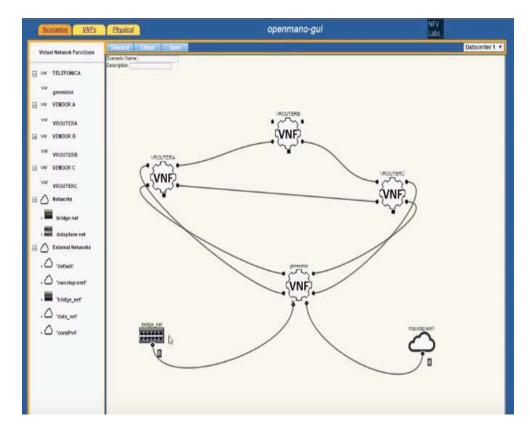


Figura 3-11: Interfaz Gráfica de OpenMano [13]

3.9 Protocolo de Comunicación Existente para SDN y NFV

En el presente capítulo se abordará el protocolo de comunicación (también llamado de control) existente para el uso de las tecnologías SDN y NFV, destacando sus características como mecanismo de comunicación entre controladores y los planos de datos.

3.10 Protocolo OpenFlow

El protocolo Openflow contiene diversas características, componentes y funcionalidades que lo componen, las cuales se detallaran a continuación

3.10.1 Componentes Switch OpenFlow

Para entender este protocolo de control asociado a OpenFlow, se presentan las características y funcionamiento en un dispositivo OpenFlow switch en la figura 3-12:

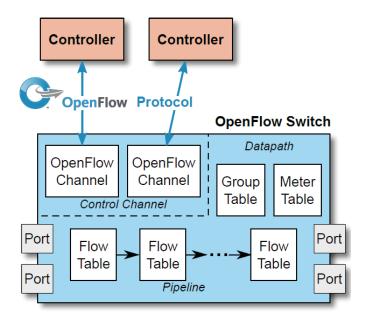


Figura 3-12: Componentes en un dispositivos OpenFlow Switch [14]

Detalladamente se puede ver que los componentes de su estructura se conforman de:

- Controller: Es una entidad que interactúa con el switch OpenFlow utilizando el protocolo OpenFlow. En la mayoría de los casos, un controlador OpenFlow es un software que controla muchos switches lógicosOpenFlow.
- Control Channel: Cumple la agregación de componentes de un switch lógico OpenFlow que gestiona Comunicación con los controladores. El canal de control incluye un canal OpenFlow por controlador OpenFlow.
- OpenFlow Channel: Interfaz entre un switch OpenFlow y un controlador OpenFlow, utilizado por el controlador para gestionar el switch.
- Pippeline: El conjunto de tablas de flujo vinculadas que proporcionan adaptación, reenvío y modificación de paquetes en un conmutador OpenFlow
- Flow Table: Una etapa de la 'pipeline' que contiene entradas de flujo
- Port:Donde los paquetes entran y salen dela 'pipeline' de OpenFlow. Puede ser un puerto físico, un puerto lógico o un puerto previamente reservado y definido por el protocolo del conmutador OpenFlow.
- Group Table: Una lista de segmentos de acción y algunos medios para elegir uno o más de esos segmentos en un paquete base.
- Meter Table: Un elemento de conmutación que puede medir y controlar la velocidad de los paquetes. El medidor activa una banda de medidores si la tasa de paquetes o la velocidad de bytes que pasa a través del medidor excede un umbral predefinido la banda del medidor deja caer el paquete, también se le llama limitador de tráfico.
- Datapath: Corresponde a la agregación de componentes de un switch lógico OpenFlow que están directamente involucrados en el procesamiento y reenvío de tráfico. El datapath incluye tablas de flujo de la 'pippeline', la 'group table' y los 'port'.

3.10.2 Funcionamiento de un Switch OpenFlow

Generalidades

En consecuencia, un switch Openflow lógico consiste en una o más tablas de flujo y una tabla de grupo, que realizan búsquedas de paquetes y reenvío, y uno o más canales OpenFlow a un controlador externo mostrado gráficamente en la figura 4-1. El switch se comunica con el controlador y el controlador gestiona el switch a través del protocolo OpenFlow. Por tanto, mediante el protocolo de conmutación OpenFlow, el controlador puede agregar, actualizar y eliminar entradas de flujo en tablas de flujo, tanto de manera reactiva (en respuesta a los paquetes) como de manera proactiva.

Tabla de flujos y Entradas de Flujo

Cada tabla de flujo en el switch contiene un conjunto de entradas de flujo; Cada entrada de flujo consiste en campos de coincidencia, contadores y un conjunto de instrucciones para aplicar a paquetes coincidentes.

En detalle se puede observar sus correspondientes descripciones en la siguiente tabla 3-3.

Tabla 3-3: Entradas de Flujos en las Tablas de Flujos de un Switch OpenFlow.

Entradas de flujo	Descripción
Campos de coincidencia	Usados para coincidir entre los paquetes. Éstos consisten en el puerto de entrada y los encabezados de paquetes y opcionalmente otros campos de canalización como metadatos especificados por una tabla anterior.
Prioridad	Verifica la coincidencia de precedencia de la entrada de flujo.
Contadores	Variables que son actualizadas cuando los paquetes coinciden.
Instrucciones	Utilizados para modificar el conjunto de acciones o el procesamiento de la 'pipeline'.
Tiempos de salida	Corresponde a la cantidad máxima de tiempo o tiempo de inactividad antes de que el flujo expire por el conmutador.
Cookie	Valor de datos elegido por el controlador. Puede ser utilizado por el controlador para filtrar el flujo Entradas afectadas por estadísticas de flujo, modificación de flujo y peticiones de eliminación de flujo. No se usa cuando se encuentra procesando paquetes.
Flags	Estos se encargan de alterar la forma en que se gestionan las entradas de flujo.

La coincidencia comienza en la primera tabla de flujo y puede continuar con las tablas de flujo adicionales de la "Pipeline". Las entradas de flujo coinciden con los paquetes en orden de prioridad, con la primera entrada coincidente en cada tabla que se utiliza

Si se encuentra una entrada coincidente, se ejecutan las instrucciones asociadas con la entrada de flujo específica. Si no se encuentra ninguna coincidencia en una tabla de flujo, el resultado depende de la configuración de la entrada de flujo de tabla, puesto que el paquete puede ser reenviado a los controladores a través del canal OpenFlow, eliminado o puede simplemente puede continuar a la siguiente tabla de flujo.

En la figura 3-13 se presenta el procesamiento de un paquete Openflow.

En consecuencia, el Paquete entra por el puerto de ingreso 'Ingress Port' encuentra su tabla de flujo coincidente 'Flow table n' realiza la respectiva acción, terminando así el proceso de ingreso del paquete, luego se agrupa en el grupo de tablas para llegar al proceso de egreso del paquete estableciendo su respectiva tabla de flujos de salida 'Flow Table e+m' realizando luego su

respectiva acción de salida, para finalmente terminar el proceso de egreso del paquete y llegar a su puerto de salida 'Output Port'.

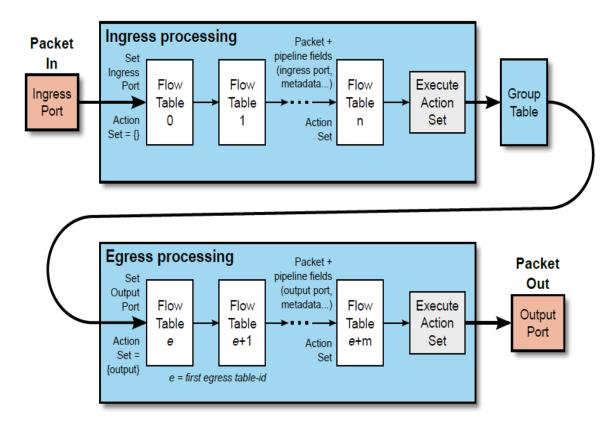


Figura 3-13: Procesamiento de un paquete en un switch OpenFlow [14]

Instrucciones

Las instrucciones asociadas con cada entrada de flujo contienen acciones o modifican el proceso de la "Pipeline". Las acciones incluidas en las instrucciones describen el reenvío de paquetes, la modificación de paquetes y el procesamiento de tablas de grupo. Las instrucciones de proceso 'Pipeline' permiten enviar paquetes a las tablas subsiguientes para procesamiento posterior y permitir que la información, en forma de metadatos, se comunique entre tablas. El procesamiento de la tabla 'pipeline' se detiene cuando el conjunto de instrucciones asociado con una entrada de flujo coincidente no especifica una tabla siguiente; En este punto el paquete es usualmente modificado y reenviado.

En consecuencia, se presentan las instrucciones asignables a las tablas de flujos por medio de las entradas de flujo, estas se encuentran divididas en instrucciones requeridas y opcionales en la tabla 3-4 y tabla 3-5 respectivamente, adicionalmente la figura 3-14 presenta las instrucciones en una tabla de flujos de un Switch OpenFlow.

Tabla 3-4: Instrucciones Requeridas para las tablas de flujo [14]

Instrucciones Requeridas	Descripción	
Clear-Actions	Borra todas las acciones del conjunto de acciones inmediatamente. Esta instrucción sólo se requiere para las entradas de flujo de donde existe una tabla perdida, y es opcional para otras entradas de flujo.	
Write-Actions action(s)	Combina el conjunto especificado de acciones en el Conjunto de acciones actuales. Si una acción del tipo dado existe en el conjunto actual lo sobrescribe, de lo contrario agrega una acción. Esta instrucción debe ser soportada en todas las tablas de flujo.	
Goto-Table next-table-id	Indica la tabla siguiente en el proceso 'Pipeline'. El id de tabla debe ser mayor que el id de tabla actual. Esta instrucción debe ser apoyada en todas las tablas de flujo excepto en laúltima, los switches OpenFlow con una sola tabla de flujo no requieren implementar esta instrucción.	

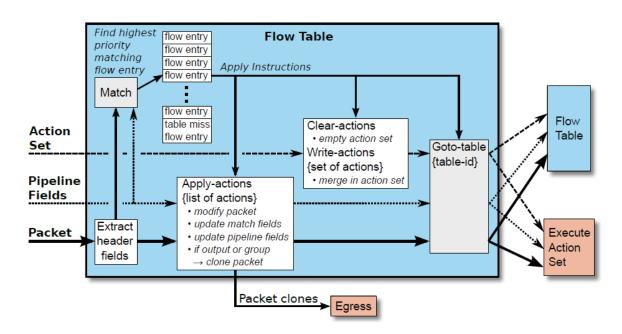


Figura 3-14: Ejecución de Instrucciones en una tabla de flujos de un Switch OpenFlow [14]

Tabla 3-5: Instrucciones Opcionales para la tablas de flujos [14]

Instrucciones Opcionales	Descripción
Apply-Actions action(s)	Aplica la(s) acción(es) específica(s) inmediatamente, Sin ningún cambio en el conjunto de acciones. Esta instrucción puede usarse para modificar el paquete entre dos tablas o para ejecutar múltiples acciones del mismo tipo. Las acciones se especifican como una lista de acciones
Write-Metadata metadata / mask	Escribe el valor de metadatos enmascarados en el campo de metadatos. La máscara especifica qué bits del registro de metadatos deben modificarse (Es decir, nuevos metadatos = metadatos antiguos && máscara máscara de valor).
Stat-Trigger stat thresholds	Generar un evento al controlador si algunas de las estadísticas de flujo cruzan uno de los valores de umbral Stat

Uso de los puertos

Las entradas de flujo pueden reenviarse a un puerto. Este es normalmente un puerto físico, pero también puede ser un puerto lógico definido por el switch o un puerto reservado definido por esta especificación. Los puertos reservados pueden especificar acciones de reenvío genéricas como enviar al controlador, inundación o reenvío utilizando métodos no OpenFlow, como el procesamiento de conmutación "normal", mientras que los puertos lógicos definidos por switch pueden especificar grupos de agregación de enlaces, túneles o Interfaces de bucle de retorno. Las acciones asociadas con las entradas de flujo también pueden dirigir paquetes a un grupo, lo que especifica un procesamiento adicional. Los grupos representan conjuntos de acciones para la inundación, así como una semántica de reenvío más compleja. Como capa general de dirección, los grupos también permiten que múltiples entradas de flujo se envíen a un solo identificador, esta abstracción permite que las acciones de salida comunes a través de entradas de flujo se cambien eficientemente. La tabla de grupo contiene entradas de grupo; Cada entrada de grupo contiene una lista de segmentos de acción con semántica específica dependiente del tipo de grupo. Las acciones en uno o más conjuntos de acciones se aplican a los paquetes enviados al grupo.

3.10.3 Canal de Control (OpenFlow Channel)

OpenFlow ejecuta cierto tipo de mensajes de comunicación, estos mensajes se clasifican en:

- Mensajes Controlador a Switch
- Mensajes Asincrónicos
- Mensajes Simétricos

Estos mensajes se detallarán a continuación:

Mensajes Controlador a Switch

Los mensajes de controlador son iniciados por el controlador y se utilizan para gestionar directamente o inspeccionar el estado del switch OpenFlow.

Features: El controlador puede solicitar la identidad y las capacidades básicas de un conmutador enviando una solicitud de características, el switch debe responder con una respuesta de características que especifica la identidad y la base del mismo switch. Esto se realiza comúnmente al establecer el canal OpenFlow.

Configuration: El controlador es capaz de establecer y consultar parámetros de configuración en el switch. Los Switches sólo responden a una consulta desde el controlador a la vez.

Modify-State: Los mensajes de estado de modificación son enviados por el controlador para gestionar el estado en los conmutadores. El propósito primario es agregar, borrar y modificar entradas de flujo / grupo e insertar / quitar segmentos de acción en las tablas de OpenFlow y para establecer las propiedades del puerto del switch.

Read-State: Los mensajes de estado de lectura son utilizados por el controlador para recopilar la configuración actual, estadísticas y capacidades. La mayoría de las peticiones y respuestas de Read-State se implementan usando secuencias de mensajes en varias partes

Packet-Out: Estos son utilizados por el controlador para enviar paquetes a un puerto específico en el switch, en respuesta a paquetes recibidos por medio de mensajes Packet-In. Mensajes de Packet-Out deben contener un paquete completo o un ID de búfer y hacer referencia a un paquete almacenado en el switch. El mensaje también debe contener una lista de acciones que se aplicarán en el orden en que son especificadas.

Barrier request/reply: Los mensajes Barrier de request/reply son utilizados por el controlador para garantizar que se cumplan las dependencias del mensaje o recibir notificaciones de las operaciones realizadas.

Role-Request: los mensajes de solicitud de función son utilizados por el controlador para establecer la función de su OpenFlowCanal, establecer su ID de controlador, o realizar consultas en estos. Esto es útil cuando el switch se conecta a Controladores múltiples.

Asynchronous-Configuration: El controlador utiliza estos mensajes para establecer un filtro adicional en los mensajes asíncronos que desea recibir en su canal OpenFlow o para consultar ese filtro. Esto es útil sobre todo cuando el conmutador se conecta a múltiples controladores y se realiza normalmente al establecer el canal OpenFlow.

Mensajes Asincrónicos

Los mensajes asíncronos son iniciados por el conmutador y utilizados para actualizar el controlador acerca de eventos de red y cambios en el estado del conmutador.

Packet-in: Para todos los paquetes que no tienen una entrada de flujo que concuerde con el paquete entrante, o si la acción que corresponde a la entrada coincidente es reenviar el paquete al controlador.

Flow-Removed: Informa al controlador acerca de la eliminación de una entrada de flujo de una tabla de flujo.

Port-status: Mensaje arrojado al cambiar la configuración de un puerto.

Controller-Status: Informar al controlador cuando cambia el estado de un canal OpenFlow. Por tanto, envía estos mensajes a todos los controladores cuando cambia el estado del canal OpenFlow ante cualquier cambio, esto facilita al procesamiento de switching por error si los controladores pierden la capacidad de comunicarse entre sí.

Flow-Monitor: Informa al controlador de un cambio en una tabla de flujo. Un controlador puede definir un conjunto de monitores para rastrear los cambios en las tablas de flujo.

Mensajes Simétricos

Los mensajes simétricos son iniciados por el switch o el controlador y se envían sin solicitud.

Hello: se intercambian mensajes entre el switch y el controlador al iniciar la conexión.

Echo request/reply: Los mensajes de solicitud / respuesta pueden enviarse desde el switch o el controlador y deben devolver una respuesta de eco. Se utilizan principalmente para verificar la una conexión controlador-switch y también se puede utilizar para medir su latencia o ancho de banda.

Error: Los mensajes de error son utilizados por el conmutador o el controlador para notificar los problemas al otro lado de la conexión. Son utilizados principalmente por el switch para indicar un fallo de una petición iniciada por el controlador.

Experimenter: Estos mensajes proporcionan una forma estándar para que los conmutadores de OpenFlow ofrezcan funcionalidad adicional dentro del espacio de tipo de mensaje OpenFlow. Se trata de un área de puesta en escena para las características de las futuras revisiones de OpenFlow.

3.10.4 Secuencia de Mensajes Protocolo OpenFlow

El correspondiente funcionamiento de las tramas para el protocolo OpenFlow entre el controlador y el switch, es decir la respectiva secuencia de mensajes que se intercambian y su respectivo funcionamiento en paralelo se presentan en la Figura 3-15.

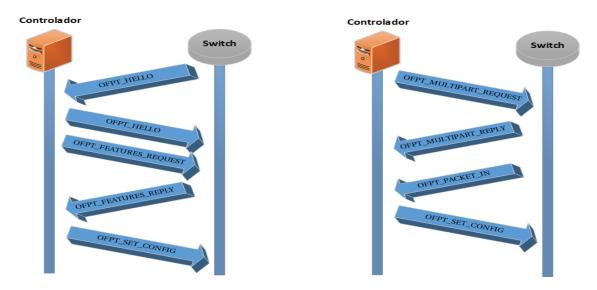


Figura 3-15: Mensajes de comunicación entre controlador y switch

A continuación, se detallan los correspondientes mensajes de comunicación

Connection Setup

El interruptor inicia una conexión TCP (o TLS) estándar al controlador. Cuando se establece una conexión OpenFlow, cada entidad debe enviar un mensaje OFPT_HELLO con la versión del protocolo establecida en la versión más alta del protocolo OpenFlow admitida por el remitente. En la figura 3-16 a continuación, se puede ver que se ha negociado la versión 1.3 de OpenFlow.

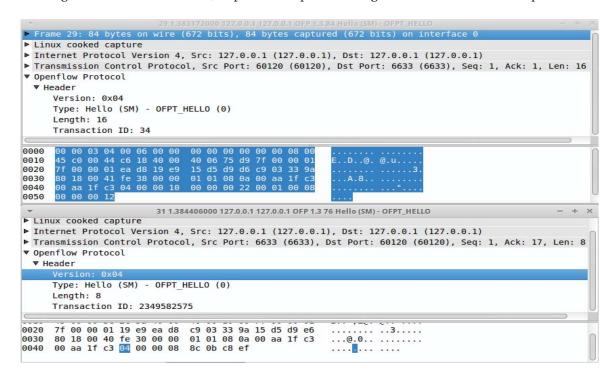


Figura 3-16: OFTP_HELLO

Feature Request - Reply

Después de establecer con éxito una sesión, el controlador envía un mensaje OFPT_FEATURES_REQUEST como se muestra en la figura 3-17. Este mensaje solo contiene un encabezado de OpenFlow y no contiene un cuerpo.

El interruptor responde con un mensaje OFPT_FEATURES_REPLY como se muestra en la figura 3-18. Se observa claramente la Id. De Datapath y las capacidades de cambio enviadas como parte del mensaje de respuesta de las respectivas funciones y su estado.

```
▼ 36 9.906392000 127.0.0.1 127.0.0.1 OFP1.3 76 Features request (CSM) - OFPT_FEATURES_REQUEST — + ×

▶ Frame 36: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0

▶ Linux cooked capture

▶ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)

▶ Transmission Control Protocol, Src Port: 6633 (6633), Dst Port: 59610 (59610), Seq: 9, Ack:

▼ Openflow Protocol

▼ Header

Version: 0x04

Type: Features request (CSM) - OFPT_FEATURES_REQUEST (5)

Length: 8

Transaction ID: 745755500
```

Figura 3-17: Mensaje OFPT_FEATURES_REQUEST

```
38 9.906693000 127.0.0.1 127.0.0.1 OFP 1.3 100 Features reply (CSM) - OFPT_FEATURES_REPLY
Frame 38: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface 0
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▶ Transmission Control Protocol, Src Port: 59610 (59610), Dst Port: 6633 (6633), Seq: 17, Ack
▼ Openflow Protocol
 ▼ Header
    Version: 0x04
    Type: Features reply (CSM) - OFPT FEATURES REPLY (6)
    Length: 32
    Transaction ID: 745755500
 ▼ Feature Reply
    Datapath ID: 0x00000000000000001
    Buffers: 256
    Tables: 254
    Auxiliary ID: 0
    Padding
  ▼ Capabilities: 0x00000047
     0000 0000 0000 0000 0000 000. 0..0 .... = Reserved: Not set
     .... 1 = Flow statistics: Set
     .... .... .... .... .1.. = Port statistics: Set
     .... - Group statistics: Not set
     .... = Can reassemble IP fragments: Not set
     .... = Queue statistics: Set
     .... = Switch will block looping ports: Not set
    Reserved: 0
```

Figura 3-18: Mensaje OFPT_FEATURES_REPLY.

Set Configuration

A continuación, en la figura 3-19 el controlador envía el mensaje OFPT_SET_CONFIG al conmutador. Esto incluye el conjunto de indicadores y los bytes máximos del paquete que el datapath debe enviar al controlador.

```
43 9.908042000 127.0.0.1 127.0.0.1 OFP 1.3 80 Set config (CSM) - OFPT_SET_CONFIG
                                                                                       - + ×
Frame 43: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface 0
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▶ Transmission Control Protocol, Src Port: 6633 (6633), Dst Port: 59610 (59610), Seq: 17, Ack
▼ Openflow Protocol
 ▼ Header
     Version: 0x04
     Type: Set config (CSM) - OFPT_SET_CONFIG (9)
     Length: 12
     Transaction ID: 745755501
 ▼ Switch Configuration
   ▼ Flags: 0x0000
      0000 0000 0000 0... = Reserved: Not set
       .... .... ...0 = Drop fragments: Not set
      .... .... ... ... ... = Reassemble (only if OFPC IP REASM set): Not set
       .... .... .0.. = Send packets with invalid TTL to the controller: Not set
     Max new flow bytes to controller: 128
```

Figura 3-19: Mensaje Set Configuration

Multipart Request - Reply

El controlador puede solicitar el estado desde la ruta de datos utilizando el mensaje OFPT_MULTIPART_REQUEST, el cual se presenta en la figura 3-20. Los tipos de mensajes manejados por este mensaje incluyen varias estadísticas (FLUJO / TABLA / PUERTO / COLA / MEDIDOR, etc.) o características de descripción (METER_CONFIG / TABLE_FEATURES / PORT_DESC, etc.).

```
45 9.947447000 127.0.0.1 127.0.0.1 OFP 1.3 84 Multipart request (CSM) - OFPT_MULTIPART_
Frame 45: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on int
► Linux cooked capture
▶ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (1
▶ Transmission Control Protocol, Src Port: 6633 (6633), Dst Port: 59610 (596

    Openflow Protocol

  ▼ Header
     Version: 0x04
     Type: Multipart request (CSM) - OFPT MULTIPART REQUEST (18)
     Length: 16
     Transaction ID: 745755502
 ▼ Multipart request
     Type: Port description - OFPMP_PORT_DESC (13)
   ▼ Flags: 0x0000
                 .... 0 = More requests to follow: Not set
     Padding
     Body: <MISSING>
```

Figura 3-20: OFPT_MULTIPART_REQUEST

El switch responde con el PORT_DESCRIPTION de todos los puertos activos en el switch mostado en la figura 3-21.

```
▼ 47 9.947761000 127.0.0.1 127.0.0.1 OFP 1.3 340 Multipart reply (CSM) - OFPT MULTIPAI - + ×
▶ Frame 47: 340 bytes on wire (2720 bits), 340 bytes captured (2720 bits)
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 59610 (59610), Dst Port: 6633
▼ Openflow Protocol
 ▼ Header
     Version: 0x04
     Type: Multipart reply (CSM) - OFPT MULTIPART REPLY (19)
     Length: 272
     Transaction ID: 745755502
 ▼ Multipart reply
     Type: Port description - OFPMP PORT DESC (13)
   ▼ Flags: 0x0000
                 .... 0 = More replies to follow: Not set
     Padding
     Body: 000000030000000768ee2cd0bd3000073312d6574683300...
```

Figura 3-21: Mensaje OFPT_MULTIPART_REPLY

Flow Mod

Los flujos pueden ser proactivos (por ejemplo, flujos de preinstalación como TableMissFlow) o reactivos (por ejemplo, reaccionar por mensajes packet_in) enviados desde el controlador. Los mensajes de modificación de la tabla de flujo pueden tener los siguientes tipos: OFPFC_ADD, OFPFC_DELETE_STRICT, OFPFC_MODIFY, OFPFC_MODIFY_STRICT. En la figura 3-22 se presenta mensaje OFPFC_ADD.

```
86 10.458036000 127.0.0.1 127.0.0.1 OFP 1.3 164 Flow mod (CSM) - OFPT_FLOW_MOD - + ×
▼ Openflow Protocol
 ▼ Header
     Type: Flow mod (CSM) - OFPT_FLOW_MOD (14)
     Length: 96
     Transaction ID: 2349582581
     Cookie: 0x0000000000000000
     Cookie Mask: 0x0000000000000000
     Table ID: 0
     Command: New flow - OFPFC_ADD (0x00)
     Idle Timeout: 0
     Hard Timeout: 0
     Priority: 1
     Buffer ID: 0xffffffff
     Output Port: 0
     Output Group: 0
     Flags: 0x0000
     Padding
    Match
       Type: OpenFlow Extensible Match - OFPMT_OXM (0x0001)
       Length: 22
      OXM field
     ► OXM field
       Padding
   ▶ Instruction
```

Figura 3-22: Mensaje OFPFC_ADD de tipo modificación OFTP_FLOW_MOD

Set Async Configuration Message

Los mensajes asíncronos se envían desde un interruptor al controlador. El conjunto de mensajes admitidos por el protocolo OpenFlow incluye mensajes "Packet-Ins, Flow-Removed, Port-Status or Error". Cuando el interruptor se conecta al controlador, el controlador puede establecer el tipo de mensajes que desea recibir en su canal OpenFlow. En la figura 3-23 se presenta como se encuentra constituido el mensaje OFTP SET ASYNC.

```
16 0.040332000 127.0.0.1 127.0.0.1 OFP 1.3 100 Set async (CSM) - OFPT_SET_ASYNC
▶ Frame 16: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface 0
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▶ Transmission Control Protocol, Src Port: 6633 (6633), Dst Port: 58294 (58294), Seq: 125, Ack: 321, Len: 32
▼ Openflow Protocol
 ▶ Header
 ▼ Async config
  ▼ Packet In Mask (for equal, master): 0x00000006
     .... 0 = No matching flow: Not set
     .... .... ... ... ... .1.. = Packet has invalid TTL: Set
  ▶ Packet In Mask (for slave): 0x00000000
  ▼ Port Status Mask (for equal, master): 0x00000007
     .... 1 = The port was added: Set
     .... .... ... ... ... ... ... .1.. = Some attribute of the port has changed: Set
     0000 0000 0000 0000 0000 0000 0000 0... = Reserved: Not set
  ▶ Port Status Mask (for slave): 0x00000000
  ▼ Flow Removed Mask (for equal, master): 0x00000007
     .... 1 = Flow idle time exceeded idle_timeout: Set
     .... .... ... ... ... .1.. = Evicted by a DELETE flow mod: Set
     .... 0... = Group was removed: Not set
     0000 0000 0000 0000 0000 0000 0000 .... = Reserved: Not set
  ▶ Flow Removed Mask (for slave): 0x00000000
```

Figura 3-23: Mensaje Set Async Configuration (OFTP_SET_ASYNC)

3.10.5 Observaciones

Los diseñadores de switches son libres de implementar los componentes internos de cualquier manera conveniente, siempre que se conserven la semántica de concordancia y de instrucciones correctas. Por ejemplo, aunque una entrada de flujo puede utilizar un grupo de todos para enviar a varios puertos, un diseñador de conmutadores puede optar por implementar esto como una máscara de bits única dentro de la tabla de reenvío de hardware. Existen dos tipos de Switches que utilizan Openflow: Switch solamente OpenFlow: y los switch OpenFlow-híbridos.

Los switches solamente OpenFlow soportan solo operaciones OpenFlow, en esos switches todos los paquetes son procesados por el Canal de OpenFlow, y no pueden ser procesados de otra manera.

Los switches OpenFlow-híbridos soportan tanto el funcionamiento OpenFlow como el funcionamiento normal de switching Ethernet, es decir, switching Ethernet L2, VLAN, routing L3, ACL y procesos de QoS. Estos switches proporcionan mecanismos de clasificación fuera de OpenFlow para encaminar el tráfico, ya sea por el canal de OpenFlow o por el canal tradicional.

4 Plataformas y Entorno de Trabajo

En esta sección se presentan las herramientas disponibles para las plataformas de simulación tanto de NFV y SDN detallando sus características, usos y modo de instalación, cabe destacar que se han seleccionado dichas herramientas enfocado en el ámbito de su implementación en la docencia para trabajar y ser implementado en un entorno Ubuntu/Linux.

4.1 Mininet

Mininet corresponde a un sistema de emulación de red enfocado en la administración y organización de un entorno SDN, pues se encarga de virtualizar hosts, Controladores SDN y switches Openflow, es decir genera una red completa con una respectiva topología.

A su vez Mininet permite crear y realizar pruebas de topologías diversas ya sean complejas o no, ya que soporta arbitrarias topologías personalizas e incluye un conjunto básico de topologías parametrizadas, se destaca también la variedad de formas para hacer andar esta plataforma puesto posee una VM(Virtual Machine) disponible para ser instalada en VirtualBox o Vmware (estos programas son herramientas de virtualización multiplataforma disponible para Windows, Linux y Mac OS X u otros sistemas operativos, siendo solo Virtual Box de código abierto), además Mininet puede ser instalado de forma nativa desde Linux.

Las redes de Mininet ejecutan código real incluyendo las aplicaciones de red Unix / Linux estándar, así como el kernel y la pila de red reales de Linux (incluidas las extensiones de kernel que pueda tener disponibles, siempre y cuando sean compatibles con espacios de nombres de red).

Mininet utiliza la virtualización basada en procesos para ejecutar muchos hosts (arrancado con éxito hasta 4096) y los switches en un solo kernel del SO, puede crear conmutadores OpenFlow del kernel o del usuario, controladores para controlar los conmutadores y hosts para comunicarse a través de la red simulada. Mininet conecta conmutadores y hosts usando pares virtuales ethernet. Aunque Mininet actualmente depende del kernel de Linux, en el futuro podrá soportar otros sistemas operativos con virtualización basada en procesos, como los contenedores de Solaris o FreeBSD.

Se acentúa como un programa Open-Source por las razones anteriores y que por tanto se encuentra en un constante perfeccionamiento pues el código de Mininet es casi enteramente Python, excepto para una utilidad C breve.

No obstante, Mininet usa un único núcleo de Linux para todos los hosts virtuales. Por lo que no se puede ejecutar software que dependa de Windows u otros sistemas operativos. Necesidad de utilizar un controlador OpenFlow independiente, tampoco posee una noción de tiempo virtual, esto quiere decir que las medidas estarán basadas en tiempo real y por lo tanto los resultados serán más rápidos que en la situación real. [15]

4.2 ODL (OpenDayLight)

El proyecto OpenDaylight es una plataforma de código abierto para Software Defined Networking (SDN) que utiliza protocolos abiertos para proporcionar control centralizado y programático y monitorización de dispositivos de red. Al igual que muchos otros controladores SDN, OpenDaylight es compatible con OpenFlow, además de ofrecer soluciones de red listas para instalar como parte de su plataforma.

Por mucho que su sistema operativo proporcione una interfaz para los dispositivos que componen su computadora, OpenDaylight proporciona una interfaz que le permite conectar dispositivos de red rápida e inteligentemente para un rendimiento óptimo de la red.

Es extremadamente útil entender que la configuración de su entorno de red con OpenDaylight no es una sola instalación de software. Mientras que su primer paso cronológico es instalar OpenDaylight, instalará funcionalidades adicionales como las características de 'Karaf' (conjunto de funciones activables en la plataforma) para satisfacer sus necesidades específicas.

Las principales distinciones de SDN de OpenDaylight en comparación con las opciones tradicionales de SDN son las siguientes:

- Posee un complemento que proporciona conectividad a dispositivos a través de los protocolos OpenFlow o BGP
- Un conmutador L2 o un servicio como Autenticación, Autorización y Contabilidad (AAA).
- Soporte para una gama amplia y creciente de protocolos de red más allá de OpenFlow, incluyendo SNMP, NETCONF, OVSDB, BGP, PCEP, LISP y más.
- Soporte para el desarrollo de nuevas funcionalidades compuestas por protocolos y servicios de red adicionales. [2]

4.3 GNS3 (Version 2.0)

GNS3 es una plataforma de emulación y configuración para probar y solucionar problemas de redes virtuales y reales. GNS3 permite ejecutar una pequeña topología que consiste en sólo unos pocos dispositivos en su computador, a aquellos que tienen muchos dispositivos alojados en varios servidores o incluso alojados en la nube.

GNS3 es software libre, con sistemas operativo multiplataforma programado en Python

Adicionalmente GNS3 prueba y verifica implementaciones del mundo real, pues originalmente se creó con el fin estudiar para certificaciones CCNP, Esto en vista que soporta muchos dispositivos de varios proveedores de red, incluyendo switchs virtuales Cisco, Cisco ASAs, vRouters de Brocade, switches Cumulus Linux, instancias Docker, VSRs HPE, múltiples aplicaciones Linux y entre otros. Las aplicaciones complementarias anteriormente nombradas se encuentran disponibles para su descarga en el sitio oficial GNS3 [16]

4.4 WireShark

Wireshark es un analizador de protocolos open source que actualmente está disponible para plataformas Windows y Unix. su principal objetivo es el análisis de tráfico, además de ser una excelente aplicación didáctica para el estudio de las comunicaciones y para la resolución de problemas de red.

Wireshark implementa una amplia gama de filtros que facilitan la definición de criterios de búsqueda para los más de 1100 protocolos soportados actualmente y todo ello por medio de una interfaz sencilla e intuitiva que permite desglosar por capas cada uno de los paquetes capturados. Gracias a que Wireshark "entiende" la estructura de los protocolos, se puede visualizar los campos de cada una de las cabeceras y capas que componen los paquetes monitorizados, proporcionando un gran abanico de posibilidades al administrador de redes a la hora de abordar ciertas tareas en el análisis de tráfico. De forma similar a Tcpdump, Wireshark incluye una versión en línea de comandos, denominada Tshark aunque el presente documento se centrará únicamente en su versión gráfica.

Es importante indicar también que las funcionalidades utilizadas en el presente informe solo representan una pequeña parte de todo el potencial que puede ofrecernos Wireshark, y cuyo objetivo principal es servir de guía orientativa para cualquier administrador que necesite detectar, analizar o solucionar anomalías de red. Pueden existir situaciones en las que Wireshark no sea capaz de interpretar ciertos protocolos debido a la falta de documentación o estandarización de los mismos, en cuyo caso la ingeniería inversa será la mejor forma de abordar la situación. Otras herramientas como SnortOSSIM así como multitud de IDS/IPS permiten alertar sobre algunos de los problemas y ataques expuestos en esta guía. No obstante, cuando se necesita analizar tráfico en profundidad o hay que auditar un entorno en el que el tiempo prima, dichas herramientas suelen carecer de la flexibilidad que nos ofrece un analizador de protocolos como Wireshark. [17]

4.5 Instalación Plataformas de Trabajo

A continuación, se presentan los procedimientos detallados para la instalación de las respectivas plataformas de trabajo en un entorno de trabajo Ubuntu/Linux todas para su uso en las futuras experiencias. Se prefiere un sistema operativo de distribución Linux (OS) con el fin de aprovechar los recursos óptimos de hardware, no obstante, se pretende abordar el uso en OS como Windows en las respectivas experiencias.

4.5.1 GNS3 (Version 2.0)

Para completar la instalación de GNS3 sobre una distribución Linux se deben ejecutar las siguientes secuencias de comandos en una terminal

```
sudo add-apt-repository ppa : gns3 /ppa
sudo apt-get update
sudo apt-get install gns3-gui
```

Una vez instalado y actualizado el repositorio se ejecutará:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com
--recv-keys A2E3EF7B
sudo apt-get update
sudo apt-get install gns3-gui
```

4.5.2 WireShark

Para la instalación del Wireshark basta con escribir en la ventana de comandos el siguiente comando:

```
1 sudo apt-get install wireshark
```

4.6 Experiencias de Laboratorio

En la presente sección se establecerán los criterios utilizados para la creación de guías para el alumnado y los docentes, con el fin de que estas formen parte del primer peldaño para abrir el mundo de posibilidades que ofrecen las tecnologías SDN y NFV en el área de redes de computadoras, y que indicaran el uso delas mismas.

4.7 Diseño de las experiencias de laboratorio

Las experiencias se encuentran definidas para desarrollar los conocimientos y habilidades del alumnado, definiendo estas mismas bajo los conceptos necesarios para realizar su ejecución y su respectiva asimilación.

Las experiencias se verán constituidas por objetivos, escenario, desarrollo practico en aula, cuestionario teórico y práctico, además de incluir pre-informes antes de la realización de las mismas experiencias con el fin de preparar al alumnado desde la sesión anterior y que obtengan noción de los temas a tratar en el aula de manera previa a su realización.

A continuación, se dispone a detallar los elementos anteriormente mencionados y su respectiva función en la ejecución de las mismas guías.

Los objetivos, corresponderán al propósito específico por el cual se realizarán las experiencias, es decir los conocimientos o competencias que se brindaran en la correspondiente guía.

El escenario, cumple la función de entender la totalidad de la tarea a realizar, es decir es una visión más general respecto a los objetivos deseados y del trabajo a realizar.

El desarrollo práctico de la experiencia, tiene la función de establecer los tópicos por medio de pasos a realizar, con el fin de generar las herramientas para desarrollar las evaluaciones posteriores y entender los tópicos que estas involucran.

En consecuencia, los cuestionarios teórico y práctico se basan en el trabajo realizado durante el desarrollo práctico, por tanto, estos no poseen un nivel de dificultad mayor a los conocimientos brindados en el desarrollo de la actividad.

Adicionalmente se dispone de pre-informes como apoyo y reforzamiento a los tópicos tratados dentro de la experiencia, estos estarán constituidos también por objetivos, escenario, contenido y un cuestionario de investigación.

Respecto al contenido, hace referencia a los temas importantes que se abordaran para la experiencia de laboratorio próxima a realizar, mientras que el cuestionario de investigación, constituye tanto un método de evaluación como a una herramienta para generar el interés en el alumnado respecto a los tópicos a tratar.

4.8 Temas seleccionados para las experiencias

Las experiencias de laboratorio y pre-informes se encuentran en el apéndice A y B de este proyecto. A continuación, se detalla a grandes rasgos sus características y los tópicos tratados.

4.8.1 Pre-informe y Experiencia N°1 (Apéndice D.1 y D.2)

Ambas guías se basan en la introducción al sistema operativo Ubuntu-Linux, considerando sus comandos, entorno de trabajo e instalación de aplicaciones para estudiar las topologías en redes SDN y NFV. Se instalarán mininet y ODL como primer contacto a este tipo de tecnología, y cuya profundidad será abordada en las siguientes experiencias. Se generan una topología básica y se dedicara a entender el entorno de trabajo en el cual se desempeña.

4.8.2 Pre-informe v Experiencia N°2 (Apéndice D.3 v D.4)

Ambas guías se dedican a asociar la tecnología NFV por medio VmWare, Gns3, ODL y mininet en un sistema operativo Windows, donde el entorno de trabajo implica una mayor complejidad en cuanto al desarrollo e instalación de estas aplicaciones, además todas estas plataformas convergen en GNS3, se generará también una red más compleja en mininet, visualizando conjuntamente en ODL.

4.8.3 Pre-informe y Experiencia N°3(Apéndice D.5 y D.6)

Ambas guías se enfocan en OpenFlow y su relación con mininet y ODL, se utilizan herramientas como Wireshark para observar las conexiones, además de ahondar en comandos más avanzados dentro de mininet, dichos comandos permiten modificar conexiones además de modificar las tablas que transmite el mismo lenguaje OpenFlow y adicionalmente se generara una topología a través de lenguaje Python.

4.9 Observaciones

Para Finalizar las guías planteadas son un apronte al potencial de trabajo y estudio que puede llevarse a cabo con esta tecnología, se recomienda, entregar de forma digital este material, debido a los diversos links que estos incluyen y que facilitarían un desarrollo más ameno de las actividades

5 Estudio Económico para la Implementacion de Laboratorio SDN

En la presente sección se realiza el estudio de la factibilidad para la implementación de un futuro laboratorio de tecnología SDN y NFV, con el fin de poder abrir un mundo de posibilidades para su estudio y desarrollo dentro del área de redes de computadores y, por tanto, seguir generando desarrollo al respecto, En el presente capitulo se prosigue a analizar y considerar todos los puntos importantes para su respectiva implementación.

5.1 Factibilidad Operacional

El fundamento de implementar SDN, se debe a los beneficios que esta ofrece con respecto a las redes tradicionales. Al optar por este tipo de tecnología se puede suplir ciertas necesidades que actualmente sufren las redes y subredes de una red establecida, siendo una oportunidad generar una red SDN para implementar en las redes de las instalaciones de la Escuela de Ingeniería Eléctrica (EIE) de la PUCV

Al seleccionar la implementación de SDN en el esquema actual ayudaría a disminuir y eliminar ciertos problemas que actualmente sufre las redes en el ámbito laboral y estudiar este campo con fines de la institución educativa y futuro de aplicación en el campo laboral.

Se puede visualizar su fácil manejo y su conveniencia según los resultados arrojados durante el proceso de simulación. Este estudio se establece como apoyo y haga más cómodo y ágil la administración de las redes de datos para los alumnos de la Escuela de Ingeniería Eléctrica de la PUCV.

5.2 Factibilidad Técnica

Desde el punto de vista técnico: A nivel Hardware la factibilidad técnica de la implementación de esta tecnología SDN considera la posibilidad de generar un laboratorio adecuado y que beneficie e incentive al alumnado y docencia en la tecnología SDN. Para esto, los equipos de red deben ser compatibles y la Universidad deberá invertir en la adquisición de equipos de comunicación que

soporten el protocolo OpenFlow que es el que permite centralizar la administración y control de acceso a los diferentes dispositivos activos de la red de datos.

Además, cabe recordar que lo ideal es utilizar equipos de comunicación completamente compatibles con el protocolo OpenFlow, pero se puede utilizar también equipos híbridos que soporten el protocolo OpenFlow y que permitan configurar los protocolos normales dentro del equipo de la red de comunicaciones.

Se debe contemplar a su vez la escalabilidad de los equipos, sus características de funcionamiento para el entorno de trabajo y su respectiva garantía y/o soporte, otorgando así la mayor calidad y durabilidad de los equipos.

5.3 Factibilidad Económica

Para la factibilidad económica se evalúa dos posibilidades para la correspondiente propuesta, es decir la implementación de un laboratorio inicial de bajo costo y fácil aplicación, y por otro lado la Instalación de un laboratorio robusto y con una alta escalabilidad. Considerando en ambos casos un entorno de trabajo OpenSource para evitar cualquier pago de licencia, que no corresponda al soporte o garantía de los equipos como tal.

5.3.1 Propuesta de bajo costo y facil implementación

En la presente propuesta se fundamenta en generar un controlador de red por medio de raspberry pi 3 modelo B y así establecer conexión SSH a la plataforma mininet instalada dentro de ella desde el sistema operativo Linux, a continuación, se presenta la topología resumida en la figura 5-1:

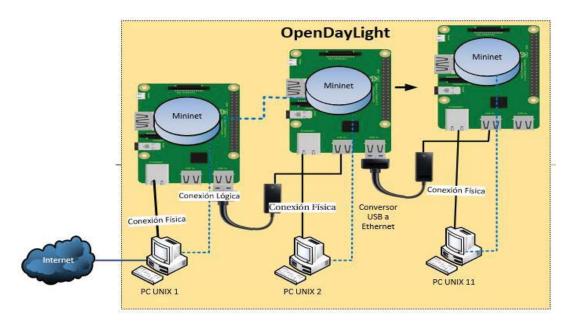


Figura 5-1: Diagrama de topología típica a implementar con Raspberry pi, mininet y ODL

Para la presente propuesta se comparará los precios tanto para las raspberry pi 3 modelo B, como para los adaptadores USB a RJ45 para agregar escalabilidad a la propuesta actual, considerando su respectivo valor en precios chilenos (CLP) presentado en las tablas 5-1 y 5-2:

Tabla 5-1: Cotizaciones para Raspberry pi 3

Componentes	Cantidad	Pcfactory	Mci Electronics	Altronics
Raspberry Pi 3 modelo B	1	\$39.000	\$33.800	\$33.790

Tabla 5-2: Cotizaciones para Conversor USB a RJ45

Componentes	Cantidad	Pcfactory	Wei Computación	Mercado Libre
Conversor USB a RJ45	1	\$9.490	\$19.990	\$3.800

Con lo cual se puede seleccionar las mejores opciones y corresponder la cantidad con los requisitos del laboratorio para el respectivo cálculo del valor final de la implementación de esta propuesta, la tabla 7-3 indica los valores finales estimados para su realización.

Tabla 5-3: Costos totales para la propuesta basada en raspberry pi

Componentes	Cantidad	Costo	Distribuidor
Raspberry Pi 3 modelo B	11	\$371.690	Altronics
Conversor USB a RJ45	22	\$83.600	Mercado Libre
	Total	\$455.290	

5.3.2 Propuesta robusta y de gran escalabilidad

La presente propuesta contempla la implementación de equipos de gama media alta para su uso en el actual laboratorio de redes digitales de la respectiva universidad, las características presentes en esta permiten no sólo virtualizar switches. Sino que también contempla la virtualización de más funcionalidades, que la anterior propuesta no permite entre ellas, como vrouter, Balance de cargas, entre otros. Dado esto último es que esta propuesta presenta una alta escalabilidad ya que puede agregar futuras funcionalidades, generando un espacio de trabajo más adecuado a la realidad presente en los datacenters actuales en la mayoría de las empresas en la

vanguardia de esta tecnología. A continuación, En la figura 5-2 se presenta el esquema topológico para el laboratorio correspondiente:

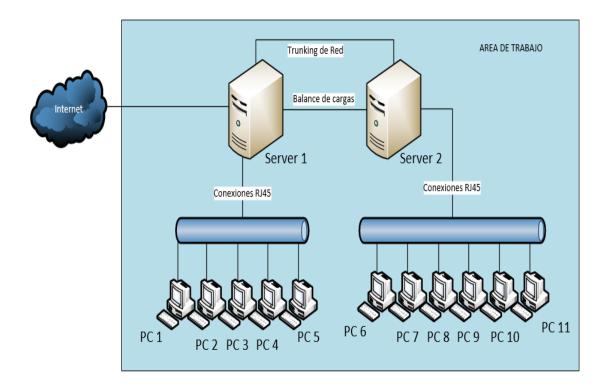


Figura 5-2: Topología de red para área de trabajo [18]

La topología presentada en la figura 7-2 corresponde a la topología idónea y más usualmente utilizada al momento de considerar la infraestructura y recursos disponibles en el actual laboratorio de sistemas 'digitales A' en la Escuela de Ingeniería Eléctrica de la PUCV. Su selección se determinó gracias a la ayuda de una consultoría dictada por Don José Antonio Cartes Valenzuela , ingeniero en sistemas y docente en el área de arquitectura y ciberseguridad en redes de computadores y cuyo contacto se encuentra en la sección bibliográfica [19] de este Proyecto y tomando en cuenta el Cabe destacar que las funciones de firewall, router y switching se encuentran virtualizadas dentro del servidor, pues he aquí el beneficio de utilizar esta tecnología, por consiguiente, la conexión con los equipos PCs son de manera lógica administrada por el correspondiente servidor por medio del hypervisor y el respectivo orquestador. A continuación, en la figura 7-3 se presenta en detalle tanto las funcionalidades empleadas como máquina virtual (VM), la relación lógica y física con los equipos para la respectiva propuesta.

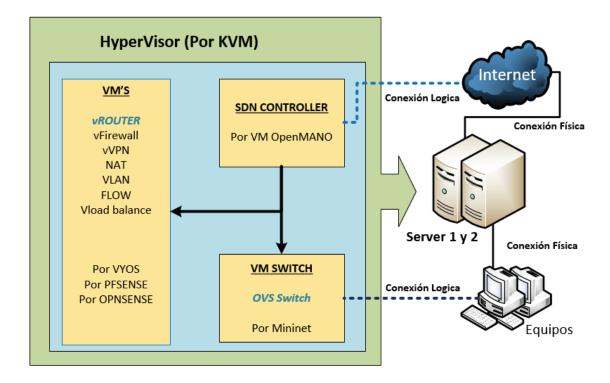


Figura 5-3: Esquema en detalle de la representación de los servidores y sus funcionalidades [18].

La figura 5-3 muestra cómo se separan los componentes que constituyen el servidor, siendo el Hypervisor el encargado de gestionar y generar las variadas virtual machine (VM), seleccionar un orquestador para estas mismas y establecer las conexiones lógicas necesarias para su funcionamiento.

Todos los elementos que componen el servidor ya sean físicos o de servicios vienen optimizados para el correcto funcionamiento y estudio, además de ser una inversión futura escalable y que no se encuentra ligada a la compra de alguna licencia y/o servicio que venga con el servidor en sí.

Para esta propuesta se presenta la opción más adecuada ofrecida por la empresa DELL, cotización que permitió la selección de las respectivas características, su dimensionamiento y requisitos para asegurar su escalabilidad dentro del entorno SDN y NFV para su aplicación en el respectivo laboratorio de redes digitales en el espacio proporcionado por la Universidad.

No obstante, los requerimientos y las características aumentan su valor considerablemente alcanzando el valor en dólares de \$US 12.936,1 dólares, lo que corresponde aproximadamente a \$CLP 8.593.000 pesos chilenos.

A continuación, en la tabla 5-4 los elementos que constituyen físicamente al servidor indicando su respectivo modelo y un detalle indicando su función, característica y precio.

Tabla 5-4: Elementos del servidor que componen la propuesta y su detalle.

Elementos del servidor		Detalle	Precio unitario e US
Pow erEdge R730xd Server (210-ADBC)	1	Tipo de Modelo de Servidor	\$ 1.85
PE R730/xd Motherboard MLK (329-BCZK)	1	Tipo de modelo de placa madre	\$
R730/xd PCle Riser 2, Center (330-BBCO)	1	Software de gestion de Puerto PCI Express	\$ 5
R730/xd PCle Riser 1, Right (330-BBCR)	1	Software de gestion de Puerto PCI Express	\$ 5
5M, LC-LC, Multi Mode Cable (331-8004)	4	Cable de fibra optica	\$ 5
US Order (332-1286)	1	orden de envio	\$
Intel Xeon E5-2650 v4 2.2GHz,30M Cache,9.60GT/s QPI,Turbo,HT,12C/24T (105W) Max Mem 2400MHz (338-BJDV)	1	Procesador Servidor	\$ 1.62
Intel Xeon E5-2650 v4 2.2GHz,30M Cache,9.60GT/s QPI,Turbo,HT,12C/24T (105W) Max Mem 2400MHz (338-BJDW)	1	Procesador Servidor	\$ 1.62
Pow erEdge R730xd Shipping (340-AKPM)	1	orden de envio	\$
Bezel (350-BBEJ)	1	Protector frontal servidor	\$ 4
Chassis with up to 12 + 4 Internal, 3.5" Hard Drives and 2, 2.5" Flex Bay Hard Drives (350-BBEX.)	1	Chassis para discos duros	\$ 30
Performance Optimized (370-AAIP)	1	software de optimizacion del servidor	\$
DIMM Blanks for System with 2 Processors (370-ABWE)	1	socket de memoria	\$
16GB RDIMM, 2400MT/s, Dual Rank, x8 Data Width (370-ACNX)	4	memoria del servidor	\$ 52
2400MT/s RDIMMs (370-ACPH)	1	memoria cache del servidor	\$ 52
Heatsink for 12 + 4 Chassis Pow er Edge R730xd (374-BBHR)	1		\$
Heatsink for 12 + 4 Chassis PowerEdge R730xd (374-bbnR)	1	disipador de calor de procesador	
		disipador de calor de procesador	\$
Performance BIOS Settings (384-BBBL)	1	Software de monitoreo BIOS	\$
iDRAC8 Enterprise, integrated Dell Remote Access Controller, Enterprise (385-BBHO)	1	controladores de acceso al servidor	\$ 46
Energy Star, PowerEdge R720 (387-BBHY)	1	certificacion fuente de poder EnergyStar	\$
400GB Solid State Drive SATA Mix Use MLC 6Gbps 2.5in Hot-plug Drive,3.5in HYB CARR, S3610 (400-AIFP)	3	disco duro de estado solido	\$ 66
300GB 10K RPM SAS 12Gbps 2.5in Flex Bay Hard Drive (400-AJPR)	2	disco duro Serial ATA	\$ 30
1TB 7.2K RPM NLSAS 12Gbps 512n 3.5in Internal Bay Hard Drive (400-ALOR)	4	disco duro Serial ATA	\$ 41
1TB 7.2K RPM NLSAS 12Gbps 512n 3.5in Hot-plug Hard Drive (400-ALQF)	8	Disco duro para Raid (6+2)	\$ 41
PERC H730P Integrated RAID Controller, 2GB Cache (405-AAEH)	1	Controlador de raid	\$ 89
SFP+, Short Range, Optical Tranceiver, LC Connector, 10Gb and 1Gb compatible for Intel and Broadcom (407-BBEQ)	1	Conector Fibra Optica	\$ 29
SFP+, Short Range, Optical Tranceiver, LC Connector, 10Gb and 1Gb compatible for Intel and Broadcom (407-BBEQ)	1	Conector Fibra Optica	\$ 29
SFP+, Short Range, Optical Tranceiver, LC Connector, 10Gb and 1Gb compatible for Intel and Broadcom (407-BBEQ)	1	Conector Fibra Optica	\$ 29
SFP+, Short Range, Optical Tranceiver, LC Connector, 10Gb and 1Gb compatible for Intel and Broadcom (407-BBEQ)	1	Conector Fibra Optica	\$ 29
No Media Required (421-5736)	1	Memoria USB con drivers	\$
Dual, Hot-plug, Redundant Pow er Supply (1+1), 750W (450-ADWS)	1	Fuente de poder	\$ 68
PE Server FIPS TPM 1.2v2 CC (461-AAED)	1	Puerta , suite de seguridad	\$ 5
C13 to C14, PDU Style, 12 AMP, 6.5 Feet (2m) Pow er Cord, North America (492-BBDI)	2	PDUM	\$ 2
Intel X520 DP 10Gb DA/SFP+, + i350 DP 1Gb Ethernet, Network Daughter Card (540-BBBB)	1	Tarjetas de Red	\$ 58
Intel X520 DP 10Gb DA/SFP+ Server Adapter, Low Profile (540-BBHY)	1	Tarjetas de Red	\$ 64
No Operating System (619-ABVR)	1	orden de envio	\$
Electronic System Documentation and OpenManage DVD Kit, Pow erEdge R730/xd (631-AAJG)	1	software de gestion del servidor completo	\$ 5
SanDisk DAS Cache, 90 Day Trial License (632-BBDC)	1	seguridad de discos cache	†
OpenManage Essentials, Server Configuration Management (634-BBWU)	1	kit de configuracion del servidor	\$ 9
ReadyRails Sliding Rails With Cable Management Arm (770-BBBR)	1	rieles del servidor	\$ 18
RAID 1+Unconfigured RAID for H330/H730/H730P (2 + 1-20 HDDs or SSDs) (780-BBLR)	1	kit de configuracion de raid	\$
UEFI BIOS (800-BBDM)	1	unidad flash que actualiza la BIOS	\$
ProDeploy Plus Dell Server R Series 1U/2U - Deployment (804-6608)	1	Parte del rack del servidor	\$ 7
ProDeploy Plus Dell Server R Series 1U/2U - Deployment Verification (804-6609)	1	Parte del rack del servidor	\$ 7
ProDeploy Plus Training Credits 200 Redeem at w.w.w.LearnDell.com Expires 1Yr from Order Date (812-4548)	1	Curso en linea	\$ 20
Dell Limited Hardw are Warranty Plus Service, Initial Year (978-4029)	1	Garantia	\$ 20
ProSupport Plus: 7x24 HW/SW Tech Support and Assistance, 3 Year (978-4011)	1		
ProSupport Hus: /x24 HW/SW Tech Support and Assistance, 3 Year (978-4131) ProSupport Plus: Mission Critical 4-Hour 7x24 On-Site Service with Emergency Dispatch, 3 Year (978-4132)	1	Soporte por 3 años	\$ 99 \$ 99
Procio Final Presupuesto rebajado en US	\$ 12.936,10	Soporte por 3 años Precio Final Presupuesto sin rebaja en US	\$ 99 \$15.307,0

Finalmente, la valoración de este proyecto se aproxima a un valor de \$US 12,937 lo que correspondería aproximadamente a ocho millones y medio de pesos chilenos aproximadamente

Para más detalles se puede observar el apéndice A para observar el valor de presupuesto en detalle, se presenta un resumen en la tabla 5-5:

Tabla 5-5: Resumen de Cotización para Propuesta Robusta y su ahorro otorgado

Detalle	Valor en CLP
Precio Final Presupuesto sin rebaja	\$15.198.296
Precio Final Presupuesto rebajado	\$ 8.589.570
Ahorro otorgado	43%

5.3.3 Valorización Productiva

Respecto a la valoración productiva de esta iniciativa esta queda sujeta a la implementación de esta, por medio del oficio de un consultor, un jefe de proyecto y un analista-programador en la creación del entorno SDN y la implementación de los 2 casos de uso. Cada rol fue designado con el fin de realizar el estudio del análisis de las tecnologías a utilizar, planificar entre los diferentes roles y la realización de la programación correspondiente. En cada caso se detalla cómo sigue en la tabla 5-6, la correspondiente tabla considera las horas en pequeñas, medianas y grandes empresas, respecto a el estudio realizado por la compañía 'It-hunter', donde se representan los sueldos de las TIC, y los cuales se calcularon por hora considerando 180 horas mensuales para cada cargo profesional.

Tabla 5-6: Valorización Productiva [20]

Rol	Horas Proyecto 1	Horas Proyecto 2	Costo por Hora P/M en \$CLP	ra P/M Hora G en Proyecto		Costo Proyecto 2 en CLP
Consultor	15	30	\$12.778	\$14.444	\$191.670	\$433.320
Jefe de Proyecto	8	20	\$11.111	\$16.111	\$88.888	\$322.220
Analista- Programador	15	50	\$8.889	\$10.556	\$133.335	\$527.800
Total Gastos Personales/ Alimentación					\$80.000	\$150.000
				Valor Final Proyecto	\$493.893	\$1.433.340

5.3.4 Herramienta de Evaluación CUNE

La evaluación de proyectos por medio de métodos matemáticos financieros es una herramienta de gran utilidad para la toma de decisiones, ya que un análisis que se anticipe al futuro puede evitar posibles desviaciones y problemas en el largo plazo

Para mayor comodidad del interesado de realizar la respectiva implementación, se efectúala comparativa de las mencionadas propuestas considerando el Costo uniforme neto equivalente (CUNE), Al cual normalmente también se le conoce como costo anual uniforme equivalente (CAUE), pues este método del CUNE consiste en convertir todos los ingresos y egresos, en una serie uniforme de pagos, dichos pagos pueden ser semanales, mensuales, trimestrales,

semestrales y anuales, siendo el indicador anual el más utilizado, por tanto se ha hecho referencia al termino CUNE para establecer su uso en cualquier periodicidad y al CAUE para su estricto uso en anualidades.

Entre las ventajas que ofrece este método de evaluación financiero, es que permite evaluar propuestas o proyectos con una periodicidad distinta, además de utilizarse mayormente en alternativas de inversión basadas en costo, es decir de alternativas que carecen de información de ingresos o beneficios económicos relevantes.

Como consecuencia de lo anterior, es importante determinar una tasa correcta para hacer los cálculos; normalmente debe usarse la tasa promedio, utilizada en el mercado financiero y hay otros que opinan que debe ser la tasa de interés, a la cual normalmente el dueño del proyecto hace sus inversiones. Esta última tasa es conocida con el nombre de Tasa de Interés de Oportunidad (TIO).

Para evaluar la tasa de interés de oportunidad (TIO), en ambos proyectos se considera la realización de cursos de las enseñanzas de la disciplina asociada a cada proyecto. Cabe destacar que la realización de dichos cursos son una opción que permita financiar el proyecto y que adicionalmente, y su correspondiente análisis es para efectos de generar la tasa correspondiente para el cálculo del CUNE, adicionalmente se muestra su método de cálculo y su respectiva tabla en la ecuación (5-1) y la tabla 5-7:

$$TIO = \frac{Total\ por\ cursos\ de\ enseñanza - Perdida}{valor\ de\ inversion} \tag{5-1}$$

•				-	-
	7 7				
	valor de inversion				
	tator ac tittorstort				
		valor de inversion	valor de inversion	valor de inversion	

Caracteristicas	Pro	yecto N°1	Pro	yecto N°2
Valor Total de inversión(Costo + VP)	\$	949.183	\$1	0.022.910
Equipos disponibles c/u (11 equipos)	\$	20.000	\$	25.000
Total por cursos de enseñanza	\$	220.000	\$	275.000
Pérdida	\$	20.000	\$	30.000
TIO		9,5%		2,4%

Tabla 5-7: Estimación de Tasa de interés de oportunidad

a continuación, para el cálculo del CUNE como tal se obtendrán los datos llevando los valores presentes y valores futuros de la inversión de cada proyecto a mensualidades en sus respectivos periodos de tiempo y cuyos cálculos se realizan según las ecuaciones (5-2) (5-3) y el resultado del análisis de CUNE se presenta como indica la tabla 5-8.

$$Valor\ Anualidad\ de\ valor\ Presente = \frac{-Valor\ de\ inversión\ presente}{\frac{1*(1+interes)^{periodo}}{interes}} \tag{5-2}$$

alor Anualidad de valor futuro =
$$\frac{-Valor de salvamento futuro}{\frac{(1+interes)^{periodo}-1}{interes}}$$
(5-3)

Característica	Proyecto N°1	Proyecto N°2
Periodo (meses)	12	36
Tasa de oportunidad (TIO)	9,5%	2,4%
Valor de inversión	\$949.183	\$10.022.910
Valor salvamento	\$350.000	\$6.000.000
Costo de reparación	n/a	n/a
Costo de operación mensual	n/a	35000
Anualidades VP	-\$135.789	-\$421.834
Anualidades VF	\$16.884	\$105.858
CUNE	-\$118.905	-\$315.976

Tabla 5-8: Calculo del CUNE para los ambos proyectos

En conclusión, a modo de comparación mediante periodos, el proyecto N°1 contempla un menor gasto y recuperación de los dineros en un periodo de tiempo menor, en cuanto el segundo proyecto cuenta con soporte incluido, garantía y mayor escalabilidad.

5.4 Análisis final de Factibilidad

En detalle al haber realizado el respectivo análisis de las condiciones actuales en el ámbito laboral y el ambiente en el cual se debe y puede implementar la tecnología SDN en conjunto con el protocolo OpenFlow, se concluye que el proyecto es factible desde los puntos de vista operacional, técnico, y económico. Dependiendo del fin necesario, es decir que no necesariamente se dependerá del proveedor de equipos o de servicios. El presente proyecto se demostró por medio de la simulación y emulación de una subred raspberry Pi, la factibilidad si se llegase a realizar la migración del diseño o esquema actual a lo revolucionario de SDN y OpenFlow para integración de redes SDN, como también la generación de una Red robusta en el mismo ámbito. Todo depende de las posibilidades e interés por el desarrollo de los respectivos temas.

5.5 Beneficios y Conclusiones de su respectiva instalación

Para finalizar, las evaluaciones de ambos proyectos contemplan ambas mejoras en contraste al actual sistema de redes tradicionales, siendo su inversión un beneficio directo para la docencia y el alumnado, además de una oportunidad de generar profesionales en el ámbito laboral. Se expone por tanto su comparativa en la tabla 5-9

Tabla 5-9: Comparativa entre redes tradicionales vs redes definidas por software (SDN)

Redes tradicionales	Redes Definidas por Software
Complejidad: uso de configuraciones de bajo nivel, despliegue distribuido, conocimiento previo del protocolo.	Abstracción: Ocultamiento de detalles de los recursos de red, programación de funciones
Dependencia del vendedor: funcionalidades provistas por el fabricante del dispositivo	Dependencia del desarrollador: funcionalidades desarrolladas por el administrador de red o especialista en software de redes
Patrón de trafico vertical: trafico cliente servidor	Patrón de trafico horizontal: tráfico de sistemas distribuidos
Control fraccionado: administración de recursos y funciones de la red de forma dispersa	Control globalizado: administración de recursos y funciones de la red de forma centralizada
Osificación: el despliegue de nuevos servicios es limitado y dependen de las funcionalidades adoptadas por los fabricantes de dispositivos	Innovación: la flexibilidad en base al software permite el fácil desarrollo de nuevas funciones y servicios
Incertidumbre: el sistema distribuido limita predecir el comportamiento de la red	Confiabilidad: el sistema centralizado permite el monitoreo de todas las funcionalidades y permite predecir el comportamiento de la red
Lenguaje de bajo nivel: especificado por el fabricante de dispositivos	Lenguaje de alto nivel : uso de lenguajes de programación como Python y Java

Discusión y conclusiones

Ante lo expuesto en los capítulos anteriores existe un gran interés por mejorar las redes actuales ante la demanda de generar mayores beneficios y reducir los costos de las mismas, pero a su vez se requiere estar al tanto de esta tecnología sus avances y modificaciones, por lo que es clara la necesidad de contar con profesionales futuros que se encarguen al respecto de estos temas.

Por consiguiente, la completa ausencia de laboratorios en casas de estudios en chile representa una debilidad, aunque también representa una oportunidad para llevar a cabo esta propuesta teórica–práctica en redes definidas por software y virtualización de funciones de red logrando mejorar el perfil de egreso de nuestra casa de estudios, así como también generar una contribución a la modernización de los sistemas de red.

Respecto a las experiencias generadas en el presente proyecto, se fundamentan en base a introducir al alumnado al estudio de SDN y NFV siendo estos un avance para el desarrollo y por consiguiente en el interés y renovación de las asignaturas y los implementos disponibles para esta área.

En detalle cada experiencia se enfocó en abordar los tópicos más esenciales para comprender las tecnologías SDN y NFV, su utilización generará las competencias necesarias para su entendimiento e utilización en el área de redes, como también serán una base para el entendimiento de la arquitectura de redes asociada, no obstante también estas dejan en evidencia que es necesario generar una asignatura, taller o modulo que tenga como tópico la arquitectura para redes de computadoras y data center para el desarrollo más avanzado de los tópicos en cuestión.

Un punto importante que se debe considerar respecto a las propuestas de instalación y desarrollo de laboratorio, tiene que ver con las posibilidades de incluir la realización de cursos a privados para reducir en gran medida el gran costo financiero que este proyecto puede llevar en sus dos opciones. No obstante, también se debería considerar esta opción a alumnos de otras escuelas dentro de la universidad con un costo reducido y accesible con el mismo fin.

Si bien el objeto de esta propuesta no es capitalizar estos recursos, se debe plantear esta posibilidad en vista de los elevados costos que esta implementación implica tanto a corto como largo plazo pueden ser poco llevaderos, si bien de ser implementado se podría generar un

desarrollo constante dentro del alumnado, en conjunto con la modernización y mejoramiento de los equipos disponibles para el área de redes dentro de la universidad.

También es deseable que el actual análisis, se presente para completar diversos proyectos que se puedan llevar a cabo dentro de la escuela de ingeniería eléctrica de la PUCV

Es importante recalcar como se comentó durante el desarrollo de este proyecto, que SDN y NFV aún siguen en desarrollo, lo que conlleva a seguir informándose al respecto de estos temas y perfeccionar en los mismos. Considerando que las guías ya propuestas permitan su estudio de manera precursora y se establezca el desarrollo e innovación en esta área, además de una posible continuidad de estas guías considerando que el factor económico no sea una barrera al momento de generar conocimientos.

Bibliografía

- [1] ONF, «Opem Networking Foundation,» [En línea]. Disponible en: https://www.opennetworking.org/mission/.
- [2] ODL, «Open Day Light,» [En línea]. Available: https://www.opendaylight.org/about.
- [3] Emaze, «Emaze,» [En línea]. Disponible en: https://www.emaze.com/@AZZIZFCF/ETHERNET%C2%A0INDUSTRIAL.
- [4] whitestack, «whitestack,» [En línea]. Disponible en: http://www.whitestack.com/sdn/.
- [5] A. Pagano, «Innovaciones de HP en redes definidas por software,» 2013.
- [6] Y. W. S. M. I. C. H. F. S. M. I. Wenfeng Xia, «A Survey on Software-Defined Networking,» 2015. [En línea]. Disponible en: http://www.ntu.edu.sg/home/ygwen/Paper/XWF-CST-15.pdf.
- [7] «Textos cientificios,» [En línea]. Disponible en: https://www.textoscientificos.com/redes/tcp-ip/comparacion-modelo-osi.
- [8] «Open Networking,» [En línea]. Disponible en: https://www.opennetworking.org/sdn-definition/.
- [9] «mundocontact.com,» [En línea]. Disponible en: https://cl.pinterest.com/pin/331507222561494610/.
- [10] brocade, «broadcom,» [En línea]. Disponible en: http://www.brocade.com/content/dam/common/documents/content-types/datasheet/brocade-sdn-controller-ds.pdf..
- [11] «texto cientificio,» [En línea]. Disponible en: https://www.textoscientificos.com/redes/tcp-ip/comparacion-modelo-osi..

- [12] A. Pagano, «Innovaciones de HP en redes definidas por software,» 2013.
- [13] «OpenMano,» [En línea]. Disponible en: http://www.tid.es/long-term-innovation/network-innovation/telefonica-nfv-reference-lab/openmano.
- [14] ONF, «Open Flow Switch Specification,» *Open Networking Fundation*, vol. vol 151, no 11, p. 283, 2015.
- [15] Mininet, «Mininet,» [En línea]. Disponible en: http://mininet.org/overview/.
- [16] gns3, «gns3,» [En línea]. Disponible en: https://docs.gns3.com/1PvtRW5eAb8RJZ11maEYD9_aLY8kkdhgaMB0wPCz8a38/index.ht ml..
- [17] «Inteco,» 2011. [En línea]. Disponible en: https://www.incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seg uridad_analisis_trafico_wireshark.pdf.
- [18] J. A. C. Valenzuela, Interviewee, *Ingeniero en Redes*. [Entrevista]. 03 marzo 2018.
- [19] J. A. C. Valenzuela, Interviewee, *Ingeniero en sistemas, mail: jcartes@dataconsulting.cl.* [Entrevista]. 10 12 2017.
- [20] It-hunter, «VI estudio publico de sueldos tic Chile,» 2016.
- [21] P. Judge, «silicon,» 30 9 2013. [En línea]. Disponible en: http://www.silicon.co.uk/workspace/hp-app-store-on-sdn-softawre-defined-networks-128219.
- [22] A. Pagano, «Innovaciones de HP en redes definidas por software,» 2013.
- [23] ONF, «Open Flow Switch Specification,» *Open Networking Fundation*, vol. 1.51, no 11, p. 283, 2015.
- [24] Mininet, «Mininet,» 2000. [En línea]. Disponible en: http://mininet.org/overview/.
- [25] Inteco, Febrero 2011. [En línea]. Disponible en: https://www.incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seg uridad_analisis_trafico_wireshark.pdf.
- [26] Brocade, «Brocade.com,» 2016. [En línea]. Disponible en: http://www.brocade.com/content/dam/common/documents/content-types/datasheet/brocade-sdn-controller-ds.pdf.

- [27] «OpenDayLight,» [En línea]. Disponible en: http://docs.opendaylight.org/en/stable-boron/getting-started-guide/introduction.html.
- [28] GNS3, «GNS3,» [En línea]. Disponible en: https://docs.gns3.com/1PvtRW5eAb8RJZ11maEYD9_aLY8kkdhgaMB0wPCz8a38/index.ht ml.

A Presupuestos

En las siguientes hojas se expondrán en detalle el presupuesto utilizado para el desarrollo de la presente guía.

Tabla A-1: Cotización Dell para propuesta robusta de SDN con detalle.

Elementos del servidor	Cantidad	Detalle		Precio itario en US	Precio Final en US	Precio Final en CLP
Pow erEdge R730xd Server (210-ADBC)	1	Tipo de Modelo de Servidor	\$	1.850	\$ 1.850	\$ 1.228.400
PE R730/xd Motherboard MLK (329-BCZK)	1	Tipo de modelo de placa madre	\$	-	\$ -	\$ -
R730/xd PCle Riser 2, Center (330-BBCO)	1	Software de gestion de Puerto PCI Express	\$	50	\$ 50	\$ 33.200
R730/xd PCle Riser 1, Right (330-BBCR)	1	Software de gestion de Puerto PCI Express	\$	50	\$ 50	\$ 33.200
5M, LC-LC, Multi Mode Cable (331-8004)	4	Cable de fibra optica	\$	55	\$ 220	\$ 146.080
US Order (332-1286)	1	orden de envio	\$	-	\$ -	\$ -
Intel Xeon E5-2650 v4 2.2GHz,30M Cache,9.60GT/s QPI,Turbo,HT,12C/24T (105W) Max Mem 2400MHz (338-BJDV)	1	Procesador Servidor	\$	1.629	\$ 1.629	\$ 1.081.656
Intel Xeon E5-2650 v4 2.2GHz,30M Cache,9.60GT/s QPI,Turbo,HT,12C/24T (105W) Max Mem 2400MHz (338-BJDW)	1	Procesador Servidor	\$	1.629	\$ 1.629	\$ 1.081.656
PowerEdge R730xd Shipping (340-AKPM)	1	orden de envio	\$	-	\$ -	\$ -
Bezel (350-BBEJ)	1	Protector frontal servidor	\$	49	\$ 49	\$ 32.536
Chassis with up to 12 + 4 Internal, 3.5" Hard Drives and 2, 2.5" Flex Bay Hard Drives (350-BBEX)	1	Chassis para discos duros	\$	300	\$ 300	\$ 199.200
Performance Optimized (370-AAIP)	1	software de optimizacion del servidor	\$	-	\$ -	\$ -
DIMM Blanks for System with 2 Processors (370-ABWE)	1	socket de memoria	\$	-	\$ -	\$ -
16GB RDIMM, 2400MT/s, Dual Rank, x8 Data Width (370-ACNX)	4	memoria del servidor	\$	520	\$ 2.080	\$ 1.381.120
2400MT/s RDIMMs (370-ACPH)	1	memoria cache del servidor	\$	-	\$ -	\$ -
Heatsink for 12 + 4 Chassis PowerEdge R730xd (374-BBHR)	1	disipador de calor de procesador	\$	-	\$ -	\$ -
Heatsink for 12 + 4 Chassis PowerEdge R730xd (374-BBHR)	1	disipador de calor de procesador	\$	-	\$ -	\$ -
Performance BIOS Settings (384-BBBL)	1	Software de monitoreo BIOS	\$	-	\$ -	\$ -
iDRAC8 Enterprise, integrated Dell Remote Access Controller, Enterprise (385-BBHO)	1	controladores de acceso al servidor	\$	469	\$ 469	\$ 311.416
Energy Star, PowerEdge R720 (387-BBHY)	1	certificacion fuente de poder EnergyStar	\$	-	\$ -	\$ -
400GB Solid State Drive SATA Mix Use MLC 6Gbps 2.5in Hot-plug Drive,3.5in HYB CARR, S3610 (400-AIFP)	3	disco duro de estado solido	\$	669	\$ 2.007	\$ 1.332.648
300GB 10K RPM SAS 12Gbps 2.5in Flex Bay Hard Drive (400-AJPR)	2	disco duro Serial ATA	\$	309	\$ 618	\$ 410.352
1TB 7.2K RPM NLSAS 12Gbps 512n 3.5in Internal Bay Hard Drive (400-ALOR)	4	disco duro Serial ATA	\$	419	\$ 1.676	\$ 1.112.864
1TB 7.2K RPM NLSAS 12Gbps 512n 3.5in Hot-plug Hard Drive (400-ALQF)	8	Disco duro para Raid (6+2)	\$	419	\$ 3.352	\$ 2.225.728
PERC H730P Integrated RAID Controller, 2GB Cache (405-AAEH)	1	Controlador de raid	\$	899	\$ 899	\$ 596.936
SFP+, Short Range, Optical Tranceiver, LC Connector, 10Gb and 1Gb compatible for Intel and Broadcom (407-BBEQ)	1	Conector Fibra Optica	\$	299	\$ 299	\$ 198.536
SFP+, Short Range, Optical Tranceiver, LC Connector, 10Gb and 1Gb compatible for Intel and Broadcom (407-BBEQ)	1	Conector Fibra Optica	\$	299	\$ 299	\$ 198.536
SFP+, Short Range, Optical Tranceiver, LC Connector, 10Gb and 1Gb compatible for Intel and Broadcom (407-BBEQ)	1	Conector Fibra Optica	\$	299	\$ 299	\$ 198.536
SFP+, Short Range, Optical Tranceiver, LC Connector, 10Gb and 1Gb compatible for Intel and Broadcom (407-BBEQ)	1	Conector Fibra Optica	\$	299	\$ 299	\$ 198.536
No Media Required (421-5736)	1	Memoria USB con drivers	\$	-	\$ -	\$ -
Dual, Hot-plug, Redundant Pow er Supply (1+1), 750W (450-ADWS)	1	Fuente de poder	\$	689	\$ 689	\$ 457.496
PE Server FIPS TPM 1.2v2 CC (461-AAED)	1	Puerta , suite de seguridad	\$	59	\$ 59	\$ 39.176
C13 to C14, PDU Style, 12 AMP, 6.5 Feet (2m) Pow er Cord, North America (492-BBDI)	2	PDUM	\$	20	\$ 40	\$ 26.560

					T .	
Intel X520 DP 10Gb DA/SFP+, + I350 DP 1Gb Ethernet, Network Daughter Card (540-BBBB)	1	Tarjetas de Red	\$	589	\$ 589	\$ 391.096
Intel X520 DP 10Gb DA/SFP+ Server Adapter, Low Profile (540-BBHY)	1	Tarjetas de Red	\$	649	\$ 649	\$ 430.936
No Operating System (619-ABVR)	1	orden de envio	\$	-	\$ -	\$ -
Electronic System Documentation and OpenManage DVD Kit, Pow erEdge R730/xd (631-AAJG)	1	software de gestion del servidor completo	\$	50	\$ 50	\$ 33.200
SanDisk DAS Cache, 90 Day Trial License (632-BBDC)	1	seguridad de discos cache			\$ -	\$ -
OpenManage Essentials, Server Configuration Management (634-BBWU)	1	kit de configuracion del servidor	\$	99	\$ 99	\$ 65.736
ReadyRails Sliding Rails With Cable Management Arm (770-BBBR)	1	rieles del servidor	\$	189	\$ 189	\$ 125.496
RAID 1+Unconfigured RAID for H330/H730/H730P (2 + 1-20 HDDs or SSDs) (780-BBLR)	1	kit de configuracion de raid	\$	-	\$ -	\$ -
UEFI BIOS (800-BBDM)	1	unidad flash que actualiza la BIOS	\$	-	\$ -	\$ -
ProDeploy Plus Dell Server R Series 1U/2U - Deployment (804-6608)	1	Parte del rack del servidor	\$	77	\$ 77	\$ 51.128
ProDeploy Plus Dell Server R Series 1U/2U - Deployment Verification (804-6609)	1	Parte del rack del servidor	\$	77	\$ 77	\$ 51.128
ProDeploy Plus Training Credits 200 Redeem at www.LearnDell.com Expires 1Yr from Order Date (812-4548)	1	Curso en linea	\$	200	\$ 200	\$ 132.800
Dell Limited Hardware Warranty Plus Service, Initial Year (978-4029)	1	Garantia	\$	99	\$ 99	\$ 65.736
ProSupport Plus: 7x24 HW/SW Tech Support and Assistance, 3 Year (978-4131)	1	Soporte por 3 años	\$	999	\$ 999	\$ 663.336
ProSupport Plus: Mission Critical 4-Hour 7x24 On-Site Service with Emergency Dispatch, 3 Year (978-4132)	1	Soporte por 3 años	\$	999	\$ 999	\$ 663.336
Precio Final Presupuesto rebajado en US	\$ 12.936,10	Precio Final Presupuesto sin rebaja en US	\$1	5.307,00	\$22.889,00	
Precio Final Presupuesto rebajado en CLP	\$ 8.589.570	Precio Final Presupuesto sin rebaja en CLP				\$ 15.198.296
http://itprice.com/dell-price-list						
Detalle	Valor en CLP					
Precio Final Presupuesto sin rebaja	\$15.198.296					
Precio Final Presupuesto rebajado	\$ 8.589.570					
Ahorro otorgado	43%					

B Apéndice – Listado de Comandos

B.1 Comandos Linux

B.1.1 Utilidad o uso común

- **sudo**: permisos especiales para ejecutar comandos como administrador (root)
- **clear**: Limpia el terminal borrando todo el texto que se ha escrito hasta ahora.
- **pwd**: Visualiza el directorio actual en el que estamos.
- history: Muestra el historial de todos los comandos que has utilizado.
- man: aprende más acerca de los comandos

B.1.2 Archivos y Directorios

- **cd /home**: entrar en el directorio "home".
- **cd** ..: retroceder un nivel.
- **cd** ../..: retroceder 2 niveles.
- cd: ir al directorio raíz.
- **cd** –: ir (regresar) al directorio anterior.
- pwd: mostrar el camino del directorio de trabajo.
- ls: ver los ficheros de un directorio.
- cp file1: copiar un fichero.
- **mv**: mover archivos y carpetas.

B.1.3 Actualizador de paquetes APT (Debian, Ubuntu y derivados)

- apt-get install package_name: instalar / actualizar un paquete deb.
- **apt-cdrom install package_name**: instalar / actualizar un paquete deb desde un cdrom.
- apt-get update: actualizar la lista de paquetes.
- apt-get upgrade: actualizar todos los paquetes instalados.

- apt-get remove package_name: eliminar un paquete deb del sistema.
- apt-get check: verificar la correcta resolución de las dependencias.
- apt-get clean: limpiar cache desde los paquetes descargados.
- **apt-cache search searched-package**: retorna lista de paquetes que corresponde a la serie «paquetes buscados».

B.1.4 Trabajo con la RED (LAN y Wi-Fi)

- **ifconfig eth0**: mostrar la configuración de una tarjeta de red Ethernet.
- **ifup eth0**: activar una interface 'eth0'.
- **ifdown eth0**: deshabilitar una interface 'eth0'.
- ifconfig eth0 192.168.1.1 netmask 255.255.255.0: configurar una dirección IP.
- **ifconfig eth0 promisc**: configurar 'eth0'en modo común para obtener los paquetes (sniffing).
- **dhclient eth0**: activar la interface 'eth0' en modo dhcp.
- route -n: mostrar mesa de recorrido.
- route add -net 0/0 gw IP_Gateway: configurar entrada predeterminada.
- **route add -net 192.168.0.0 netmask 255.255.0.0 gw 192.168.1.1**: configurar ruta estática para buscar la red '192.168.0.0/16'.
- route del 0/0 gw IP_gateway: eliminar la ruta estática.
- echo "1" > /proc/sys/net/ipv4/ip_forward: activar el recorrido ip.
- **hostname**: mostrar el nombre del host del sistema.
- **host** www.example.com: buscar el nombre del host para resolver el nombre a una dirección ip(1).
- **nslookup** <u>www.example.com</u>: buscar el nombre del host para resolver el nombre a una direccióm ip y viceversa(2).
- **ip link show**: mostar el estado de enlace de todas las interfaces.

- mii-tool eth0: mostar el estado de enlace de 'eth0'.
- **ethtool eth0**: mostrar las estadísticas de tarjeta de red 'eth0'.
- **netstat -tup**: mostrar todas las conexiones de red activas y sus PID.
- **netstat -tupl**: mostrar todos los servicios de escucha de red en el sistema y sus PID.
- **tcpdump tcp port 80**: mostrar todo el tráfico HTTP.
- **iwlist scan**: mostrar las redes inalámbricas.
- iwconfig eth1: mostrar la configuración de una tarjeta de red inalámbrica.
- whois www.example.com: buscar en base de datos Whois.

B.2 Comandos Mininet

El comando 'sudo mn', ejecutado desde una terminal (consola) de Linux inicia el emulador Mininet. Permite personalizar la topología y el funcionamiento de esta mediante la introducción de diferentes opciones. Se compone de la siguiente estructura:

sudo mn --[OPCION]=[PARAMETRO], [ARGUMENTOS]

Las opciones que hay disponibles son:

- "-h", "--help": muestra en pantalla un listado de las posibles opciones que se pueden invocar con el comando 'sudo mn'."
- "--switch=[PARÁMETRO]": invoca un tipo de switch compatible con Mininet de entre los siguientes parámetros:
 - default: usa un switch Open vSwitch por defecto.
 - ivs: IVSSwitch, es un switch OpenFlow que usa tecnología Undigo VirtualSwitch y requiere instalación previa.
 - ovs: Open vSwitch, usa tecnología Open vSwitch compatible con OpenFlow.
 - ovsbr: OVSBridge, usa un switch Ethernet implementado a partir de OpenvSwitch.
 - ovsk: usa Open vSwitch en modo kernel para cada switch. •
 - ovsl: Open vSwitch legacy kernel-space, actualmente solo trabaja con elespacio de nombres principal Root.
 - user: switch con implementación OpenFlow invocado desde el espacio de usuario, ósea, externo a Mininet.
 - lxbr: Linux Bridge, switch implementado en código abierto.

- "--host=[PARAMETRO]": `limita el ancho de banda del procesador deun host virtual:
- cfs (Completely Fair Scheduler): planificador de uso de recursos de procesamiento en Linux basado en Fair Queuing.
- rt: planificador POSIX real-time (interfaz de sistema operativo portablede tiempo real), este planificador ha sido deshabilitado por defecto entodos los kernel Linux. Se debe habilitar RT-GROUP-SCHED.
- "--controller=[PARAMETRO]": permite la creación de un tipo de controlador compatible con Mininet de entre los siguientes:
 - default: usa un controlador por defecto compatible con OpenFlow.
 - none: deshabilita el uso de un controller.
 - NOX: habilita un controlador tipo NOX (requiere instalación previa)
 - ovsc: usa el controlados de prueba de Open vSwitch.
 - remote: permite utilizar un controlador externo a Mininet y compatible con OpenFlow. Usa argumentos como:
 - o ip= [IP del CONTROLADOR]
 - o port=[PUERTO]
 - ryu: usa el controlador Ryu. (requiere instalación previa)
- "--link=[PARAMETRO]": ´ permite variar parámetros como ancho debanda y latencia de los enlaces:
 - default: configura un enlace con ancho de banda, latencia y perdida de paquetes por defecto.
 - tc: personaliza las interfaces por medio de la utilidad Traffic Control, permitiendo especificaciones de limites de ancho de banda, latencia, perdidas y máxima longitud de colas manejadas. Usa argumentos como:
 - o bw= [ANCHO DE BANDA]
 - o delay= [TIEMPO ms]
 - o loss= [PORCENTAJE]
- "--topo=[PARAMETRO]": ´permite cambiar el tamaño y el tipo de topología a emular:
 - linear: genera una topología de k switches en serie con n hosts conectados a cada switch. Usa argumentos como:
 - k=[NUMERO DE SWITCHES]
 - o n=[NUMERO DE HOSTS POR SWITCH]
 - minimal: genera una topología simple de dos hosts y un switch.
 - $\bullet \quad$ single: genera una topología simple de un switch y N hosts. Usa el argumento: \circ

- k= [NUMERO DE HOSTS]
- reversed: similar a una topología single, pero esta invierte el orden de los puertos en el switch. Una topología single asigna los puertos del switch en orden ascendente mientras que ´esta lo hace de forma descendente. Utiliza argumentos como:
 - o k= [NUMERO DE HOSTS]
- tree: genera una topología de árbol, compuesta de N niveles (depth), Nramas (fanout) y dos hosts conectados a cada switch hoja. Usa argumentos como
 - o depth=[PROFUNDIDAD]
 - o fanout=[NUMERO DE RAMAS] ´
- torus: genera una topología en forma de malla MxN donde cada switch se conecta con sus vecinos más cercanos y los switches del borde se conectan con el opuesto. Mininet solo soporta topologías 2D, este tipo de redes tiene bucles y podría no funcionar con dispositivos que no soporten el protocoloSpanning Tree, por lo tanto, se recomienda utilizarla con switches lxbr,compatibles con STP. Recibe como argumentos:
 - \circ x=[M]
 - \circ y=[N]

"--custom=[PARAMETRO]": lee archivos de configuración de red escritos en Python, con extensión .py, para crear topologías personalizadas. Recibe como parámetro el nombre y extensión del archivo junto con su ruta relativa absoluta.

"--test=[PARAMETRO]": permite realizar diferentes pruebas a la redemulada según el parámetro ingresado, estos pueden ser:

- cli: inicia la emulación y permite el uso de la línea de comandos.
- none: inicia la emulación e inmediatamente la finaliza, este parámetro puede ser usado para probar si una topología personalizada opera correctamente en el emulador.
- build: inicia la emulación hasta que la virtualización este completamente operativa, seguidamente, se finaliza mostrando por pantalla el tiempo de duración de la ejecución.
- pingpair: inicia la emulación, realiza prueba de conectividad entre los dosprimeros hosts de la emulación y la finaliza. Se usa generalmente parapruebas.
- pingall: inicia la emulación, realiza pruebas de conectividad entre todos los hosts por medio de ping y finalmente se finaliza la emulación.
- iperf: es una herramienta que mide el máximo ancho de banda entre dos hosts basado en el protocolo TCP, devolviendo información como latencia, pérdida de datagramas y ancho de banda entre dos hosts.
- iperfudp: similar al anterior, pero mide el máximo ancho de banda entredos hosts basados en el protocolo UDP, retornando información como latencia, perdida de datagramas y ancho de banda. Al terminar finaliza laemulación.

[&]quot;-c", "--clean": limpia los registros de emulación y cierra el emuladorMininet.

- all: inicia la simulación, realiza una prueba de conectividad entre <u>todos los</u> hosts, similar al funcionamiento de pingall, realiza una medición del ancho de banda entre dos hosts como lo hace iperf, y finaliza la emulación.
- "-x", "--xterms": inicia la emulación y abre una terminal independientepara cada dispositivo emulado en Mininet.
- "-i [PARAMETRO]", "´--ipbase=[PARAMETRO]": defina el espacio de direcciones que usar a le red a emular, por defecto, Mininet asigna la red10.0.0.0/8
- "--mac": por defecto los valores de las direcciones MAC de los hosts emuladosson aleatorios, este parámetro permite asignar direcciones MAC de formaordenada.
- "--arp": Este parámetro inicializa las tablas ARP de los hosts emulados.
- "-v [PARAMETRO]"," ´--verbosity=[PARAMETRO]": esta opciónmuestra información interna del emulador con el objetivo de depurar la operación de la virtualización, la información está clasificada en los niveles critical, error, warning, info, debug y output ordenados de mayor a menos relevancia, además, los parámetros que recibe esta opción son:
 - critical, error, warning: estos parámetros retornan el mismo resultado:
 ***WARNING: selected verbosity level (C, E, W) will hide CLI output! Please restart Mininet with -v [debug, inf, output].
 - info: este parámetro es ejecutado por defecto, permite visualizar en consolalo que el emulador está realizando durante el inicio y finalización de laemulación.
 - output: este parámetro reduce la información visualizada en consola a solo los datos necesarios.
 - debug: este parámetro habilita una visualización muy detallada sobre elcomportamiento del emulador Mininet durante toda la ejecución.
- "--innamespace": por defecto los hosts están puestos en su propio espacio de nombres, mientras que switches y controladores están en el espacio de nombres principal (root), este parámetro ubica a los switches en su propio espacio de nombres permitiendo la separación en el sistema de comunicaciónentre switches y controladores.
- "--listenport=[PARAMETRO]": ´esta opción recibe como parámetro un número que ser a la base para estableces los puertos lógicos de los switches. Pordefecto, el puerto lógico asignado a un switch es el 6634 e incrementa según elnúmero de switches usados.
- "--nolistenport": deshabilita el uso de un puerto lógico en los switchesemulados inclusive el puerto lógico por defecto 6634.
- "--nat": agrega servicio NAT entre la red emulada y el host anfitrión, esteparámetro ofrece conectividad entre los dispositivos emulados en Mininet y elhost anfitrión.

"--version": muestra por pantalla el número de versión de Mininet.Se detallan a continuación los comandos para la CLI (línea de introducción decomandos) de Mininet.

mininet> EOF: el comando "EOF" finaliza la emulación de Mininet.

mininet> exit: el comando "exit" finaliza la emulación y cierra el programa.

mininet> quit: el comando "quit" finaliza la emulación actual.

mininet> help: el comando "help" muestra en pantalla documentación einformación del uso de comandos.

mininet> dump: el comando "dump" muestra en pantalla información detallada de la red, datos como tipo de dispositivo, nombre, puerto usado, direcciónIP e ID de proceso.

mininet> net: el comando "net" muestra en pantalla los enlaces y los respectivos puertos usados por los dispositivos emulados.

mininet> intfs: el comando "intfs" detalla las interfaces usadas por losdispositivos emulados.

mininet> nodes: el comando "nodes" muestra en pantalla los nodos emulados.

mininet> ports: el comando "ports" muestra en pantalla los puertos e interfaces de cada switch emulado.

mininet> time [comando]: el comando "time" muestra en pantalla el tiempoque usa cualquier comando Mininet para ejecutarse.

mininet> switch [switch] [start/stop]: el comando "switch" inicia o detieneel funcionamiento del switch especificado.

mininet> links: el comando "links" reporta los enlaces que presentan unestado correcto y funcional.

mininet> link [nodo1] [nodo2] [up/down]: el comando "link" habilita odeshabilita el enlace entre dos nodos.

mininet> noecho [host] [cmd args]: el comando "noecho" ejecuta accionesy comandos directamente en el dispositivo virtual sin realizar eco en elhost anfitrión de Mininet, evita el uso de herramientas como xterm y gterm.

mininet> sh [cmd args]: el comando "sh" ejecuta un comando Shell delsistema operativo Ubuntu.

mininet> source file: el comando "source" lee comandos Mininet desdeun archivo de entrada; habrá que tener en cuenta el manejo de rutas absolutaso relativas.

mininet> pingall: el comando "pingall" realiza una prueba de conectividadentre todos los hosts emulados.

mininet> pingallfull: el comando "pingallfull" realiza una prueba de conectividad entre todos los hosts emulados, mostrando en pantalla los resultados.

mininet> pingpair: el comando "pingpair" realiza una prueba de conectividadentre los dos primeros hosts emulados (h1 y h2).

mininet> pingpairfull: el comando "pingpairfull" realiza una prueba deconectividad entre los dos primeros hosts emulados (h1 y h2) mostrando enpantalla los resultados.

mininet> iperf [host1] [host2]: el comando "iperf" es una herramienta queprueba del rendimiento de ancho de banda TCP entre dos hosts específicos.

mininet> iperfudp [bw] [host1] [host2]: el comando "iperfudp" es unaherramienta que prueba el rendimiento de ancho de banda UDP entre dos hosts específicos.

mininet> px [PYTHON]: el comando "px" ejecuta declaraciones en lenguajede programación Python, es posible usar variables y funciones de las librerías de Mininet.

mininet> py [OBJETO.FUNCION()]: el comando "py" permite evaluary ejecutar desde la consola Mininet expresiones en lenguaje Python y basadasen librerías Mininet.

mininet> xterm [nodo1] [nodo...]: el comando "xterm" abre una nuevaterminal para los nodos especificados.

mininet> x [host] [cmd args]: el comando "x" crea un t'unel X11 a un hostespecifico.

mininet> gterm [nodo1] [nodo. . .]: el comando "gterm" abre terminalesgnome para los host solicitados.

mininet> dpctl [COMANDO][ARGUMENTOS]: dpctl (o ovs-ofctl) esuna herramienta de administración y monitoreo que se ejecuta en todos losswitches OpenFlow emulados cuando es invocado desde la consola Mininet.

C Apéndice – Instalaciones

C.1 Instalación Mininet en Raspberry Pi 3

Paso nº1: en la Raspberry Pi desde un terminal, clonar Mininet desde github, para ello se debe utilizar el siguiente comando:

pi@raspberrypi1 ~ \$ sudo git clone git://github.com/mininet/mininet.git

Paso nº 2: ir hasta el directorio 'util' de la carpeta mininet generada luego de la descarga via terminal:

pi@raspberrypi1 ~ \$ cd mininet/util/

Paso nº 3: Ejecutar e instalar el 'install.sh script' usando alguna de las siguientes opciones:

Opción nº1: instalación básica:

pi@raspberrypi1 ~/mininet/util \$ sudo ./install.sh -fnv

Opción 2: Instalación completa:

pi@raspberrypi1 ~/mininet/util \$ sudo ./install.sh -a

OBS: si realiza este procedimiento por medio de terminal SSH, favor usar los siguientes comandos

ssh remote username@remote host

donde username corresponde al nombre del usuario de la raspberry, normalmente 'pi' Y EL remote_host corresponde a la dirección ip del host que se desea ingresar.

En caso de que raspberry pi requiera contraseña para establecer conexión, esta corresponde a 'raspberry' caso contrario revise el sistema raspbian instalando en ella para obtener dicha password

D Experiencias

D.1 Pre informe Nº1

Guía Alumno

"Laboratorio Introductorio a Tecnología

Pre-informe N°1 SDN"

Nombre:		
_		
Rut:	Fecha:	

Objetivos:

- Introducir a los conceptos del entorno Ubuntu-Linux, SDN, Mininet y OpenDayLight
- Reconocer los comandos existentes para cada una de las plataformas

Escenario:

Se plantea estudiar los conocimientos para diseñar una topología de redes definida por software (SDN) por lo que se comprenderán los entornos de trabajo que permiten su diseño y estudio, para así llevar a cabo su implementación en una próxima experiencia en el aula.

Entendiendo Ubuntu-Linux

¿Qué es Ubuntu?

Ubuntu corresponde a una distribución del sistema operativo Linux el cual es un software de carácter libre. La principal característica de Ubuntu es que logra hacer a Linux un sistema operativo más accesible y fácil de usar.

Conceptos Básicos

Dentro del sistema operativo Ubuntu-Linux existe la aplicación llamada 'Terminal' o 'Consola' a la cual se puede acceder presionando "Ctrl+Alt+t" o buscando dicha aplicación en la barra del sistema.

La aplicación consola permite manejar el sistema en todos los aspectos, y es posible utilizar algunos comandos para instalar programas, borrar, copiar y mover archivos, editar archivos de texto y solucionar prácticamente cualquier error del sistema. Sus beneficios radican en la velocidad de ejecución por medio de comandos, ya que pueden realizar eficientemente en una línea de comando lo que tardaría varios clics en una interfaz gráfica.

Comandos Basicos

Dentro de la aplicación Consola es posible escribir los siguientes comandos para realizar las respectivas acciones descritas:

- **apt-get install** *nombre_programa*: instalar / actualizar un programa. (requiere conexión a internet para descargar e instalar el programa)
- apt-get update: actualizar la lista de paquetes (programas).
- apt-get remove nombre_programa: eliminar un programa del sistema.
- apt-get clean: limpiar cache desde los paquetes descargados.
- cd: ir al directorio raíz.
- ls: ver los ficheros de un directorio.
- **cd /home**: entrar en el directorio "home".
- **cd** ..: retroceder un nivel dentro del directorio actual.
- **clear:** Limpia el terminal borrando todo el texto que se ha escrito hasta ahora.
- Exit: cierra el terminal (consola)

Entendiendo SDN

¿Qué es SDN?

SDN (Software-Defined Networking) es la separación en la gestión de los dispositivos de red del plano de control del plano de datos, en otras palabras, se trata de la separación del hardware y del software, y donde el control de la red es llevado a cabo por dispositivos denominados controladores.

Conceptos Básicos

El objetivo principal que persigue SDN es facilitar la implementación e implantaciónde servicios de red de una manera determinista, dinámica, económica y escalable, evitandoal administrador de red gestionar dichos servicios a bajo nivel.

Protocolo de Comunicación

El primer protocolo de comunicación estándar para la tecnología SDN se denomina Openflow. No obstante, también existen protocolos como Netconfig/ Yang o BGP.

Entendiendo Mininet

¿Qué es Mininet?

Mininet es un emulador de red que crea una red de hosts virtuales, conmutadores, controladores y enlaces. Los hosts de Mininet ejecutan un software de red Linux estándar, y sus switches admiten OpenFlow para un enrutamiento personalizado altamente flexible y redes definidas por software

Conceptos Básicos

Mininet es rápidamente reconfigurable y reiniciable, ofreciendo un buen rendimiento de trabajo bajo una pequeña cantidad de comandos, este programa puede ser usado como una máquina virtual (VM) o directamente desde la consola de Ubuntu-Linux.

La topología por defecto para Mininet consta de un switch, un controlador (local) y dos hosts.

Comandos Básico

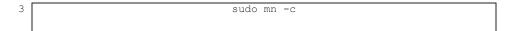
Mininet permite generar diversas topologías, no obstante, su configuración más básica viene dada por el siguiente comando:

1	sudo mn

Comando que genera una topología de 2 hosts, un switch y un controlador SDN, además esta puede modificarse agregando más opciones, las cuales son visibles al agregar '-h'a dicho comando, es decir:

2	sudo mn -h

Finalmente, también existe un comando que regresa todas las configuraciones a cero llamado clear:



Entendiendo OpenDayLight

¿Qué es OpenDaylight?

Este proyecto es promovido por *The Linux Foundation* con el cual se impulsa al desarrollo, distribución y adopción global de la plataforma SDN Open Daylight (ODL) de código abierto y de índole neutral

¿Cómo se relaciona con SDN y Mininet?

ODL corresponde a la interfaz gráfica que funciona como administrador para el controlador SDN observando la topología creada por mininet, sus protocolos de comunicación, entre otras aplicaciones.

Cuestionario de investigación:
¿Cuál es la función del comando 'sudo' dentro de linux?
¿Cuál es el conjunto de teclas que permiten copiar y pegar dentro del terminal de Linux?
Detalle las topologías proporcionadas por mininet al momento de generar redes.

- Escriba los comandos para dos topologías distintas a ejecutar con mininet, asigne también una dirección Mac para cada host dentro de la red.
¿A que corresponden el apartado 'features' dentro de ODL al momento de ejecutarlo?
¿Cómo es que se conectan mininet y ODL a través de Linux?

D.2 Experiencia Nº1

Guía Alumno

"Laboratorio Introductorio a Tecnología SDN"

Experiencia N°1

Nombre:	 	
Rut:	 Fecha:	

Objetivo:

• implementar y comprender topologías para una red SDN y su respectiva visualización grafica a través de aplicaciones en el sistema operativo Ubuntu-Linux.

Requerimientos:

- 1 PC (Ram: 6GB como mínimo, puerto Ethernet y/o Tarjeta WLAN)
- Conexión a Internet
- Sistema Ubuntu-Linux (Versión 16.04 o superior) como partición del equipo

Nota: Es importante tener en cuenta que la sintaxis del código en el sistema operativo Ubuntu-Linux diferencia entre mayúsculas y minúsculas, también que la consola de terminal requiere introducir la clave de la cuenta de usuario del sistema entre comandos.

Escenario:

Se debe implementar un entorno de trabajo de tecnología SDN en el sistema Ubuntu-Linux desde cero a partir del terminal (consola) del sistema operativo, esto implica instalar Mininet y OpenDayLight en conjunto con sus requerimientos para posteriormente configurar y generar una red SDN y así visualizarla.

Desarrollo Experiencia

Paso Nº1: Conociendo El terminal de Ubuntu-Linux

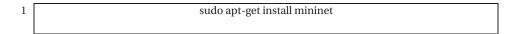
Una vez estando en el entorno gráfico de Ubuntu, es necesario abrir un terminal del sistema, lo cual se puede lograr presionando "Ctrl+Alt+t" o buscando dicha aplicación en la barra del sistema.

El terminal o consola es una aplicación que permite realizar múltiples acciones por medio de breves comandos, para esta actividad se necesita instalar las aplicaciones llamadas Mininet y OpenDayLight.

Paso Nº2: Instalación de Mininet y sus requerimientos

A través del terminal se procede a la instalación nativa (a través de comandos) de Mininet en Ubuntu:

Con este comando se instalarán los núcleos para Mininet:



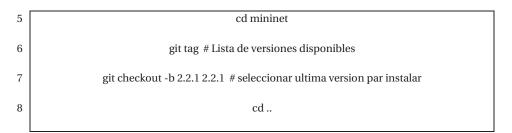
Este paso permite eliminar cualquier controlador que Mininet pueda haber activado anteriormente:

```
2 sudo mn –c
```

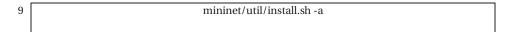
Con esto se instala git para que Mininet pueda ser descargado desde GitHub (Foro de la comunidad dedicada a Mininet) y crear una estructura de ceros en nuestra máquina.

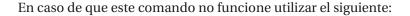


Se descarga la aplicación "git", ya que esta permite ver el control de versiones para los respectivos programas desarrollados para Linux, con los siguientes comandos, lo que hará esto es cambiar a la última versión de Mininet.



El siguiente paso es el más importante ya que permite instalar todos los elementos que ofrece Mininet. Si este paso no es llevado a cabo Mininet no podrá conectarse correctamente con el controlador y no funcionará.





10 mininet/util/install.sh -a Ubuntu trusty i386 Ubuntu

Con esto ya se tendrá instalado Mininet.

Paso N°3: Instalación OpenDayLight y sus requerimientos

A través de un segundo terminal prosiga mediante la siguiente manera para la instalación nativa del controlador OpenDayLight en Ubuntu:

Primero se debe instalar la aplicación Java ya que este controlador es un programa Java.

Para instalar Java se deben usar los siguientes comandos:

sudo apt-get update

sudo apt-get install default-jre-headless

Tras esto hay que editar el fichero ".bashrc" el cual está encargado de los comandos que se ejecutan al momento de cargar la consola como tal:

nano /.bashrc

export JAVA HOME=/usr/lib/jvm/default-java

Una vez se hayan realizado estos pasos se debe ejecutar el fichero:

5 Source ~/.bashrc

A continuación, se empezará por descargar la última versión de ODL, en este caso, Boron:

En el momento que ya se haya instalado Java se podrá proceder a la instalación de OpenDayLight:

6 Wget

https://nexus.opendaylight.org/content/repositories/public/org/opendaylight/integration/distributionkaraf/0.5.3-Boron-SR3/distribution-karaf-0.5.3-Boron-SR3.zip

OBS: Puede encontrar otras versiones en: https://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight/i-ntegration/distribution-karaf/

Una vez descargado se extrae el archivo instalando unzip:

7 apt-get install unzip
8 unzip distribution-karaf-0.5.3-Boron-SR3.zip

Esto creará una carpeta llamada "distribution-karaf-0.5.3-Boron-SR3 "que es la que contiene el software y los plugins de OpenDayLight en esta carpeta personal de linux. Karaf es un contenedor tecnológico que permite a los desarrolladores situar todo el software requerido en una misma carpeta, permitiendo instalar o reinstalar ODL de una forma más fácil, ya que todo se encuentra en una misma ubicación.

Luego, se procede a arrancar ODL:



Ahora se deben instalar las características requeridas para la prueba de OpenDayLight y la GUI de OpenDayLight, los siguientes comandos se deben introducir en la ventana de comandos del usuario de ODL:

11 feature:install odl-restconf odl-l2switch-switch odl-mdsal-apidocs odl-dlux-all

Las características que se han instalado son las siguientes:

- odl-restconf: permite el acceso a RESTCONF API
- odl l2switch-switch: provee funcionalidades de red similares a un switch Ethernet.
- odl-mdsal-apidocs: permite el acceso a la Yang API.
- odl-dlux-all: la interfaz de usuario grafica de OpenDayLight.

Para ver las posibles características disponibles a instalar dentro de ODL basta con el ejecutar el comando:

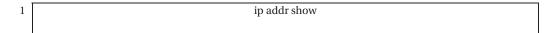
12	feature:list

Para ver las características ya instaladas dentro de ODL:

13 feature:list --installed

Paso Nº4: Verificación de dirección IP

Tras instalar ODL, se ha de buscar la dirección en la que se encontrara el controlador, para ello se teclea en una consola el comando:



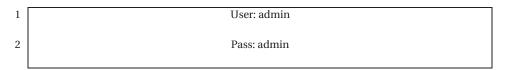
Paso Nº5: Accediendo a ODL

A continuación, será necesario dirigirse al navegador e introducir la siguiente URL:

(OBS: La respectiva ip XXX.YYY.Z.A corresponde a la puerta de enlace de su red y debe ser cambiada en dicha URL)



Finalmente se ingresará el usuario y contraseña en dicha URL para OpenDayLight como:



Luego será necesario un segundo terminal para establecer la topología inicial de la red SDN:

```
sudo mn –c

sudo mn –c

sudo mn --controller=remote,ip=XXX.YYY.Z.A

--switch=ovsk,protocols=OpenFlow13 --mac --topo=linear,3
```

Luego basta presionar el botón 'reload' para visualizar la red generada por Mininet, se presenta el entorno grafico generado por ODL en la figura D-1:

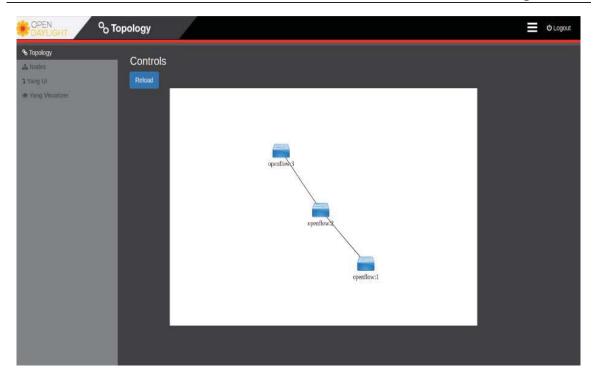
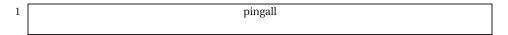


Figura D-1: Red SDN de tipo Lineal generada por Mininet

Luego para generar Hosts conectados a dichos switches basta con agregar el siguiente comando en la ventana donde se está ejecutando Mininet, el cual establecerá la conexión exitosa con los hosts y podrán visualizarse en nuestro controlador ODL:



Finalmente se puede visualizar la topología correspondiente en la siguiente figura D-2:

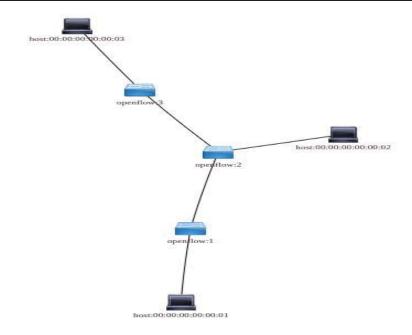


Figura D-2: Topología de red lineal de 3 switch y 3 host generados por Mininet

11gart D 2. Topologia de red inicai de 3 switch y 3 nost generados por iniminet
Evaluación:
Teórico:
Responda las siguientes preguntas respecto a la actividad realizada anteriormente:
1 ¿Cuál es el beneficio y desventaja de la aplicación de terminal (consola) en Linux (OS)?
2 ¿Qué función cumple Mininet en la experiencia realizada?

		D Experiencias
3 ¿Qué fun	nción Cumple OpenDayLight en la experiencia realizada?	
Práctico:		
4 Dados el	comando:	
1	Sudo mn -h	
	las opciones de comandos ahí mostrados para configurar su propia nexión en árbol con 20 switches con OpenFlow 1.0. Observe ht	
R:		

5.- Dada la siguiente Figura D-3:

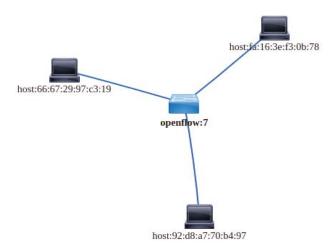


figura D-3: Topología de 1 switch OpenFlow en Mininet

Configure la topología de su red para que sea idéntica a la de la imagen

(Obs: Las direcciones MAC pueden ser distintas)

R:

6.- Para Finalizar elimine la última topología creada en Mininet y finalice la ejecución de OpenDayLight, en ambos casos consulte al menú de ayuda para saber que comandos realizan dichas acciones. Posteriormente termine por apagar el equipo.

R:

D.3 Pre-informe N°2

Guía Alumno

"Laboratorio sobre tecnología SDN y

Pre-informe N°2 NFV"

Nombre:	 	
Rut:	 Fecha:	

Objetivos:

- Introducir a los programas GNS3 y VMware para Simular topologías SDN
- Comprender el concepto de máquina virtual

Escenario:

Se plantea estudiar los programas GNS3 y VMware para diseñar una topología de redes definida por software (SDN) en un entorno Windows, por lo que se comprenderán las características de cada uno y su función al momento de trabajarlos próximamente en el aula.

¿Qué es GNS3?

GNS3 es un simulador gráfico de redes, routing y switching cuya principal característica es que es multiplataforma, se puede ejecutar tanto en Microsoft Windows, Linux como en Mac OS X, y todo ello de forma completamente gratuita.

¿Cómo se relaciona con SDN?

Con las configuraciones adecuadas y las correspondientes máquinas virtuales es capaz de generar topologías SDN, ya que permite una gran variedad de protocolos, entre ellos OpenFlow.

¿Qué es VMware?

VMware es un software de virtualización de sistemasoperativos, al cual le otorga componentes físicos con unas características de hardware determinadas. Cuando se ejecuta proporciona un ambiente de ejecución similar a todos los efectos a un computador físico, con CPU, BIOS, tarjeta gráfica, memoria RAM, tarjeta de red, sistema de sonido, conexión USB, disco duro, etc.

¿Cómo se relaciona con SDN?

Un virtualizador por software permite simular la ejecución de varios computadores dentro de un mismo hardware de manera simultánea, permitiendo así el mayor aprovechamiento de recursos, es decir permite optimizar por medio de la virtualización por lo que SDN es un enfoque similar. No obstante, y al ser una capa intermedia entre el sistema físico y el sistema operativo que funciona en el hardware emulado, la velocidad de ejecución de este último es menor, pero en la mayoría de los casos suficiente para usarse en entornos de producción.

Cuestionario de investigación :
¿Qué son las 'appliances'que nos ofrece GNS3 en su sitio web?
¿Cómo instalo un sistema operativo dentro de VMware?
¿VMware y GNS3 permiten su interconexión? De ser afirmativo ¿Cómo lo hacen?

D.4 Experiencia Nº2

Guía Alumno

"Laboratorio sobre tecnología SDN y

Experiencia N°2

N	ΕŢ	79
1.74	г, 1	/

Nombre:		
Rut:	Fecha:	

Objetivo:

- Comprender el uso de las máquinas virtuales y las funciones de red virtualizadas
- Entender las plataformas de trabajo GNS3 y VMware
- Configurar y visualizar redes definidas por software en un entorno Windows

Requerimientos:

- 1 PC (recomendado mínimo 6GB de RAM y 60GB de disco duro libre)
- Conexión a internet
- Windows (OS)
- Imágenes Virtuales de Mininet, Linux

Escenario:

En la presente práctica de Laboratorio, usted configurará una red SDN por medio de una máquina virtual en el entorno de Windows para visualizarla finalmente en OpenDayLight (ODL)

Desarrollo Experiencia:

Paso Nº1: Instalacion de Software GNS3 y VMware

Para la virtualización de sistemas operativos y su respectiva configuración en el sistema, es necesario instalar las siguientes aplicaciones en Windows (OS):

GNS3 (Plataforma de Emulación de redes)

Para su instalación es necesario ir a:

En esta URL se debe realizar el registro de su correo, para luego descargar el archivo .exe de GNS3 e instalar.

IMPORTANTE: Al finalizar la instalación no se debe ejecutar la aplicación GNS3, sino que debe abrir la siguiente URL:

https://gns3.com/marketplace/appliances

La URL corresponde a las aplicaciones adicionales a instalar dentro de nuestro emulador, es decir, ya sean sistemas operativos, routers, switch, navegadores, etc.

Las aplicaciones a descargar serán:

- Ubuntu
- Tiny-core Linux
- Firefox

VmWare Workstation:

Para su instalación es necesario ir a:

https://www.dropbox.com/sh/ixy7utkvj1wdtk3/AAA8_IVjxGh19jI5DRL1ALZra?dl=0

Se realizan los pasos adjuntos en la carpeta, es decir su instalación y registro

Paso Nº2: Descargar Máquinas Virtuales

A continuación, se descargan las máquinas virtuales a importar en VMware, siendo la primera de GNS3 VM y la segunda de Mininet

- https://github.com/GNS3/gns3gui/releases/download/v2.0.3/GNS3.VM.VMware.Workstation.2.0.3.zip
- 2 <u>https://github.com/mininet/mininet/wiki/Mininet-VM-Images</u>

Una vez descargadas se proseguirá a importar dichas máquinas.

Paso Nº3: Importar Máquinas Virtuales

Se prosigue importando estas máquinas virtuales a VMware, para ello es necesario seleccionar ambas imágenes virtuales y mover estas a la pestaña librería de la aplicación VMware, dando OK a las características predeterminadas propias de cada máquina virtual que aparecen al realizar esta acción.

Paso Nº4: Configuracion de Compatibilidad Windows

Abrir la configuración de firewall de Windows en el panel de control, y luego hacer click en la sección de 'permitir una aplicación o característica a través de firewall de Windows'.

En cambiar configuración debe seleccionar ambas casillas en la sección privada y pública en GNS3 y GNS3server para permitir la conexión de GNS3 a sus respectivos servidores.

Se dispone a ejecutar la aplicación GNS3 para definir la correspondiente configuración para la experiencia deseada

Al iniciar GNS3 la configuración inicial a elegir es como aparece en figura D-4:

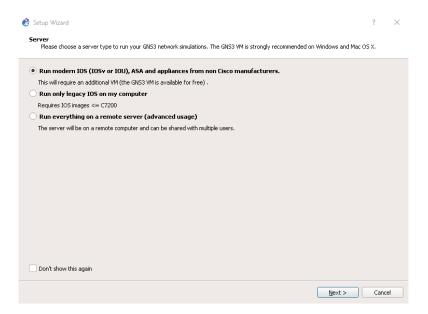


Figura D-4: Pantalla inicial GNS3

Presione Next hasta crear un nuevo proyecto

Presione "Ctrl+Shift+P" para abrir las configuraciones, en la pestaña GNS3 VM seleccione VMware y la imagen de GNS3 VM para ejecutar, posteriormente seleccione 'apply' y finalmente en OK

Luego en la pestaña File, y luego la sección import appliance

Y debe seleccionar una por una las aplicaciones descargadas anteriormente, descargando a través de GNS3 los paquetes adicionales necesarios.

OBS: la aplicación de Firefox permitirá acceder a su navegador web con lo cual se podra ver ODL mas adelante.

Se importará Mininet desde VMware para ello Presiona "Ctrl+Shift+P" para abrir las configuraciones, en la pestaña VMwares VMs seleccionamos VMware y seleccionamos "New" para luego agregar la VM mininet, presionamos apply y ok

Finalmente, las aplicaciones instaladas se encuentran configuradas para realizar la emulación de una topología de red funcional.

Paso Nº5: Configuración topología en GNS3

A continuación, se establecerá la siguiente topología arrastrando desde la pestaña "All Device" en la figura D-5:

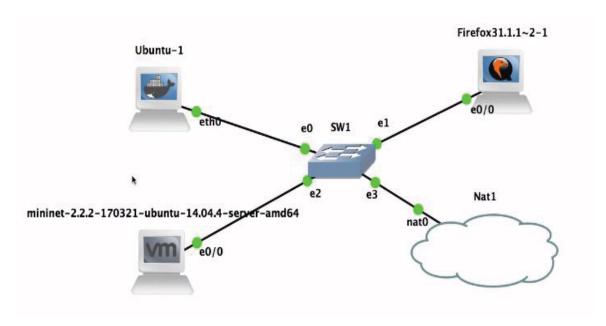


figura D-5: Topología para emular SDN en GNS3

se proseguirá a configurar Ubuntu-1:

- Clic derecho sobre Ubuntu-1
- Edit config
- Se quitará los signos "#" de las dos últimas líneas (sin borrar el texto siguiente)
- Save v OK

Finalmente, clic secundario en Ubuntu-1 y abrir consola

En consola se ejecutarán los siguientes comandos, con el fin de instalar ODL en la VM de Ubuntu:

	D Experie
en conso	la:
1	apt-get update
2	apt-get install default-jre-headless export
3	JAVA_HOME=/usr/lib/jvm/default-java apt-get
4	install wget
5	wget https://nexus.opendaylight.org/conten
6	apt-get install unzip
7	unzip distribution-karaf-0.5.2-Boron-SR3.zip
8	cd distribution-karaf-0.5.2-Boron-SR3/
9	./bin/karaf
	feature:install odl-restconf odl-l2switch-switch odl-mdsal-apidocs odl-dlux-all
10	
Una vez f	inalizado dicho proceso tenemos nuestra red configurada con éxit
Evaluaci	ón:

Teórico:
1 ¿Cuál es la función de una Máquina virtual?

¿Qué permite que se pueda realizar la topología de red propuesta?
¿la Aplicación Ubuntu-1 También es una VM? ¿porque no se usó desde VMWare?
Practico:
Con los conocimientos adquiridos de ambas experiencias configure la emulación en gns3 para enerar una red SDN con topología "Torus "y 9 switch con protocolo OpenFlow a elección en el M de mininet

5.- Visualice Por medio de la aplicación Firefox la interfaz de ODL y $\,$ verifique que la configuración de Mininet es la correcta

D.5 Pre-informe N3

Guía Alumno

"Laboratorio sobre configuración de Pre-informe N°3 elementos en SDN y Protocolo OpenFlow"

Nombre:		
Rut:	 Fecha:	

Objetivos:

- Ampliar los comandos utilizados para mininet
- Conocer la Plataforma WireShark

Escenario:

Se plantea estudiar la plataforma Mininet con más detalle para configurar los elementos dentro de la topología de manera independiente, además de visualizar los protocolos utilizados por wireshark para una próxima experiencia en el aula.

Comandos avanzados de mininet

Existen comandos dentro de mininet que nos permiten modificar características de elementos específicos dentro de topología ya establecida en la plataforma, desde los hosts o switches, hasta los enlaces asociados a su conexión., entre los más destacados se encuentran:

Comando	Descripción
mininet> link [nodo1] [nodo2] [up/down]	el comando "link" habilita o deshabilita el enlace entre dos nodos.
mininet> xterm [nodo1] [nodo]	el comando "xterm" abre una nueva terminal para los nodos especificados.

¿Qué es WireShark?

Wireshark es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica, disponible para las plataformas de Windows y Linux.

Cuestionario:
¿Es posible ver las conexiones en ODL al utilizar los comandos avanzados de mininet?
Detalle otros comandos que permitan verificar el estado de la conexión de los elementos en mininet.
Detalle los métodos de instalación para WireShark en Ubuntu-Linux
¿Cuál es la estructura del protocolo Openflow?

D.6 Experiencia Nº3

Guía Alumno

Experiencia N°3

"Laboratorio sobre configuración de elementos en SDN y Protocolo OpenFlow"

Nombre:	 	
Rut:	 Fecha:	

Objetivos:

- Configurar elementos dentro de las topologías SDN en mininet.
- Verificar, quitar y restablecer conexiones de enlaces entre componentes.
- Visualizar el Protocolo de comunicación OpenFlow empleado.
- Visualizar por medio de ODL las respectivas topologías

Requerimientos:

- 1 PC (RAM: 6GB como mínimo)
- Conexión a Internet
- Sistema Ubuntu-Linux (Versión 16.04 o superior) como partición del equipo
- Mininet y OpenDayLight ya instalados adecuadamente en Ubuntu-Linux

Nota: Es importante tener en cuenta que la sintaxis del código en el sistema operativo Ubuntu-Linux diferencia entre Mayúsculas y minúsculas, también que la consola de terminal requiera introducir la clave de la cuenta de usuario del sistema entre comandos.

Escenario:

Se establece comprender las funcionalidades del entorno Mininet y la configuración autónoma de cada dispositivo dentro de la red virtual, observando sus enlaces de comunicación y protocolos asociados, permitiendo así generar y facilitar el estudio de estas tecnologías.

Desarrollo Experiencia:

Paso Nº1: Comandos disponibles

Se abre un terminal (consola) en Ubuntu, en el cual se iniciará una topología sencilla por medio de mininet con el fin de estudiar las distintas opciones disponibles, para ello digite el siguiente comando:

```
1 Sudo mn --topo single,2
```

Dicha topología corresponde a solo un controlador SDN y dos hosts conectados por medio de un switch, a continuación, con intención de saber las diversas opciones disponibles una vez emulada la red teclee el comando help:

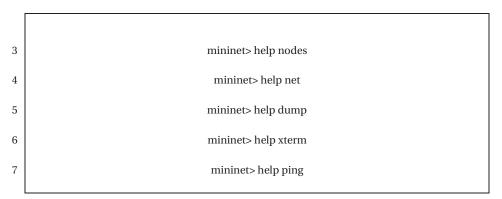
```
2 mininet> help
```

Lo que se mostrará en pantalla serán los diversos comandos disponibles para ejecutar en la topología de red simulada, como se observa en la 'figura D-6':

```
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> help
Documented commands (type help <topic>):
______
EOF
      gterm
             iperfudp nodes
                                                       switch
                                   pingpair
dpctl
      help
             link
                      noecho
                                   pingpairfull
                                                quit
                                                       time
dump
                      pingall
                                   ports
exit
                      pingallfull
                                                source
                                                       xterm
```

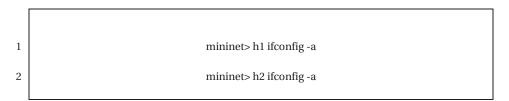
Figura D-6: comandos disponibles entorno mininet

Ahora es de interés comprender el uso de los comandos, para ello basta con anteponer la palabra 'help' en cada uno de ellos, eventualmente

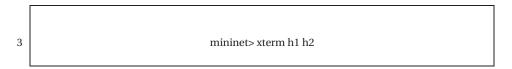


Paso Nº2: Verificación y configuración de elementos

Una vez ya comprendidos los comandos disponibles, procede a la configuración de los respectivos hosts y su estado, en consecuencia, verificará la configuración de cada uno con los comandos:



Lo que se mostrará en pantalla corresponderá a la conexión ethernet y loopback del host, es decir su dirección IP, máscara y estado de comunicación. A continuación, se procederá a verificar la conexión de los mismos por medio de sus respectivas consolas:



El comando anterior desplegara las consolas 'node h1' y 'node h2' respectivamente tal como se muestra en la 'figura d-7':

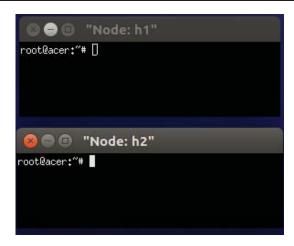


Figura D-7: Consolas hosts h1 y h2 para topología single de mininet.

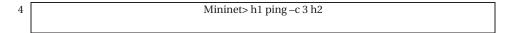
Por tanto, se realizará un ping desde estas consolas y la consola de mininet para ver su estado.

Primeramente, desde la consola 'node h1' se realizará el ping a la dirección IP de la interface h2-eth0 del host 'node h2' representada como XXX.YYY.Z.A. Por consiguiente, el nodo

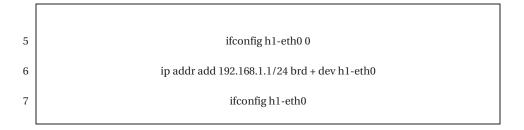
```
3 ping XXX.YYY.Z.A
```

La solicitud de ping se puede detener al presionar el conjunto de teclas 'CRTL+C' en la respectiva consola.

Mientras que para el caso de la consola de mininet es posible limitar la cantidad de ping y no se requiere el conocimiento de la dirección IP del host con el siguiente comando:



Ya verificadas las conexiones, se prosigue a modificar las direcciones IP y MAC de ambos hosts, de manera remota y al mismo tiempo por medio del controlador SDN. Ahora es necesario aplicar los siguientes comandos en el caso de la consola 'node h1':



El comando en la línea '5' selecciona la conexión a modificar, el comando en la línea '6', establece la nueva dirección y su respectiva mascara y permite modificar en h1-eth0, mientras que el

comando en la línea '7' verifica que el resultado sea el correcto, como se presenta en la figura d-8:

```
"Node: h1"

root@acer: "# ifconfig h1-eth0 0

root@acer: "# ip addr add 192,168.1.1/24 brd + dev h1-eth0

root@acer: "# ip addr add 192,168.1.1/24 brd + dev h1-eth0

root@acer: "# ifconfig h1-eth0 direcciónHW 7a:30:90:d5:b6:fd

Direc. inet:192.168.1.1 Difus:132.168.1.255 Másc:255.255.255.0

Dirección inet6: fe80::7830:30ff:fed5:b6fd/64 Alcance:Enlace

ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST HTU:1500 Métrica:1

Paquetes RX:42 errores:0 perdidos:0 overruns:0 frame:0

Paquetes TX:16 errores:0 perdidos:0 overruns:0 carrier:0

colisiones:0 long.colalX:1000

Bytes RX:5068 (5.0 KB) TX bytes:1244 (1.2 KB)

lo

Link encap:Bucle local

Direc. inet:127.0.0.1 Másc:255.0.0.0

Dirección inet6: ::1/128 Alcance:Anfitrión

ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1

Paquetes RX:0 errores:0 perdidos:0 overruns:0 frame:0

Paquetes TX:0 errores:0 perdidos:0 overruns:0 carrier:0

colisiones:0 long.colaTX:1000

Bytes RX:0 (0.0 B) TX bytes:0 (0.0 B)
```

Figura D-8: modificaciones a host por medio de consola en Ubuntu.

También se espera realizar dicha acción de forma análoga en la consola 'Node h2':

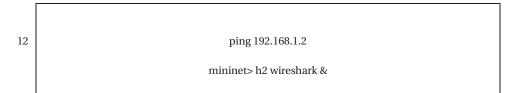
```
8 ifconfig h2-eth0 0
9 Ip addr add 192.168.1.2/24 brd + dev h2-eth0
```

Prosigue el cambio de dirección Mac para ambos hosts por medio del siguiente comando dentro de la consola 'Node:h1' y 'node:h2':

```
ifconfig h1-eth0 hw ether aa:bb:cc:dd:ee:ff
liconfig h2-eth0 hw ether gg:hh:ii:jj:kk:ll
```

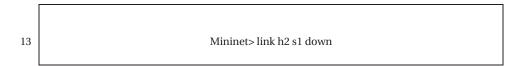
Paso Nº3: Enlaces de comunicación

En esta sección se evaluará el estado de la conexión del 'host h2' luego de bajar uno de los enlaces de la topología a través del comando 'link'. Para esta tarea en una primera instancia se realizará un ping desde la consola del host h1, además se utilizará Wireshark para observar el protocolo de comunicación OpenFlow:



Mientras el ping esté operativo, podrá ser visualizado, la transferencia a través de Wireshark al momento de desplegarse seleccione la visualización para h1-eth0 o en su defecto h2-eth0.

Mientras desde la consola 'node h1', se dirigirá a la consola de mininet para bajar el enlace del switch al host h2 'node h2' por medio del siguiente comando:



Una vez realizado el comando observe la consola 'node h1' donde estaba realizando el ping, este dejo de realizar el ping satisfactoriamente debido a que ha eliminado el único enlace disponible, esto se puede ver tanto en la consola como en wireshark.

Para regresar el enlace a la normalidad solo debe cambiar el estado down a up:

```
14 Mininet> link h2 s1 up
```

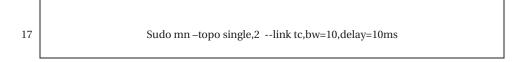
Podrá notar que el ping volvió a ejecutarse con normalidad, restableciéndose la conexión.

Paso Nº4: Prueba con ancho de banda

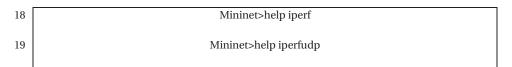
A continuación, se realizará la misma topología, pero esta vez se modificará el ancho de banda de los respectivos enlaces nuevamente con el comando 'link', por tanto, es necesario terminar la simulación actual y limpiar cualquier configuración guardada, para dicha tarea se tiene que:



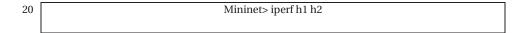
Se prosigue con la simulación de la red por medio del comando:



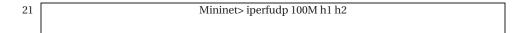
Luego para sí poder evaluar lo que ocurre con el ancho de banda se realizaran dos test que se encuentran dentro de las opciones de mininet, hablamos de iperf y iperfudp probando el rendimiento del ancho de banda UTP Y UDP respectivamente. Utilice el comando help para comprender su estructura:



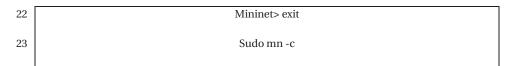
Comience por el comando 'iperf' entre los respectivos host



Dado el resultado podrá confirmar si el uso del protocolo UTP fue exitoso, mientras que para UDP no se garantiza que los paquetes lleguen exitosamente ya que estos pueden adelantarse unos con otros, a continuación, observe lo mencionado con el siguiente comando



Luego de finalizada la experiencia, recuerde finalizar mininet y limpiar posibles configuraciones guardas por medio de los comandos:



Paso Na5: Crear Topologias personalidas por Script Python y Tablas de Flujo OpenFlow

A continuación, para la creación de topologías personalizadas en mininet existe la posibilidad de crearlas por medio de lenguaje de programación Python, lenguaje en el cual está basado Mininet por consiguiente se procederá a generar un código que genere la siguiente topología de la Figura d-9:



Figura D-9: Topología personalizada de dos switch y dos hosts

Si bien dicha topología es creable con el comando:

```
01 Sudo mn –topo: linear,2
```

Este viene con la opción dpctl deshabilitada para visualizar las tablas de flujos que están operando en los switch a través del mismo emulador mininet además de la imposibilidad de agregar tablas de flujos manualmente, por tanto, se procede a generar la topología solicitada habilitando la administración de los flujos para los puertos respectivos.

Es consecuencia se debe utilizar la aplicación Notepadqq (editor de texto y programación), cuya funcionalidad nos permite generar el código en lenguaje de programación Python, realizando su instalación y ejecución de la siguiente manera:

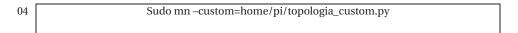
```
Sudo apt-get instal notepadqq
Sudo notepadqq
```

Una vez dentro de Notepadqq se prosigue a la generación del siguiente código, el cual se detalla a continuación en la ilustración D con los comentarios respectivos para su entendimiento.

```
#!/usr/bin/python
    from mininet.topo import Topo
    from mininet.net import Mininet
 4
    from mininet.cli import CLI
    from mininet.node import OVSSwitch
 6
    from mininet.log import setLogLevel
 7
    net = Mininet()
    print"+++++ Creando nodos en la red"
 8
 9
    c0 =net.addController('c1', port=6633)
    h1 =net.addHost("h1")
10
    h2 =net.addHost("h2")
11
    s1 =net.addSwitch("s1",listenPort=6634)
12
    s2 =net.addSwitch("s2",listenPort=6635)
13
14
15
    print"+++++ Creando enlaces"
    net.addLink(h1,s1)
16
17
    net.addLink(h2,s2)
18
    net.addLink(s1,s2)
19
20
    print"+++++Configurando direcciones IP"
    h1.setIP("192.168.0.1",24)
h2.setIP("192.168.0.2",24)
21
22
23
24
    print"+++++Corriendo red"
25
    net.build()
26
    sl.start (
                [ c0 ])
27
                [ c0 ])
    s2.start (
28
    net.start()
29
    net.pingAll()
30 v CLI(net)
```

Figura D-10: Código Python para topología personalizada en mininet.

Una vez generado, es guardado en la carpeta personal de la sesión de Ubuntu. Con el nombre "topologia_custom.py", luego en una nueva ventana de terminal de Ubuntu se prosigue a implementar la topología creada en la ubicación correspondiente:



Finalmente se procederá a estudiar las tablas de flujo del lenguaje de comunicación OpenFlow, para ello se utilizará el comando 'dpctl', dpctl es una utilidad de línea de comandos que transmite rápidamente mensajes OpenFlow, aplicación útil para la visualización del puerto de switch y de los estados de flujos, además permite también la inserción de entradas de flujo desde el terminal del switch.

Por tanto, se prosigue a abrir la configuración la consola de cada switch

```
05 Mininet> xterm s1 s2
```

En las consolas de s1 y s2 se puede visualizar las tablas de flujos con el siguiente comando que se detalla a continuación respectivamente:

```
04 # dpctl dump-flows tcp:192.168.0.1:6634
05 # dpctl dump-flows tcp:192.168.0.2:6635
```

Donde la sentencia 'dump-flows' nos permitirá visualizar los flujos de programación OpenFlow ejecutados en la dirección ip y la conexión del puerto TCP con la que dpctl realiza la conexión TCP con el switch.

A continuación, se agregarán flujos OpenFlow en las respectivas consolas de s1 y s2

```
6 # dpctl add-flow tcp: 192.168.0.1:6634 ip,nw_src=198.160.0.1,nw_dst=10.0.0.4,priority=10,idle_timeout=180,hard_timeout=0,actions=output:2
```

```
$dpctl add-flow tcp: 192.168.0.1:6634
ip,nw_src=10.0.0.4,nw_dst=10.0.0.3,priority=10,idle_timeout=180,hard_timeout=0,actions=output:1
```

Donde 'add-flow' es la sentencia que indica que se añadirá un flujo, donde se debe indicar la dirección IP del switch y el puerto con el que el dpctl creara la conexión TCP con el switch, en este caso como todo el tráfico es local se usa la dirección de la *Loopback*, a continuación, se ingresan las direcciones de IP origen y destino mediante los parámetros **nw_src** y **nw_dst**respectivamente. Opcionalmente se agrega la prioridad que se le dará al paquete, en el caso que se desee implementar calidad de servicio (QoS). Finalmente se define la acción a realizar por el switch para este flujo, en este caso la redirección a un puerto de salida.

Es posible visualizar los respectivos cambios a través wireshark, por tanto, primeramente, se abrirá la aplicación en un nuevo terminal de Linux, para luego ejecutar un segundo comando en otro terminal de Linux con el cual generaremos una list de flows para visualizar las tramas correspondientes en wireshark, es así como:

8	Sudo wireshark &	
9	sudo ovs-ofctl dump-flows s1 -O OpenFlow13	

¿A que corresponden los	Protocolos UTP y UDP?
Considere los diferentes se recomienda el apoyo	flujos, mostrados en wireshark y detalle cómo funciona el protocolo (obs: de un esquema)

Práctico:

Según lo aplicado en la experiencia anterior genere una topología de red por medio de mininet que se pueda visualizar en ODL, teniendo las siguientes características:

- Topología tipo árbol (tree) y que posea una profundidad de 3 y una cantidad de 3 ramas
- Debe tener un Ancho de banda de 100, una latencia de 21ms y una pérdida de 1%
- Una de las sub ramas debe tener su enlace abajo
- Debe configurar la dirección ip y mascara para cada host

B Desarrollo Experiencias para Docentes

E.1 Pre-informe Nº1

Guía Profesor

"Laboratorio Introductorio a Tecnología

Pre-informe N°1 SDN"

Nombre:	 	
Rut:	 Fecha:	

Objetivos:

- Introducir a los conceptos del entorno Ubuntu-Linux, SDN, Mininet y OpenDayLight
- Reconocer los comandos existentes para cada una de las plataformas

Escenario:

Se plantea estudiar los conocimientos para diseñar una topología de redes definida por software (SDN) por lo que se comprenderán los entornos de trabajo que permiten su diseño y estudio, para así llevar a cabo su implementación en una próxima experiencia en el aula.

Entendiendo Ubuntu-Linux

¿Qué es Ubuntu?

Ubuntu corresponde a una distribución del sistema operativo Linux el cual es un software de carácter libre. La principal característica de Ubuntu es que logra hacer a Linux un sistema operativo más accesible y fácil de usar.

Conceptos Básicos

Dentro del sistema operativo Ubuntu-Linux existe la aplicación llamada 'Terminal' o 'Consola' a la cual se puede acceder presionando "Ctrl+Alt+t" o buscando dicha aplicación en la barra del sistema.

La aplicación consola permite manejar el sistema en todos los aspectos, y es posible utilizar algunos comandos para instalar programas, borrar, copiar y mover archivos, editar archivos de texto y solucionar prácticamente cualquier error del sistema. Sus beneficios radican en la velocidad de ejecución por medio de comandos, ya que pueden realizar eficientemente en una línea de comando lo que tardaría varios clics en una interfaz gráfica.

Comandos Basicos

Dentro de la aplicación Consola es posible escribir los siguientes comandos para realizar las respectivas acciones descritas:

- **apt-get install** *nombre_programa*: instalar / actualizar un programa. (requiere conexión a internet para descargar e instalar el programa)
- apt-get update: actualizar la lista de paquetes (programas).
- apt-get remove nombre_programa: eliminar un programa del sistema.
- apt-get clean: limpiar cache desde los paquetes descargados.
- cd: ir al directorio raíz.
- **ls**: ver los ficheros de un directorio.
- cd /home: entrar en el directorio "home".
- **cd** ..: retroceder un nivel dentro del directorio actual.
- **clear:** Limpia el terminal borrando todo el texto que se ha escrito hasta ahora.
- Exit: cierra el terminal (consola)

Entendiendo SDN

¿Qué es SDN?

SDN (Software-Defined Networking) es la separación en la gestión de los dispositivos de red del plano de control del plano de datos, en otras palabras, se trata de la separación del hardware y del software, y donde el control de la red es llevado a cabo por dispositivos denominados controladores.

Conceptos Básicos

El objetivo principal que persigue SDN es facilitar la implementación e implantaciónde servicios de red de una manera determinista, dinámica, económica y escalable, evitandoal administrador de red gestionar dichos servicios a bajo nivel.

Protocolo de Comunicación

El primer protocolo de comunicación estándar para la tecnología SDN se denomina Openflow. No obstante, también existen protocolos como Netconfig/ Yang o BGP.

Entendiendo Mininet

¿Qué es Mininet?

Mininet es un emulador de red que crea una red de hosts virtuales, conmutadores, controladores y enlaces. Los hosts de Mininet ejecutan un software de red Linux estándar, y sus switches admiten OpenFlow para un enrutamiento personalizado altamente flexible y redes definidas por software

Conceptos Básicos

Mininet es rápidamente reconfigurable y reiniciable, ofreciendo un buen rendimiento de trabajo bajo una pequeña cantidad de comandos, este programa puede ser usado como una máquina virtual (VM) o directamente desde la consola de Ubuntu-Linux.

La topología por defecto para Mininet consta de un switch, un controlador (local) y dos hosts.

Comandos Básico

Mininet permite generar diversas topologías, no obstante, su configuración más básica viene dada por el siguiente comando:



Comando que genera una topología de 2 hosts, un switch y un controlador SDN, además esta puede modificarse agregando más opciones, las cuales son visibles al agregar '-h'a dicho comando, es decir:



Finalmente, también existe un comando que regresa todas las configuraciones a cero llamado clear:



Entendiendo OpenDayLight

¿Qué es OpenDaylight?

Este proyecto es promovido por *The Linux Foundation* con el cual se impulsa al desarrollo, distribución y adopción global de la plataforma SDN Open Daylight (ODL) de código abierto y de índole neutral

¿Cómo se relaciona con SDN y Mininet?

ODL corresponde a la interfaz gráfica que funciona como administrador para el controlador SDN observando la topología creada por mininet, sus protocolos de comunicación, entre otras aplicaciones.

Cuestionario:

¿Cuál es la función del comando 'sudo' dentro de linux?

Corresponde a la característica de administrador, lo que permite ejecutar comandos que están restringidos para el usuario standard

¿Cuál es el conjunto de teclas que permiten copiar y pegar dentro del terminal de Linux?

Corresponden a CTRL+shift+C para copiar y Ctrl+Shift+V para pegar dentro del terminal de linux

-Detalle las topologías proporcionadas por mininet al momento de generar redes.

Es necesario agregar la sentencia "--topo=[PARAMETRO]": lo que permite cambiar el tamaño y el tipo de topología a emular:

- linear: genera una topología de k switches en serie con n hosts conectados a cada switch. Usa argumentos como:
 - o k=[NUMERO DE SWITCHES]
 - o n=[NUMERO DE HOSTS POR SWITCH]
- minimal: genera una topología simple de dos hosts y un switch.
- single: genera una topología simple de un switch y N hosts. Usa el argumento:
 - k= [NUMERO DE HOSTS]
- reversed: similar a una topología single, pero esta invierte el orden de los puertos en el switch. Una topología single asigna los puertos del switch en orden ascendente mientras que 'esta lo hace de forma descendente. Utiliza argumentos como:
 - k= [NUMERO DE HOSTS]
- tree: genera una topología de árbol, compuesta de N niveles (depth), Nramas (fanout) y dos hosts conectados a cada switch hoja. Usa argumentos como
 - o depth=[PROFUNDIDAD]
 - o fanout=[NUMERO DE RAMAS] ´
- torus: genera una topología en forma de malla MxN donde cada switch se conecta con sus vecinos más cercanos y los switches del borde se conectan con el opuesto. Mininet solo

soporta topologías 2D, este tipo de redes tiene bucles y podría no funcionar con dispositivos que no soporten el protocoloSpanning Tree, por lo tanto, se recomienda utilizarla con switches lxbr,compatibles con STP. Recibe como argumentos:

- \circ x=[M]
- \circ y=[N]

- Escriba los comandos para dos topologías distintas a ejecutar con mininet, asigne también una dirección Mac para cada host dentro de la red.

```
sudo mn --controller=remote,ip=192.168.0.1 --switch=ovsk,protocols=OpenFlow13 --mac --
topo=linear,3

sudo mn --controller=remote,ip=192.168.0.1 --switch=ovsk,protocols=OpenFlow13 --mac --
topo=single,5
```

¿A que corresponden el apartado 'features' dentro de ODL al momento de ejecutarlo?

Corresponden a las características que están dentro de ODL, al ver estas, nos indica aquellas que se encuentran activadas

¿Cómo es que se conectan mininet y ODL a través de Linux?

Utilizan la misma dirección ip brindada por medio de la puerta de enlace otorgada por los servicios de red, y configurando estos correctamente a través de sus respectivos terminales.

E.2 Pre-informe N°2

Guía Profesor

"Laboratorio sobre tecnología SDN y

Pre-informe N°2 NFV"

Nombre:	 	
Rut:	 Fecha:	

Objetivos:

- Introducir a los programas GNS3 y VMware para Simular topologías SDN
- Comprender el concepto de máquina virtual

Escenario:

Se plantea estudiar los programas GNS3 y VMware para diseñar una topología de redes definida por software (SDN) en un entorno Windows, por lo que se comprenderán las características de cada uno y su función al momento de trabajarlos próximamente en el aula.

¿Qué es GNS3?

GNS3 es un simulador gráfico de redes, routing y switching cuya principal característica es que es multiplataforma, se puede ejecutar tanto en Microsoft Windows, Linux como en Mac OS X, y todo ello de forma completamente gratuita.

¿Cómo se relaciona con SDN?

Con las configuraciones adecuadas y las correspondientes máquinas virtuales es capaz de generar topologías SDN, ya que permite una gran variedad de protocolos, entre ellos OpenFlow.

¿Qué es VMware?

VMware es un software de virtualización de sistemasoperativos, al cual le otorga componentes físicos con unas características de hardware determinadas. Cuando se ejecuta proporciona un ambiente de ejecución similar a todos los efectos a un computador físico, con CPU, BIOS, tarjeta gráfica, memoria RAM, tarjeta de red, sistema de sonido, conexión USB, disco duro, etc.

¿Cómo se relaciona con SDN?

Un virtualizador por software permite simular la ejecución de varios computadores dentro de un mismo hardware de manera simultánea, permitiendo así el mayor aprovechamiento de recursos, es decir permite optimizar por medio de la virtualización por lo que SDN es un enfoque similar.

No obstante, y al ser una capa intermedia entre el sistema físico y el sistema operativo que funciona en el hardware emulado, la velocidad de ejecución de este último es menor, pero en la mayoría de los casos suficiente para usarse en entornos de producción.

Cuestionario:

¿Qué son las 'appliances' que nos ofrece GNS3 en su sitio web?

Corresponde a aplicaciones que permiten simular diferentes equipos o características dentro dentro del programa gns3, agregando estas 'appliances' por medio del menú disponible en la pestaña de archivo y luego importar aplicaciones.

¿Cómo instalo un sistema operativo .dentro de VMware?

Es necesario tener la máquina virtual del equipo a virtualizar, o en su defecto, su disco de instalación y seleccionar sus respectivas características en VMware, para eso se debe ir a pestaña archivo e importar nueva máquina virtual y luego seguir los pasos que se despliegan seleccionando el sistema operativo correspondiente.

¿VMware y GNS3 permiten su interconexión? De ser afirmativo ¿Cómo lo hacen?

Si lo permiten, esto se logra a traves de la pestaña de configuración de los mismo, ya que a su vez también GNS3 posee su propia máquina virtual que puede ser instalada dentro de VMware

E.3 Pre-informe No3

Guía Profesor

"Laboratorio sobre configuración de Pre-informe N°3 elementos en SDN y Protocolo OpenFlow"

Nombre:	 	
Rut:	 Fecha:	

Objetivos:

- Ampliar los comandos utilizados para mininet
- Conocer la Plataforma WireShark

Escenario:

Se plantea estudiar la plataforma Mininet con más detalle para configurar los elementos dentro de la topología de manera independiente, además de visualizar los protocolos utilizados por wireshark para una próxima experiencia en el aula.

Comandos avanzados de mininet

Existen comandos dentro de mininet que nos permiten modificar características de elementos específicos dentro de topología ya establecida en la plataforma, desde los hosts o switches, hasta los enlaces asociados a su conexión., entre los más destacados se encuentran:

Comando	Descripción
mininet> link [nodo1] [nodo2] [up/down]	el comando "link" habilita o deshabilita el enlace entre dos nodos.
mininet> xterm [nodo1] [nodo]	el comando "xterm" abre una nueva terminal para los nodos especificados.

¿Qué es WireShark?

Wireshark es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica, disponible para las plataformas de Windows y Linux.

Cuestionario:

¿Es posible ver las conexiones en ODL al utilizar los comandos avanzados de mininet?

Efectivamente se pueden observar, incluso se pueden generar topologías a través de lenguaje de programación Python.

Detalle otros comandos que permitan verificar el estado de la conexión de los elementos en mininet.

Existen comandos como

Mininet> h1 ping -c 3 h2

Lo que permite verificar el estado de conexión de otro dispositivo sin necesidad de saber su dirección ip, no obstante '-c3' es necesario para solo hacer 3 peticiones, de lo contrario al no indicar algún numero este comando estará realizando ping constantemente entre ambos dispositivos

Detalle los métodos de instalación para WireShark en Ubuntu-Linux

Para la instalación del Wireshark basta con escribir en la ventana de comandos terminal, el siguiente comando:

1 sudo apt-get install wireshark

¿Cuál es la estructura del protocolo Openflow?

Detalladamente se puede ver que los componentes de su estructura se conforman de:

- Controller: Es una entidad que interactúa con el switch OpenFlow utilizando el protocoloOpenFlow. En la mayoría de los casos, un controlador OpenFlow es un software que controla muchos switcheslógicosOpenFlow.
- Control Channel: Cumple la agregación de componentes de un switch lógico OpenFlow que gestiona Comunicación con los controladores. El canal de control incluye un canal OpenFlow por controlador OpenFlow.
- OpenFlow Channel: Interfaz entre un switch OpenFlow y un controlador OpenFlow, utilizado por el controlador para gestionar el switch.
- Pippeline: El conjunto de tablas de flujo vinculadas que proporcionan adaptación, reenvío y modificación de paquetes en un conmutador OpenFlow
- Flow Table:Una etapa de la 'pipeline' que contiene entradas de flujo
- Port:Donde los paquetes entran y salen dela 'pipeline' de OpenFlow. Puede ser un puerto físico, un puerto lógico o un puerto previamente reservado y definido por el protocolo del conmutador OpenFlow.

- Group Table:Una lista de segmentos de acción y algunos medios para elegir uno o más de esos segmentos en un paquete base.
- Meter Table: Un elemento de conmutación que puede medir y controlar la velocidad de los paquetes. El medidor activa una banda de medidores si la tasa de paquetes o la velocidad de bytes que pasa a través del medidor excede un umbral predefinido la banda del medidor deja caer el paquete, también se le llama limitador de tráfico.
- Datapath: Corresponde a la agregación de componentes de un switch lógico OpenFlow que están directamente involucrados en el procesamiento y reenvío de tráfico. El datapath incluye tablas de flujo de la 'pippeline', la 'group table' y los 'port'.

E.4 Experiencia Nº1

Guía Profesor Experiencia N°1

"Laboratorio Introductorio a Tecnología SDN"

Nombre			
:	_		
Rut:			
	_	Fecha:	

Objetivo:

• implementar y comprender topologías para una red SDN y su respectiva visualización grafica a través de aplicaciones en el sistema operativo Ubuntu-Linux.

Requerimientos:

- 1 PC (Ram: 6GB como mínimo, puerto Ethernet y/o Tarjeta WLAN)
- Conexión a Internet
- Sistema Ubuntu-Linux (Versión 16.04 o superior) como partición del equipo

Nota: Es importante tener en cuenta que la sintaxis del código en el sistema operativo Ubuntu-Linux diferencia entre mayúsculas y minúsculas, también que la consola de terminal requiere introducir la clave de la cuenta de usuario del sistema entre comandos.

Escenario:

Se debe implementar un entorno de trabajo de tecnología SDN en el sistema Ubuntu-Linux desde cero a partir del terminal (consola) del sistema operativo, esto implica instalar Mininet y OpenDayLight en conjunto con sus requerimientos para posteriormente configurar y generar una red SDN y así visualizarla.

Desarrollo Experiencia

Paso Nº1: Conociendo El terminal de Ubuntu-Linux

Una vez estando en el entorno gráfico de Ubuntu, es necesario abrir un terminal del sistema, lo cual se puede lograr presionando "Ctrl+Alt+t" o buscando dicha aplicación en la barra del sistema.

El terminal o consola es una aplicación que permite realizar múltiples acciones por medio de breves comandos, para esta actividad se necesita instalar las aplicaciones llamadas Mininet y OpenDayLight.

Paso Nº2: Instalación de Mininet y sus requerimientos

A través del terminal se procede a la instalación nativa (a través de comandos) de Mininet en Ubuntu:

Con este comando se instalarán los núcleos para Mininet:

```
1 sudo apt-get install mininet
```

Este paso permite eliminar cualquier controlador que Mininet pueda haber activado anteriormente:

```
2 sudo mn -c
```

Con esto se instala git para que Mininet pueda ser descargado desde GitHub (Foro de la comunidad dedicada a Mininet) y crear una estructura de ceros en nuestra máquina.

```
sudo apt-get install git
sudo git clone git://github.com/mininet/mininet
```

Se descarga la aplicación "git", ya que esta permite ver el control de versiones para los respectivos programas desarrollados para Linux, con los siguientes comandos, lo que hará esto es cambiar a la última versión de Mininet.

```
cd mininet
git tag # Lista de versiones disponibles
git checkout -b 2.2.1 2.2.1 # seleccionar ultima version par instalar
cd ..
```

El siguiente paso es el más importante ya que permite instalar todos los elementos que ofrece Mininet. Si este paso no es llevado a cabo Mininet no podrá conectarse correctamente con el controlador y no funcionará.

```
9 mininet/util/install.sh -a
```

En caso de que este comando no funcione utilizar el siguiente:

```
mininet/util/install.sh -a Ubuntu trusty i386 Ubuntu
```

Con esto ya se tendrá instalado Mininet.

Paso Nº3: Instalación OpenDayLight y sus requerimientos

A través de un segundo terminal prosiga mediante la siguiente manera para la instalación nativa del controlador OpenDayLight en Ubuntu:

Primero se debe instalar la aplicación Java ya que este controlador es un programa Java.

Para instalar Java se deben usar los siguientes comandos:

```
sudo apt-get update
sudo apt-get install default-jre-headless
```

Tras esto hay que editar el fichero ".bashrc" el cual está encargado de los comandos que se ejecutan al momento de cargar la consola como tal:

```
nano /.bashrc
export JAVA HOME=/usr/lib/jvm/default-java
```

Una vez se hayan realizado estos pasos se debe ejecutar el fichero:

```
5 Source ~/.bashrc
```

A continuación, se empezará por descargar la última versión de ODL, en este caso, Boron:

En el momento que ya se haya instalado Java se podrá proceder a la instalación de OpenDayLight:

```
Wget
https://nexus.opendaylight.org/content/repositories/public/org/opendaylight/inte
gration/distribution-karaf/0.5.3-Boron-SR3/distribution-karaf-0.5.3-Boron-
SR3.zip
```

OBS: Puede encontrar otras versiones en:

https://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight/integration/distribution-karaf/

Una vez descargado se extrae el archivo instalando unzip:

```
apt-get install unzip
unzip distribution-karaf-0.5.3-Boron-SR3.zip
```

Esto creará una carpeta llamada "distribution-karaf-0.5.3-Boron-SR3 "que es la que contiene el software y los plugins de OpenDayLight en esta carpeta personal de linux. Karaf es un contenedor tecnológico que permite a los desarrolladores situar todo el software requerido en una misma

carpeta, permitiendo instalar o reinstalar ODL de una forma más fácil, ya que todo se encuentra en una misma ubicación.

Luego, se procede a arrancar ODL:

```
10 cd distribution-karaf-0.5.3-Boron-SR3/bim/
11 ./karaf
```

Ahora se deben instalar las características requeridas para la prueba de OpenDayLight y la GUI de OpenDayLight, los siguientes comandos se deben introducir en la ventana de comandos del usuario de ODL:

```
feature:install odl-restconf odl-12switch-switch odl-mdsal-apidocs odl-dlux-
all
```

Las características que se han instalado son las siguientes:

- odl-restconf: permite el acceso a RESTCONF API
- odl l2switch-switch: provee funcionalidades de red similares a un switch Ethernet.
- odl-mdsal-apidocs: permite el acceso a la Yang API.
- odl-dlux-all: la interfaz de usuario grafica de OpenDayLight.

Para ver las posibles características disponibles a instalar dentro de ODL basta con el ejecutar el comando:

```
13 feature:list
```

Para ver las características ya instaladas dentro de ODL:

```
14 feature:list --installed
```

Paso Nº4: Verificación de direccción IP

Tras instalar ODL, se ha de buscar la dirección en la que se encontrara el controlador, para ello se teclea en una consola el comando:

```
1 ip addr show
```

Paso No5: Accediendo a ODL

A continuación, será necesario dirigirse al navegador e introducir la siguiente URL:

(OBS: La respectiva ip XXX.YYY.Z.A corresponde a la puerta de enlace de su red y debe ser cambiada en dicha URL)

```
1 <a href="http://XXX.YYY.Z.A:8181/index.html#/login">http://XXX.YYY.Z.A:8181/index.html#/login</a>
```

Finalmente se ingresará el usuario y contraseña en dicha URL para OpenDayLight como:

```
2 User: admin
3 Pass: admin
```

Luego será necesario un segundo terminal para establecer la topología inicial de la red SDN:

```
sudo mn -c
sudo mn --controller=remote,ip=XXX.YYY.Z.A
--switch=ovsk,protocols=OpenFlow13 --mac --topo=linear,3
```

Luego basta presionar el botón 'reload' para visualizar la red generada por Mininet, se presenta el entorno grafico generado por ODL en la figura E-1:

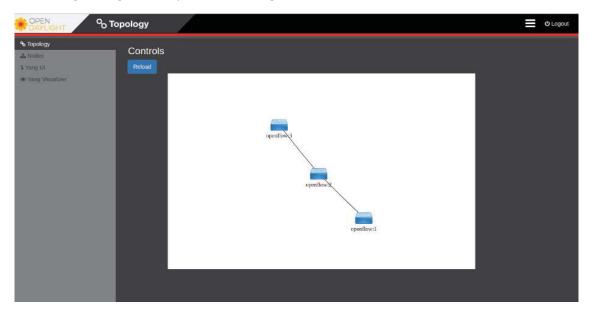


Figura E-1: Red SDN de tipo Lineal generada por Mininet

Luego para generar Hosts conectados a dichos switches basta con agregar el siguiente comando en la ventana donde se está ejecutando Mininet, el cual establecerá la conexión exitosa con los hosts y podrán visualizarse en nuestro controlador ODL:

```
1 pingall
```

Finalmente se puede visualizar la topología correspondiente en la siguiente figura E-2:

Figura E-2: Topología de red lineal de 3 switch y 3 host generados por Mininet

Evaluación:

Teórico:

Responda las siguientes preguntas respecto a la actividad realizada anteriormente:

host:00:00:00:00:00:01

1.- ¿Cuál es el beneficio y desventaja de la aplicación de terminal (consola) en Linux (OS)?

R: El no poseer una interfaz gráfica como el común de las aplicaciones es tanto una ventaja como desventaja, ya que, por una parte, se ahorran muchos procedimientos al utilizar solo un comando de línea, pero al mismo tiempo no es muy intuitivo el comando adecuado a utilizar para realizar una determinada acción sin la experiencia previa.

2.- ¿Qué función cumple Mininet en la experiencia realizada?

R: Genera la topología para la red SDN, es decir genera la cantidad de switches y Hosts y la forma en que estos están conectados

3.- ¿Qué función Cumple OpenDayLight en la experiencia realizada?

R: Genera la interfaz gráfica para visualizar la topología asociada a la dirección IP correspondiente, facilitando su comprensión y análisis.

Práctico:

4.- Dados el comando:

Sudo mn -h

Comprenda las opciones de comandos ahí mostrados para configurar su propia red SDN con un tipo de conexión en árbol con 20 switches con OpenFlow 1.0. Observe el resultado por OpenDaylight

R: \$ sudo mn --controller=remote --topo tree, 20 --mac --switchovsk, protocols=OpenFlow10

5.- Dada la siguiente la figura E-3

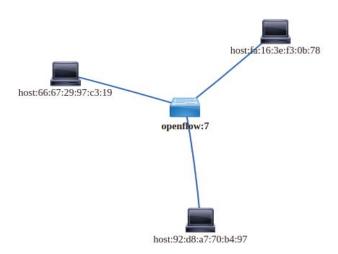


figura E-3: Topología de 1 switch OpenFlow en Mininet

Configure la topología de su red para que sea idéntica a la de la figura E-3

(Obs: Las direcciones MAC pueden ser distintas)

R: \$ sudo mn --controller=remote --topo single, 3 --mac --switchovsk, protocols=OpenFlow10

6.- Para Finalizar elimine la última topología creada en Mininet y finalice la ejecución de OpenDayLight, en ambos casos consulte al menú de ayuda para saber que comandos realizan dichas acciones. Posteriormente termine por apagar el equipo.

R:

Mininet → sudo mn –c / luego exit.

Opendaylight → logout /luego exit.

E.5 Experiencia Nº2

Guía Profesor	
Experiencia N°2	"Laboratorio sobre tecnología SDN y NF
Nombre	
: _ Rut:	

Fecha:

Objetivo:

- Comprender el uso de las máquinas virtuales y las funciones de red virtualizadas
- Entender las plataformas de trabajo GNS3 y VMware
- Configurar y visualizar redes definidas por software en un entorno Windows

Requerimientos:

- 1 PC (recomendado mínimo 6GB de RAM y 60GB de disco duro libre)
- Conexión a internet
- Windows (OS)
- Imágenes Virtuales de Mininet, Linux

Escenario:

En la presente práctica de Laboratorio, usted configurará una red SDN por medio de una máquina virtual en el entorno de Windows para visualizarla finalmente en OpenDayLight (ODL)

Desarrollo Experiencia:

Paso Nº1: Instalacion de Software GNS3 y VMware

Para la virtualización de sistemas operativos y su respectiva configuración en el sistema, es necesario instalar las siguientes aplicaciones en Windows (OS):

GNS3 (Plataforma de Emulación de redes)

Para su instalación es necesario ir a:

1 https://gns3.com/software/download

En esta URL se debe realizar el registro de su correo, para luego descargar el archivo .exe de GNS3 e instalar.

IMPORTANTE: Al finalizar la instalación no se debe ejecutar la aplicación GNS3, si no que debe abrir la siguiente URL:

https://gns3.com/marketplace/appliances

La URL corresponde a las aplicaciones adicionales a instalar dentro de nuestro emulador, es decir, ya sean sistemas operativos, routers, switch, navegadores, etc.

Las aplicaciones a descargar serán:

- Ubuntu

1

- Tiny-core Linux
- Firefox

VmWare Workstation:

Para su instalación es necesario ir a:

 $^{1} \quad \underline{\text{https://www.dropbox.com/sh/ixy7utkvj1wdtk3/AAA8_IVjxGh19jI5DRL1ALZra?dl=0}}\\$

Se realizan los pasos adjuntos en la carpeta, es decir su instalación y registro

Paso Nº2: Descargar Máquinas Virtuales

A continuación, se descargan las máquinas virtuales a importar en VMware, siendo la primera de GNS3 VM y la segunda de Mininet

- https://github.com/GNS3/gns3gui/releases/download/v2.0.3/GNS3.VM.VMware.Workstation.2.0.3.zip
- https://github.com/mininet/mininet/wiki/Mininet-VM-Images

Una vez descargadas se proseguirá a importar dichas máquinas.

Paso Nº3: Importar Máquinas Virtuales

Se prosigue importando estas máquinas virtuales a VMware, para ello es necesario seleccionar ambas imágenes virtuales y mover estas a la pestaña librería de la aplicación VMware, dando OK a las características predeterminadas propias de cada máquina virtual que aparecen al realizar esta acción.

Paso Nº4: Configuracion de Compatibilidad Windows

Abrir la configuración de firewall de Windows en el panel de control, y luego hacer click en la sección de 'permitir una aplicación o característica a través de firewall de Windows'.

En cambiar configuración debe seleccionar ambas casillas en la sección privada y pública en GNS3 y GNS3server para permitir la conexión de GNS3 a sus respectivos servidores.

Se dispone a ejecutar la aplicación GNS3 para definir la correspondiente configuración para la experiencia deseada

Al iniciar GNS3 la configuración inicial a elegir es la que se muestra en la figura E-4:

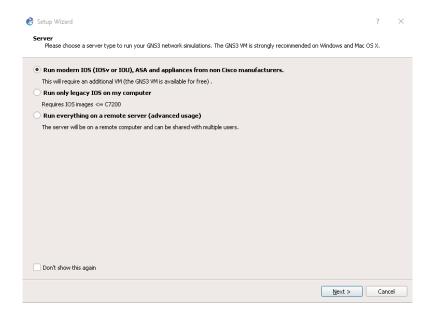


Figura E-4: Pantalla inicial GNS3

Presione Next hasta crear un nuevo proyecto

Presione "Ctrl+Shift+P" para abrir las configuraciones, en la pestaña GNS3 VM seleccione VMware y la imagen de GNS3 VM para ejecutar, posteriormente seleccione 'apply' y finalmente en OK

Luego en la pestaña File, y luego la sección import appliance

Y debe seleccionar una por una las aplicaciones descargadas anteriormente, descargando a través de GNS3 los paquetes adicionales necesarios.

OBS: la aplicación de Firefox permitirá acceder a su navegador web con lo cual se podrá ver ODL mas adelante.

Se importará Mininet desde VMware para ello Presiona "Ctrl+Shift+P" para abrir las configuraciones, en la pestaña VMwares VMs seleccionamos VMware y seleccionamos "New" para luego agregar la VM mininet, presionamos apply y ok

Finalmente, las aplicaciones instaladas se encuentran configuradas para realizar la emulación de una topología de red funcional.

Paso Nº5: Configuración topología en GNS3

A continuación, se establecerá la siguiente topología arrastrando desde la pestaña "All Device" en la figura E-5

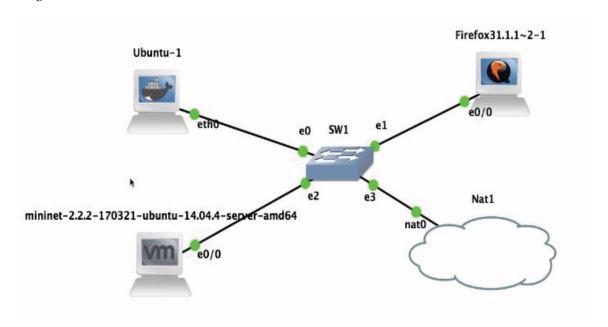


Figura E-5: Topología para emular SDN en GNS3

se proseguirá a configurar Ubuntu-1:

- Clic derecho sobre Ubuntu-1
- Edit config
- Se quitará los signos "#" de las dos últimas líneas (sin borrar el texto siguiente)
- Save y OK

Finalmente, clic secundario en Ubuntu-1 y abrir consola

En consola se ejecutarán los siguientes comandos, con el fin de instalar ODL en la VM de Ubuntu: en consola:

apt-get update
 apt-get install default-jre-headless export

JAVA_HOME=/usr/lib/jvm/default-java apt-get

install wget

wget https://nexus.opendaylight.org/conten...

apt-get install unzip

unzip distribution-karaf-0.5.2-Boron-SR3.zip

cd distribution-karaf-0.5.2-Boron-SR3/

./bin/karaf

feature:install odl-restconf odl-l2switch-switch odl-mdsal-apidocs odl-dlux-all

Una vez finalizado dicho proceso tenemos nuestra red configurada con éxito.

Evaluación:

Teórico:

1.- ¿Cuál es la función de una Máquina virtual?

R: Una máquina virtual es un software que crea una capa independiente donde se emula el funcionamiento de un ordenador real con todos los componentes de hardware que necesita para funcionar (disco duro, memoria RAM, tarjetas de red, tarjeta gráfica, etc.) y que puede ejecutar cualquier sistema operativo o programa, tal y como lo haría un ordenador real.

2.- ¿Qué permite que se pueda realizar la topología de red propuesta?

R: Emular una Red SDN por medio de VM de ODL y Mininet, utilizando una conexión NAT y la aplicación de Firefox para poder visualizar ODL.

3.- ¿la Aplicación Ubuntu-1 También es una VM? ¿porque no se usó desde VMWare?

R: Si también lo es, con la diferencia que fue creada por los programadores de gns3 para mejorar sus características y rendimiento y así dejar sus configuraciones más fáciles de modificar.

Practico:

- 4.- Con los conocimientos adquiridos de ambas experiencias configure la emulación en gns3 para generar una red SDN con topología "Torus "y 9 switch con protocolo OpenFlow a elección en el VM de mininet
- R: --sudo mn --controller=remote, ip=192.168.0.1 --topo=torus,3,3 --mac --switchovsk, protocols=OpenFlow10 --nat
- 5.- Visualice Por medio de la aplicación Firefox la interfaz de ODL y verifique que la configuración de Mininet es la correcta para lafigura E-6

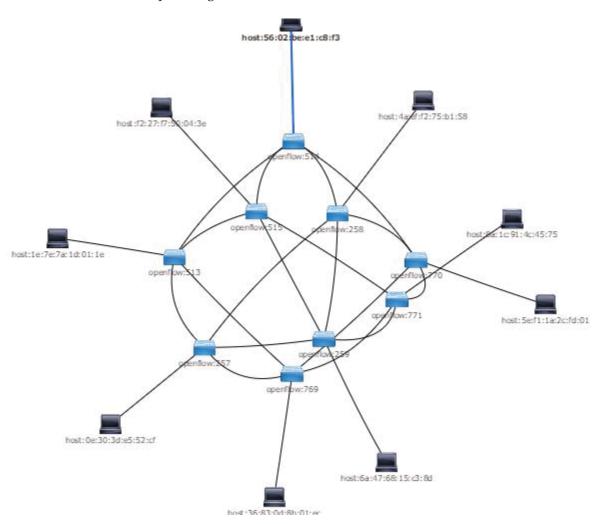


Figura E-6: Topología tipo torus de profundidad 3 por 3

E.6 Experiencia Nº3

Guía Profesor Experiencia N°3

"Laboratorio sobre configuración de elementos en SDN y Protocolo OpenFlow"

Nombre			
:			
Rut:			
	_	Fecha: _	

Objetivos:

- Configurar elementos dentro de las topologías SDN en mininet.
- Verificar, quitar y restablecer conexiones de enlaces entre componentes.
- Visualizar el Protocolo de comunicación OpenFlow empleado.
- Visualizar por medio de ODL las respectivas topologías

Requerimientos:

- 1 PC (RAM: 6GB como mínimo)
- Conexión a Internet
- Sistema Ubuntu-Linux (Versión 16.04 o superior) como partición del equipo
- Mininet y OpenDayLight ya instalados adecuadamente en Ubuntu-Linux

Nota: Es importante tener en cuenta que la sintaxis del código en el sistema operativo Ubuntu-Linux diferencia entre Mayúsculas y minúsculas, también que la consola de terminal requiera introducir la clave de la cuenta de usuario del sistema entre comandos.

Escenario:

Se establece comprender las funcionalidades del entorno Mininet y la configuración autónoma de cada dispositivo dentro de la red virtual, observando sus enlaces de comunicación y protocolos asociados, permitiendo así generar y facilitar el estudio de estas tecnologías.

Desarrollo Experiencia:

Paso Nº1: Comandos disponibles

Se abre un terminal (consola) en Ubuntu, en el cual se iniciará una topología sencilla por medio de mininet con el fin de estudiar las distintas opciones disponibles, para ello digite el siguiente comando:

-		

```
1 Sudo mn --topo single,2
```

Dicha topología corresponde a solo un controlador SDN y dos hosts conectados por medio de un switch, a continuación, con intención de saber las diversas opciones disponibles una vez emulada la red teclee el comando help:

```
2 mininet> help
```

Lo que se mostrará en pantalla serán los diversos comandos disponibles para ejecutar en la topología de red simulada, como se observa en la figura E-7:

```
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> help
Documented commands (type help <topic>):
EOF
       gterm iperfudp nodes
                                      pingpair
                                                             switch
                                                     рц
dpctl
      help
                        noecho
                                      pingpairfull
              link
                                                     quit
                                                             time
              links
                        pingall
                                      ports
       intfs
                                                     sh
dump
                                                             ¥
                         pingallfull
exit
       iperf
              net
                                      рx
                                                     source
                                                             xterm
```

Figura E-7 comandos disponibles entorno mininet

Ahora es de interés comprender el uso de los comandos, para ello basta con anteponer la palabra 'help' en cada uno de ellos, eventualmente

```
mininet> help nodes
mininet> help net
mininet> help dump
mininet> help xterm
mininet> help ping
```

Paso Nº2: Verificación y configuración de elementos

Una vez ya comprendidos los comandos disponibles, procede a la configuración de los respectivos hosts y su estado, en consecuencia, verificará la configuración de cada uno con los comandos:

```
1 mininet> h1 ifconfig -a
2 mininet> h2 ifconfig -a
```

Lo que se mostrará en pantalla corresponderá a la conexión ethernet y loopback del host, es decir su dirección IP, máscara y estado de comunicación. A continuación, se procederá a verificar la conexión de los mismos por medio de sus respectivas consolas:

```
3 mininet> xterm h1 h2
```

El comando anterior desplegara las consolas 'node h1' y 'node h2' respectivamente tal como se muestra en la Figura E-8



Figura E -8: consolas hosts h1 y h2 para topología single de mininet.

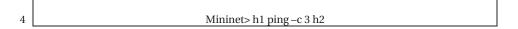
Por tanto, se realizará un ping desde estas consolas y la consola de mininet para ver su estado.

Primeramente, desde la consola 'node h1' se realizará el ping a la dirección IP de la interface h2-eth0 del host 'node h2' representada como XXX.YYY.Z.A. Por consiguiente, el nodo

```
3 ping XXX.YYY.Z.A
```

La solicitud de ping se puede detener al presionar el conjunto de teclas 'CRTL+C' en la respectiva consola.

Mientras que para el caso de la consola de mininet es posible limitar la cantidad de ping y no se requiere el conocimiento de la dirección IP del host con el siguiente comando:



Ya verificadas las conexiones, se prosigue a modificar las direcciones IP y MAC de ambos hosts, de manera remota y al mismo tiempo por medio del controlador SDN. Ahora es necesario aplicar los siguientes comandos en el caso de la consola 'node h1':

```
5 ifconfig h1-eth0 0
6 ip addr add 192.168.1.1/24 brd + dev h1-eth0
7 ifconfig h1-eth0
```

El comando en la línea '5' selecciona la conexión a modificar, el comando en la línea '6', establece la nueva dirección y su respectiva mascara y permite modificar en h1-eth0, mientras que el comando en la línea '7' verifica que el resultado sea el correcto, como se presenta en la 'figura E-9':

Figura E-9: modificaciones a host por medio de consola en Ubuntu.

También se espera realizar dicha acción de forma análoga en la consola 'Node h2':

```
8 ifconfig h2-eth0 0
9 Ip addr add 192.168.1.2/24 brd + dev h2-eth0
```

Prosigue el cambio de dirección Mac para ambos hosts por medio del siguiente comando dentro de la consola 'Node:h1' y 'node:h2':

```
ifconfig h1-eth0 hw ether aa:bb:cc:dd:ee:ff
ifconfig h2-eth0 hw ether gg:hh:ii:jj:kk:ll
```

Paso Nº3: Enlaces de comunicación

En esta sección se evaluará el estado de la conexión del 'host h2' luego de bajar uno de los enlaces de la topología a través del comando 'link'. Para esta tarea en una primera instancia se realizará un ping desde la consola del host h1, además se utilizará Wireshark para observar el protocolo de comunicación OpenFlow:

ping 192.168.1.2 mininet> h2 wireshark &

Mientras el ping esté operativo, podrá ser visualizado, la transferencia a través de Wireshark al momento de desplegarse seleccione la visualización para h1-eth0 o en su defecto h2-eth0.

Mientras desde la consola 'node h1', se dirigirá a la consola de mininet para bajar el enlace del switch al host h2 'node h2' por medio del siguiente comando:

13 Mininet> link h2 s1 down

Una vez realizado el comando observe la consola 'node h1' donde estaba realizando el ping, este dejo de realizar el ping satisfactoriamente debido a que ha eliminado el único enlace disponible, esto se puede ver tanto en la consola como en wireshark.

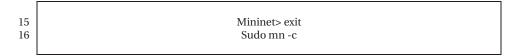
Para regresar el enlace a la normalidad solo debe cambiar el estado down a up:

14 Mininet> link h2 s1 up

Podrá notar que el ping volvió a ejecutarse con normalidad, restableciéndose la conexión.

Paso Nº4: Prueba con ancho de banda

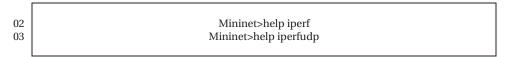
A continuación, se realizará la misma topología, pero esta vez se modificará el ancho de banda de los respectivos enlaces nuevamente con el comando 'link', por tanto, es necesario terminar la simulación actual y limpiar cualquier configuración guardada, para dicha tarea se tiene que:



Se prosigue con la simulación de la red por medio del comando:

01 Sudo mn –topo single,2 --link tc,bw=10,delay=10ms

Luego para sí poder evaluar lo que ocurre con el ancho de banda se realizaran dos test que se encuentran dentro de las opciones de mininet, hablamos de iperf y iperfudp probando el rendimiento del ancho de banda UTP Y UDP respectivamente. Utilice el comando help para comprender su estructura:



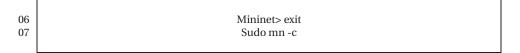
Comience por el comando 'iperf' entre los respectivos host



Dado el resultado podrá confirmar si el uso del protocolo UTP fue exitoso, mientras que para UDP no se garantiza que los paquetes lleguen exitosamente ya que estos pueden adelantarse unos con otros, a continuación, observe lo mencionado con el siguiente comando



Luego de finalizada la experiencia, recuerde finalizar mininet y limpiar posibles configuraciones guardas por medio de los comandos:



Paso Na5: Crear Topologias personalidas por Script Python y Tablas de Flujo OpenFlow

A continuación, para la creación de topologías personalizadas en mininet existe la posibilidad de crearlas por medio de lenguaje de programación Python, lenguaje en el cual está basado Mininet por consiguiente se procederá a generar un código que genere la siguiente topología de la Figura E-10:



Figura E-10: Topología personalizada de dos switch y dos hosts

Si bien dicha topología es creable con el comando:

```
01 Sudo mn –topo: linear,2
```

Este viene con la opción dpctl deshabilitada para visualizar las tablas de flujos que están operando en los switch a través del mismo emulador mininet además de la imposibilidad de agregar tablas de flujos manualmente, por tanto, se procede a generar la topología solicitada habilitando la administración de los flujos para los puertos respectivos.

Es consecuencia se debe utilizar la aplicación Notepadqq (editor de texto y programación), cuya funcionalidad nos permite generar el código en lenguaje de programación Python, realizando su instalación y ejecución de la siguiente manera:

```
O2 Sudo apt-get instal notepadqq
O3 Sudo notepadqq
```

Una vez dentro de Notepadqq se prosigue a la generación del siguiente código, el cual se detalla a continuación en la figura E-11 con los comentarios respectivos para su entendimiento.

```
#!/usr/bin/python
    from mininet.topo import Topo
 3
    from mininet.net import Mininet
 4
    from mininet.cli import CLI
    from mininet.node import OVSSwitch
   from mininet.log import setLogLevel
    net = Mininet()
 8
    print"+++++ Creando nodos en la red"
    c0 =net.addController('c1', port=6633)
    h1 =net.addHost("h1")
1.0
    h2 =net.addHost("h2")
11
    s1 =net.addSwitch("s1",listenPort=6634)
s2 =net.addSwitch("s2",listenPort=6635)
12
13
14
15
    print"+++++ Creando enlaces"
16
    net.addLink(h1,s1)
    net.addLink(h2,s2)
17
18
    net.addLink(s1,s2)
19
20
    print"+++++Configurando direcciones IP"
    h1.setIP("192.168.0.1",24)
21
22
    h2.setIP("192.168.0.2",24)
23
24
    print"+++++Corriendo red"
25
    net.build()
    s1.start ( [ c0 ])
26
27
    s2.start (
                [ c0 ])
28
    net.start()
29
    net.pingAll()
30 v CLI(net)
```

Figura E-11: Código Python para topología personalizada en mininet.

Una vez generado, es guardado en la carpeta personal de la sesión de Ubuntu. Con el nombre "topologia_custom.py", luego en una nueva ventana de terminal de Ubuntu se prosigue a implementar la topología creada en la ubicación correspondiente:

04 Sudo mn –custom=home/pi/topologia_custom.py

Finalmente se procederá a estudiar las tablas de flujo del lenguaje de comunicación OpenFlow, para ello se utilizará el comando 'dpctl', dpctl es una utilidad de línea de comandos que transmite rápidamente mensajes OpenFlow, aplicación útil para la visualización del puerto de switch y de los estados de flujos, además permite también la inserción de entradas de flujo desde el terminal del switch.

Por tanto, se prosigue a abrir la configuración la consola de cada switch

05 Mininet> xterm s1 s2

En las consolas de s1 y s2 se puede visualizar las tablas de flujos con el siguiente comando que se detalla a continuación respectivamente:

04 # dpctl dump-flows tcp:192.168.0.1:6634 05 # dpctl dump-flows tcp:192.168.0.2:6635

Donde la sentencia 'dump-flows' nos permitirá visualizar los flujos de programación OpenFlow ejecutados en la dirección ip y la conexión del puerto TCP con la que dpctl realiza la conexión TCP con el switch.

A continuación, se agregarán flujos OpenFlow en las respectivas consolas de s1 y s2

```
6 # dpctl add-flow tcp: 192.168.0.1:6634 ip,nw_src=198.160.0.1,nw_dst=10.0.0.4,priority=10,idle_timeout=180,hard_timeout=0,actions=output:2
```

\$dpctl add-flow tcp: 192.168.0.1:6634 ip,nw_src=10.0.0.4,nw_dst=10.0.0.3,priority=10,idle_timeout=180,hard_timeout=0,actions=output:1

Donde 'add-flow' es la sentencia que indica que se añadirá un flujo, donde se debe indicar la dirección IP del switch y el puerto con el que el dpctl creara la conexión TCP con el switch, en este caso como todo el tráfico es local se usa la dirección de la *Loopback*, a continuación, se ingresan las direcciones de IP origen y destino mediante los parámetros nw_src y nw_dstrespectivamente. Opcionalmente se agrega la prioridad que se le dará al paquete, en el caso que se desee implementar calidad de servicio (QoS).

Finalmente se define la acción a realizar por el switch para este flujo, en este caso la redirección a un puerto de salida.

Es posible visualizar los respectivos cambios a traves wireshark, por tanto, primeramente, se abrirá la aplicación en un nuevo terminal de Linux, para luego ejecutar un segundo comando en otro terminal de Linux con el cual generaremos una list de flows para visualizar las tramas correspondientes en wireshark, es así como:

```
Sudo wireshark & sudo ovs-ofctl dump-flows s1 -O OpenFlow13
```

Por consiguiente, finalmente se podrá analizar esta demostración de trafico de flujo.

Evaluación:

Teórico:

¿Qué función cumple wireshark?

R= Capturar los paquetes desde la interfaz de red y los almacena en el disco, para después analizarlos, también permite visualizar la transferencia por medio de gráficos.

¿Cuál es la función del comando 'noecho'?

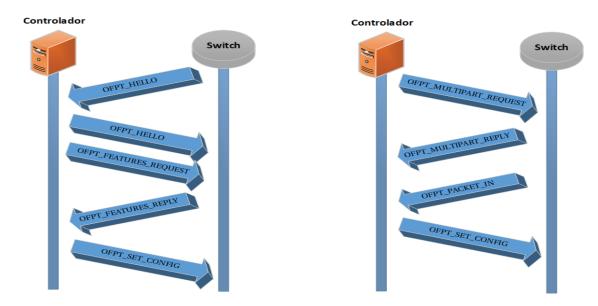
R= Ejecuta acciones y comandos directamente en el dispositivo virtual sin realizar eco en el host anfitrión de Mininet, evita el uso de herramientas como xterm.

¿A que corresponden los Protocolos UTP y UDP?

R= Son protocolos orientados a causas opuestas. Por un lado, UDP es un protocolo no orientado a conexión. Es decir, cuando una maquina A envía paquetes a una maquina B, el flujo es unidireccional. el destinatario recibirá los datos sin enviar una confirmación al emisor (la maquina A). mientras que el protocolo TCP está orientado a conexión. Cuando una máquina A envía datos a una máquina B, la máquina B es informada de la llegada de datos, y confirma su buena recepción.

Considere los diferentes flujos, mostrados en wireshark y detalle cómo funciona el protocolo

R= El funcionamiento del protocolo está definido por el siguiente esquema



Práctico:

Según lo aplicado en la experiencia anterior genere una topología de red por medio de mininet que se pueda visualizar en ODL, teniendo las siguientes características:

- Topología tipo árbol (tree) y que posea una profundidad de 3 y una cantidad de 3 ramas
- Una de las sub ramas debe tener su enlace abajo
- Debe configurar la dirección ip y mascara para cada host

R= Sudo mn -topo o depth=3,fanout=3 --link tc,bw=100,delay=21ms -nat --mac

Mininet: link h1 s2 down