FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

DISEÑO DE UNA RED SDN EN LA DIRECCIÓN PROVINCIAL DE PICHINCHA DEL CONSEJO DE LA JUDICATURA

Autor

Giancarlo Ianni Bermudes

Año
2018

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

DISEÑO DE UNA RED SDN EN LA DIRECCIÓN PROVINCIAL DE PICHINCHA DEL CONSEJO DE LA JUDICATURA

Trabajo de Titulación presentado en conformidad con los requisitos establecidos para optar por el título de ingeniero en Electrónica y Redes de Información

Profesor Guía

Mgs. Iván Patricio Ortiz Garcés

Autor

Giancarlo Ianni Bermudes

Año

2018

DECLARACIÓN DEL PROFESOR GUÍA

"Declaro haber dirigido el trabajo, Diseño de una red SDN en la Dirección Provincial de Pichincha del Consejo de la Judicatura, a través de reuniones periódicas con el estudiante Giancarlo Ianni Bermudes, en el semestre 2018-2, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los trabajos de titulación".

_____

Iván Patricio Ortiz Garcés
Magister en Redes de Comunicaciones
C.I.: 0602356776

DECLARACIÓN DEL PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, Diseño de una red SDN en la Dirección Provincial de Pichincha del Consejo de la Judicatura, del estudiante Giancarlo Ianni Bermudes, en el semestre 2018-2, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

_____

Luis Santiago Criollo Caizaguano
Magister en Redes de Comunicación
C.I.: 1717112955

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

_____

Giancarlo Ianni Bermudes
C.I.: 1715143697

AGRADECIMIENTOS

Agradezco a mis padres que han estado conmigo a lo largo de mi vida apoyándome de todas las maneras posibles. Agradezco a mis hermanos que han sido parte fundamental de mi vida en general y sin los cuales esto no habría sido posible. Agradezco a mis amigos que son para mí otra familia y han estado ahí siempre. Y a la música, que es parte de mí.

DEDICATORIA

Dedico este trabajo a mi familia y amigos que estuvieron conmigo a lo largo de mis estudios en las buenas y en las malas siempre dándome fuerzas para seguir, nunca rendirme y ser un bacán. Esto es para ustedes.

RESUMEN

El siguiente documento explica el concepto de la tecnología SDN (Redes definidas por *Software*), antecedentes para su surgimiento, estructura y varios ámbitos más que describen su funcionalidad. También es la propuesta de un diseño de red con la utilización de tecnologías SDN para la Dirección Provincial de Pichincha del Consejo de la Judicatura, posterior a la realización un análisis de su infraestructura de red y de reconocer los puntos en los cuales la infraestructura actual se vería mejorada con la propuesta SDN. También se detallan las características del diseño de red actual y el diseño propuesto, y se analizan las ventajas y/o desventajas que presente cada red para suplir las necesidades de esta institución.

ABSTRACT

The following document explains the concept of SDN (*Software* Defined Networking) technology, background for its emergence, structure and several more topics that describe its functionality. It is also the proposal of a network design making use of SDN technologies for the Provincial Direction of Pichincha of the Judicature's Council, after carrying out an analysis of its network infrastructure and after recognizing the points in which the current infrastructure would be improved with the SDN proposal. The characteristics of the current network design and the proposed design are also detailed, and the advantages and / or disadvantages that each network presents to meet the needs of this institution are analyzed.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN

Como parte fundamental del desarrollo tecnológico de la presente generación están las redes informáticas las cuales a su vez también han pasado por una gran evolución en busca de optimizar recursos y brindar un servicio cada vez mejor y ajustándose de mejor manera a las distintas limitaciones de la vida cotidiana. Esto con el fin general de simplificar procesos y facilitar de cierta manera su implementación, mantenimiento, etc., dependiendo de las necesidades específicas de cada red.

Las redes definidas por *software* (SDN) por sus siglas en inglés, son una alternativa para crear redes en la cual el control se desliga del *hardware* y se transfiere a una aplicación de *software* llamada controlador. Este enfoque presenta gran innovación en cuanto a la utilización de recursos en dicha red.

Como indica (Fiscó, 2014) cuando un paquete llega a un *switch* en una red convencional, las reglas integradas al firmware propietario del *switch* le dicen al *switch* a dónde transferir el paquete. El *switch* envía cada paquete al mismo destino por la misma ruta y trata a todos los paquetes de la misma manera. En una empresa, los *switches* que normalmente están diseñados con circuitos integrados de aplicación específica (ASICs) son lo suficientemente sofisticados para reconocer diferentes tipos de paquetes y tratarlos de forma diferente. Sin embargo, estos *switches* pueden ser bastante costosos y al ser de distintos propietarios existen problemas de compatibilidad entre equipos.

En una SDN, un administrador de red puede darle forma al tráfico desde una consola de control centralizada sin tener que tocar *switches* individuales. El administrador puede cambiar cualquier regla de los *switches* cuando sea necesario, dando o quitando prioridad, o incluso bloqueando tipos específicos de paquetes con un nivel de control muy detallado.

Las SDN permiten a los ingenieros de redes soportar un tejido de conmutación a través de *hardware* de múltiples vendedores y circuitos integrados de aplicación específica.

En enero del 2017 aproximadamente se inauguró un nuevo complejo para la Dirección Provincial de Pichincha del Consejo de la Judicatura, una gran estructura situada en el norte de Quito. Este gran edificio cuenta con una infraestructura de red bastante amplia que ocupa un gran número de equipos , la cual podría ser representada utilizando redes SDN utilizando *switches* virtuales, habilitados y dedicados que se comunican con controladores que ejecutan módulos que definen el comportamiento de la SDN.

1.1 Alcance

Se realizará un análisis en sitio de la infraestructura de red actual de la Dirección Provincial de Pichincha del Consejo de la Judicatura para. Posteriormente, después de realizado este análisis, se procederá al diseño de una red SDN que simule esta infraestructura para conocer en que secciones de esta red existe alguna ventaja en cuanto a su administración, o en qué circunstancias es más adecuada la infraestructura de red actual, o la propuesta. Se conocerá el funcionamiento de la red actual, así también como su infraestructura y se evaluará en que porciones de red una SDN podría aportar a la optimización de recursos, tomando en cuenta que partes podrían ser obviadas y qué partes necesitarían ser aumentadas para este propósito.

1.2 Justificación

La Dirección Provincial de Pichincha del Consejo de la Judicatura ha realizado movimientos de edificios que ha concentrado a más funcionarios y más equipos en menos localidades, debido a eso en este complejo se cuenta con una gran cantidad de *switches* configurables los cuales podrían disminuir en cantidad y configurarse su función de manera centralizada, dando practicidad al

mantenimiento y versatilidad de la red, de manera que el diseño de una red SDN podría facilitar de gran manera el mantenimiento y funcionamiento de la misma.

Se realizará el diseño de una nueva propuesta de la red para lograr de cierta manera tener una red mejor estructurada y que aproveche las características y facilidades que la tecnología SDN propone, para luego poder seguir implementando sobre esa misma red servicios que sean necesarios en el Complejo judicial diseñando una red escalable y que supla la gran cantidad de usuarios administrativos y dispositivos que esta institución requiere.

Con las utilidades que ofrece una red SDN se tendrá como beneficio un control más minucioso del tráfico de la red en general y un procesamiento de este tráfico de acuerdo a las necesidades del complejo para manejar la información de una manera más organizada.

El impacto que este diseño puede tener, después de su implementación es el control de la información y la automatización de procesos que a largo plazo lograrían que ciertos procesos lleguen a realizarse solos o de una manera mucho más simple con un sistema centralizado que controle todos los dispositivos desde un mismo lugar, logrando así ahorrar recursos y minimizar tiempos.

1.3 Objetivos

1.3.1 Objetivo General

Diseñar una propuesta de red para la Dirección Provincial de Pichincha del Consejo de la Judicatura utilizando SDN.

1.3.2 Objetivos Específicos

- Analizar la infraestructura de red actual.

- Analizar cuáles son las ventajas y limitaciones de utilizar redes SDN en lugar de otro tipo de redes.
- Proponer un diseño de una red SDN que supla todas las necesidades de la Dirección Provincial de Pichincha del Complejo de la Judicatura enfocándose en optimizar servicios y simplificar procesos.

2. MARCO TEÓRICO

2.1 Generalidades

Para un mejor entendimiento del funcionamiento de la tecnología SDN y para una mejor comprensión de la infraestructura de red actual del complejo judicial, es necesario conocer ciertos componentes una infraestructura de red. Estas definiciones serán enfocadas al cableado, dispositivos utilizados, tipos de dispositivos y su clasificación, etc. describiéndose los más importantes para este trabajo. Además se describirán las características y propiedades de los dispositivos que actualmente forman parte de la infraestructura de red del complejo judicial.

2.1.1 Cableado estructurado

El cableado estructurado es la distribución de cable en uno o varios edificios que permite la interconexión entre equipos activos, de diferente o igual tecnología permitiendo así la integración de varios servicios que dependen de la utilización de tendido de cable, como los datos, telefonía, control, entre otros.

El cableado estructurado es además, una metodología basada en estándares que permiten diseñar e instalar un sistema de cableado que integre la transmisión de voz, datos y vídeo. Un sistema de cableado estructurad (SCE) por sus siglas en inglés proporciona una infraestructura de cableado que suministra un desempeño predefinido y la flexibilidad de acomodar futuros crecimientos por un período extendido de tiempo. Adaptado de (Cableado Estructurado, s.f)

Ésta metodología tiene como objetivo integrar los sistemas de automatización con los sistemas del lugar en el cual se implemente la red. También dar una infraestructura flexible de cables que puede aceptar y soportar múltiples sistemas de computación y de comunicación, buscando además optimizar la cantidad de cable utilizado. Adaptado de (Universidad de Buenos Aires, s.f)

2.1.1.1 Cableado Horizontal

El cableado horizontal es una disposición del cableado que se extiende desde la salida del punto de red del usuario final hasta el cuarto de telecomunicaciones, *Data center*, etc.
En un *Data center* el cableado horizontal es el cableado que se extiende desde el punto de área de distribución principal o MDA hasta la salida en el área de distribución de equipo activo. Adaptado de (Axioma, s.f).

El cableado horizontal consiste de dos elementos básicos:

2.1.1.2 Cable Horizontal y *Hardware* de Conexión.

El cableado horizontal es un tipo de cableado que sirve para proporcionar el medio de transmisión de datos entre el área de trabajo y el cuarto de telecomunicaciones, como se muestra en la figura 1.

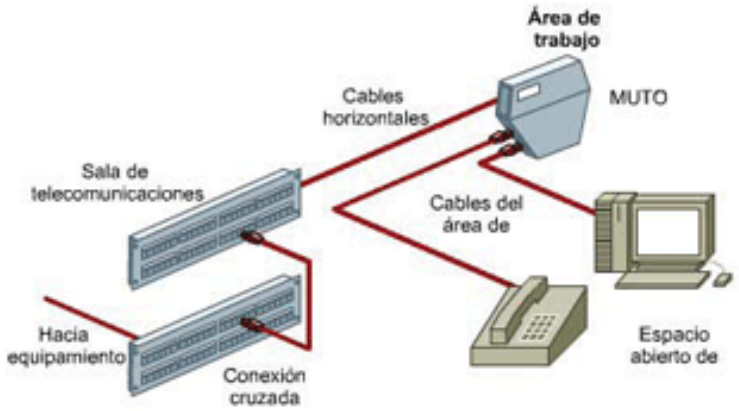


Figura 1. Cableado horizontal

2.1.1.3 Cableado backbone

El cableado *backbone* se refiere a la distribución vertical entre pisos de una estructura en la cual se maneje cableado estructurado. Su propósito es interconectar cuartos de entrada de servicios, cuartos de equipos y cuartos de telecomunicaciones como se puede apreciar en la figura 2.

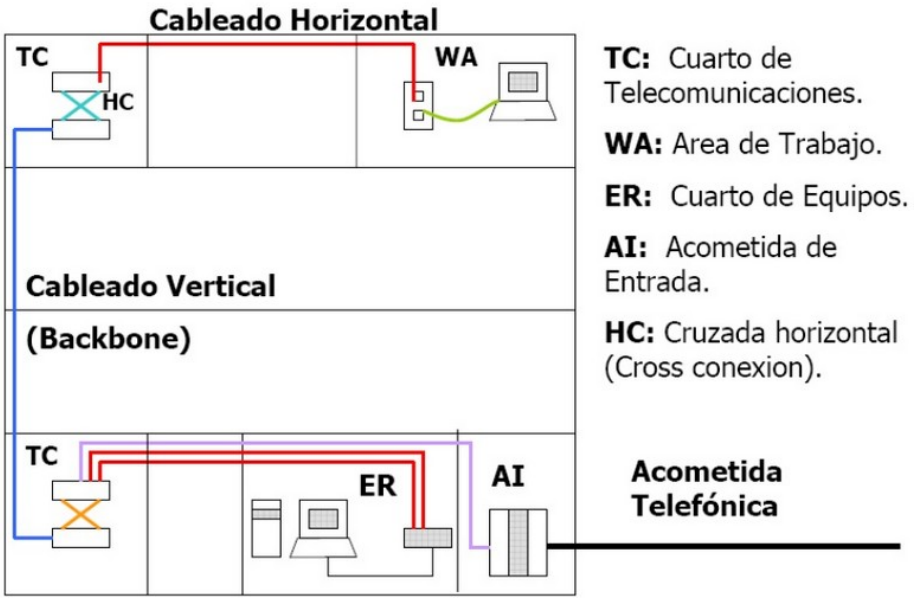


Figura 2. Cableado Estructurado (Backbone).
Tomado de (Jiménez, 2017)

Para este tipo de cableado se pueden utilizar cableado Multipar UTP y STP, y también, Fibra óptica Multimodo y Monomodo.

Como explica (M., 2017), existen 2 tipos de backbone: cascada y colapsado. En cascada, todos los *hosts* o terminales están conectados a un enlace troncal con el cuarto de equipos; esta arquitectura es casi obsoleta y genera mucho tráfico innecesario en la red. El colapsado funciona, en cambio, con tramos que salen del cuarto de equipos distribuyendo los servicios manera tal que existe una buena conexión hasta en los puntos más alejados del cuarto de equipos.

2.1.2 Power over Ethernet (PoE)

Power over Ethernet (PoE) por sus siglas en inglés se refiere a la especificación IEEE 802.3af, más comúnmente referida como *Power-over-Ethernet*. Básicamente es un protocolo analógico de transmisión de potencia, y no de transmisión de datos, que permite alimentar remotamente a dispositivos de red a través del mismo cable por donde se produce la transmisión de datos, sin distorsionarla. Está enmarcado dentro de la familia de protocolos IEEE 802.3 (cláusula 33 del IEEE 802.3-2005) que se dedica al estudio del nivel físico y de enlace de la tan extendida tecnología Ethernet. Fue aprobado en Junio del 2003 y revisado en el año 2005. Adaptado de (Wheelers Lane Technology College, s.f)

Un sistema PoE está compuesto principalmente por dos tipos de dispositivos:

2.1.2.1 Powered devices (PD)

Los *Power devices* o PD por sus siglas en inglés son los dispositivos de red que son alimentados por el *Power Sourcing Equipment* (PSE por sus siglas en inglés). El estándar PoE especifica una potencia máxima recibida de 12.95W por cada PD, incluyendo las pérdidas. Existen a su vez dos clases de PD:

2.1.2.1.1 PDs no compatibles con PoE

No están definidos por la norma PoE sin embargo esta hace referencia a ellos. Al no ser compatibles con el estándar necesitan un *tap*, *splitter* o derivador que separe la corriente continua (DC por sus siglas en inglés) de la transmisión de datos, y se la inyecte al dispositivo a través del tradicional conector jack de potencia. Adaptado de (Wheelers Lane Technology College, s.f)

2.1.2.1.2 PDs compatibles con PoE

Estos dispositivos son quienes reciben la potencia eléctrica directamente del conector RJ45 sin necesidad de un *splitter*. Que dichos dispositivos sean compatibles con PoE no significa que lo sean con la norma. Habrá dispositivos compatibles con PoE 802.3af, y habrá dispositivos compatibles con PoE de otra solución no estandarizada e incluso ambas. Adaptado de (Wheelers Lane Technology College, s.f)

2.1.2.2 PSE (Power Sourcing Equipment)

Es el equipo principal encargado de suministrar potencia eléctrica al resto de dispositivos de red o PDs. El PSE inyecta corriente continua en 2 de los 4 pares del RJ45. En concreto, el estándar establece un voltaje de 48V DC, con una intensidad máxima de 400mA, para una carga máxima de potencia de 15.4W por cada puerto. Es el cerebro del sistema, ya que es el que detecta, clasifica y controla la potencia eléctrica suministrada. El estándar diferencia a su vez dos tipos de PSEs:

2.1.2.2.1 Endpoint

Es un PSE que combina la función de generar corriente eléctrica para alimentar a los PD, con la función de *Data Terminal End* (DTE por sus siglas en inglés), *switch* o *hub*. Generalizando, se puede decir que se trata de un *switch* que incluye en su interior la circuitería necesaria para poder implementar PoE. En el mercado se encuentra con la denominación de PoE *switch* como ilustra la figura 3. (Wheelers Lane Technology College, s.f)

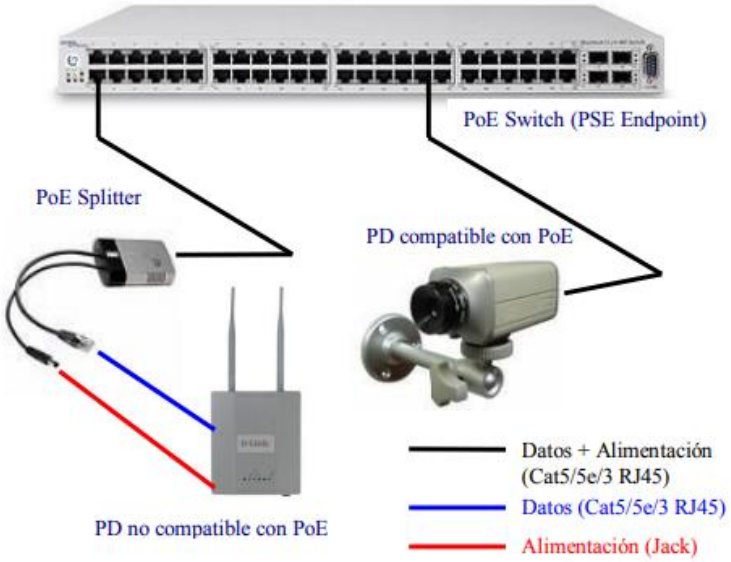


Figura 3. Ejemplo de un sistema PoE com un PSE Enpoint.
Tomado de (Wheelers Lane Technology College, s.f)

2.1.2.2.2 Midspan

El midspan es un PSE que sólo tiene la función de alimentación. Es decir, un inyector de potencia que se sitúa entre el *switch* Ethernet tradicional y el PD como se aprecia en la figura 4.

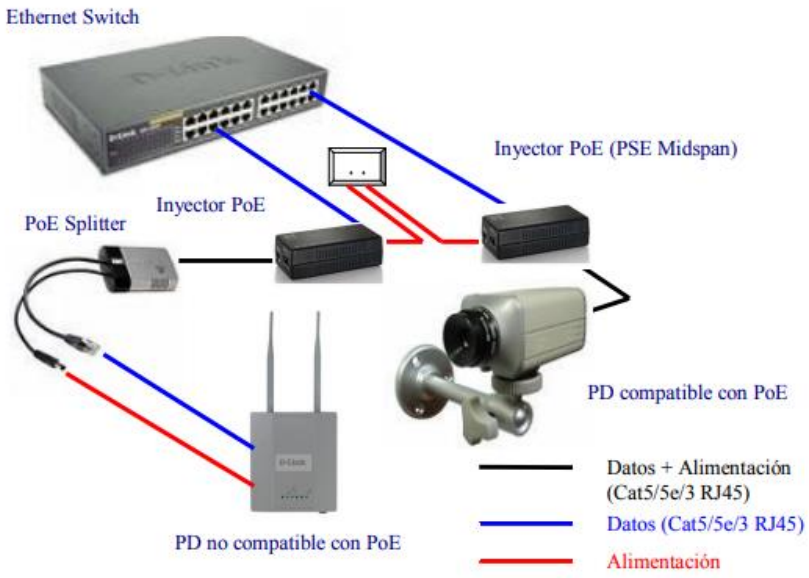


Figura 4. Ejemplo de sistemas PoE con PSE Midspan.
Tomado de (Wheelers Lane Technology College, s.f)

Este estándar también defina la interfaz PSE – PD: *Power Interface* (PI por sus siglas en inglés) que físicamente es un cable de red UTP RJ45 de ocho pines de categoría 5, 4 o 3 y una longitud menor a los 100 metros. En la figura 5 se pueden apreciar dichos pines. Adaptado de (Wheelers Lane Technology College, s.f)

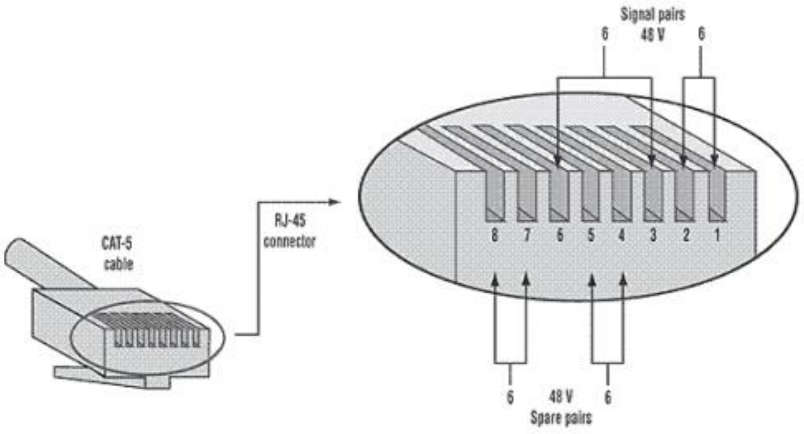


Figura 5. Pines de cable RJ45 Categoría 5.
Tomado de (Wheelers Lane Technology College, s.f)

2.1.3 *Switch*

El *switch* es un dispositivo de interconexión utilizado para conectar equipos en red formando una red de área local (LAN) y cuyas especificaciones técnicas siguen el estándar Ethernet (o técnicamente IEEE 802.3).

Estos componentes permiten la comunicación utilizando técnicas de conmutación por *hardware* y consiguen velocidades que llegan a los 10Gbps. Debido la flexibilidad que ofrecía Ethernet antes de que surja la conmutación con *switches*, se consiguió una enorme flexibilidad para establecer las configuraciones y topologías de las redes basadas en Ethernet. Esta es la principal razón de que existan tantos tipos y variedades de *switches*, cubrir las necesidades de todos los posibles tipos de redes que puedan existir. Como explica (Redes Telemáticas , 2012).

Generalmente, los *switches* pueden ser clasificados en:

2.1.3.1 *Switches* de Core o Troncales

Son los *switches* que se ubican en el centro o *core* de grandes redes. A ellos están conectados otros *switches* de jerarquía inferior, además de *routers*, servidores, etc.

2.1.3.2 *Switches* de distribución

Los *switches* de distribución son los *switches* que proveen una ruta de acceso para que los *switches* de acceso puedan enviarse datos entre sí. Normalmente se conecta un *switch* de acceso a un *switch* de distribución pero para conseguir redundancia se suele levantar dos *uplinks* por cada *switch* de acceso a dos *switches* de distribución.

2.1.3.3 *Switches* de acceso

A estos *switches* están conectados dispositivos de usuario final directamente y no conmutan con otros como se logra apreciar en la figura 6.

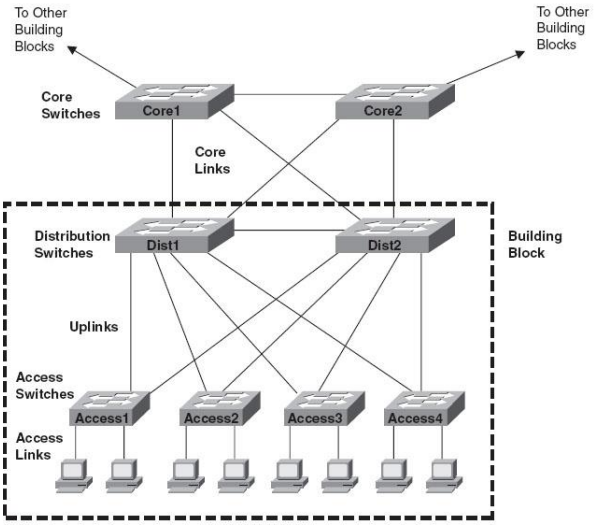


Figura 6. Tipos de *Switches*.

Tomado de (Switches de acceso, 2008)

2.1.3.4 Apilamiento de *switches*

El apilamiento de *switches*, o "*switch stacking*" consiste en agrupar diferentes *switches* haciendo que la red los perciba como uno solo. Esta práctica no se encuentra implementada en la red del complejo judicial actualmente, lo que dificulta de gran manera la gestión de la red al tener que administrar cada dispositivo individualmente. Para que el apilamiento de *switches* sea posible los dispositivos deben contar con unos puertos específicos (de cobre o fibra óptica) para ser enlazados unos con otros.

Figura 7. Switches Apliados.
Tomada de (Switch Stacking, 2013)

2.1.3.5 *Switch* Cisco 2960

El *switch* Cisco 2960 es uno de varios *switches* de la serie Catalyst de Cisco utilizados como *switches* de acceso en la red actual del Complejo Judicial. Estos dispositivos son *switches Gigabit Ethernet* apilables de configuración fija que ofrecen conectividad de red para grandes y medianas empresas, y sucursales. Los *switches* Cisco 2960 permiten realizar operaciones empresariales de manera confiable y segura. Estos dispositivos cuentan con varias características, tales como Cisco FlexStack-Plus, visibilidad y control de aplicaciones, *Power over Ethernet Plus* (PoE+ por sus siglas en inglés), revolucionarias funciones de administración de energía y *Smart Operations*. En la figura 8 se muestra un *switch* Cisco 2960 físicamente. Tomado de (Cisco, s.f)

Figura 8. Switches Cisco Catalyst 2960

Esta variedad de *switches* ofrece *switching* de capa 2 y cuentan con alimentación fija con una alimentación externa redundante. Cuentan también con 24 o 48 puertos Gigabit Ethernet compatibles con PoE y cuatro enlaces de subida *Small Form-Factor Pluggable* (SFP por sus siglas en inglés) de 1G o dos enlaces de subida SFP+ de 10G. Tomado de (Cisco, s.f)

2.1.3.6 *Switch* Cisco 4500

El Cisco 4500 es un *switch* de alta disponibilidad, utilizado como *switch* de core en la red del complejo judicial. Este dispositivo es parte de la serie Catalyst de Cisco que permite redes sin borde, brindando un alto rendimiento, movilidad, etc. Este *switch* tiene una arquitectura de reenvió centralizada que permite la colaboración, virtualización y capacidad de administración operativa a través de operaciones simplificadas. Adaptado de (Cisco, 2017)

Figura 9. Switches Catalyst 4500

El *switch* 4500 extiende el control al extremo de la red con servicios de red inteligentes, que incluyen una sofisticada calidad de servicio (QoS), rendimiento

predecible, seguridad avanzada, administración integral y resiliencia integrada. La escalabilidad de estos servicios de red inteligentes se hace posible con recursos dedicados y especializados conocidos como memoria ternaria de contenido direccionable (TCAM). Los amplios recursos de TCAM (hasta 384,000 entradas) habilitan la "capacidad de altas prestaciones", que proporciona un rendimiento de enrutamiento y conmutación a velocidad de cable independiente del aprovisionamiento de servicios como QoS y seguridad. Adaptado de (Cisco, 2017)

2.1.4 EtherChannel y Port Channel

EtherChannel es una tecnología desarrollada por Cisco de acuerdo a los estándares 802.3 full dúplex *Fast Ethernet*, la cual permite agrupar varios *switches* físicos en un solo *switch* lógico. Al hacer esto la red percibe a varios *switches* como uno solo y permite aumentar la velocidad nominal de cada puerto Ethernet utilizado para de esta manera obtener un enlace troncal de mayor velocidad. Adapado de (Cisco, 2017).

Según (Cisco, s.f) un *Port Channel* es una agregación de múltiples interfaces físicas que crea una interfaz lógica. Se puede agrupar hasta ocho enlaces activos individuales en un *Port Channel* para proporcionar un mayor ancho de banda y redundancia. Usar *Port Channels* también balancea la carga del tráfico a lo largo de estas interfaces físicas. El *Port Channel* permanece operacional mientras al menos una interfaz dentro del *Port Channel* permanezca operacional.

2.2 Introducción a SDN

En el planteamiento tradicional del *networking* la mayoría de las funciones de la red son implementadas en un dispositivo específico como un *switch*, router, etc. Normalmente estos equipos cuentan con dos capas: la capa de datos que es la capa en la cual datos reales son únicamente conmutados mediante circuitos

integrados ASIC y una capa de control que es la que se encarga de determinar cómo ésta información es enviada, recibida, conmutada, etc. utilizando la gran variedad de protocolos de red que existe actualmente.

Tradicionalmente, un router trabaja de la siguiente manera (Ver figura 10): un paquete entra al flujo y dentro el dispositivo decide cómo tratarlo (el control yace en el router):

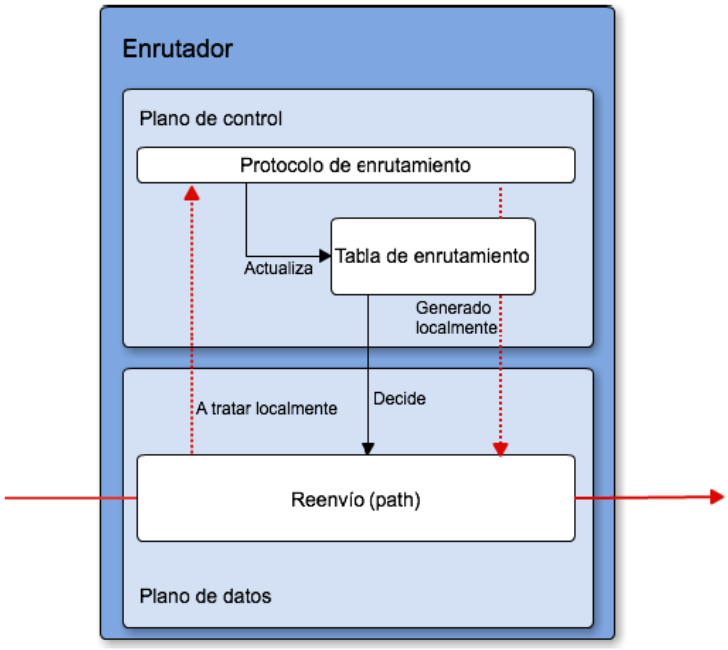


Figura 10. Procesamiento de un paquete entrante en un router.
Tomado de (Serrano, 2015)

En la figura 10 se puede apreciar la división del plano de control con el plano de datos dentro del dispositivo en el cual el plano de control decide cómo tratar a los paquetes recibidos y después instruye al plano de datos a proceder con el paquete según su tabla de enrutamiento. Es de esta manera que funcionan tradicionalmente las redes, con equipos inteligentes, configurados para decidir sobre la información que reciben, que sin embargo son costosos y dependientes de su fabricante en cuanto a su desarrollo. Adaptado de (Serrano, 2015).

Este enfoque tiene varias características que a su vez son desventajas:

- Los circuitos integrados ASICS son relativamente costosos y su evolución es lenta.
- La complejidad de las redes ha crecido mucho, son más grandes y requieren un despliegue de servicios más dinámico ya que las redes tradicionales no permiten programabilidad, flexibilidad ni portabilidad.
- La evolución de estos equipos depende directamente de su propietario.
- Cada dispositivo debe ser configurado y administrado individualmente.

Estas desventajas permiten concluir que la red tradicional evoluciona lentamente. Además está limitada en funcionalidades las cuales van evolucionando dependiendo de los proveedores de ASIC y los proveedores de los dispositivos de red; tiene relativamente un alto nivel de OPEX (gastos de operación) y es relativamente estática en su naturaleza.

Debido a esto y también a que con el paso del tiempo y a la evolución de las tecnologías surge la necesidad de redes más administrables, programables y que requieren implementar más y más servicios surge el concepto SDN que promete superar estas limitaciones.

2.2.1 *Software* Defined Networks (SDN)

Las redes SDN (*Software* Defined Networks), como sus siglas indican, son redes definidas por *software*, esto más específicamente significa que su funcionamiento no depende fundamentalmente de los equipos físicos como routers, *switches*, etc. sino más bien su funcionamiento se basa en políticas automáticas y en desligar el plano de datos del plano de control, centralizando el control de la red en un dispositivo llamado controlador SDN que define las políticas y gestiona a toda la red.

Según (Cisco, s.f)*: "Las redes SDN son redes programables y permiten a las organizaciones acelerar la implementación y la distribución de aplicaciones*

reduciendo drásticamente los costos de TI mediante la automatización del flujo de trabajo basada en políticas”.

La idea detrás del concepto SDN es separar el plano de datos del plano de control, es decir, que las decisiones tomadas para gestionar el flujo de datos en la red y el manejo de los paquetes se generan en un dispositivo llamado controlador, y los *switches* se convierten en dispositivos que siguen las instrucciones del controlador y dejan de tomar decisiones o procesar la información individualmente. De esta manera el controlador puede configurar equipos como routers, *switches*, traductores de direcciones de red NAT, cortafuegos, etc. Adaptado de (Espinosa, 2016). Las ventajas de esta tecnología se describen más a detalle en capítulos posteriores de este trabajo.

La manera estándar de trabajar con SDN es con un solo controlador que administre la actuación de los *switches* en una red. Así se percibe que toda la red es una sola unidad, ya que el controlador precisa la conducta de la red como un todo. Esta particularidad permite mejoras en la seguridad y monitorización. (Albán Ruiz & Brito López, 2015)

De esta manera la red tiene un control centralizado ya que el controlador permite visualizar y tratar a la red como un todo, simplificando de gran manera su administración.

2.2.1.1 Estructura

La estructura general de las redes SDN consta de 3 capas: La capa de aplicación, la capa de control y la capa de infraestructura como se muestra en la figura 11:

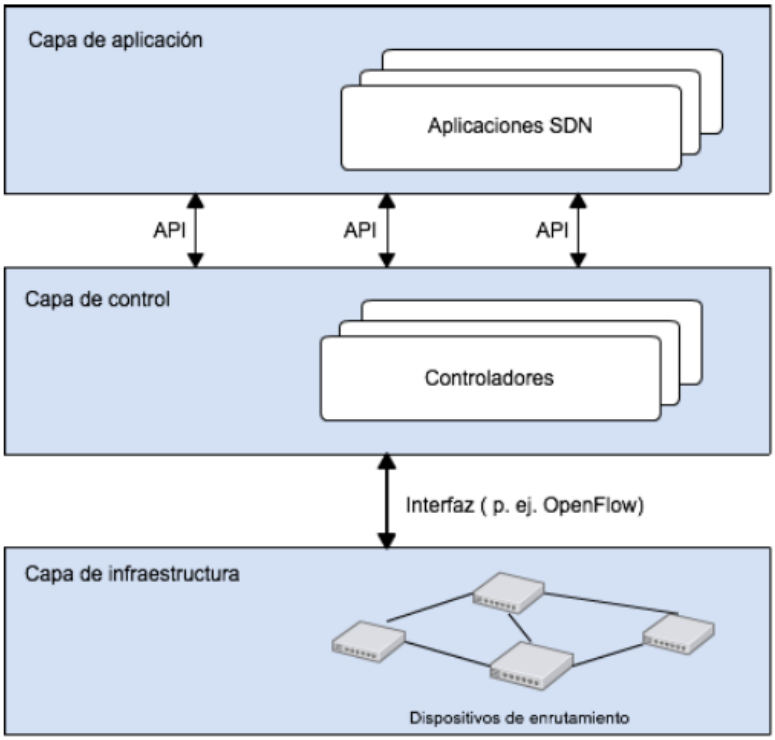


Figura 11. Arquitectura SDN.
Tomado de (Serrano, 2015)

2.2.1.1.1 Capa de aplicación

Es la capa superior de la arquitectura en la cual se encuentran aplicaciones SDN conectadas a la capa de control a través de APIs (*Northbound Interface*) y que permiten satisfacer las necesidades del usuario con ejemplos como: enrutamiento adaptivo, itinerancia sin interrupciones, seguridad de la red, mantenimiento de la red, etc.

En esta capa se puede contrastar el cambio de la filosofía respecto a las redes tradicionales. Estas APIs incorporan los patrones de uso de la red de cada aplicación, y su función es comunicar esos patrones a la capa de control, donde se toman las decisiones y se cierra el ciclo: las aplicaciones definen el uso que se va a dar a la red, lo comunican al controlador SDN, el cuál toma las decisiones oportunas y las comunica a la infraestructura de red (capa de

infraestructura) mediante la *Southbound Interface* que es en donde se involucra directamente Open Flow.

2.2.1.1.2 Capa de Control

La capa de control es la más importante de esta arquitectura ya que es el centro de la misma. Esta se conecta con las capas de aplicación e infraestructura a través de las interfaces *Northbound* y *Southbound* respectivamente. En ella se encuentra el controlador SDN el cual administra toda la red, gestionando la capa de aplicación y la capa de infraestructura y además monitorizando todos los elementos de la red haciéndola ver como un todo.

El controlador SDN es una entidad *software* que tiene control exclusivo sobre un conjunto abstracto de recursos de plano de control, es decir, es la entidad que controla y configura los nodos de la red para dirigir adecuadamente los flujos de tráfico. El controlador SDN elimina la inteligencia de conmutación y encaminamiento de datos de los nodos que realizan dicha función, delegando el control al controlador SDN, que toma esas decisiones y selecciona el mejor camino para el tráfico basándose en lo que ha aprendido de la red y lo que ha aprendido de las APIs. La arquitectura describe una serie de funciones internas al controlador SDN y al elemento de red, pero sólo se especifica el comportamiento de aquellos aspectos que son necesarios para asegurar la interoperabilidad. La arquitectura permite que un controlador SDN gestione un amplio rango de recursos de plano de datos, lo cual ofrece el potencial de unificar y simplificar su configuración. Adaptado de (Tejedor, 2014)

2.2.1.1.3 Capa de Infraestructura

Por último, está la capa de Infraestructura la cual consta de los dispositivos *hardware* de conmutación que forman la red y que realizan dos tareas de acuerdo a sus componentes lógicos:

- Control: recoger el estado de toda la red y mantener al controlador informado de la misma, el cual a su vez le dará instrucciones de reenvió de paquetes.
- Datos: el procesador de red reenvía los paquetes en base a las decisiones del plano de control. Cuando se recibe un paquete, el dispositivo identifica la decisión de reenvío que coincide con el paquete a través de tablas de flujo.

La historia de las redes SDN se remonta a los inicios del internet, hace unos 20 años aproximadamente, Estas son como tal el producto de un largo periodo de desarrollo en el cual gradualmente se separó el plano de datos (*Hardware*) del plano de control (*Software*).

La necesidad de separar el plano de control del plano de datos se generó con el despliegue del internet en la época de los 90, por ello la utilización de redes aumento y fue necesario que antiguas tecnologías fueran quedando atrás y nuevos protocolos de red que se adapten a las necesidades de aquel entonces fueran creados. Con esto, investigadores fueron desarrollando nuevas tecnologías gradualmente, y con el pasar del tiempo y debido a la evolución de las redes y su necesidad de tener mayor flexibilidad, programabilidad y la capacidad de probar nuevas ideas sin interrumpir los servicios que se estén ejecutando. En el 2011, Deutsche Telekom, Facebook, Google, Microsoft, Verizon y Yahoo! crearon la Open Networking Foundation (ONF). Esta es una organización sin fines de lucro que se dedica a la promoción y adopción del paradigma SDN a través de estándares abiertos, ya que en este tipo de investigaciones una estandarización oficial de la IETF eran procesos que podían durar varios años frustrando así el desarrollo.

La ONF además propone OpenFlow como el despliegue comercial de este tipo de tecnología. Un estándar abierto que permite realizar pruebas y empezar a desarrollar y explotar las prestaciones de SDN. Adaptado de (Serrano, 2015).

2.2.1.2 OpenFlow

OpenFlow es el protocolo de comunicaciones utilizado en redes SDN. Es uno de los primeros estándares con los que trabajan las SDN. Originalmente definía el protocolo de comunicación en ambientes SDN que habilitaba a un controlador SDN para interactuar directamente con el plano de datos de dispositivos de red como *switches* y routers, física y virtualmente (basado en un Hipervisor) para que se puedan adaptar fácilmente a los requerimientos de un negocio. Adaptado de (SDX Central, s.f)

De acuerdo con (Spera, 2013): *"La falta de una interfaz abierta para la capa de forwarding hace que los dispositivos de Networking sean monolíticos, al estilo de un mainframe. Un protocolo como OpenFlow es necesario para poder mover el control de la red fuera de los dispositivos de red para centralizarlo en un software. Este protocolo es implementado en ambos lados de la interfaz, entre la infraestructura de red y el software de control de SDN."*

La idea general de OpenFlow es simple, se aprovecha el hecho de que la mayor parte de los *switches* y routers Ethernet contienen tablas de flujo (típicamente construidas de memorias TCAM y RAM que realizan gestión interna) que corren en línea para implementar firewalls, NATs, QoS (Quality of Service) y para recolectar estadísticas. Mientras que la tabla de cada vendedor sea diferente, existen un conjunto interesante de funciones que corren en muchos *switches* y routers, el objetivo principal de OpenFlow es explotar estas funcionalidades comunes.

Cada dispositivo OpenFlow consta de al menos estos 3 elementos:

- Tabla de Flujos: Cada tabla de flujo en el *switch* contiene un conjunto de entradas, cada una de estas representan campos coincidentes (match fields), contadores (counters) o una serie de instrucciones que aplicar a los paquetes coincidentes. Cada tabla de flujo tiene los siguientes componentes:

    - Match Fields (Campos coincidentes): consisten en el puerto de entrada y cabeceras y también, ocasionalmente metadata especificada por una tabla previa.
    - Counters (Contadores): permiten actualizar los paquetes coincidentes.
    - Instructions (Instrucciones): Modifican el conjunto de acciones o el procesamiento canalizado.
- Canal Seguro/Canal OpenFlow: Es un canal exclusivo para la comunicación entre el dispositivo OpenFlow y el controlador que permite, por default, que el controlador aprenda acerca de la red desde su estado inicial, aprendiendo como es la topología, que dispositivos contiene, etc.

  Entre el camino de datos y el canal OpenFlow, la interfaz se implementa de forma específica, sin embargo todos los mensajes del canal OpenFlow deben poseer un formato acorde al protocolo OpenFlow. El canal OpenFlow es usualmente encriptado utilizando TLS (Transport Layer Security), pero este canal puede correr directamente en TCP (Transmission Control Protocol). (Serrano, 2015)
- Protocolo OpenFlow: Permite una manera abierta y estándar de comunicación entre el *switch* y el controlador, permitiendo al controlador insertar, eliminar, modificar y buscar en las entradas de la tabla de flujos a través del canal seguro.

OpenFlow, al ser relativamente joven tiene varias versiones con especificaciones y prestaciones distintas, a medida que se iba desarrollando esta tecnología. La primera implementación de referencia fue lanzada en el año 2007 por un grupo colaborativo de investigadores universitarios y administradores de red, llamado el OpenFlow Consortium con la versión 0.1.0. Se continuó con versiones aún experimentales: 0.2.0, 0.8.0, 0.8.1, 0.8.2, 0.8.9, publicadas en 2008 y 0.9 a mediados de 2009. Adaptado de (Serrano, 2015).
En diciembre del 2009 se lanzó la versión 1.0.0, la más ampliamente adoptada, que para cada paquete Ethernet evalúa los siguientes campos de su cabecera como se muestra en la tabla 1:

Tabla 1.
Campos de cabecera evaluados en versión 1.0.0

Puerto entrante
Dirección MAC Origen
Dirección MAC Destino
Tipo Ethernet
ID VLAN
Prioridad QoS VLAN
IP Origen
IP Destino
Protocolo IP
Bits ToS (tipo de servicio) IP
Puerto TCP/UDP Origen
Puerto TCP/UDP Destino

Las operaciones realizadas cuando entra un paquete a un *switch* son las siguientes:

- Los campos de cabecera son extraídos de los paquetes y utilizados para encontrar coincidencias.
- Se compara con las reglas definidas para cada entrada en la tabla de flujos OpenFlow, teniendo en cuenta el orden descendente de prioridad. Un paquete puede ser coincidente con una entrada particular de la tabla de flujos usando uno o más campos del paquete. Un campo en la tabla de flujos puede tener el valor “ANY” o cualquiera, que coincidirá con todos los paquetes.
- Si hay una coincidencia, las acciones especificadas en esa entrada se realizan sobre el paquete. Si no, los primeros 200 bytes del paquete son enviados al controlador para su procesado.

En cuanto a simulaciones, existen varias herramientas que son compatibles con el estándar OpenFlow: Kiva, Packet Tracer (Cisco), NS2, OMNet++, etc.

Entre estas varias herramientas de *software*, se situa Mininet. Adaptado de (Serrano, 2015)

2.2.1.3 Mininet

De acuerdo con (Kaur, Singh, & Singh Ghumman, 2014), Mininet es un emulador para el despliegue de redes sobre limitados recursos de una computadora simple o una máquina virtual. Esta herramienta trabaja utilizando el kernel de Linux y otros recursos para simular todos los elementos de una SDN como el controlador SDN, los *hosts* y los *switches* OpenFlow.

Las principales ventajas de Mininet son su rapidez para levantar toda una red, su capacidad para crear topologías personalizadas y la posibilidad de interactuar con otros programas.

Mininet permite crear topologías de gran escala de hasta miles de nodos y realizar pruebas a través de un gestor de comandos y su API como ilustra la figura 12.

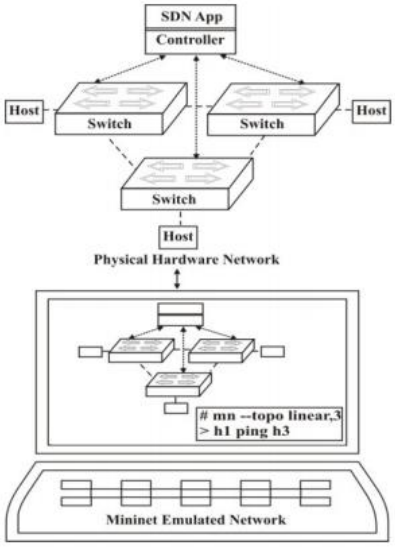


Figura 12. Emulando redes SDN reales en Mininet.
Tomado de (Kaur, Singh, & Singh Ghumman, 2014)

Utiliza *hosts* virtuales, *switches* y enlaces para crear una red en un solo núcleo del sistema operativo, y utiliza la pila de red real para procesar paquetes y conectarse a las redes reales. Además, las aplicaciones de red basadas en Unix/Linux, también se pueden ejecutar en los *hosts* virtuales. En una red OpenFlow emulada por Mininet, una aplicación de controlador real OpenFlow se puede ejecutar en una máquina externa o en el mismo equipo en el que se emulan los *hosts* virtuales. A continuación una tabla comparativa entre Mininet y dos simuladores de red (Estinet y NS-3):

Tabla 2.
Comparación entre Estinet, NS-3 y Mininet.

	Estinet	NS-3	Mininet
Modo de Simulación	Sí	Sí	Sí
Modo de Emulación	Sí	No	Sí
Compatible con controladores reales	Sí	No	Sí
Escalabilidad	Alta (para un solo proceso)	Alta (para un solo proceso)	Media (para múltiples procesos)
Exactitud en los resultados de rendimiento	Sí	No admite protocolo de árbol y ningún controlador real	Sí
Soporte de GUI	Sí	Sólo para observación	Sólo para observación

Tomado de: (Serrano, 2015)

Las topologías disponibles en Mininet son: minimal, single, linear, tree y personalizadas. En cualquiera de éstas existe un controlador, varían en el número de *hosts*, *switches* y los enlaces entre éstos. Minimal, por ejemplo, está compuesta por dos *hosts* conectados a un *switch*; single también es una topología con un único *switch* pero la cantidad de *host* puede indicarse por parámetro, como describe (Valencia, Santacruz, Becerra, & Padilla, 2015).

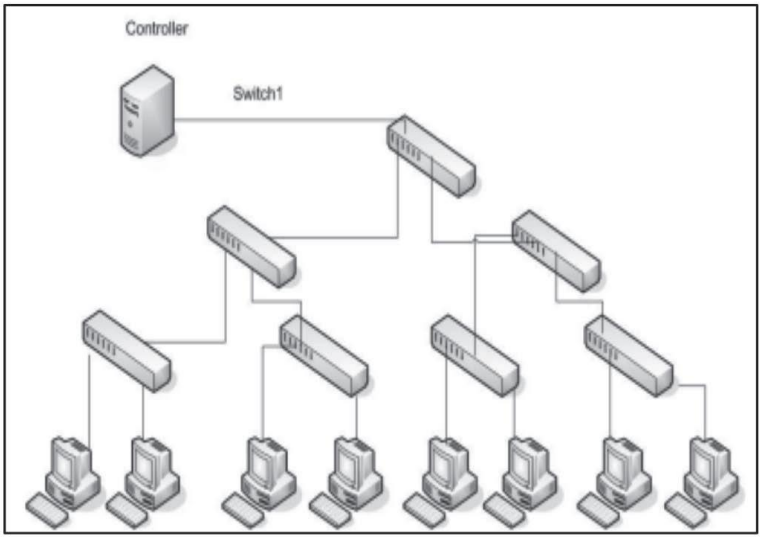


Figura 13. Topología Tree (Mininet)

Mininet utiliza un controlador por defecto incorporado a la distribución OpenFlow que por defecto actúa como un *switch* clásico. Esto significa que el controlador va creando una tabla donde se asocian direcciones MAC con puertos, según va recibiendo tramas. Lo que hace después de asociar la MAC con un puerto es enviar un mensaje OpenFlow al *switch* para crear una nueva entrada en su tabla de flujo. OpenFlow nombra a este tipo de controlador Ethernet *Learning Switch* y su fin es entender el flujo de mensajes. Al separar el plano de control del de datos no admite componentes creados por el usuario. Como se mostró en la tabla 3, Mininet acepta el uso de controladores reales y al iniciar una red en este emulador, cada *switch* se puede conectar a un controlador remoto, que podría estar en la máquina virtual, fuera de la máquina virtual, en el equipo local, o en cualquier parte del mundo. Para esto, cada vez que se inicie una topología debe indicarse que se usará un controlador remoto. El controlador puede ser NOX, POX, RYU, *Foodlight* o *Beacon*.

Utilizar un controlador diferente al predefinido por Mininet significa la ejecución de aplicaciones desarrolladas por el usuario y además de la mano de la evolución de OpenFlow se podría requerir el soporte de versiones nuevas de este protocolo. Adaptado de (Valencia, Santacruz, Becerra, & Padilla, 2015).

2.2.1.4 OpenDaylight (ODL)

OpenDaylight es un proyecto de código abierto que nació a raíz del surgimiento de las SDN y cuyo objetivo es difundir la innovación en el diseño e implementación de un estándar abierto y transparente de SDN. Este proyecto cuenta con una plataforma modular del mismo nombre que permite customizar y automatizar SDN de cualquier tamaño, enfocándose en la programabilidad de la misma. Para este trabajo se utilizó la versión Lithium de su plataforma, y se añadieron varios componentes a la instalación que serán descritos más adelante.

2.2.1.4.1 Arquitectura ODL

El núcleo de la plataforma es la capa enfocada al modelo de abstracción de servicios (MD-SAL), de sus siglas en inglés (*Model-Driven Service Abstraction Layer*). Los dispositivos de red subyacentes y las aplicaciones de red son representados como objetos, o modelos, cuyas interacciones se manejan en la capa de abstracción de servicios (*Service Abstraction Layer* o SAL por sus siglas en inglés).

La capa SAL a su vez es un mecanismo de intercambio y adaptación de datos entre modelos que representan dispositivos de red y aplicaciones. Los modelos proporcionan descripciones generalizadas de las capacidades de un dispositivo o aplicación sin que sea necesario conocer los detalles específicos de implementación del otro dispositivo.

Dentro de esta capa, los modelos se definen simplemente por sus roles respectivos en una interacción dada. Un modelo “productor” implementa una API y proporciona los datos de la API; un modelo “consumidor” usa la API y consume los datos de la API.

2.2.1.5 Cisco OpenFlow Manager

Cisco OpenFLow Manager (OFM por sus siglas en inglés) es una aplicación desarrollada para correr sobre un controlador *SDN OpenDaylight* para visualizar topologías OpenFlow y gestionar los flujos en la red, es decir, filtrar y realizar operaciones con el tráfico de la red, administrándola y automatizando procesos. OFM es una aplicación *Northbound*, es decir una API, que se conecta al controlador para definir las reglas con las que trabaja la red, y a su vez el controlador se conecta a la capa de infraestructura con los *switches* OpenFlow que tienen como objetivo realizar ciertas acciones cuando se cumplan las condiciones definidas en los flujos.

3. SITUACIÓN ACTUAL DEL COMPLEJO JUDICIAL

En enero del 2017 se inauguró el nuevo complejo para la Dirección Provincial de Pichincha del Consejo de la Judicatura. Una gran construcción de 12 pisos ubicada en la avenida Amazonas y Villalengua en la ciudad de Quito. Esta gran estructura cuenta con 12 plantas por las cuales se encuentra distribuida una red ampliamente extensa que brinda diversos servicios y que cuenta con una gran cantidad de equipos de red sobrepasando los 60 *switches* Cisco 2960 incluidos dos *switches* de core Cisco 4500, y más de 2000 puntos de red incluyen: computadores, teléfonos IP, impresoras, puntos de acceso y servidores.

La mayoría de dispositivos en esta red son equipos que necesitan una administración individual ya que, en el caso de los *switches* se tiene ubicados entre 4 y 7 *switches* por cada piso divididos entre el ala norte y el ala sur del Complejo (Ver Figura 4). Estos no se encuentran configurados con EtherChannel o FiberChannel, que son tecnologías de Cisco para agregación de enlaces y agrupación de enlaces físicos en un solo enlace lógico. Esto representa una gran utilización de recursos de fibra óptica, ancho de banda y

demás, ya que cada uno de estos *switches* se conecta directamente al MDF a través de Fibra óptica.

Debido a motivos de seguridad y a la confidencialidad que se requiere en una entidad Judicial de este tipo, no se mostrara a detalle el diseño de la infraestructura de red actual. Sin embargo, a continuación la figura 14 representa el diseño general de cómo se distribuyen los dispositivos en dicha red:

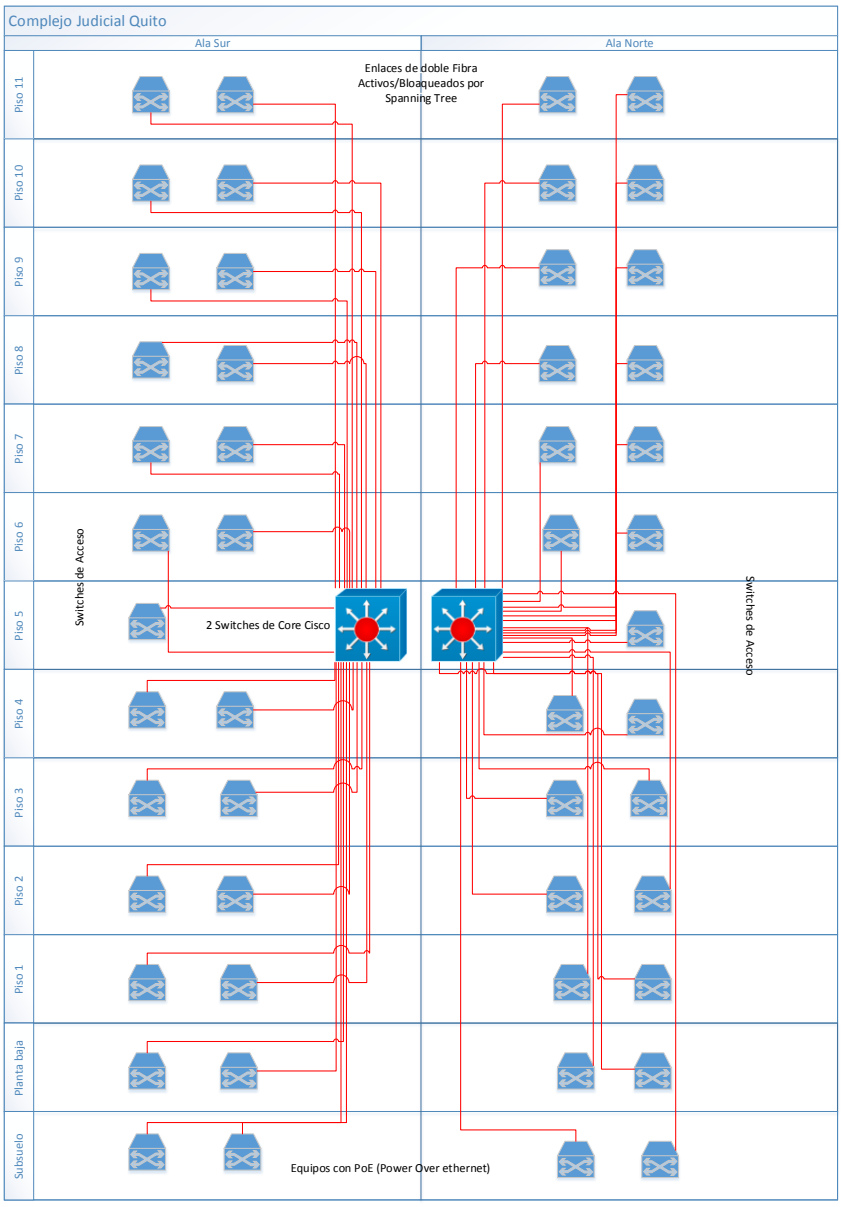


Figura 14. Diagrama de red complejo judicial

Al ser una red tan amplia, con tantos puntos de red y *switches*, su administración es una tarea bastante compleja ya que varios de estos equipos requieren ser configurados y administrados uno por uno lo cual alarga y dificulta el trabajo para su administrador.

En base a los estándares y definiciones detallados en el marco teórico, contrastados con la información de la red actual del complejo judicial se puede evidenciar que actualmente la red cuenta con un backbone de fibra óptica. Esta red cuenta además con una gran cantidad de *switches* de acceso conectados directamente a los swiches de core a través de enlaces de fibra óptica, sin estar apilados entre sí, lo cual incrementa la cantidad de equipos que requieren administración. Utilizando además cableado horizontal y vertical y varias configuraciones que podrían ser mejoradas con la propuesta SDN.

Además, la red del complejo judicial cuenta en cada piso con dos cuartos de comunicaciones, uno en ala norte del edificio y otro en el ala sur. En cada uno de estas cuartos, se ubica un *Intermediate Distribution Frame* (IDF por sus siglas en inglés) en el cual se ubican entre 2 y 7 *switches* y cada uno de estos *switches* se conecta al *Main Distribution Frame* (MDF por sus siglas en inglés) a través de una fibra óptica, es decir una fibra por cada uno de los 2 a 7 *switches* ubicados en cada piso del complejo judicial como se aprecia en la figura 15.

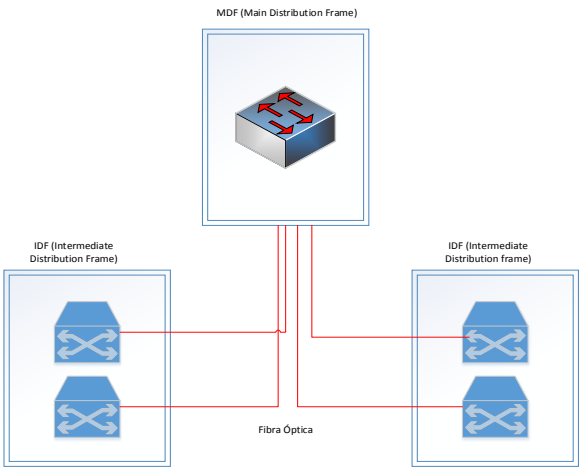


Figura 15. Distribución de *switches* por piso

Es evidente que la configuración actual es complicada al verificarse que cada equipo necesita una administración individual, también existe una falta de optimización de recursos en cuanto al uso de fibra óptica ya que basta con un solo enlace de fibra para cada IDF para suplir las necesidades de la red. Con una propuesta SDN se podría administrar la red únicamente a través del controlador sin la necesidad de entrar en la configuración interna de cada equipo que puede estar sujeta a una configuración distinta en el *software* de cada dispositivo.

La red cuenta con varios direccionamientos (direccionamiento red inalámbrica, WAN de datos, WAN de telefonía y WAN de los servidores), estos a su vez se conectan a una red LAN, la cual está conectada con 2 *switches* de core a la red WAN del ISP (CNT). A los *switches* de core se encuentran conectados aproximadamente 69 *switches* mediante enlaces dobles de fibra activo/bloqueados por *spanning tree* y los cuales se encuentran distribuidos por el complejo de la siguiente manera: 3 *switches* en el subsuelo, 7 en la planta baja, 5 en el piso 1, 6 en el piso 2, 6 en el piso 3, 5 en el piso 4, 7 en el piso 5, 4 en el piso 6, 6 en el piso 7, 6 en el piso 8, 5 en el piso 9, 7 en el piso 10 y dos en el piso 11.

La administración de los *switches* de la red del complejo judicial es una tarea laboriosa ya que al tener tantos usuarios en la red y brindar tantos servicios con dos redes LAN y dos redes de voz en la mayoría de los pisos, su mantenimiento se dificulta, es costoso y la implementación de nuevos servicios es una tarea que conlleva mucho tiempo y un gran estudio incluso después de su implementación para comprobar que no existan fallos en la red.

Para optimizar todos los recursos con los que cuenta actualmente esta red, en primer lugar es necesario apilar los *switches* de manera que lógicamente se reduzca el número de *switches* por administrar como muestra la figura 16.

Figura 16. Switches Apilados.

Además de esto se podría implementar EtherChannel, que consiste en una tecnología que permite agrupar varios enlaces físicos en un solo enlace lógico, lo cual optimizaría el uso de fibra, además se podría implementar PortChannel que permite agregar enlaces para tener enlaces de hasta 10Gbps en cada uno de los puestos de trabajo, es decir, se tendrían solo dos fibras ópticas conectadas al MDF por cada piso, una de cada cuarto de comunicaciones, agregados ahí los varios enlaces lógicos de la red.

Esta mejora sería incluso aún más efectiva con la implementación de una red SDN que facilite la gestión, ya que se tendría una menor cantidad de *switches* lógicos los cuales no tendrían que ser administrados uno por uno, sino ser abstraídos por el controlador SDN, desde donde se gestionaría toda la red, facilitando la su administración.

Además, se tendría la posibilidad de implementar diversos servicios, que permitan gestionar la red de una manera más eficiente, reducir su complejidad y costos de operación.

Gracias a las tablas de flujo en el protocolo *Open Flow*, utilizado en las SDN, sería posible implantar políticas que filtren paquetes, redirijan tráfico, bloqueen tráfico no deseado, etiqueten paquetes, redirijan cierto tipo de tráfico a usuarios específicos. Con la ayuda de las APIs las cuales conectadas al controlador brindaran servicios como los mencionados anteriormente y permitirán implementar más servicios.

Estos y más beneficios del uso de una red SDN serán descritos más a detalle en el capítulo posterior.

4. VENTAJAS Y LIMITACIONES DE UTILIZAR REDES SDN EN LUGAR DE REDES TRADICIONALES

En una red tradicional los *switches* están organizados de manera jerárquica, son programados antes de su operación y cada uno, con políticas individuales para lograr el funcionamiento adecuado de la red. Sin embargo, esta necesidad de ser programados independientemente es la que actualmente representa un estancamiento para diversas empresas o instituciones como la Dirección provincial de Pichincha del Consejo de la Judicatura que requieren innovar con nuevas aplicaciones, nuevos servicios o procesos tecnológicos dinámicos.

Las redes son un punto clave en el desarrollo tecnológico de la presente generación, debido a que son fundamentales para poder ofrecer distintos servicios de IT a la gran cantidad de usuarios que una red puede tener, tomando como ejemplo una institución como la Dirección provincial de Pichincha del Consejo de la Judicatura. Para esta institución se necesita de una red confiable para poder distribuir datos internamente y al exterior de la misma, asegurando la seguridad de la información y llevando un control granular de la misma para poder brindar los servicios que una institución judicial requiere. Las redes tradicionales no están pensadas para afrontar las necesidades de las distintas tendencias tecnológicas que están surgiendo día a día, en cuanto a los nuevos servicios y aplicaciones que el usuario requiere, como el Cloud Computing, el uso de Big Data y la virtualización de servicios en una institución. Estas funcionalidades que se hacen cada vez más necesarias y por lo tanto utilizadas en todo tipo de empresas e instituciones con grandes cantidades de usuarios.

En contraste con las redes tradicionales que en su mayoría operan con tecnologías y protocolos propietarios, las SDN tienen a su favor: Cuerpos internacionales de estándares (ONF) que definen protocolos y estándares esenciales para las tecnologías de SDN, fabricantes de equipo que desarrollan y mejoran productos compatibles con SDN; universidades que implementan proyectos iniciales de SDN para validar los nuevos diseños de red, sus

tecnologías, y sus aplicaciones. Adaptado de (Parra Loera, Morales Rocha, & Hernández Hernández, 2015).

Las ventajas de una red SDN en comparación a una red tradicional han sido detalladas en la encuesta realizada por *Gigaom Research* en el 2014 (ver figura 17), donde se indican los beneficios más importantes en orden ascendente del uso de SDN´s:

1. Habilitación de la programación de la red bajo demanda.
2. Aprovisionamiento acelerado de nuevos clientes y servicios.
3. Bajo costo de inversión.
4. Bajo costo de operación.
5. Simplificación en el despliegue y operación de la red.
6. Altos niveles de utilización de la red.
7. Fortalecimiento de la seguridad de la red.

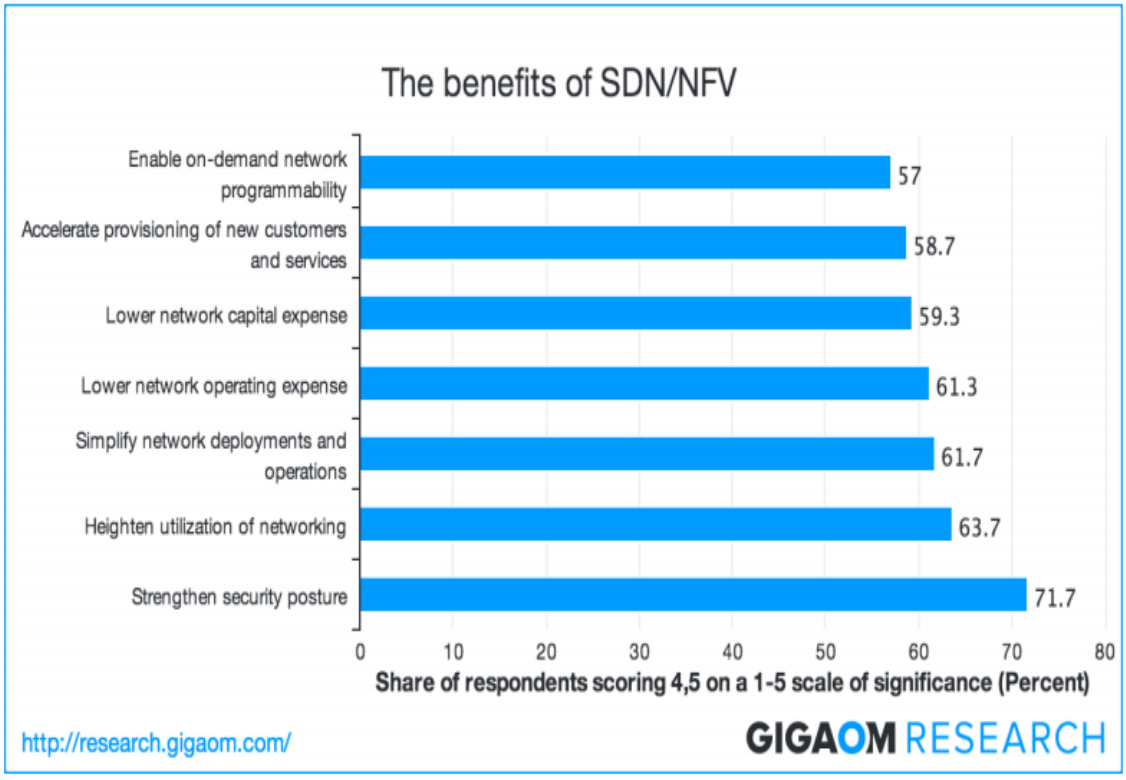


Figura 17. Beneficios de las redes SDN.
Tomado de (Parra Loera, Morales Rocha, & Hernández Hernández, 2015).

Las ventajas mencionadas anteriormente son solo algunas de todas las ventajas que se tiene al utilizar este tipo de tecnología.

A continuación se muestran ventajas de la tecnología SDN orientadas a una red como la del Complejo Judicial:

4.1 Seguridad de la red

En una institución como el complejo judicial, es particularmente necesario contar con una red segura debido a la fragilidad de la información que viaja por ella. Para esto la alternativa SDN tiene como ventaja un controlador centralizado permite crear un punto central de control para distribuir información de seguridad y de políticas de una manera sólida.

Al tener un control centralizado de la red, es posible mejorar la seguridad de la red con la provisión de un sistema de monitorización, análisis y respuesta de seguridad que reaccione rápidamente ante cualquier anomalía. El análisis del tráfico de la red y métodos de detección de anomalías a través de sondas en la red, generan datos relacionados con la seguridad, que son transferidos al controlador central. La capa de aplicación puede utilizar APIs que se pueden ejecutar en el controlador para analizar y correlacionar estos datos de la red. Con base en el análisis, la política de seguridad nueva o actualizada se puede propagar a través del controlador de la red en forma de reglas de flujo. Este enfoque consolidado puede acelerar de manera eficiente el control y la contención de las amenazas a la red y su seguridad. Adaptado de (Seguridad en las redes definidas por Software (SDN), 2017)

Esto sería de gran utilidad en el complejo judicial al poder redirigir tráfico sensible exclusivamente a los usuarios que lo requieran y generar políticas para automatizar estos procesos.

4.2 Reducción de la complejidad

Uno de los principales problemas de la arquitectura de redes tradicionales es la complejidad, generada principalmente por la necesidad de “apilamiento” de

protocolos creados para atender a diversas demandas. De esa manera para cada alteración en la infraestructura son necesarias configuraciones en diversos niveles, en cada elemento. Además de que, para redes como la del Complejo judicial en particular, que cuentan con una gran cantidad de dispositivos sin ser apilados, es mucho más extensa la configuración que requiere cada equipo.

Con el enfoque SDN no hay necesidad de usar protocolos, ya que los controles no son hechos en el nivel de los equipos, sino a nivel de controlador. Además, la tecnología SDN hace posible el desarrollo de herramientas que automatizan muchas actividades de gestión de la red que hoy es realizada manualmente. De esta manera, la complejidad de la red es reducida significativamente, posibilitando la reducción de la mano de obra, al mismo tiempo que disminuye la inestabilidad (causada por errores de configuración) y permite modelos de suministro mucho más ágiles. Adaptado de (Logicalis, 2014)

4.3 Reducción de costos

Con una infraestructura más simple, que disminuye el número de equipos como *switches* de acceso y fibra óptica utilizada para cada uno de estos es evidente que los precios de implementación se aminoran en gran cantidad. El nivel de especialización también es reducido con la independencia en relación a los grandes proveedores, ya que al ser una tecnología estandarizada con protocolos libres, su desarrollo es más veloz, y barato. Además de ser necesario un menor número de profesionales.

El costo total de propiedad (TCO por sus siglas en inglés) y los costos operacionales de la infraestructura también son reducidos, proporcionando una mejor economía a los administradores. En un *data center*, esa reducción se ve reflejada directamente en los costos de interfaces, permitiendo la adopción de velocidades mayores en el acceso, como 40G Ethernet. Este tipo de

velocidades permiten la reducción en la cantidad de interfaces, lo que resulta en una reducción de costos de cableado y OPEX.

4.4 Control centralizado y Granularidad

Por un lado, la arquitectura de las SDN garantiza el control centralizado de la infraestructura permitiendo controlar varios dispositivos distintos, de distintos proveedores, con diferentes protocolos a partir de un punto central. También es posible implementar políticas a un nivel extremadamente granular, permitiendo de esta manera, no solo controlar la información de una manera ampliamente efectiva, sino que a su vez esto proporciona seguridad en cuanto a qué acciones se deben tomar con cierto tipo de tráfico o cierto tipo de paquetes. En el complejo judicial, debido a que existe comunicación con otras instituciones, es necesario un mejor control del tráfico y una mayor seguridad del mismo.

4.5 Agilidad en el desarrollo de aplicaciones

Debido a su configuración más simple y control centralizado, es posible adecuar la infraestructura de la red a las necesidades del negocio o del usuario, y gracias a la virtualización del ambiente de redes pueden definir políticas escalables y flexibles basadas en la aplicación. De esta manera es posible editar aplicaciones o desplegarlas directamente en la capa de red, aumentando la rapidez en su desarrollo. Esto sería beneficioso para una institución judicial, para poder crear aplicaciones de manera rápida y segura, teniendo así una red altamente escalable.

5. DISENO DE RED SDN PARA EL COMPLEJO DE LA JUDICATURA

Este capítulo describe detalladamente como se realizó la simulación de una red SDN a escala en base a la red actual del Complejo Judicial, simulando una proporción de toda su infraestructura, demostrando la funcionalidad de la tecnología SDN para suplir las necesidades de esta institución.

Para la realización de esta simulación se tomó en cuenta las porciones de red con mayor número de usuarios y mayor cantidad de tráfico dando como resultado que: las plantas congestionadas en cuanto a tráfico de red son el subsuelo, en donde se encuentra el área de Archivo, a su vez la planta baja en donde se encuentra la mayor cantidad de equipos, llegando a 160 computadoras, sin contar impresoras ni teléfonos IP para el área de atención al cliente y ventanillas.

También existe congestión de tráfico en el piso 5, en donde se ubican los MDF con los *switches* de core a donde se conectan la gran cantidad de *switches* de acceso a través de fibra óptica. Por último se tiene el piso 10 del complejo judicial, en donde se encuentra ubicada el área de juzgados y jueces, en donde existe congestión en el tráfico al necesitarse un alto nivel de seguridad, debido a la sensibilidad de la información. Debido a los puntos expuestos anteriormente se realizó el diseño de una red que cuente con 20 *switches OpenFlow* con un *host* conectado a cada uno, para simular tráfico a través de esta red y demostrar cómo se puede controlar el tráfico de una manera granular.

Para el diseño de la red se necesitó realizar varias configuraciones previas a la creación de la red, incluyendo la instalación y configuración de máquinas virtuales, ya que se trabajó con los siguientes sistemas operativos: Mininet en su versión 2.2.2 y Ubuntu Server en su versión 14.04.3 LTS, en donde se realizaron varias configuraciones, y en donde se instaló el controlador SDN OpenDaylight.

Se trabajó con el sistema operativo Ubuntu Server debido a que Mininet es un sistema operativo basado en Ubuntu y que aprovecha su kernel, al igual que el *software* controlador OpenDaylight.

Para este trabajo y por facilitar el despliegue de los sistemas operativos se trabajó con Oracle VirtualBox que es un *software* hipervisor que permite trabajar con máquinas virtuales de distintos sistemas operativos. Se puede descargar este hipervisor desde su página web: https://www.virtualbox.org/
Su instalación es sencilla y con ayuda del asistente de instalación se puede finalizar en minutos.

Después de haber realizado la instalación del hipervisor, se procedió a descargar el sistema operativo Ubuntu Server, que funcionara como controlador ya que en él se instala el *software* controlador SDN *OpenDaylight*. Se utilizó la versión 14.04.3 LTS y se descargó de su sitio oficial: https://www.ubuntu.com/. Se descarga un archivo tipo iso el cual se instala en una máquina virtual de VirtualBox. Para esto es necesario dar click en “Nueva” en VirtualBox y definir el nombre que se dará al sistema, en este caso ODL ya que ahí se instaló el *software* controlador OpenDaylight como muestra la figura 18:

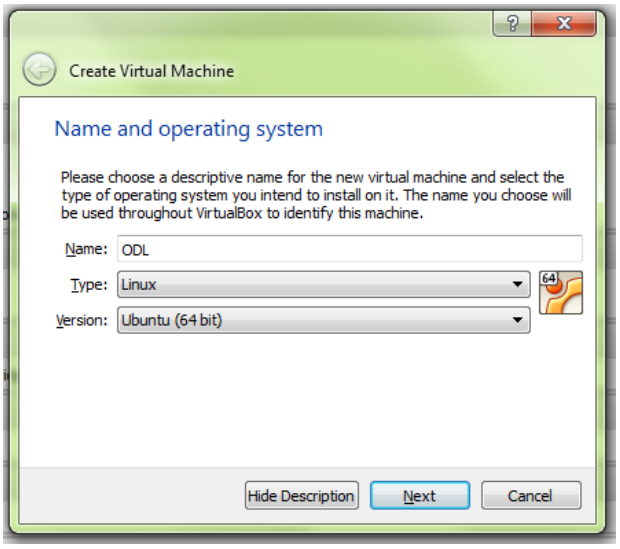


Figura 18. Creación de máquina virtual Ubuntu.

Luego, se definieron los parámetros para la máquina virtual en cuanto a sus prestaciones de memoria, disco duro, sistema operativo y su versión. Es importante recalcar el hecho de que ya que se trabajó con una computadora *host* y dos máquinas virtuales, fue necesario que exista una comunicación

entre ellas, para esto se instaló un servidor SSH durante la instalación de Ubuntu Server (ver figura 19):

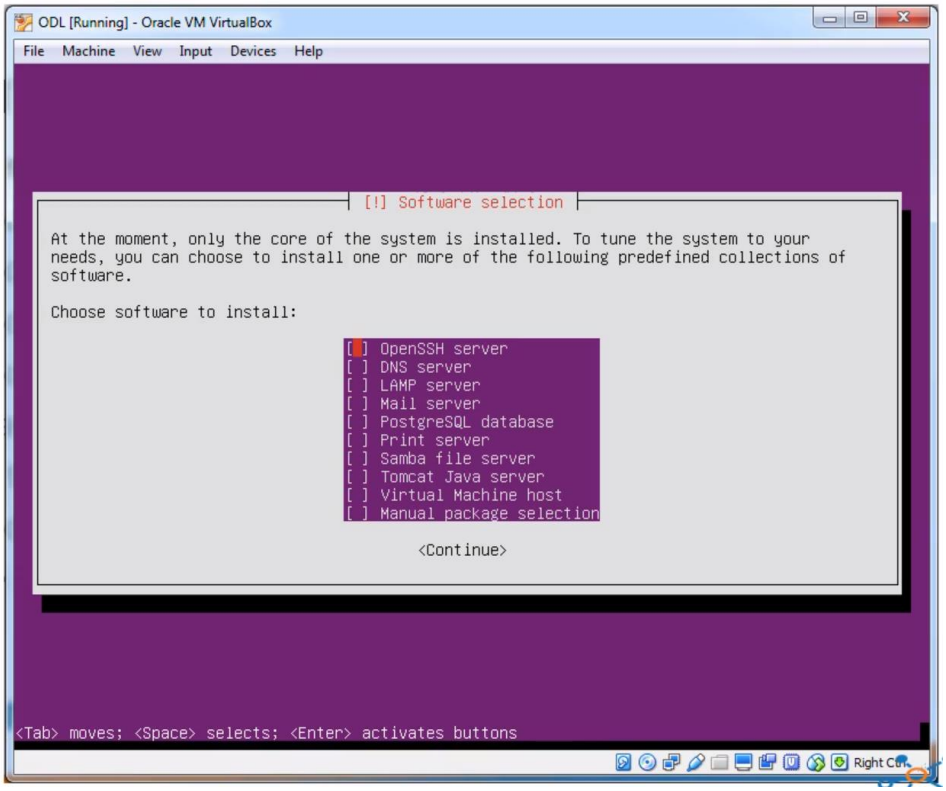


Figura 19. Servidor SSH en Ububtu Server

Después de realizar la instalación de Ubuntu Server se procedió a realizar configuraciones en el sistema previas a la instalación del controlador SDN.

Para esta simulación se necesitó configurar una dirección IP estática al sistema para que sea más fácil su integración con el sistema operativo Mininet y con la computadora *host*. Para esto se utilizó el comando: *"sudo nano /etc/network/interfaces"* en donde se muestran las interfaces de red y en donde se configura una direccionamiento estático para el servidor, definiendo su dirección IP, mascara de red, Gateway y demás como se ve en la figura 20:

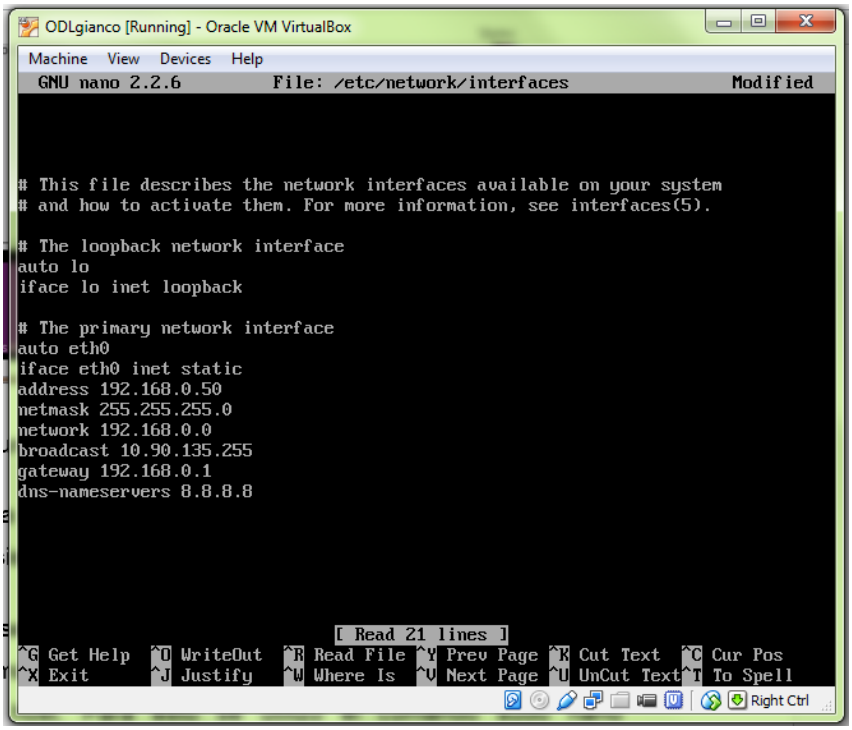


Figura 20. Dirección IP estática para Ubuntu Server

Después, en la configuración de red de la máquina virtual se configuro el adaptador de red en modo puente para comunicarse de esta manera con la computadora *host* (ver figura 21):

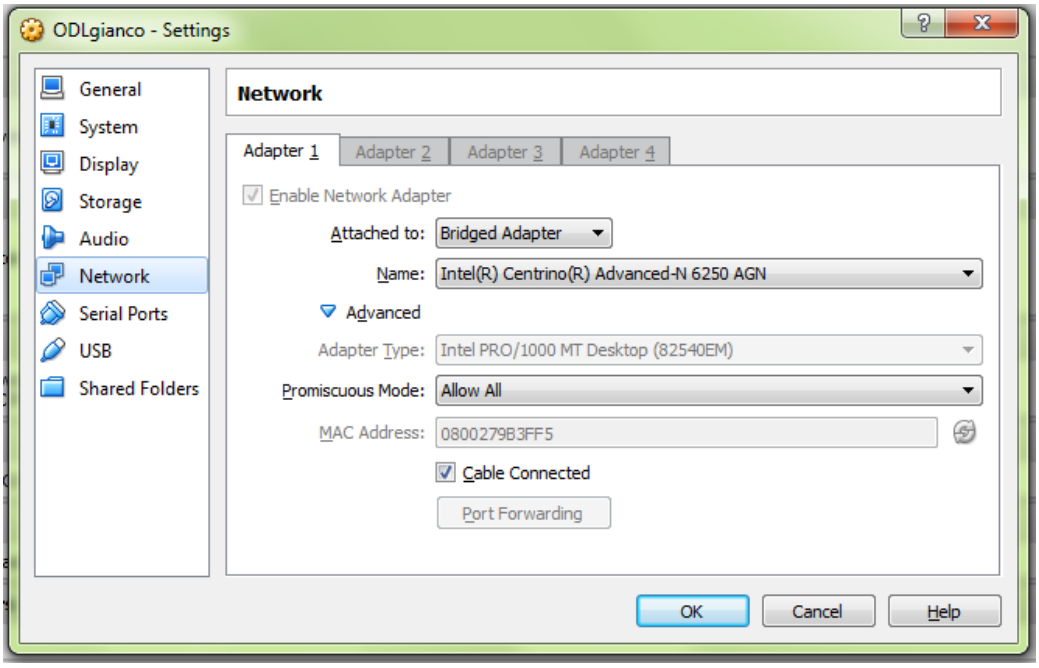


Figura 21. Configuración de red Ubuntu Server

Una vez realizadas las configuraciones previas a la instalación del controlador SDN, se procedió a descargar el mismo de su página oficial, copiando la dirección del link y luego instalándola en Ubuntu Server como se ilustra en la figura 22:

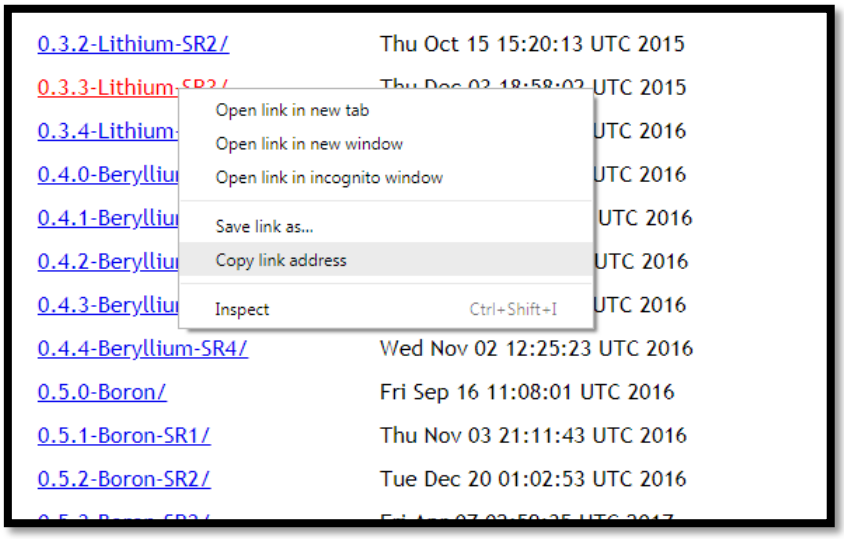


Figura 22. Enlace de descarga OpenDaylight

Para esto fue necesario iniciar una sesión SSH a Ubuntu server a través de la herramienta PuTTy:

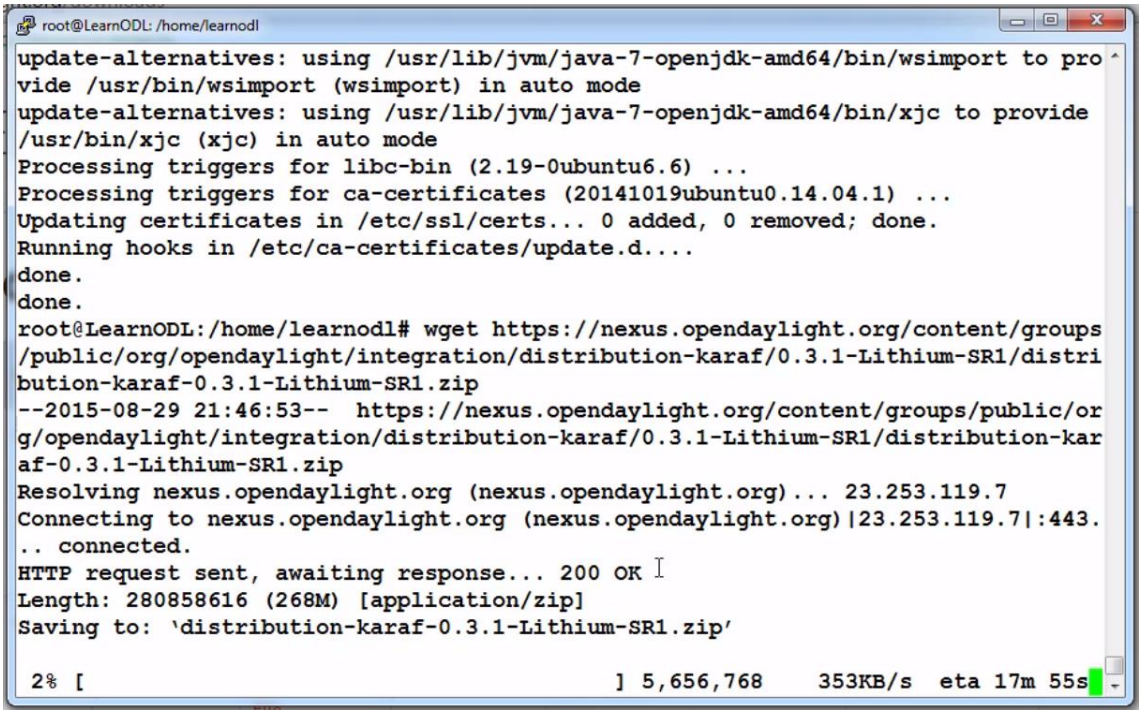


Figura 23. Descarga e instalación de ODL

Después se procedió a descomprimir el archivo del controlador SDN de nombre "*distribution-karaf-0.3.1-Lithium-SR1.zip*". Después, se instaló el directorio Java necesario para el funcionamiento del controlador utilizando el comando: *"export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-amd64"* y finalmente, para iniciar el controlador se accedió al archivo "karaf" dentro de la carpeta "bin" en el directorio del controlador utilizó el comando: *"./bin/karaf"* como se muestra en la figura 24:

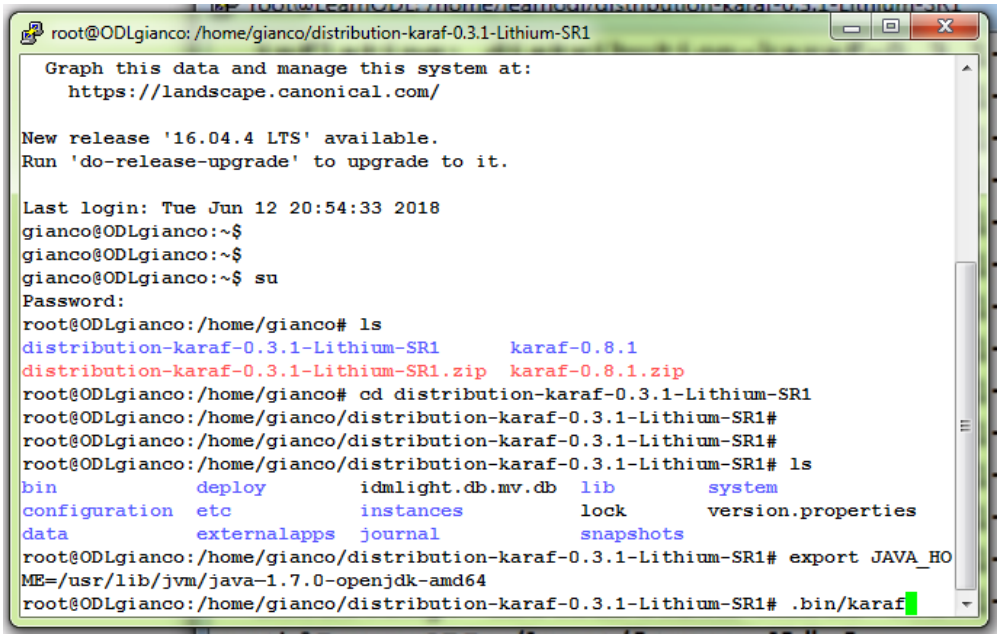


Figura 24. Inicialización de controlador SDN

Al inicializarse el controlador se muestra de la siguiente manera (ver figura 25):

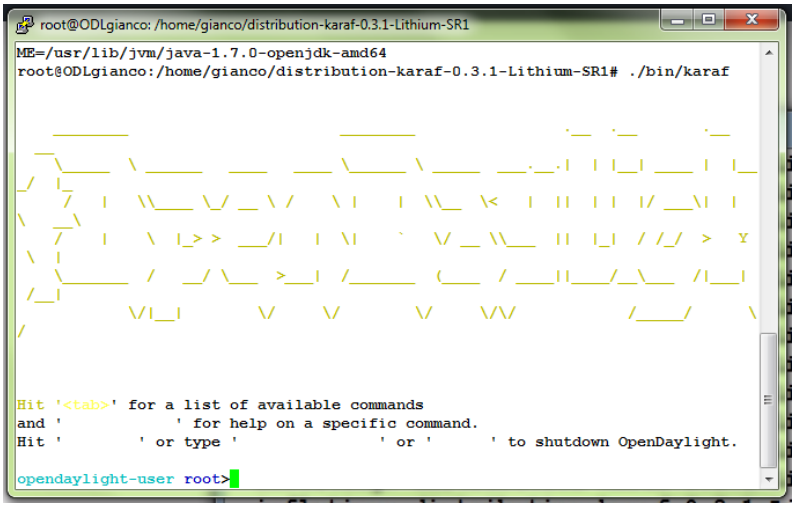


Figura 25. Controlador OpenDaylight Inicializado.

Después de haber inicializado el controlador SDN, es necesario instalar varios componentes que permiten su funcionamiento, incluida una interfaz gráfica para poder visualizar las redes simuladas en Mininet de manera gráfica en el controlador y desde donde se manejan las tablas de flujo con las cuales se controlara el tráfico de la red.

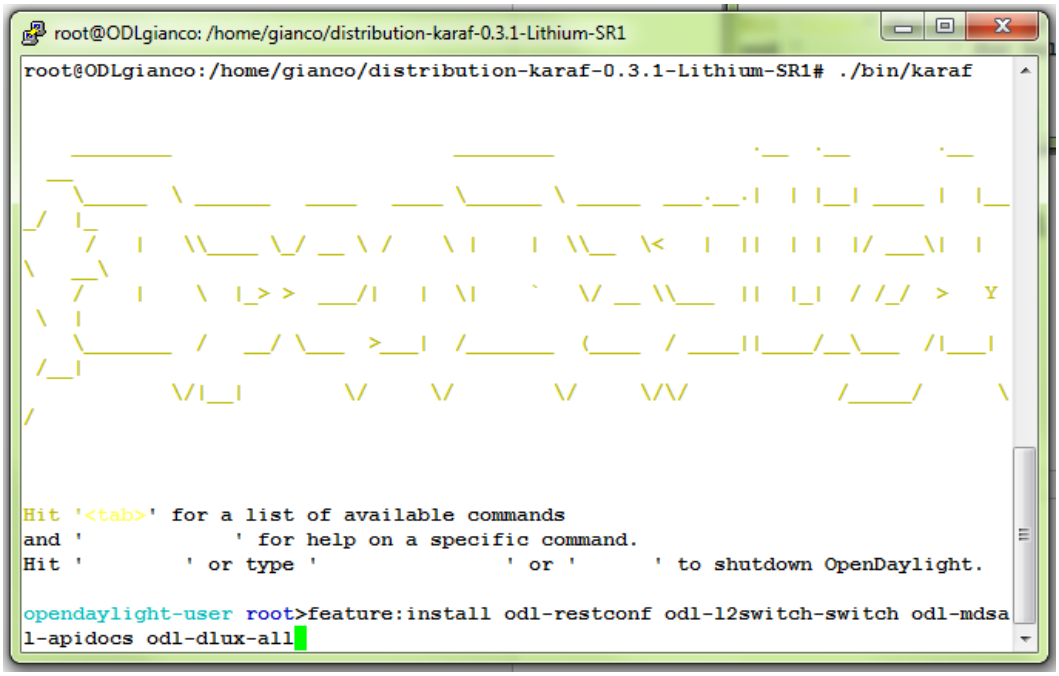


Figura 26. Instalación de componentes en ODL

Después de realizar las configuraciones mencionadas anteriormente, se tendrá a disposición un componente *switch* de capa 2 instalado y que a su vez es un "*learning switch*" que aprende del tráfico que pasa por él. También una interfaz gráfica a través de los componentes DLUX de *OpenDaylight*. La instalación de los componentes DLUX permite visualizar la red, las diferentes topologías y controlar las tablas de flujos que permiten obtener estadísticas acerca del desempeño de la red como se muestra en la figura 27:

Figura 27. Estadísticas de nodo *OpenDaylight*

Para acceder a la interfaz gráfica del controlador hay que acceder a través del navegador web de la computadora *host* accediendo su dirección IP con la URL: *192.168.0.50:8181/index.html#/login*. ODL utiliza por defecto el puerto 8181.

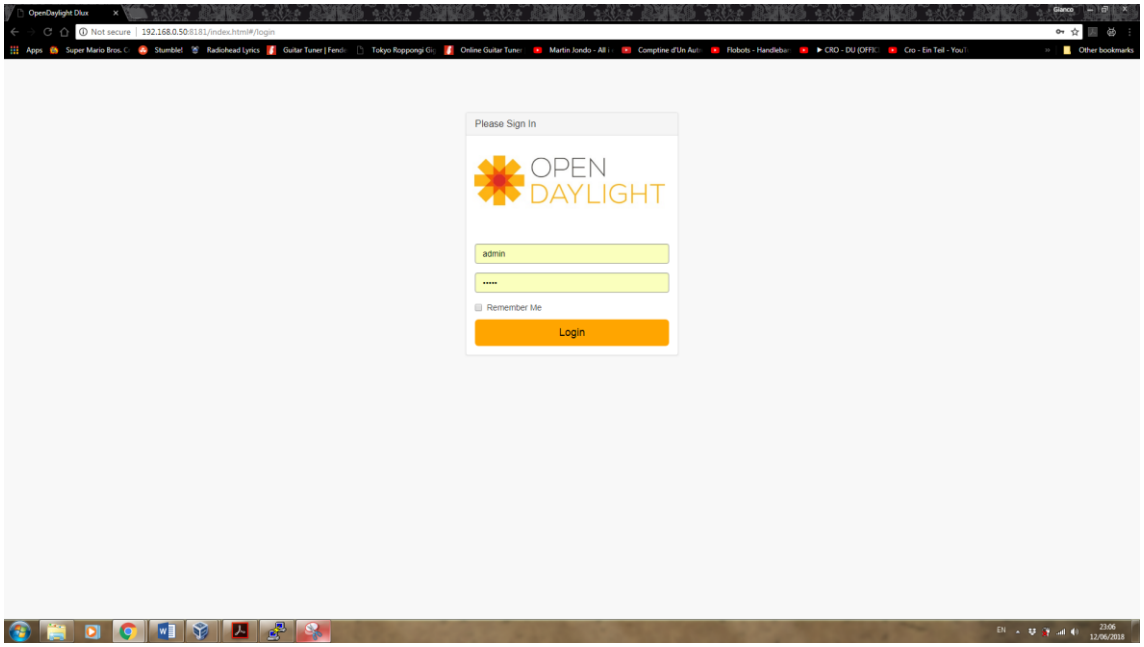


Figura 28. Pantalla de inicio interfaz ODL

Como se aprecia en la figura 28, se puede acceder al controlador *OpenDaylight* a través de la interfaz DLUX instalada. Por defecto se accede con las credenciales: *Username*: admin, *Password*: admin.

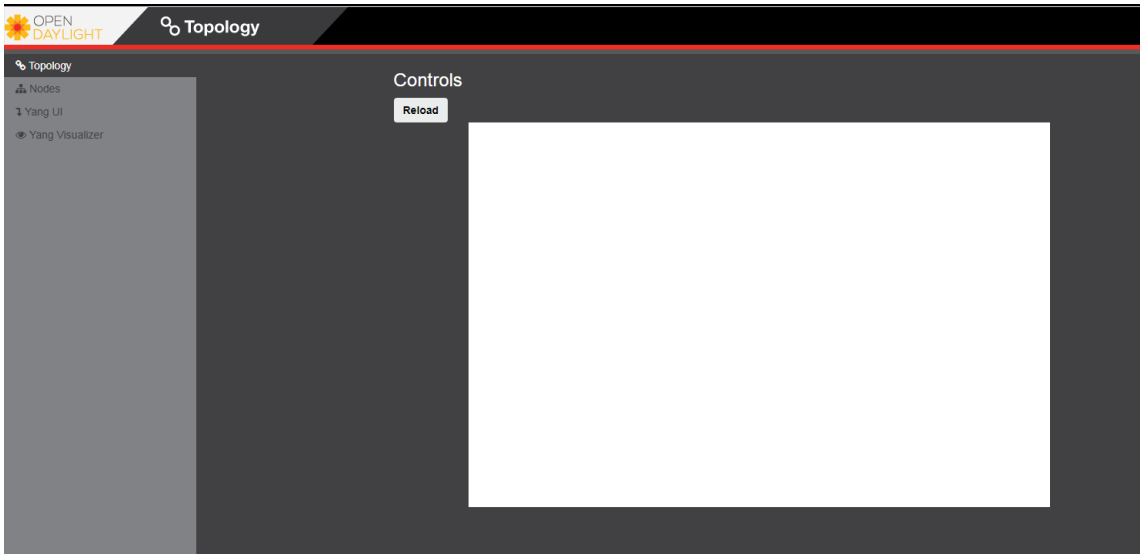


Figura 29. Controlador SDN

En principio no se muestra ningún dispositivo en la topología debido a que el controlador funciona activamente con Mininet, el *software* simulador de la red SDN, es decir, a través de comandos se crea una red desde el sistema operativo Mininet y esta red es visualizada por el controlador SDN.

El siguiente paso es descargar la imagen del sistema operativo Mininet de su sitio web oficial (http://mininet.org/):

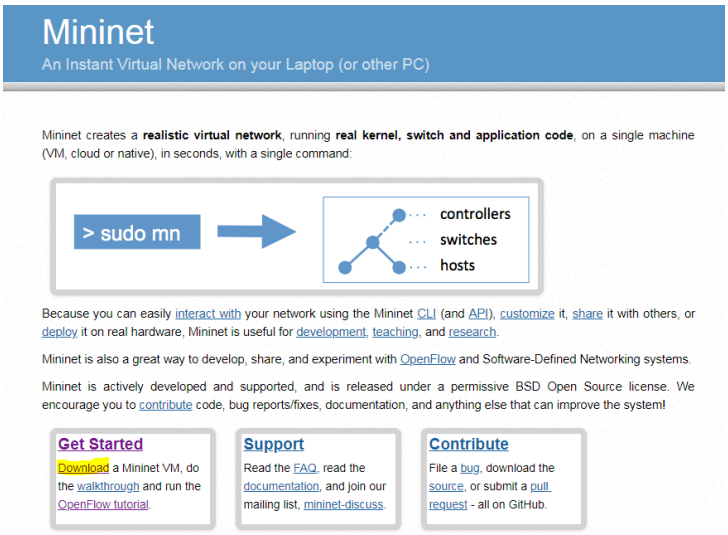


Figura 30. Descarga de Mininet

Una vez descargada la imagen del SO Mininet se procedió a levantar la máquina virtual de Mininet definiendo características de capacidad de memoria RAM y utilización de disco duro básicas, debido a que Mininet es un sistema operativo que no requiere de altas prestaciones en cuanto a *hardware*:

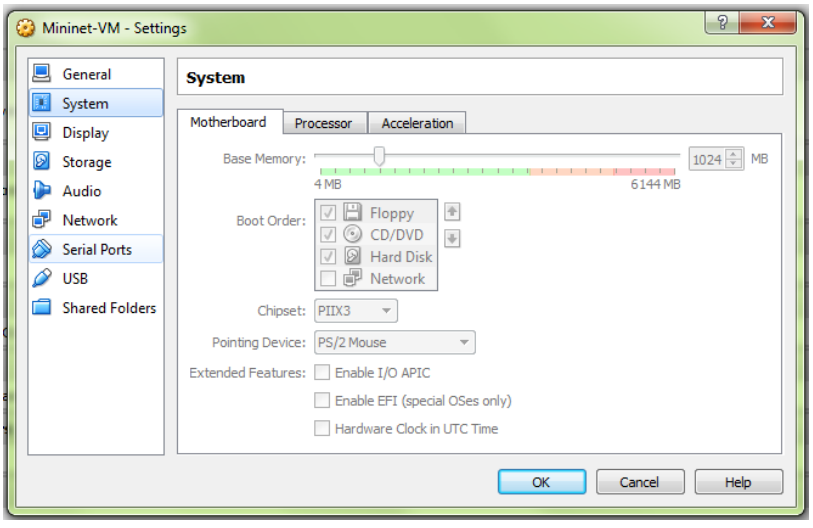


Figura 31. Prestaciones de *Hardware* requeridas por Mininet

Una vez instalado Mininet se procedió a inicializarlo para generar la red, ingresando con las credenciales por default: mininet – mininet:

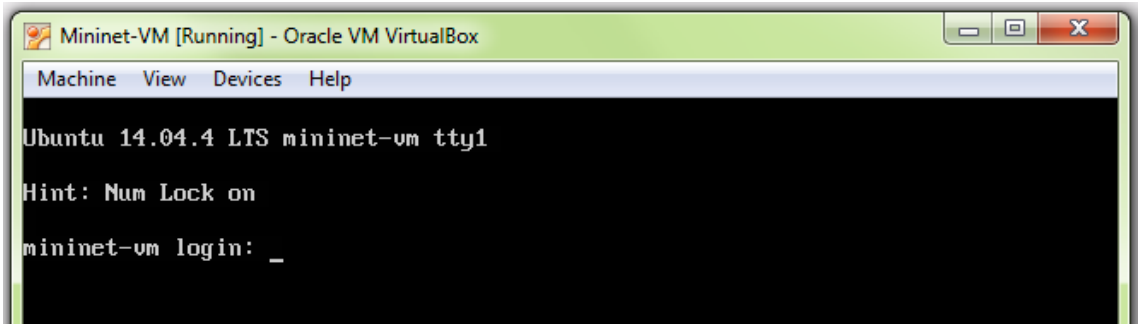


Figura 32. Mininet inicializado

El siguiente paso es verificar la dirección IP de Mininet para poder conectarse a las demás máquinas como muestra la figura 33:

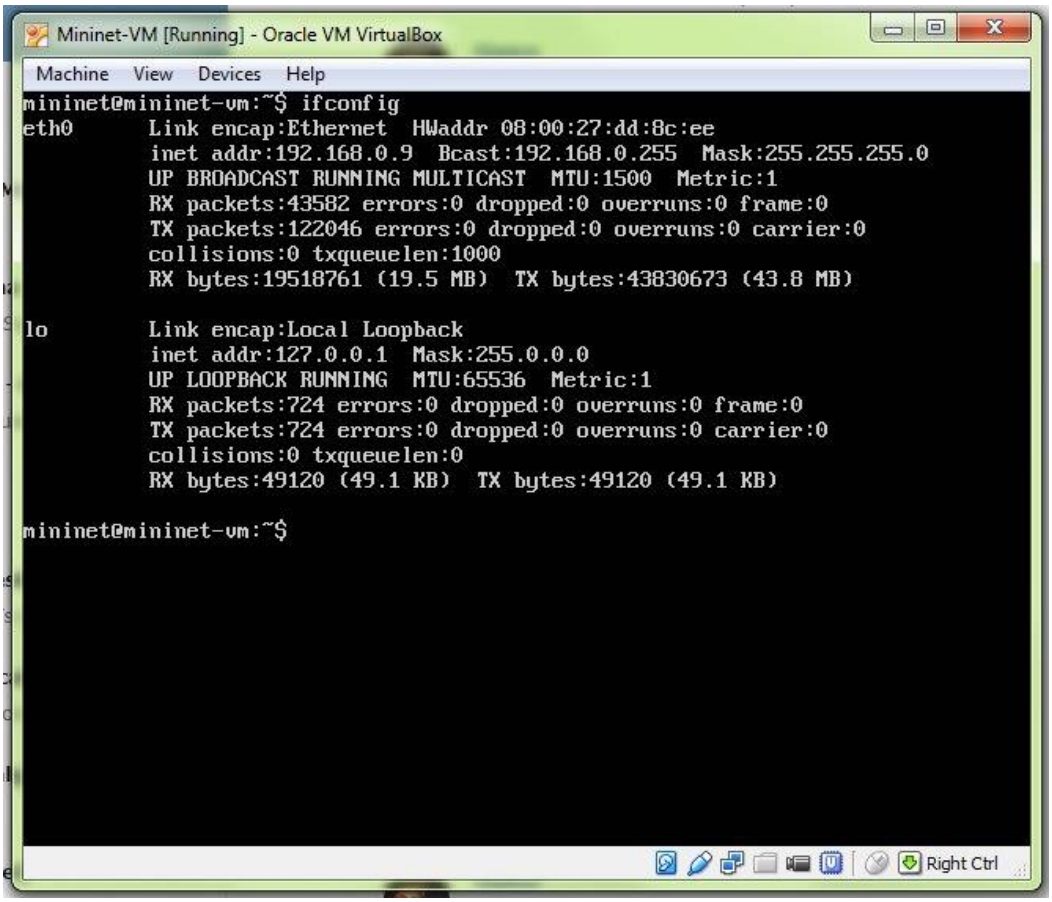


Figura 33. Dirección IP de Mininet

Mininet es una herramienta relativamente fácil de operar y solo con el comando "sudo mn" basta para crear una red simple con un *switch* y dos *hosts*. Además viene configurado un controlador SDN por defecto, que si no se especifica de otra manera, es con el cual se trabaja.

Para este trabajo se utilizó un controlador distinto al configurado por defecto, el controlador SDN *OpenDaylight*; así que para crear una red en Mininet con el controlador es necesario definir que el controlador que se utilizará en la red es remoto y a su vez su dirección IP con el comando: "*sudo mn –controller=remote,ip=192.168.0.50*". Esto creara una red básica que consta de dos *hosts* y un solo *switch* administrados por el controlador *OpenDaylight*:

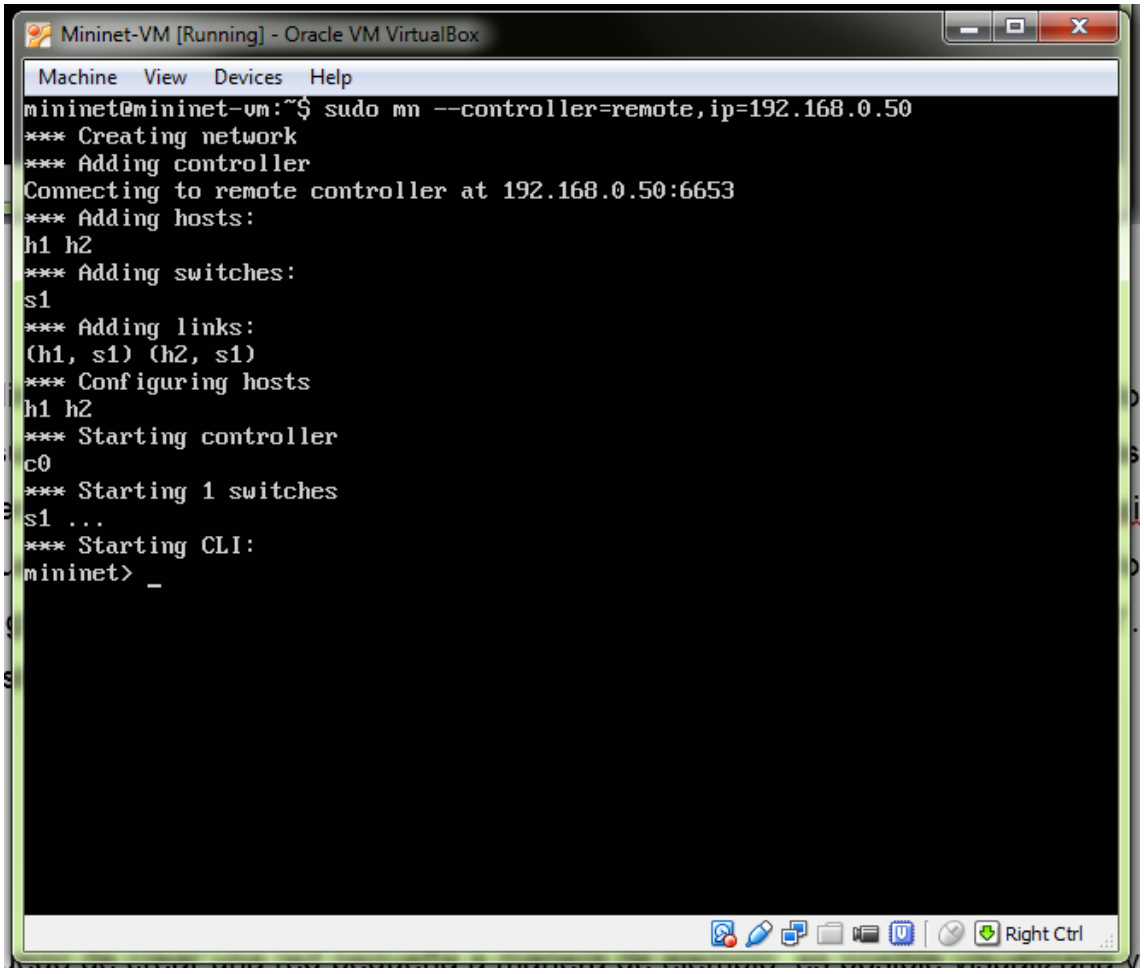


Figura 34. Creación de SDN en Mininet

Luego de crear una red pequeña a manera de ejemplo, es posible visualizarla y monitorearla desde el controlador *OpenDaylight*, sin embargo el controlador SDN solo conoce como está estructurada la red al pasar tráfico por la misma, por lo cual es necesario enviar tráfico a través de la red haciendo *ping* entre todos los *hosts* para ver en donde se ubican y conocer cuáles son sus direcciones mac a través del comando: "*pingall"* dentro de Mininet como se puede apreciar en la figura 35:

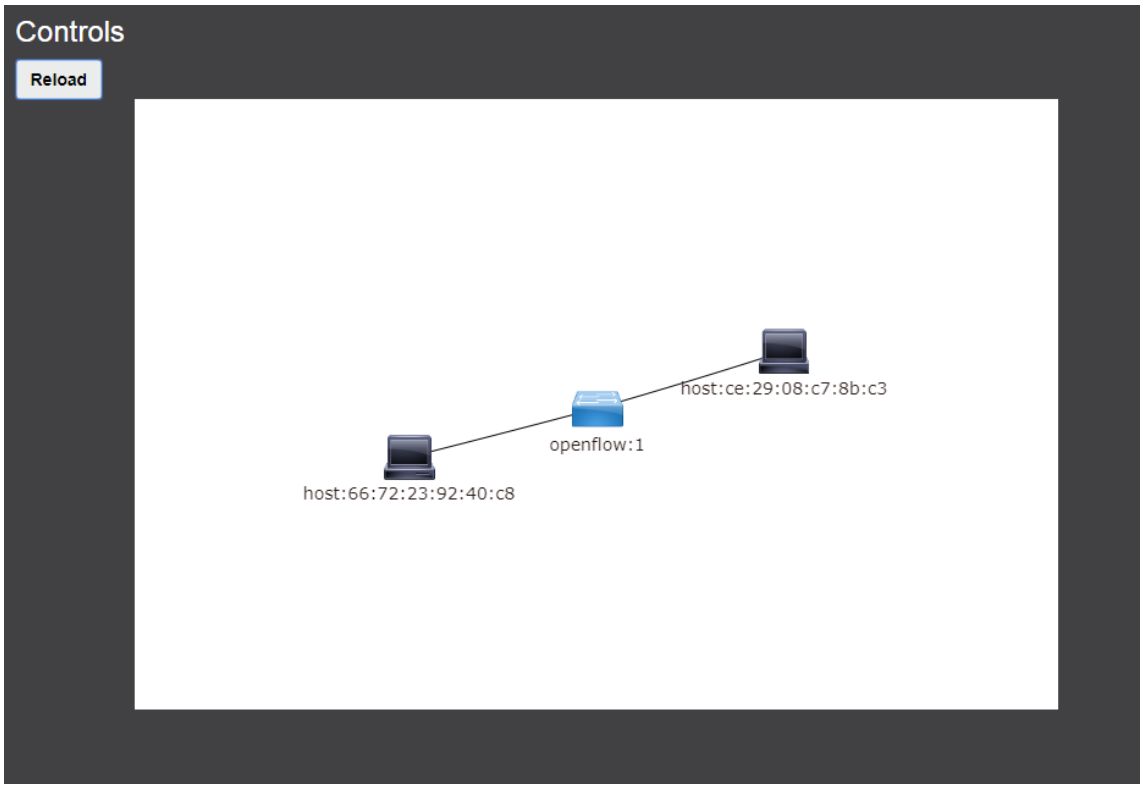


Figura 35. SDN básica después del ping.

Existen varias otras posibilidades para la creación de una red SDN a través de Mininet y a su vez existen varios comandos muy útiles para este motivo:

"*help*": Permite visualizar los comandos de CLI disponibles
"*node*": Muestra todos los dispositivos de red desplegados por Mininet.
"*dump*"; Permite mostrar la información de la red relacionada a cada dispositivo de red desplegado.
"*net*": Comando para ver cómo se encuentran interconectados los dispositivos de red.

Ahora, accediendo a la utilidad mn se muestran varias opciones más para el manejo de SDN, utilizando el comando: *“sudo mn –h”* se enlistan las siguientes opciones:
--*switch* = *SWITCH*: Por defecto trabaja con Open v*Switch* y se le puede indicar también con la opción ovsk. Si en cambio se escribe user, se implementa un *switch* más lento.
--controller = CONTROLLER: Por defecto trabaja con un controlador OpenFlow y que está preinstalado en la máquina virtual. Se puede trabajar también con controladores externos a Mininet detallando la dirección IP del mismo.

--topo = TOPO: Se crea una topología de red virtual, teniendo diferentes opciones: **minimal** crea una topología por defecto de un *switch* y dos *hosts*; **single**,X un solo *switch* conectado a X *hosts*; **linear**,X crea X *switches* conectados uno detrás de otro, cada uno con un *host* conectado; y **tree**,X un árbol con fanout X, o puntos hijos por cada nodo.

--mac: Asigna direcciones MAC fáciles a los *hosts* de forma automática.

Como se mencionó anteriormente, enfocándose en las porciones de red más congestionadas (Subsuelo, Planta baja y piso 10) es necesario simular una red que cuente con una gran cantidad de *switches*, ver cómo pasa el tráfico por estos, y a su vez verificar el control centralizado de la red, la disminución de la congestión del tráfico y la seguridad de la red.

Para esto primero es necesario eliminar la topología actual, utilizando el comando: *“sudo mn –c”*. Después, para crear una red en escala de la red actual del Complejo judicial se trabaja creando una red que contenga 20 *switches* OpenFlow con dos *hosts* conectados a cada uno. Esta red es una red customizada que no se crea con las topologías básicas debido a que se tiene 2 *hosts* por cada *switch*. Es por esto que para la creación de dicha red, se tiene que acceder a un archivo Python dentro de Mininet, el cual puede ser configurado a nuestra conveniencia.

Para acceder al archivo Python en Mininet es necesario acceder al directorio: “/Mininet/custom” en donde se encuentra el archivo “topo-2sw-2*host*.py” de Python que es un archivo de ejemplo que viene por defecto en Mininet y que será editado para simular la red necesitada:

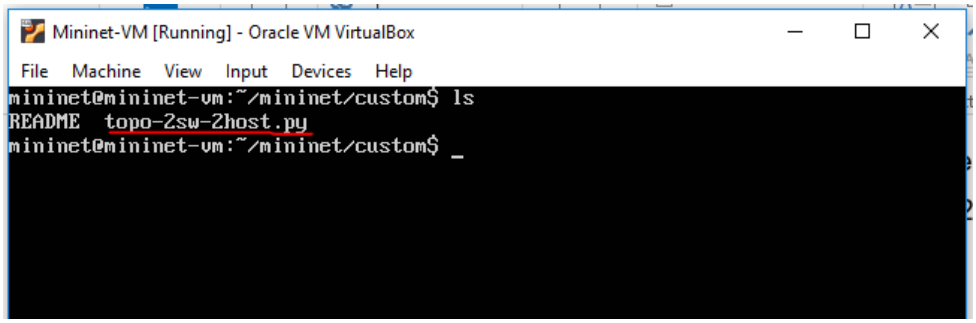


Figura 36. Archivo Python para creación de SDN customizada.

Originalmente este archivo demuestra de manera básica como se crea una red a partir de comandos Python, y crea una red con dos *switches* Openflow conectados entre sí y 2 *hosts* conectados a cada uno de los *switches* respectivamente:

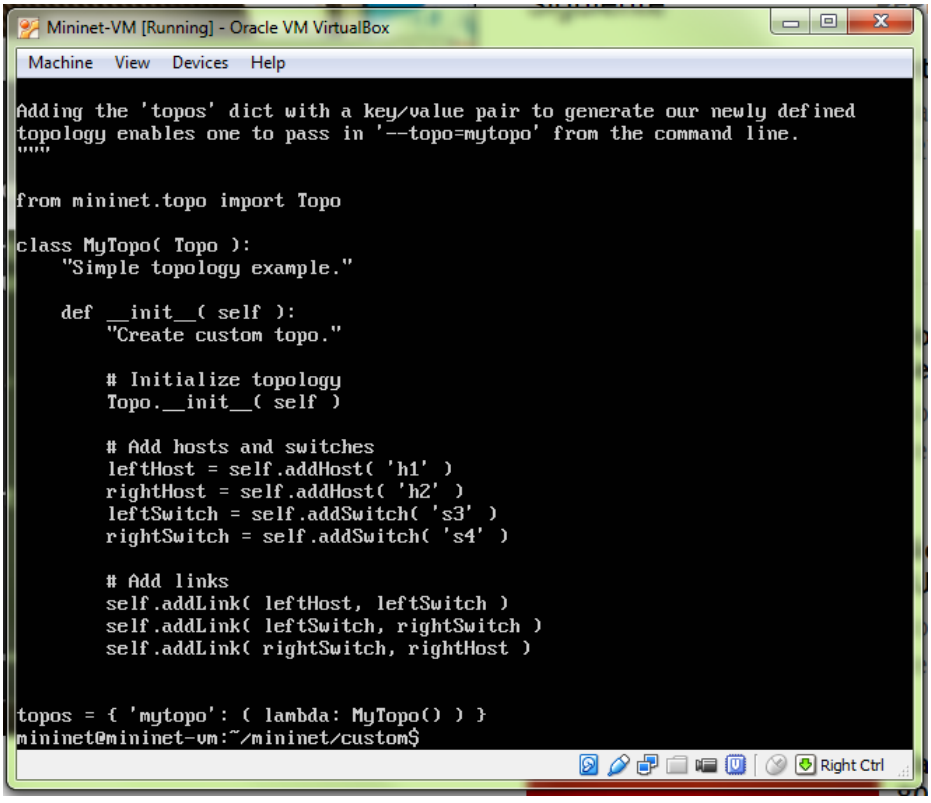


Figura 37. Archivo Python con red básica.

En la figura 37 se puede apreciar el código Python utilizado para crear una red SDN con 2 *hosts* y 2 *switches*. Se utiliza la clase Python “MyTopo” para crear topologías personalizadas, luego se añaden los *hosts* y *switches*, y luego se definen los enlaces entre los mismos.

Para nuestra simulación se editó el archivo Python agregando 20 *Switches* OpenFlow y 40 *Hosts*. Luego de ello se agregaron los enlaces por cada *switch*, conectándose los *switches* entre si y a sus 2 *hosts* (ver figuras 38 y 39):

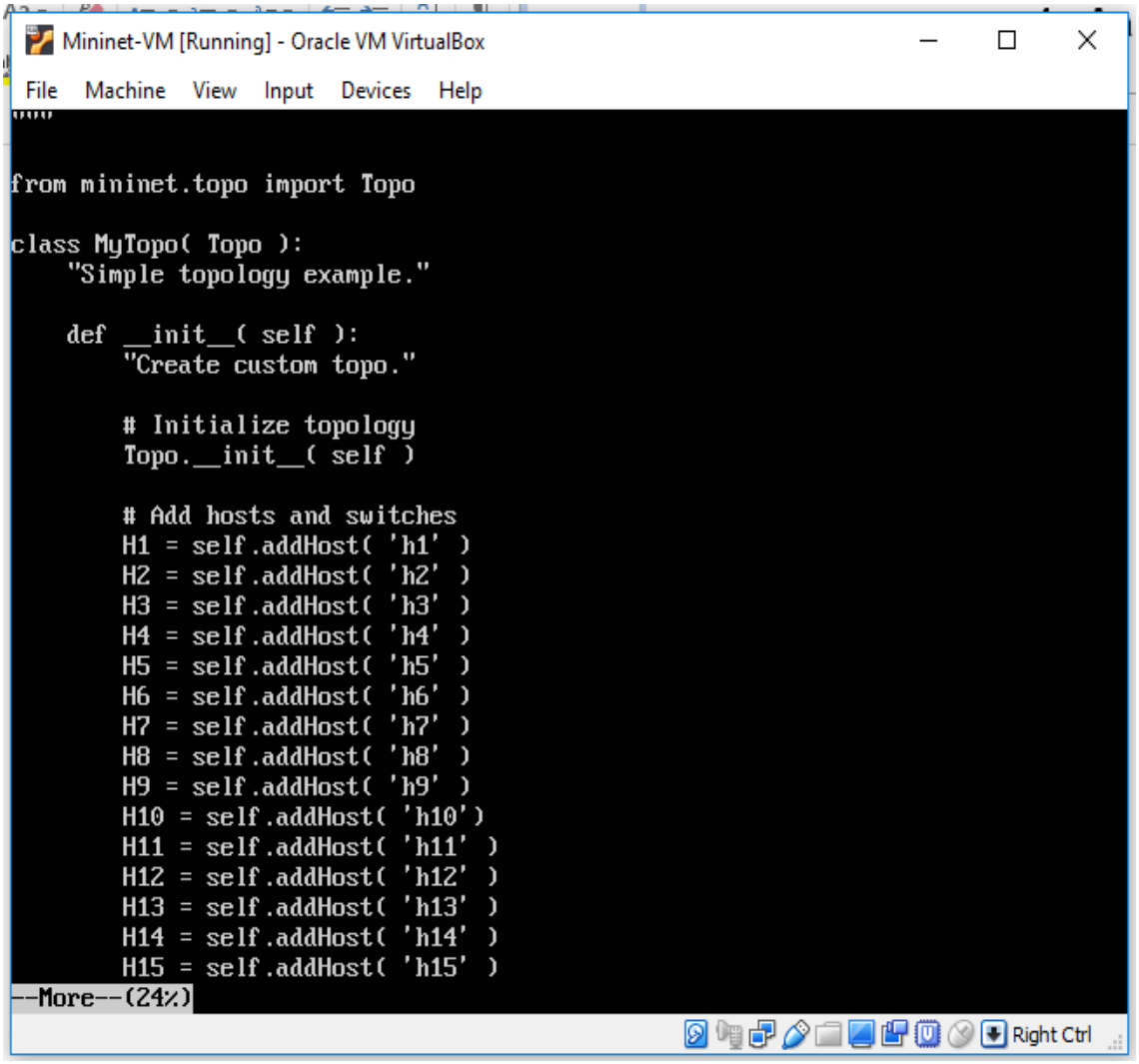


Figura 38. Creación de *hosts* y *switches*.

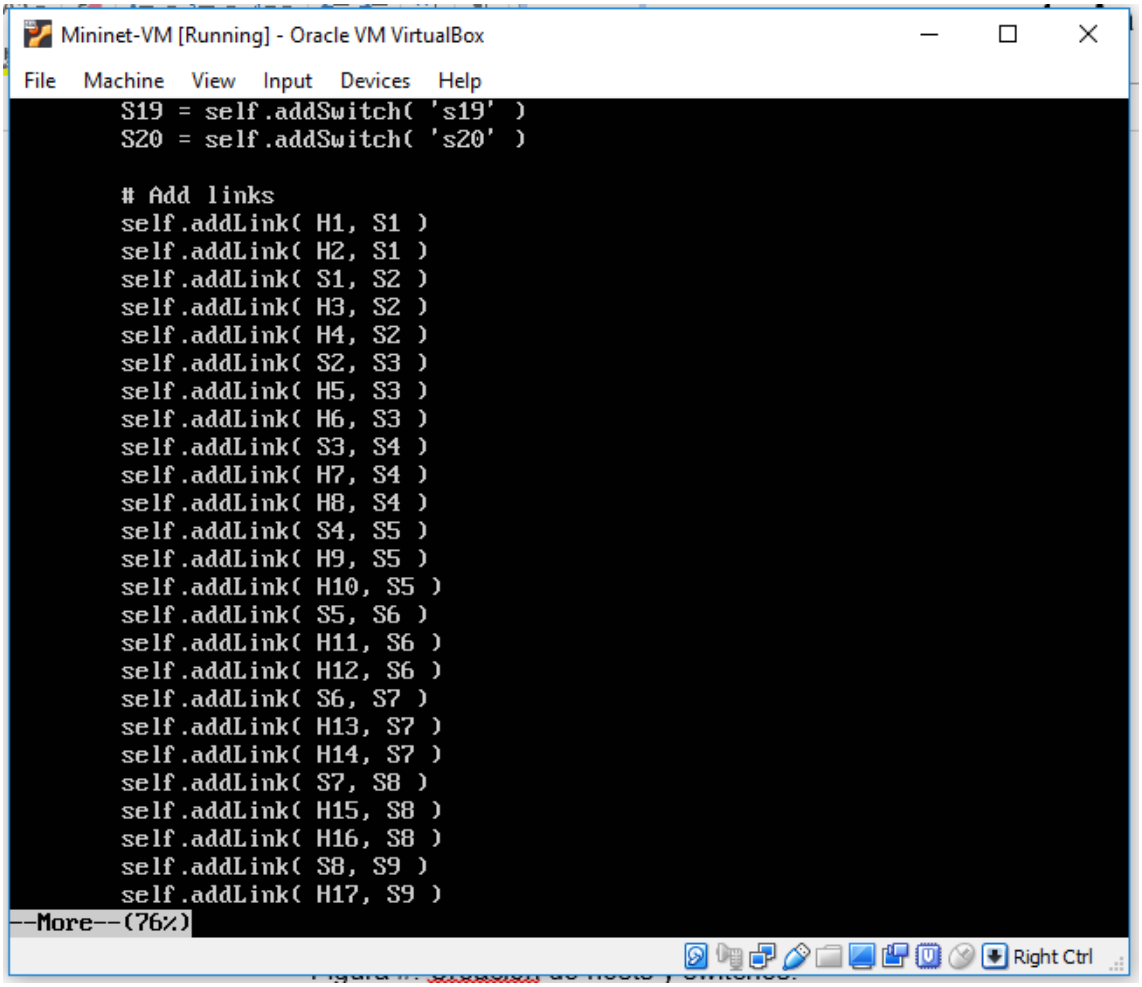


Figura 39. Creación de enlaces.

Una vez editado el archivo, se procede a guardarlo digitando “Ctrl”+X, luego “Y” para confirmar.

Después que el archivo ha sido guardado, se ejecutó definiendo el archivo personalizado para la red, y definiendo la dirección IP de nuestro controlador OpenDaylight a través del comando: *“sudo mn --custom topo-2sw-2host.py --topo mytopo --controller=remote,ip=192.168.0.51”*

Al ejecutar el comando se crea la red SDN deseada. Luego se envía tráfico a través de la misma para que el controlador identifique en donde se encuentran los *hosts* utilizando el comando *“pingall”* y despues se visualiza la red desde la interfaz DLUX de OpenDaylight quedando la red de la siguiente manera:

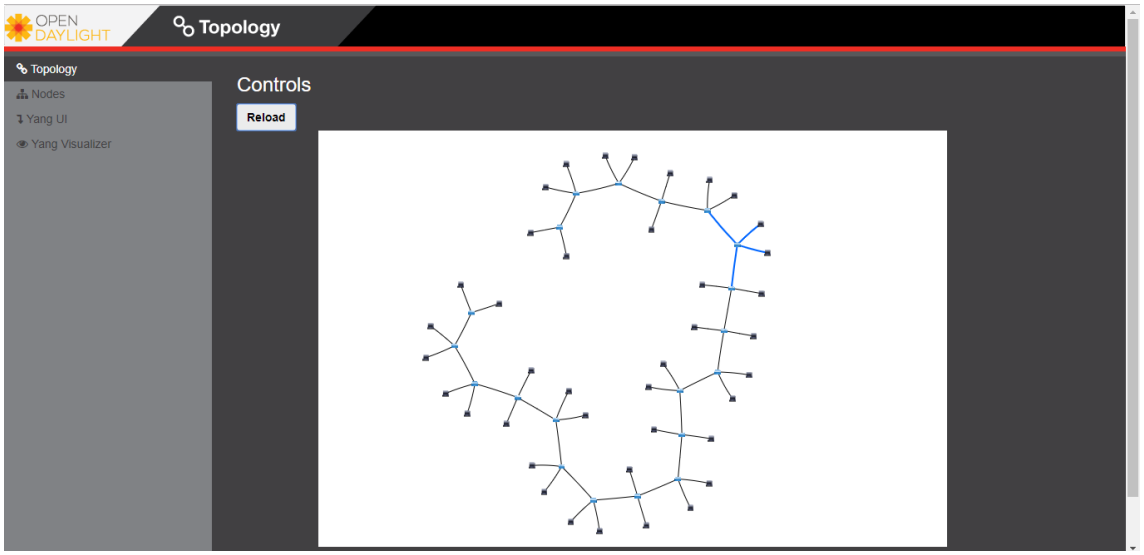
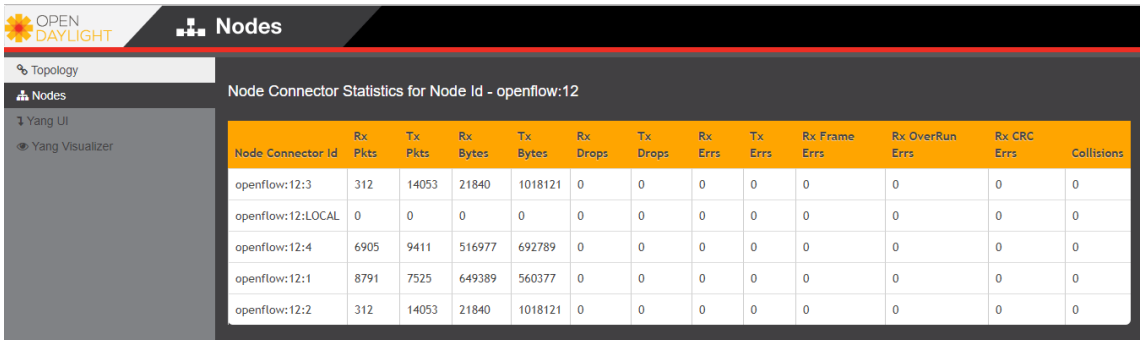


Figura 40. SDN personalizada.

Una vez creada la red (ver figura 40) es necesario enviar tráfico por la misma para que el controlador pueda reconocer en donde se encuentran los dispositivos, utilizando el comando *"pingall":*

Al enviar tráfico por la red, se puede visualizar por dónde se envían los paquetes, a dónde se envían, su cantidad e incluso si existen errores en su envío, accediendo en *OpenDaylight* al menú "Nodos":



Node Connector Id	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisions
openflow:12:3	312	14053	21840	1018121	0	0	0	0	0	0	0	0
openflow:12:LOCAL	0	0	0	0	0	0	0	0	0	0	0	0
openflow:12:4	6905	9411	516977	692789	0	0	0	0	0	0	0	0
openflow:12:1	8791	7525	649389	560377	0	0	0	0	0	0	0	0
openflow:12:2	312	14053	21840	1018121	0	0	0	0	0	0	0	0

Figura 41. Switch OpenFlow 12

En la imagen se muestran específicamente las interfaces conectadas en el *switch* OpenFlow 12. Se Cuenta con cuatro interfaces activas y una interfaz de administración local (LOCAL) por la cual no pasa tráfico. Se puede apreciar cuantos paquetes han sido enviados por cada interfaz, cuantos recibidos y su cantidad en bites.

Para manejar las tablas de flujos con las cuales se controlaran los paquetes y el tráfico en general, es necesario instalar la API *Northbound* con la que se comunicara el controlador y la cual permitirá administrar los flujos de tráfico en la red. Esta aplicación es Cisco OpenFlow Manager (OFM). Debido a esto, se instalaron los componentes restconf en el controlador, porque es la manera en la que se comunican el controlador y la API *Northbound* OFM.

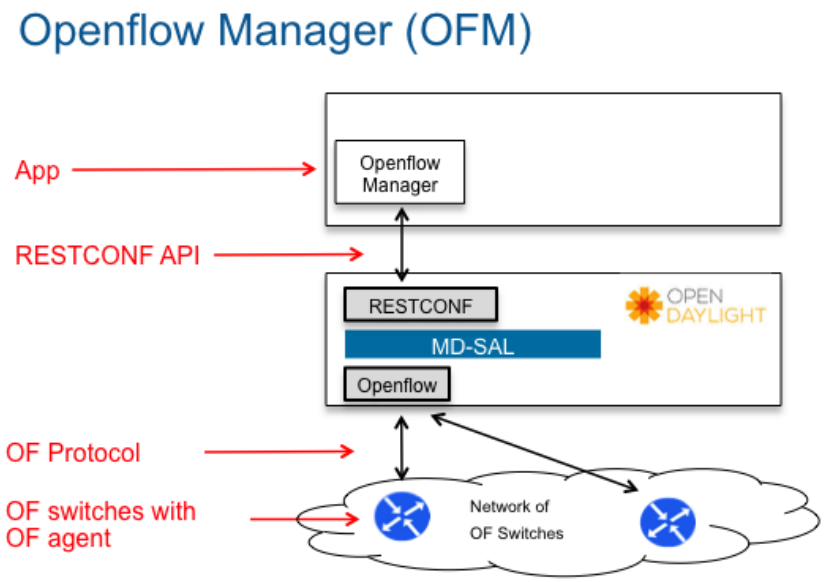


Figura 42. Arquitectura OFM. Tomado de
Tomado de: (GitHub, 2018)

Esta API fue instalada dentro del mismo sistema operativo en el cual se instaló el controlador OpenDaylight. A su vez, para que exista una comunicación entre esta API y el controlador hubo que realizar varias configuraciones. Mientras el controlador estaba levantado, se inició una sesión utilizando la herramienta Putty para conectarse a Ubuntu:

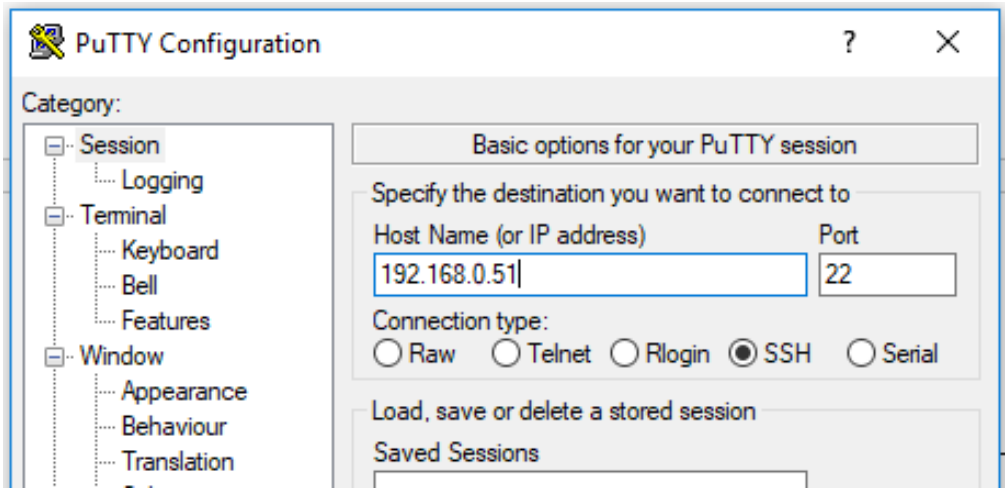


Figura 43. Sesión de Putty al controlador vía SSH

Después, se instaló la herramienta nmap dentro de Ubuntu para conocer que puertos estaban abiertos:

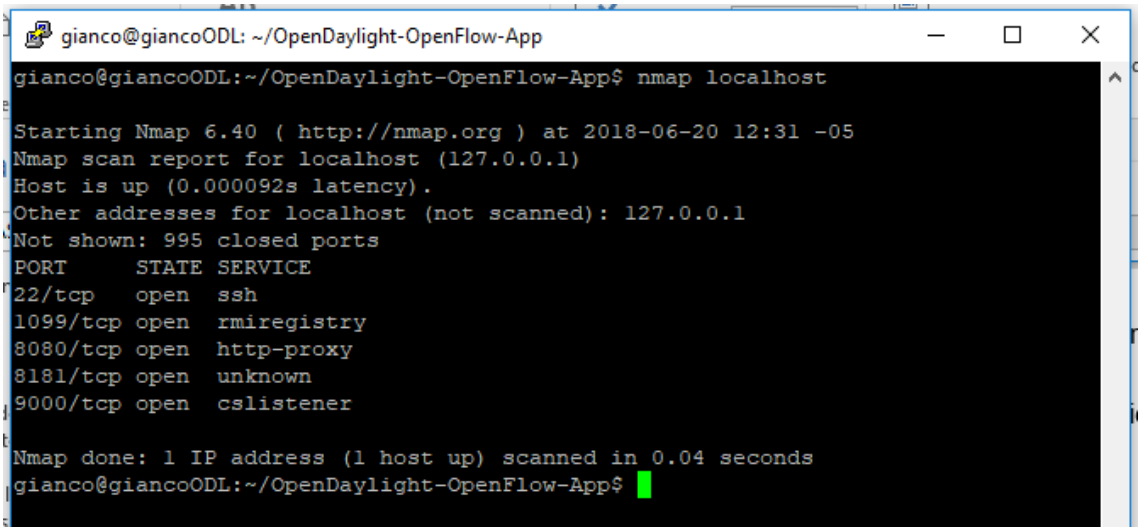


Figura 44. Puertos abiertos Ubuntu.

Luego se instalaron dentro de Ubuntu componentes previos necesarios para el funcionamiento de OFM:

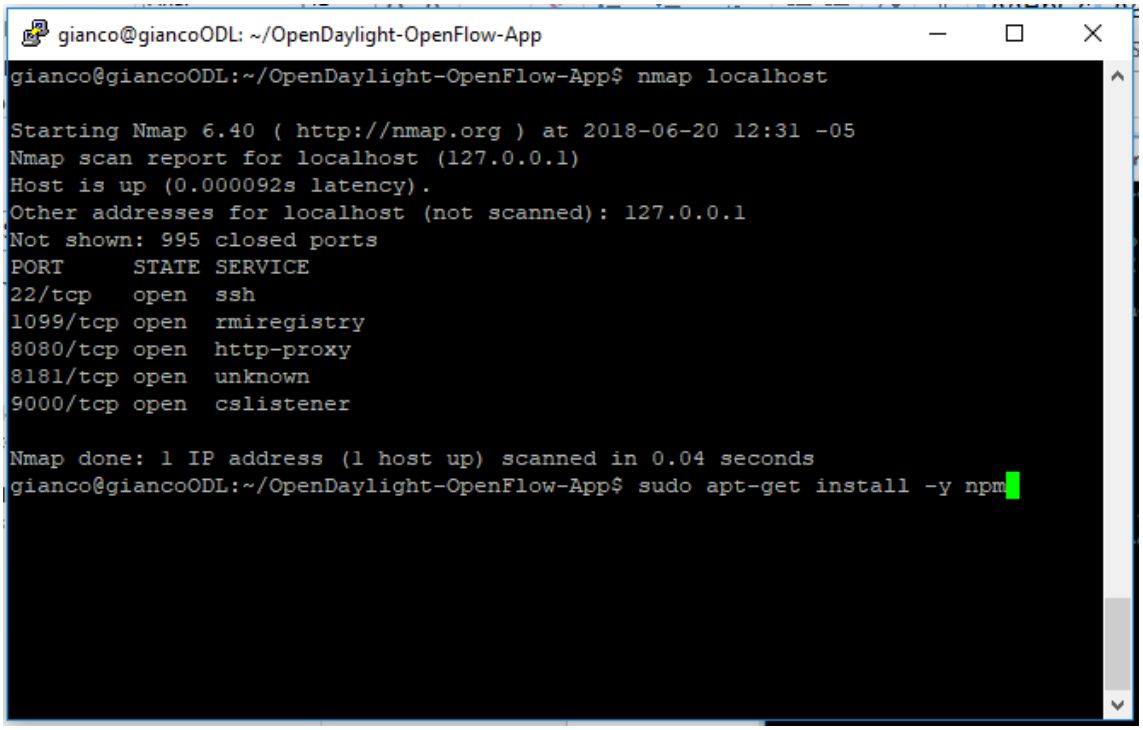


Figura 45. Instalacion “npm”

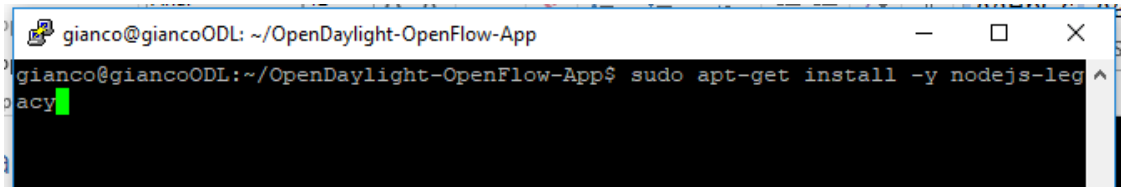


Figura 46. Instalación de Nodejs

Se descargó desde GitHub a través de Ubuntu la API OpenFlow Manager de cisco a través del comando *“git clone http://github.com/CiscoDevNet/OpenDaylight-OpenFLow-App.git”*

Una vez descargada la API, fue necesario ingresar a su archivo “env.module.js” para definir como se conectaría la API al controlador detallando la IP de nuestro controlador SDN:

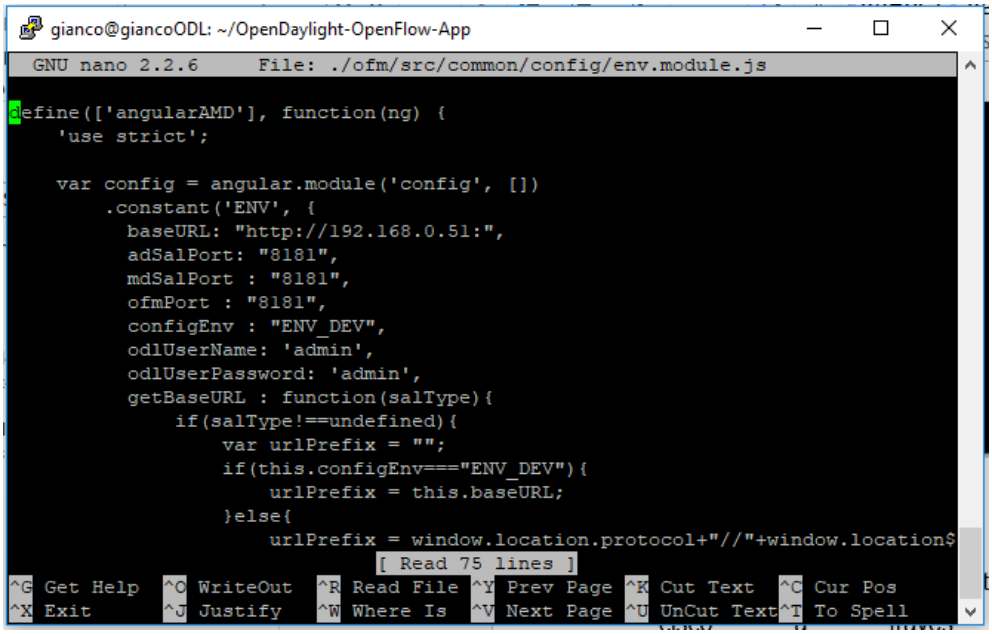


Figura 47. Archivo env.modules.js

Como se aprecia en la figura anterior, se detalló la dirección IP de nuestro controlador en la propiedad "*baseURL*".

Por último se instaló un servidor web grunt desde donde se inicializa la API OFM a través del comando: *"sudo npm install –g grunt-cli"* y se arrancó el mismo a través del comando: *"sudo grunt"*

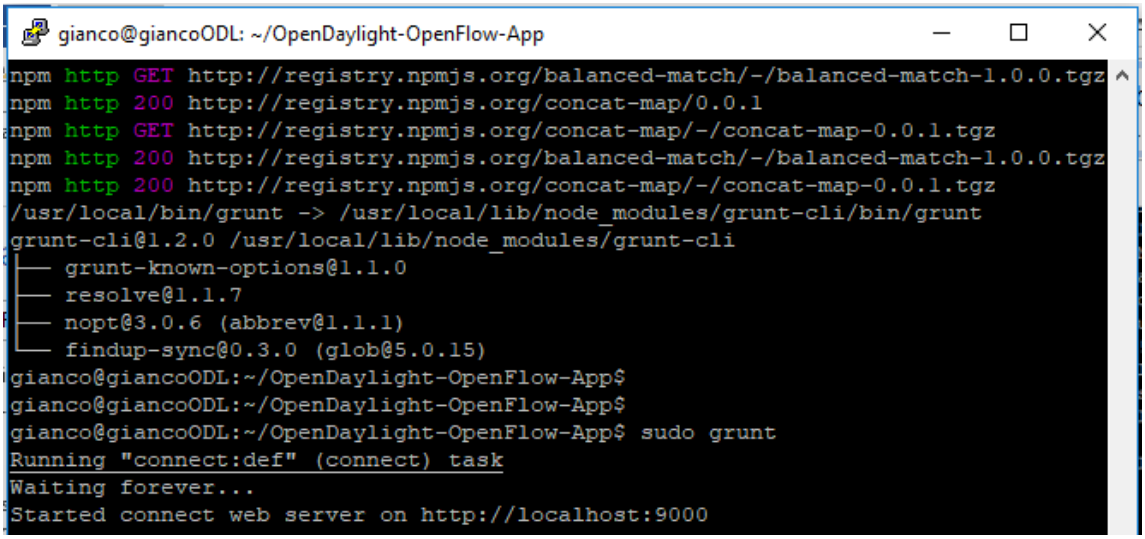


Figura 48. Servidor Web inicializado.

Una vez realizadas estas configuraciones se pudo acceder a la interfaz gráfica de la API OFM a través del navegador conectándose a la IP de Ubuntu y a su vez, del controlador a través del puerto 9000 del servidor *grunt* con la dirección: 192.168.0.51:9000.

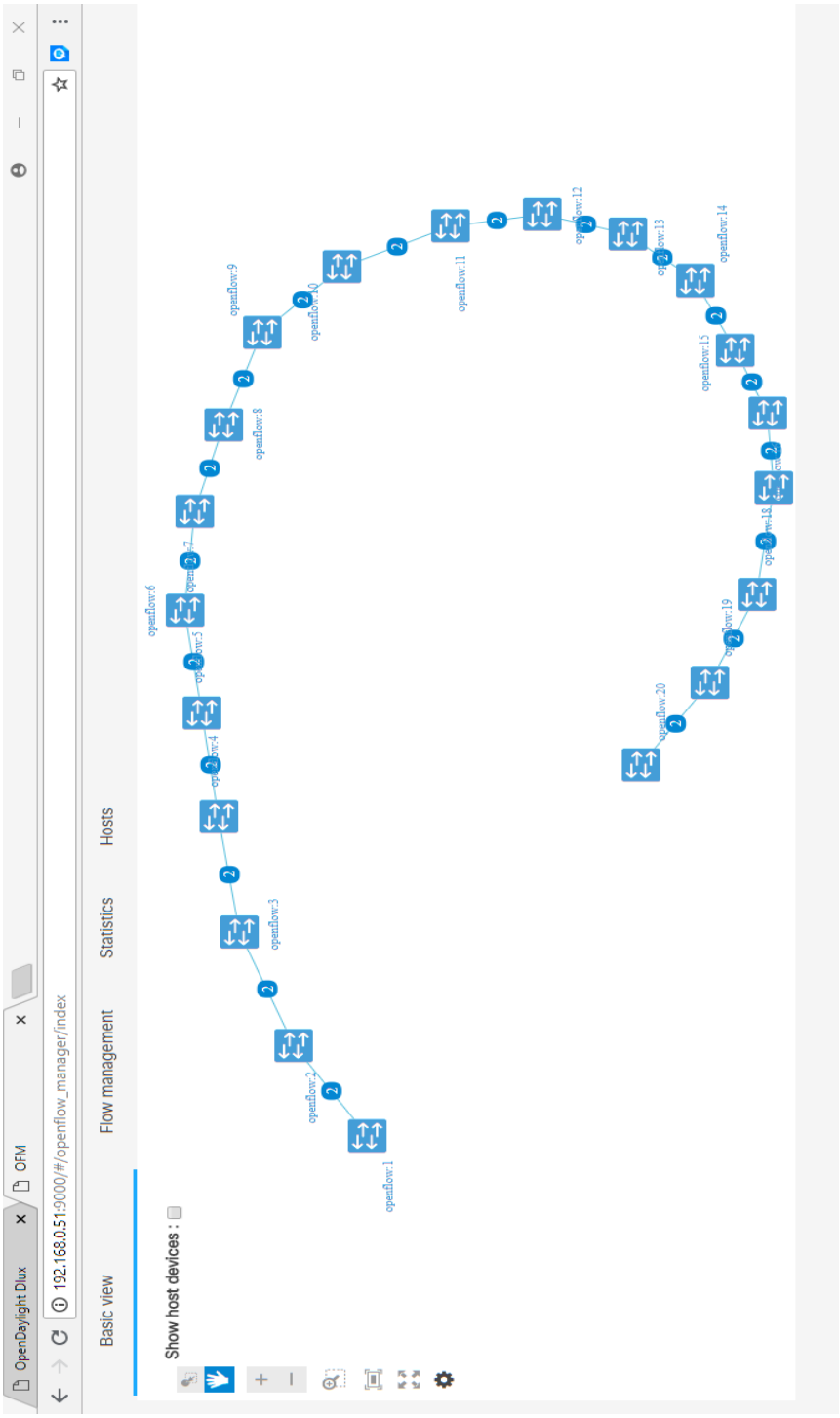


Figura 49. OFM inicializado.

Como se aprecia en la figura 49, se inicializó la API OFM y se muestra la red creada anteriormente en Mininet, de esta manera se integran como la arquitectura SDN indica, la API *Northbound* OFM con el controlador SDN, y este a su vez con la interfaz *Southbound* que es Mininet.

Dentro de esta API es en donde se definen específicamente las tablas de flujo, que pueden ser visualizadas también desde el controlador SDN como se mostró en figuras anteriores.

Dentro de OFM se pueden configurar tablas de flujo para cada uno de los *switches* presentes en la red, en este caso 20. Dando click a un *switch* y accediendo a la pestaña "*Flow Management*":

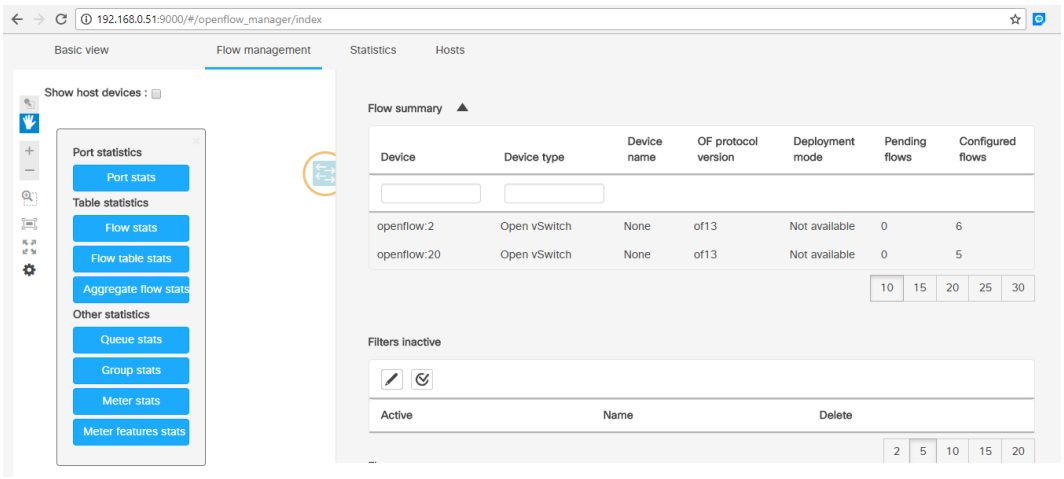


Figura 50. Flow Management Switch 1

Como se muestra en la figura 50, esa es la administración de los flujos para el *switch* numero uno que está conectado en un extremo de la topología, por lo cual se refleja en "*Flow Summary*" las dos conexiones que tiene a otros dos *switches*.

También se verifica que los dispositivos de la simulación trabajan con la versión 1.3 del protocolo OpenFlow.

Bajando más en la configuración, se pueden visualizar las tablas de flujo presentes en este *switch* (ver figura 51):

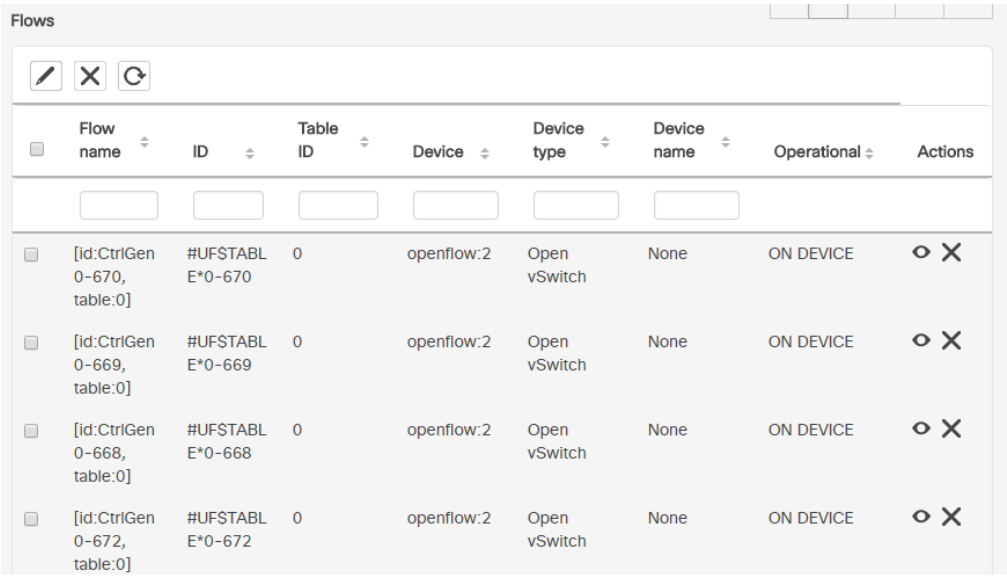


Figura 51. Flujos en *Switch* 1

Tomando el primer flujo presente en el *switch* se puede ver mucha información, como el dispositivo, su tipo, si esta operacional o no, y por último en "*Actions*" se tiene la opción de ver las configuraciones de la tabla de flujos:

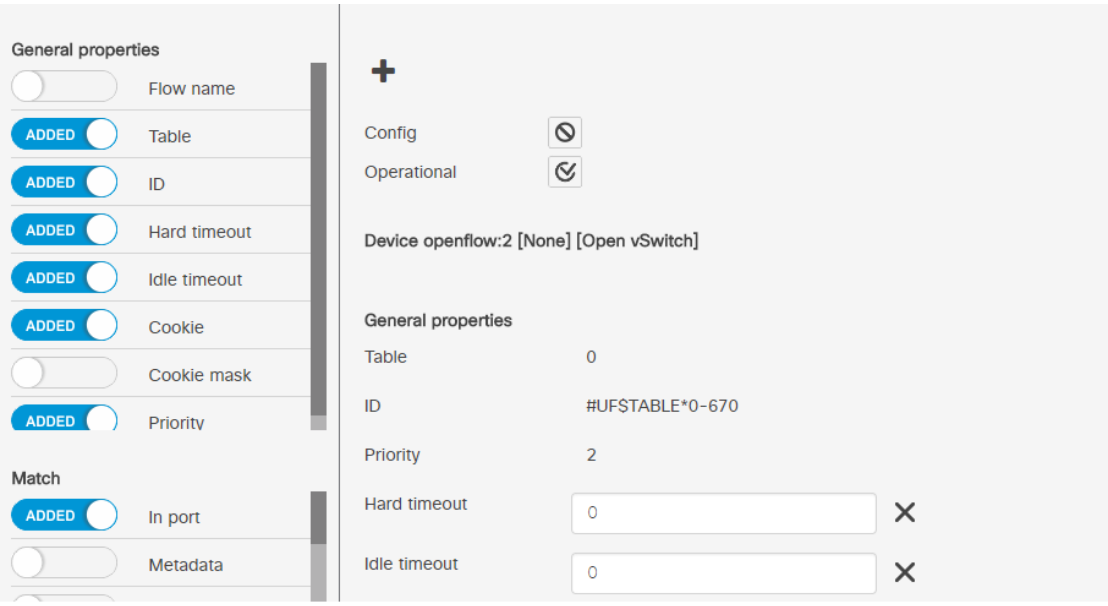


Figura 52. Propiedades del Flujo *Switch* 1

Como se aprecia en la figura 52, existen muchas configuraciones que se pueden realizar en cada flujo, verificando así el control granular que se tiene sobre la red.

El protocolo OpenFlow y las tablas de flujo funcionan básicamente con los *switches* de manera que en el flujo se define qué acciones debe realizar el *switch* si es que el tráfico cumple con la condición especificada en el mismo. En el lado izquierdo de la figura se puede ver las propiedades de “Match”. Estas son las condiciones que se definen en el flujo es decir, se detalla que características del flujo se quieren identificar, y cuando alguna porción de tráfico cumpla estas condiciones o “haga match” se realizan las acciones especificadas.

En la parte de “Actions” se define que se debe hacer con el tráfico una vez que se cumplan las condiciones de “Match”. De esta manera se puede redirigir el tráfico al controlador, descartarlo, enviarlo por uno o varios puertos específicos y muchas opciones más. Esto significaría una gran mejora en la seguridad de la red ya que por ejemplo se puede reconocer tráfico malicioso en la red y simplemente descartarlo. A continuación se muestran opciones de “Actions” disponibles en la API OFM:

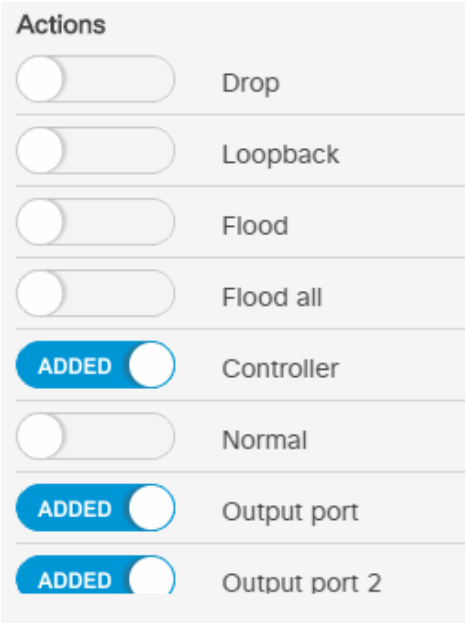


Figura 53. Acciones en tabla de flujo

Se puede concluir y queda demostrado con la simulación propuesta, que el control de la red seria centralizado ya que todo se controla desde la API y se puede visualizar en el controlador. OFM es solo una de las muchas APIs disponibles para redes SDN, sin embargo es una muy poderos ya que define flujos para cada uno de los *switches* conectados en la red. Se tendría una gran escalabilidad al poder crear más y más tablas de flujos de acuerdo a las necesidades del negocio y se asegura una seguridad robusta.

Es necesario reconocer que el único punto débil de este enfoque es a su vez la centralización del control, ya que, si se ataca al controlador se caería toda la red, por lo cual se recomienda tomar precauciones debidas en cuanto a donde se ubica el controlador y que políticas de seguridad se implementan con el mismo.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Se puede concluir que la tecnología SDN es una tecnología creciente que promete ser algo grande y utilizado por una gran diversidad de tipos de instituciones con diferentes requerimientos de red.

Se verifica que el uso de este tipo de tecnología en una institución como el Complejo judicial puede ser de mucha ayuda debido a la granularidad con la que se pueden controlar los datos, automatizando procesos para hacer más simple la administración de la misma y a su vez escalable para el despliegue de nuevos servicios con la capa de aplicación de la infraestructura SDN.

Con la implementación de una red SDN para el Complejo judicial, se puede filtrar el tráfico y definir tablas de flujo que automaticen procesos, como por ejemplo, hacer que cierto tipo de tráfico sea descartado, o que tráfico utilizando ciertos protocolos sea redirigido al controlador para su gestión, o a su vez a cualquier otro punto de la red.
Se puede evidenciar los beneficios que trae consigo una implementación SDN debido a su simplicidad y escalabilidad en todo sentido. Esta tecnología creciente y de *software* libre promete desarrollar APIs que estén enfocadas a la seguridad de una red judicial como la estudiada en este trabajo.

Para la implementación de una red SDN en el complejo judicial se debe conseguir *hardware* y *software* con suficiente aprovisionamiento en cuanto a prestaciones para su correcto funcionamiento.

6.2. Recomendaciones

Se recomienda implementar las características de *EtherChannel* y *Port Channel* para aprovechar de esa manera el ancho de banda y las prestaciones

que ofrece la fibra óptica, ya que al tenerse una fibra por cada dispositivo no se optimiza cada Fibra óptica a su mayor capacidad.

A su vez se recomienda, utilizando SDN filtrar el tipo de tráfico para usuarios específicos desde el controlador SDN debido a que es una funcionalidad muy útil en la utilización de SDN y también porque sería de mucha utilidad para enviar información sensible a usuarios previamente seleccionados.
Se recomienda utilizar para su implementación el controlador OpenDaylight debido a que al ser un proyecto de código abierto, se están implementando nuevas y mejores características a la plataforma OpenDaylight añadiendo más características que pueden ser de mucha utilidad en la administración de una red de gran tamaño.

Se recomienda también utilizar maquinas distintas para el despliegue del controlador SDN y la API para el control de los flujos. En este trabajo se utilizaron OpenDaylight y OpenFlow Manager de Cisco ya que interactúan muy bien al ser contemporáneos.

REFERENCIAS

Albán Ruiz, P. F., & Brito López, D. X. (2015). *En las SDN la manera estándar de emplearlas es con un solo controlador que administra la actuación de los conmutadores en una red. Así se percibe que toda la red es una sola unidad, ya que el controlador precisa la conducta de la red como un todo. Esta p.* Sangolquí: ESPE.

Axioma. (s.f). *Cableado Horizontal*. Recuperado el 20 de Junio de 2018, de http://www.axioma.co.cr/cableado_horizontal.html

Cableado Estructurado. (s.f). *Cableado Estructurado.* Recuperado el 4 de Junio de 2018, de https://ldc.usb.ve/~rgonzalez/telematica/CableadoEstructurado.pdf

Cisco. (23 de Marzo de 2017). *Cisco Catalyst 4500 Series Switch Data Sheet.* Recuperado el 6 de Junio de 2018, de https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-4500-series-switches/product_data_sheet0900aecd801792b1.html

Cisco. (s.f). *Configuring Port Channels.* Recuperado el 6 de Junio de 2018, de https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus1000/sw/4_0_4_s_v_1_3/interface/configuration/guide/n1000v_if/n1000v_if_5portchannel.pdf

Cisco. (s.f). *Redes definidas por software (SDN)*. Recuperado el 7 de Junio de 2018, de https://www.cisco.com/c/es_ec/solutions/software-defined-networking/overview.html

Cisco. (s.f). *Switches Cisco Catalyst serie 2960-X.* Recuperado el 6 de Junio de 2018, de https://www.cisco.com/c/dam/global/es_mx/assets/ofertas/catalyst/pdfs/switches_cisco_catalyst_serie_2960_x.pdf

Espinosa, C. S. (2016). *Protocolos Multicast en redes SDN.* Granada - España: Departamento de Teoría de la Señal, telemática y comunicaciones.

Fiscó, D. A. (2014). *IMPLEMENTACIÓN DE UN MÓDULO DE COMUNICACIONES.* Bogotá. Recuperado el 05 de Junio de 2018

GitHub. (2018). *OpenDaylight OpenFlow Manager (OFM) App*. Retrieved Junio 20, 2018, from https://github.com/CiscoDevNet/OpenDayLight-OpenFlow-App

Jiménez, J. A. (2017). *Planificación y Administración de Redes.* Recuperado el 5 de Junio de 2017, de http://planificacionadministracionredes.readthedocs.io/es/latest/Tema04/Teoria.html

Kaur, K., Singh, J., & Singh Ghumman, N. (2014). *Mininet as Software Defined Networking.* Ferozepur: Shaheed Bhagat Singh State Technical Campus.

Logicalis. (Junio de 2014). *SDN Cómo el nuevo universo trazado por las redes definidas por software impactará en los negocios.* Recuperado el 7 de Junio de 2018, de https://www.la.logicalis.com/globalassets/latin-america/advisors/es/advisor_sdn.pdf

M., J. V. (2017). *Introducción a las Redes.* Recuperado el 5 de Junio de 2018, de https://www.scribd.com/document/174948862/Introduccion-a-las-Redes

Parra Loera, R., Morales Rocha, V. M., & Hernández Hernández, J. I. (Noviembre de 2015). *Redes Definidas por Software: beneficios y riesgos de su implementación en.* Recuperado el 7 de Junio de 2018, de https://www.conaic.net/revista/publicaciones/ANIEI2015Articulo%208.pdf

Redes Telemáticas . (2012). *Tipos de switches*. Recuperado el 6 de Junio de 2018, de http://redestelematicas.com/tipos-de-switches/

SDX Central. (s.f). *What is OpenFlow? Definition and How it Relates to SDN*. Recuperado el 6 de Junio de 2018, de SDX Central: https://www.sdxcentral.com/sdn/definitions/what-is-openflow/

Seguridad en las redes definidas por Software (SDN). (Diciembre de 2017). *Seguridad en las redes definidas por Software (SDN)*. Recuperado el 7 de Junio de 2018, de https://ciberseguridad.blog/seguridad-en-las-redes-definidas-por-software-sdn/

Serrano, D. A. (2015). *Redes Definidas por Software (SDN): OpenFlow.* Valencia: Universidad Politécnica de Valencia.

Spera, C. (Marzo de 2013). *Software Defined Network:.* Recuperado el 07 de Junio de 2018, de https://www.la.logicalis.com/globalassets/latin-america/logicalisnow/revista-20/lnow20-nota-42-45.pdf

Switch Stacking. (2013). Recuperado el 06 de Junio de 2018, de https://networkingcontrol.wordpress.com/2013/08/18/switch-stacked/

Switches de acceso. (2008). Recuperado el 6 de Junio de 2018, de http://stweeker.blogspot.com/2008/03/switches-de-acceso.html

Tejedor, R. J. (2014). *SDN: el futuro de las redes inteligentes.* Conectrónica.

Universidad de Buenos Aires. (s.f). *Cableado Estructurado.* Recuperado el 5 de Junio de 2018, de http://materias.fi.uba.ar/6679/apuntes/CABLEADO_ESTRUC.pdf

Valencia, B., Santacruz, S., Becerra, L., & Padilla, J. (2015). *Mininet: una herramienta versátil para emulación y prototipado de Redes Definidas por Software.* Recuperado el 7 de Junio de 2018, de http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1909-83672015000100009

Wheelers Lane Technology College. (s.f). *Power-over-Ethernet (PoE).* Recuperado el 5 de Junio de 2018, de http://bibing.us.es/proyectos/abreproy/11579/fichero/h.+Cap%C3%ADtulo+4+-+Power+over+Ethernet.pdf+

ANEXOS

ANEXO 1: SOLICITUD DE INFORMACION A LA DIRECCIÓN PROVINCIAL DE PICHINCHA DEL CONSEJO DE LA JUDICATURA

Ingeniería en Electrónica y Redes de Información

Quito, 14 de noviembre del 2017

Abogado
Santiago Páez
Director Provincial del Consejo de la Judicatura de Pichincha.

Presente

Reciba un cordial saludo, por medio de la presente me permito informarle que dentro de las carreras que oferta la Universidad de las Américas se encuentra Ingeniera Electrónica y Redes de la Información, del cual soy egresado y que me encuentro realizando mi proyecto de titulación el cual consiste en una propuesta de diseño, y comparación de tecnologías de red.

En razón de que es de público conocimiento que el Complejo Judicial Norte presenta tecnología de punta y sistemas de innovación, le solicito comedidamente, se me brinde información básica de la red actual y su infraestructura para realizar el diseño utilizando tecnologías SDN de la red y luego realizar una comparativa de la misma contra el diseño de red actual.

Debido a lo delicado de la información que el Consejo de la Judicatura administra no se realizará ninguna implementación en sitio, adicionalmente que la información proporcionada será utilizada para fines académicos de acuerdo a mi propuesta del proyecto de titulación, además que en el desarrollo de este proyecto me encuentro supervisado y asesorado por mi profesor guía Ing. Iván Ortiz, profesor de la carrera.

Le reitero mis sentimientos de gratitud

36617

VENTANILLA DE RECEPCIÓN DE DOCUMENTOS
FECHA: 14 NOV 2017
HORA: 11:00
DIRECCIÓN PROVINCIAL DE PICHINCHA
ANEXOS:

Giancarlo Ianni
CI: 1715143697

Ing. Iván Ortiz
Tutor

Contacto: Giancarlo.ianni@udla.edu.ec

ANEXO 2: OFICIO DE ACEPTACION DE SOLICITUD DE INFORMACION POR PARTE DEL COMPLEJO JUDICIAL

Código descarga documento firmado electrónicamente.

Oficio-DP17-2017-0019-OF **TR: DP17-INT-2017-00029**

QUITO D.M., lunes 27 de noviembre de 2017

Asunto: SOLICITUD DE INFORMACION DE REDES

EGRESADO
Ianni Bermudes Giancarlo

En atención a su solicitud de fecha 14 de noviembre de 2017, adjunto el memorando N° CJ-DNTICS-SNIT-2017-0020-M, con el cual el Ing. Patricio Alvarez Gordón, Subdirector Nacional de Infraestructura de la Dirección Nacional de Tecnologías de la Información y Comunicaciones del Consejo de la Judicatura, remite la información requerida por usted.

Atentamente,

Dr.Esteban Morales Moncayo
Director Provincial del Consejo de la Judicatura de Pichincha, Subrogante
Dirección Provincial de Pichincha

MEMORANDO CJ-DNTICS-SNIT-2017-0020-M DIRECCIONAMIENTO DE RED COMPLEJO JUDICIAL NORTE
ESQUEMA DE RED COMPLEJO JUDICIAL NORTE

DISEÑO DE UNA RED SDN EN LA DIRECCIÓN PROVINCIAL DE PICHINCHA DEL CONSEJO DE LA JUDICATURA

Universidad de las Américas

Memoria Técnica

Quito, 22 de Junio de 2018

Realizado por: Giancarlo Ianni B	Aprobado por: Mgs. Iván Patricio Ortiz Garcés

ÍNDICE

1. **Situación Actual**

La construcción del nuevo complejo para la Dirección Provincial de Pichincha del Consejo de la Judicatura realizada a inicios de enero del 2017 representó para el Complejo una concentración de más funcionarios y equipos en menos localidades, y debido a esto el Complejo Judicial cuenta con una gran cantidad de equipos de red los cuales deben ser configurados individualmente ya que en el caso de los *switches*, no existe apilamiento entre ellos y cada uno de los 4 a 8 *switches* de acceso ubicados en cada una de las 12 plantas del Complejo Judicial dificulta de gran manera la administración de su red. Además, siendo una institución a la que concurren muchas personas diariamente debido a los trámites que en esta se realizan, se tiene una gran congestión en el tráfico de los datos.

Esta institución cuenta con dos *switches* de acceso en un datacenter ubicado en el quinto piso, en donde también existe una gran cantidad de tráfico, el cual debe ser manejado de una manera muy meticulosa debido a las decenas de redes WAN que existen en la red.

También, en los pisos superiores se encuentran las áreas de juzgados y altos mandos, en las cuales la información es confidencial y debe ser tratada delicadamente por un canal seguro y se solo transferida a los usuarios necesarios.

El Complejo Judicial es una gran construcción con 12 pisos con una gran cantidad de usuarios, puntos de red, Access points, *switches* y demás dispositivos que debido a la red actual, deben ser configurados individualmente, esto podría ser optimizado con la propuesta de la utilización de redes SDN para esta institución, debido a que esta creciente tecnología promete suplir todas las necesidades para este tipo de institución judicial.

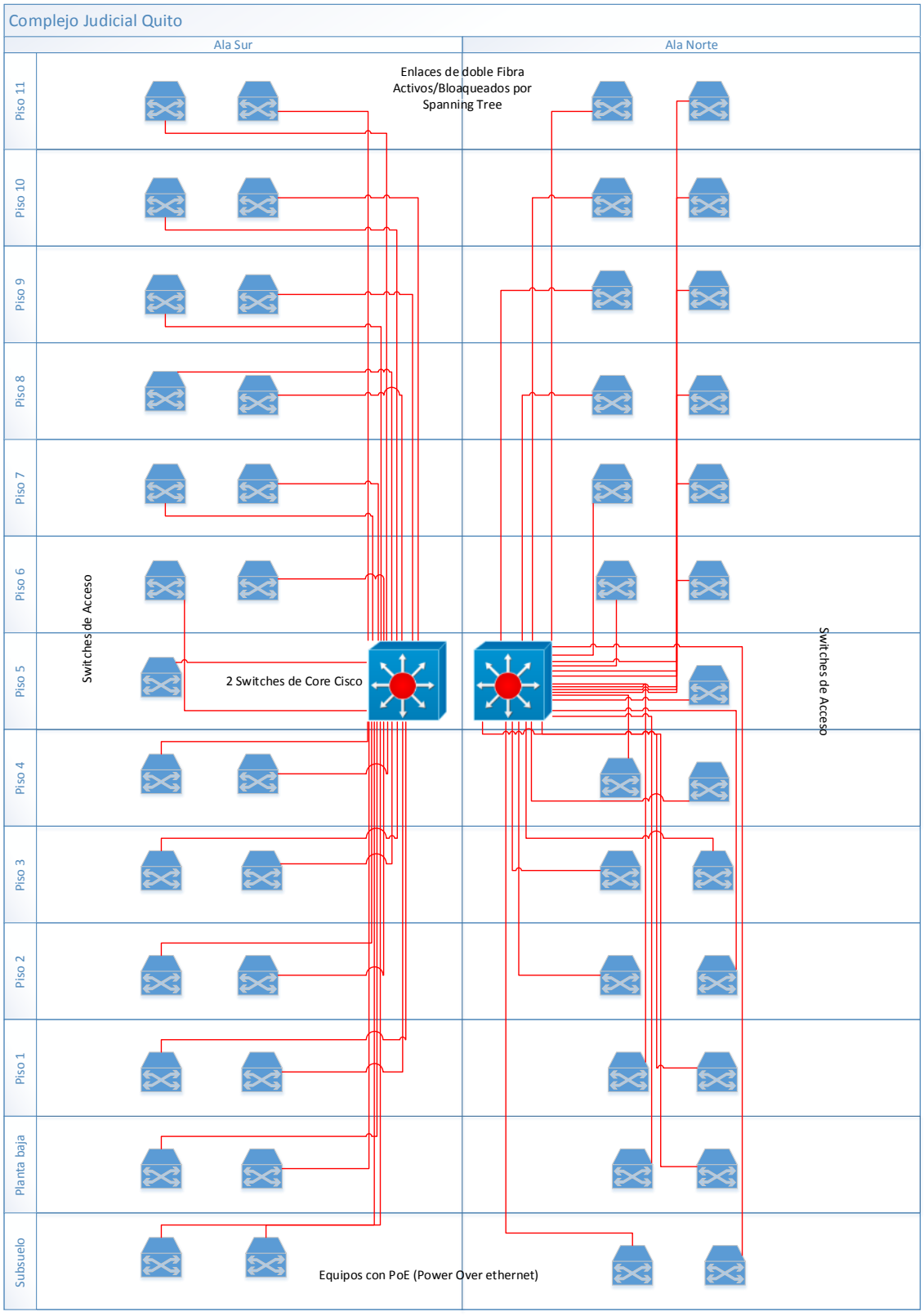


Figura 1. Esquema de red actual Complejo Judicial

Esta situación podría ser facilitada para el administrador de red con un concepto que está surgiendo más y más con el pasar del tiempo, las redes SDN que prometen optimizar los recursos de red y facilitar las administración de la misma.

2. **Marco Teórico**

1.1 **Redes definidas por *Software* (SDN)**

Las redes definidas por *software*, son un nuevo concepto que se basa esencialmente en separar el plano de datos del plano de control dentro de una red, lo cual significa que el funcionamiento de este tipo de red no está ligado al funcionamiento de equipos físicos como routers y *switches*. Sino más bien el control de la red se centraliza y los dispositivos de red como routers o *switches* solo se encargan de ejecutar las decisiones tomadas en el plano de control, es decir, el controlador SDN, parte central de este tipo de redes.

La arquitectura SDN está dispuesta en 3 capas:

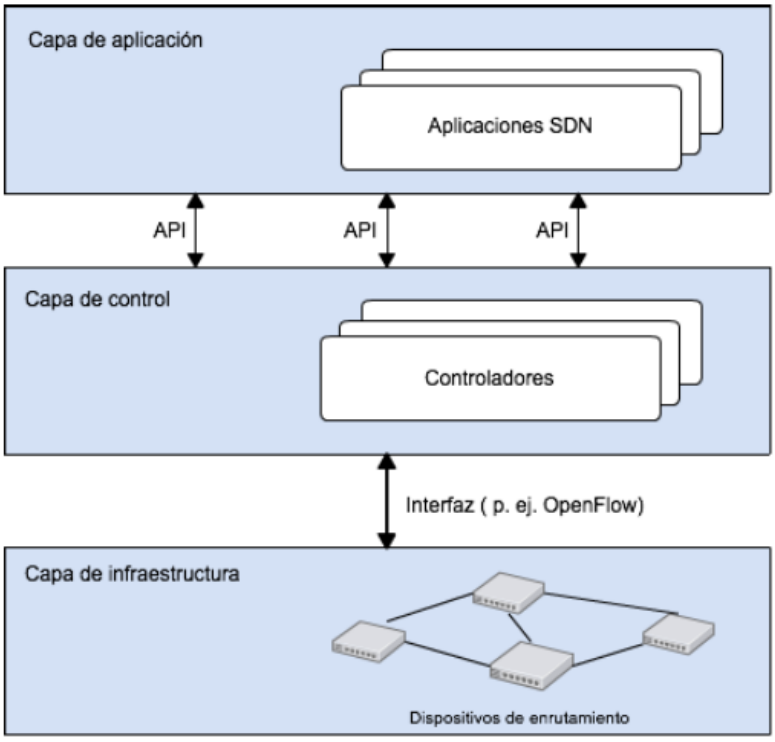


Figura 2. Arquitectura SDN

La capa de aplicación es la capa en la cual se desarrollan aplicaciones SDN conectadas a la capa de control a través de APIs (Northbound Interface) y que permiten satisfacer las necesidades del usuario con ejemplos como: enrutamiento adaptivo, itinerancia sin interrupciones, seguridad de la red, mantenimiento de la red, etc.

Dentro de esta capa es en donde se crean tablas de flujos y demás aplicaciones que permiten controlar a la red filtrando el tráfico dependiendo de su tipo o de muchos otros parámetros configurables en cada una de las distintas APIs que existen en la actualidad.

La capa de control, como se menciona anteriormente, es la más importante de esta arquitectura ya que es el centro de la misma, conectándose con las capas de aplicación e infraestructura a través de las interfaces Northbound y Southbound respectivamente. En ella se encuentra el controlador SDN el cual administra toda la red, gestionando la capa de aplicación y la capa de infraestructura y además monitorizando todos los elementos de la red haciéndola ver como un todo.

El controlador es desde donde se puede visualizar la red, ver estadísticas del tráfico y definir detalles en so estructura interna, dependiendo del controlador SDN que se utilice, en esta propuesta se utiliza el controlador SDN OpenDaylight, el cual es *software* libre y permite controlar la red a un nivel bastante granular.

La capa de Infraestructura consta de los dispositivos *hardware* de conmutación que forman la red y que realizan dos tareas de acuerdo a sus componentes lógicos:

- Control: recoger el estado de toda la red y mantener al controlador informado de la misma, el cual a su vez le dará instrucciones de reenvió de paquetes.

- Datos: el procesador de red reenvía los paquetes en base a las decisiones del plano de control. Cuando se recibe un paquete, el dispositivo identifica la decisión de reenvío que coincide con el paquete a través de tablas de flujo o flow tables.

1.2 **OpenFlow**

OpenFlow es el protocolo de comunicaciones utilizado en redes SDN. Es uno de los primeros estándares con los que trabajan las SDNs. Originalmente definía el protocolo de comunicación en ambientes SDN que habilitaba a un controlador SDN para interactuar directamente con el plano de datos de dispositivos de red como *switches* y routers, física y virtualmente (basado en un Hypervisor) para que se puedan adaptar fácilmente a los requerimientos de un negocio.

Este es el protocolo mediante el cual se comunican la capa de infraestructur, es decir los *switches* y equipos de enrutamiento, y el controlador SDN. Es mediante este protocolo que se aplican las reglas y flujos definidos en capas superiores.

1.3 **Mininet**

Mininet es el simulador utilizado para realizar pruebas con redes SDNs de una manera bastante apegada a la realidad. Se pueden simular grandes redes con gran detalle, enviar tráfico por las mismas y realizar un sinnúmero de pruebas acerca de esta tecnología.

1.4 **OpenDaylight**

OpenDaylight es un proyecto de código abierto que nació a raíz del surgimiento de las SDNs y cuyo objetivo difundir la innovación en el diseño e implementación de un estándar abierto y transparente de Redes Definidas por *Software* (SDN). Este proyecto cuenta con una plataforma modular del mismo

nombre que permite customizar y automatizar redes definidas por *software* de cualquier tamaño, enfocándose en la programabilidad de la misma.

Opendaylight es en si el controlador utilizado en este trabajo debido a su constante desarrollo y al hecho de que nació a raíz de OpenFlow y debido a eso su manejo es bastante sólido y confiable.

3. **Propuesta de Mejora**

Para la propuesta de mejora considerando lo investigado acerca de SDNs contrastado con la información de red otorgada por el Complejo Judicial, se propone realizar varios cambios previos a la utilización de una red SDN, para que después de implementados, la utilización de una red SDN simplifique aún más la administración de la red, y permita un control más personalizado de la información y como esta es distribuida a través de esta gran institución con gran cantidad de usuarios.

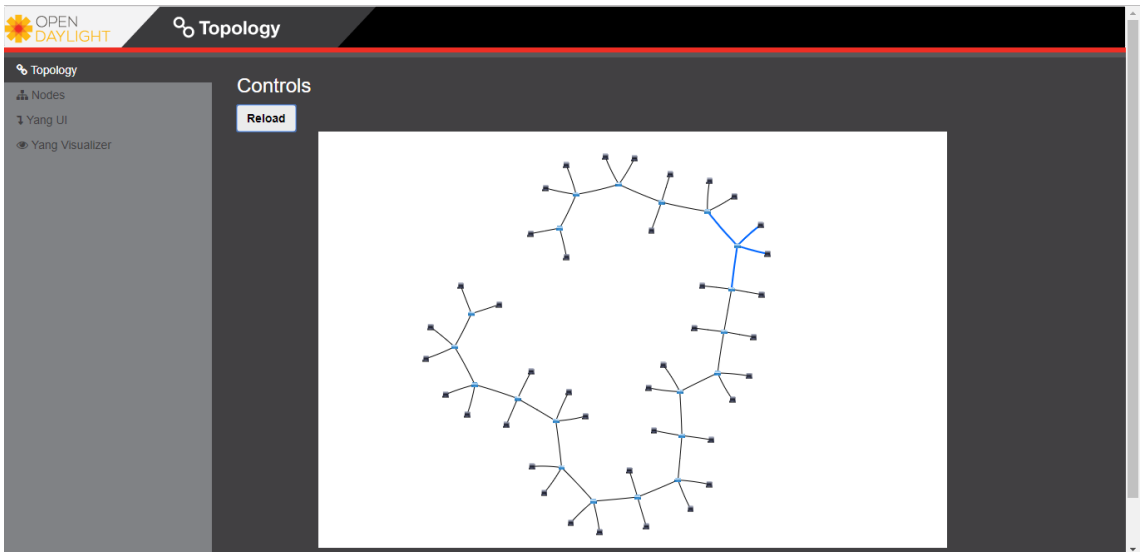


Figura 1. Red SDN – Interfaz gráfica Controlador OpenDayilight

Se propone realizar un apilamiento de *switches* para disminuir su cantidad lógica facilitando su administración. También se propone la implementación de EtherChannel y Portchannel en los *switches* de acceso para optimizar la utilización de fibra óptica dentro de la red ya que actualmente se cuenta con

una fibra óptica por cada *switch* de acceso lo cual puede ser optimizado de gran manera debido a las prestaciones de la fibra óptica..

Las mejoras propuestas anteriormente se verían potenciadas con la adopción de una arquitectura SDN que desligue los planos de control y de datos. Con las prestaciones de la tecnología SDN es posible llevar un control de la red filtrando el tráfico de diferentes maneras, pudiendo manejar la red de una mejor manera.

Se propone implementar una red SDN para el complejo judicial controlada a través del controlador SDN OpenDaylight, instalado en un servidor confiable al cual solo se puede el administrador de la red.

A su vez es necesaria la implementación de un servidor Web para dentro de este, poder realizar el despliegue de la APIs de la capa de aplicación, específicamente se propone la utilización de la aplicación de Cisco OFM (OpenFlow manager).

También se propone la utilización de *switches* que permitan la utilización del protocolo Openflow. A continuación una tabla con varios fabricantes:

Fabricante	**Producto**
Arista	Arista EOS, Arista 7124FX
Ciena	Actualización Core Director firmware 6.1.1 en dispositivos Ciena
Cisco	Cisco cat6k, series catalyst 3750, 6500
Juniper	Juniper MX-240, T-640
HP	HP series procurve: 5400 zl, 8200 zl, 6200 yl, 3500 yl, 6600
Pronto	Pronto 3240, 3290
Toroki	Toroki Light*switch* 4810
Dell	Dell Z900 y S4810

Quanta	Quanta LB4G

Tabla 1. Conmutadores *Hardware* OpenFlow

Gracias a las flow tables o tablas de flujo en el protocolo OpenFlow, utilizado en el controlador SDN OpenDaylight, sería posible, a través de OFM implantar políticas que filtren paquetes, redirijan tráfico, bloqueen tráfico no deseado, etiqueten paquetes, redirijan cierto tipo de tráfico a usuarios específicos.

Se podrían implantar después de la implementación de la SDN, mas aplicaciones que permitan añadir mayor funcionalidades a la red para asegurar su confiabilidad, mejorar su seguridad y facilitar su administración.