Universidad de las Ciencias Informáticas

www.uci.cu

PROPUESTA DE ARQUITECTURA PARA LA GESTIÓN DE REDES DEFINIDAS POR SOFTWARES HÍBRIDAS

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

Autor: Yoel Díaz Herrera
Tutoras: MSc. Mónica Peña Casanova
Ing. Evelyn Yanez Clark

La Habana
2016

A mis padres...

AGRADECIMIENTOS

A mis padres, a quienes debo todo lo que soy. Gracias por mostrarme los valores que deben acompañar a un ser humano y por ser mis ejemplos en la vida.

A mi bella familia, quien siempre me ha dado su cariño y apoyo. Gracias por creer en mí.

A mis tutoras, mis grandes aliadas en este logro. Gracias por motivarme e inspirar en mí el esfuerzo para convertirme en un profesional. Ambas significan mucho para mí.

A todos mis amigos y amigas de la UCI, con los que he tenido el placer de compartir cinco años en donde hay tantas experiencias e historias que contar, y otras que mejor no tanto. Gracias por ayudarme a lograr mis metas y por permitirme ser parte de sus vidas.

A mis amigos y amigas de los años, con los que he compartido tantas cosas. Gracias por estar conmigo en las buenas y en las malas.

A todas aquellas personas que han contribuido de una forma u otra a la culminación de mis estudios y a que pronto me gradúe como ingeniero, a todas las que han estado a mi lado y con las que he compartido inolvidables momentos… GRACIAS.

DECLARACIÓN DE AUTORÍA

Declaro ser autor del presente trabajo de diploma y autorizo a la Universidad de las Ciencias Informáticas (UCI) y al Centro de Identificación y Seguridad Digital (CISED) a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año_____.

Autor: Yoel Díaz Herrera _____

Firma

Tutora: MSc. Mónica Peña Casanova _____

Firma

Tutora: Ing. Evelyn Yanez Clark _____

Firma

RESUMEN

En el contexto actual de las Tecnologías de la Información y las Comunicaciones (TIC) en donde existe una creciente demanda hacia los servicios de información debido a la masificación del uso de dispositivos móviles, la computación en la nube y el aumento del acceso a internet a nivel global, se hace necesario reexaminar la eficiencia de los sistemas comunicación a fin de ajustar estas realidades a las redes de datos existentes. Para ello, las Redes Definidas por Software o SDN (*Software-Defined Networking*, por sus siglas en inglés) constituyen una tecnología que brinda grandes ventajas respecto a las redes tradicionales por las facilidades y potencialidades que brinda en cuanto a agilidad, flexibilidad y adaptación a las nuevas condiciones.

En Cuba comienza a introducirse equipamiento de red que trabaja en el formato convencional y soporta la tecnología SDN; sin embargo, a pesar de las potencialidades que ofrece esta tecnología, no existe una implementación de SDN que explote las funcionalidades de dicho equipamiento, ya que la mayoría de los estudios que se han realizado de estas tecnologías son a nivel académico, de ahí que este trabajo se proyecta, a través de la propuesta de una arquitectura para la gestión de SDN híbrida que emplea la gestión de red basada en políticas, sentar las bases que faciliten la integración de manera ordenada de dicha tecnología en Cuba, así como su coexistencia con las redes tradicionales. Dicha propuesta es evaluada mediante su despliegue en un entorno de trabajo simulado en el que se expone la factibilidad de la solución.

Palabras Claves: Gestión de Red Basada en Políticas, Red Definida por Software, Red Definida por Software Híbrida.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Con el surgimiento de Internet a finales de la década de 1960 y la introducción de nuevas facilidades de interconexión de redes de información y herramientas para su uso, se inició una revolución de las comunicaciones que actualmente experimenta cada día la integración de nuevas redes y usuarios, extendiendo su amplitud y dominio, al tiempo que surgen nuevos mercados, tecnologías, instituciones y empresas que aprovechan este nuevo medio. La Gestión de Redes y Servicios de telecomunicaciones se ha convertido por ello en una creciente y compleja tarea debido a la heterogeneidad de redes y servicios que coexisten. El gran volumen de información que demanda la sociedad actual a partir de la masificación a nivel mundial del uso de Internet, el aumento del uso de los dispositivos móviles, la virtualización de servidores, la llegada de los servicios de la nube y al mismo tiempo, el crecimiento de tráfico por parte de los usuarios hacia los centros de datos, son algunos de los factores que han impulsado a los desarrolladores en la industria de las telecomunicaciones a reexaminar las tecnologías de redes tradicionales.

Este contexto ha traído consigo que las empresas y proveedores de servicios de red tengan que invertir gran cantidad de recursos en aumentar las capacidades de sus sistemas, intentando aprovecharlos al máximo. Esto se evidencia en que la infraestructura de red en el mundo aloja cerca de tres dispositivos interconectados y 15 gigabytes de datos per cápita en 2016, por encima de la cifra que existía en 2014 de un dispositivo interconectado y 4 gigabytes de datos per cápita (1). Sin embargo, esta solución no deja de ser temporal, ya que de ninguna manera las arquitecturas de redes actuales están diseñadas para soportar los requerimientos reales de usuarios y negocios. La expansión de las infraestructuras conlleva a un aumento de su complejidad y por consecuente a medida que las redes se tornan más grandes y complejas el costo de la gestión de las mismas va aumentando proporcionalmente.

Estos elementos hacen que la necesidad de implementar y desplegar una nueva arquitectura de red sea indispensable para mantener la calidad de los servicios informáticos de las redes de datos, por lo que muchas propuestas han sido introducidas para el mejor diseño en futuras redes, entre ellas NDN (*Named Data Networking,* por sus siglas en inglés) (2), Redes Programadas (3) y Redes Definidas por Software o SDN (*Software-Defined Networking*, por sus siglas en inglés) (4). En particular, SDN constituye la tecnología prominente con el potencial de revolucionar todo el mundo de las redes, otorgando una manera flexible de controlarlas (5).

INTRODUCCIÓN

SDN es una arquitectura de red que ofrece gran control sobre la red al traer beneficios en cuanto a configuración y rendimiento. A partir de la separación del plano de control (software) y el plano de datos (hardware) esta arquitectura brinda la posibilidad de implementar servicios telemáticos de una manera determinista, dinámica y escalable, evitando al administrador de red gestionar dichos servicios a bajo nivel.

Actualmente en Cuba asumir esta tecnología en los entornos de redes nacionales traería los beneficios antes descritos ya que nuestro país experimenta un proceso de informatización de la sociedad que aumenta a un ritmo acelerado, lo que significa que las infraestructuras de redes que existen se ven cada vez más limitadas debido a que necesitan mayores y mejores recursos. La Figura A representa un gráfico de líneas que ejemplifica, mediante el incremento de la cantidad de usuarios de servicios a Internet, el crecimiento en los últimos años del acceso y utilización de las TIC en Cuba y la ampliación de los requerimientos de calidad de servicios que esto conlleva.

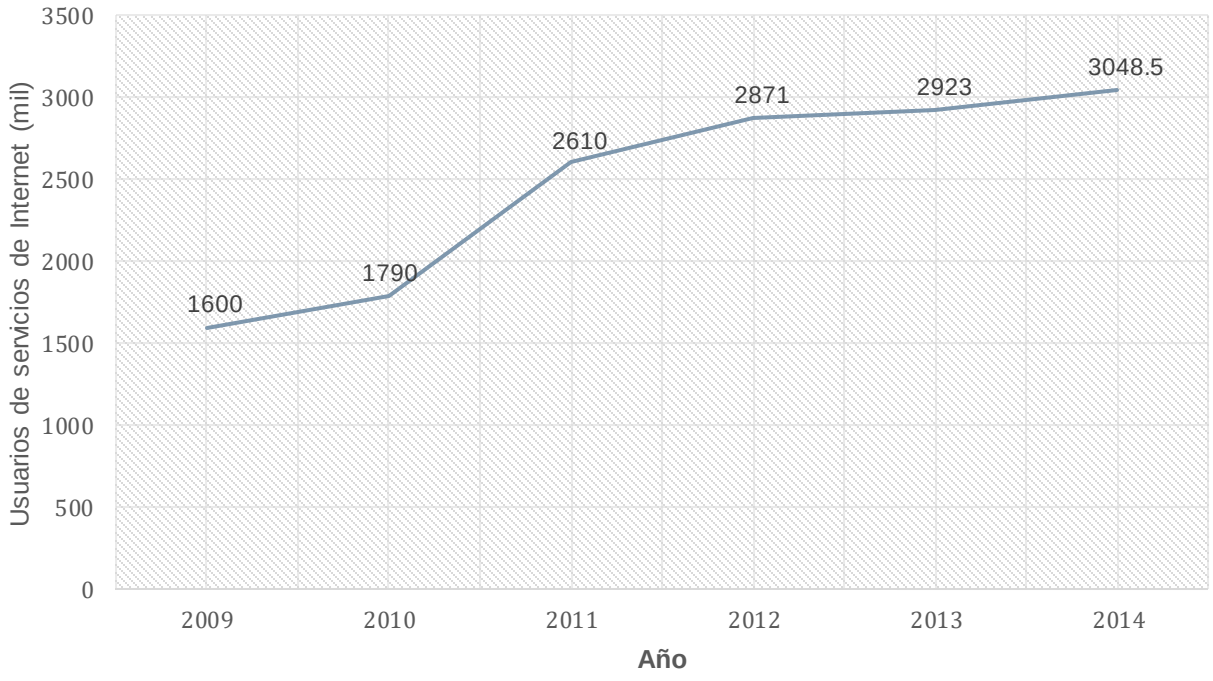

Figura 0.1 – Cantidad de usuarios de servicio a Internet (6)

La implementación de SDN permitiría aumentar las capacidades de gestión de los sistemas informáticos en las redes cubanas, las que en su mayoría carecen de posibilidades de autoconfiguración de los servicios,

en las que existe equipamiento de red activo heterogéneo, con implementaciones de diferentes fabricantes que dificultan la gestión integrada del mismo, y en las que se realiza el monitoreo y control de los dispositivos de manera local, lo que conlleva errores en la configuración e ineficiencia. Sin embargo, la mayoría del equipamiento de red que existe en el país no cuenta con soporte para SDN. Por tanto, la asimilación en las redes cubanas de esta arquitectura y su vinculación a las mismas es más viable a través de equipamiento híbrido que combine la tecnología SDN y los protocolos estándares de comunicación convencionales. En Cuba actualmente comienza a introducirse equipamiento híbrido; no obstante, como resultado de la investigación realizada no se ha encontrado ningún caso en el país de implementación de SDN y los estudios que se han realizado de esta tecnología son a nivel académico sin aplicación práctica, de ahí que se hace necesario sentar las bases que faciliten su integración, gestión y coexistencia con las tecnologías tradicionales.

Se plantea entonces como **problema de investigación**:

¿Cómo gestionar redes definidas por software híbridas en el entorno cubano?

Presentado el problema de la investigación se define como **objeto de estudio** la gestión de redes que se enfoca en el **campo de acción** la gestión de Redes Definidas por Software híbridas.

Para dar solución al problema planteado el **objetivo general** de la investigación se centra en elaborar una arquitectura para la gestión de red basada en la integración de la tecnología SDN y la gestión basada en políticas.

Con el propósito de complementar el objetivo general se han establecido los siguientes **objetivos específicos**:

- Definir el marco teórico mediante el análisis de las redes tradicionales, la arquitectura SDN y la gestión basada en políticas.
- Realizar la modelación de una propuesta de arquitectura para la gestión de SDN híbrida que integre las tecnologías SDN y la gestión basada en políticas.
- Elaborar un procedimiento que permita el despliegue de la arquitectura para la gestión de SDN híbrida.

- Evaluar la arquitectura para la gestión de SDN híbrida propuesta en un entorno de trabajo simulado.

A partir de lo expuesto anteriormente se puede plantear la **idea a defender** de que la elaboración de una arquitectura basada en la integración de la tecnología SDN y la gestión basada en políticas contribuirá a la gestión de redes definidas por software híbridas en el entorno cubano.

Para darle cumplimiento a los objetivos específicos antes mencionados se proyectan las siguientes **tareas de investigación**:

- Análisis del estado de las redes tradicionales.
- Análisis de la arquitectura SDN.
- Análisis de la gestión de red basada en políticas.
- Análisis de herramientas de gestión de configuración, controladores SDN y herramientas de simulación de red.
- Análisis de sistema de operación centralizada de red basado en políticas
- Modelación de la arquitectura la SDN-PBNM.
- Descripción de los pasos a seguir para desplegar la arquitectura.
- Evaluación de la arquitectura SDN-PBNM en un entorno de trabajo simulado.

Para desarrollar estas tareas se utilizan los siguientes métodos científicos de investigación:

Dentro de los **métodos teóricos**:

- **Analítico - Sintético**: Se utiliza en el análisis e interpretación de los conceptos y elementos asociados a la arquitectura SDN y las relaciones y características generales que existen entre ellos.
- **Histórico - Lógico**: Se identifica la evolución histórica de la tecnología de SDN y el desarrollo y potencialidades que con esta arquitectura han alcanzado las redes de información que implementan diferentes organizaciones.
- **Hipotético - Deductivo**: Se pone en práctica en la formulación de la idea a defender y su verificación a partir del análisis de los resultados obtenidos en la aplicación de la propuesta de arquitectura SDN híbrida.

Dentro de los **métodos empíricos**:

- **Entrevistas**: Se utiliza para recuperar información relacionada con la gestión de redes que se realiza los entornos cubanos y las tecnologías que están implementadas en los mismos. Las entrevistas previstas no son estructuradas, es decir, serán entrevistas en las cuales se define un tema, pero no llevan un cuestionario rígido.
- **Estadística**: Se utiliza para analizar el comportamiento y el estado de desarrollo de las TIC en la sociedad.

Justificación de la investigación

Las Redes Definidas por Software en la actualidad se presentan como una solución a muchos de los problemas que existen en las redes tradicionales al tener como principales ventajas mayor flexibilidad, capacidad de programación, gestión y rentabilidad. SDN posibilita hacer la gestión de la red más eficiente y automatizada con el fin de disminuir los gastos en operaciones y capitales.

En las redes cubanas se necesita perfeccionar la gestión de los servicios y recursos debido a las condiciones económicas actuales y al desarrollo acelerado de las tecnologías de la información y las comunicaciones en el país y en el mundo en general, que hacen que las tecnologías tradicionales de comunicación sean llevadas al límite. Por ello es necesario el despliegue de un nuevo paradigma como SDN que mediante una arquitectura híbrida sea adaptada al entorno cubano.

Estructura de la investigación

El presente trabajo de diploma se rige bajo la estructura de: introducción, tres capítulos, conclusiones, recomendaciones, glosario de términos, referencias bibliográficas y anexos.

En el **Capítulo 1** se realiza un análisis del contexto actual de las redes informáticas tradicionales y de sus limitaciones y se definen las características generales de la arquitectura SDN teniendo en cuenta sus conceptos asociados y elementos que la componen, explicando sus ventajas, consideraciones de seguridad e implementaciones, así como las características de la variante SDN híbrida. Además, se investiga la Gestión de Red Basada en Políticas y sus principales conceptos asociados y se realiza un análisis y

valoración de herramientas de monitorización de red, herramientas de simulación de red y controladores SDN.

En el **Capítulo 2** se describen cada uno de los elementos que conforman un sistema de operación centralizada de red basado en políticas y se introduce la propuesta de arquitectura SDN-PBNM a la que se le realiza un análisis de los elementos que la componen y la relación que existe entre los mismos. Además, se desarrolla un esquema metodológico para la aplicación de la arquitectura propuesta.

En el **Capítulo 3** se evalúa la arquitectura SDN-PBNM propuesta en una herramienta de simulación a fin de validar su factibilidad en función de métricas definidas en un entorno de trabajo que simula el escenario real de una red LAN.

CAPÍTULO 1 : Fundamentación Teórica

1.1 INTRODUCCIÓN

En el contexto actual de las tecnologías de la información y las comunicaciones los servicios de red surgidos en los últimos tiempos están llevando a las redes tradicionales a su límite por lo que la implementación de una nueva arquitectura como SDN haría posible crear una infraestructura de red mucho más ágil y flexible a través de la transformación de la misma a una más programable.

En el presente capítulo se realiza un análisis de la situación en la que se encuentran las redes informáticas y las limitaciones que presentan. Posteriormente se definen las características generales de la arquitectura SDN teniendo en cuenta sus conceptos asociados, elementos que la componen, protocolos, ventajas, consideraciones de seguridad, así como las características de la variante SDN híbrida. Además, se estudia la Gestión de Red Basada en Políticas y sus principales conceptos asociados y se realiza un análisis y valoración de herramientas de monitorización, herramientas de simulación de red y controladores SDN.

1.2 LIMITACIONES DE REDES ACTUALES

El desarrollo tecnológico de los últimos años y el creciente ritmo evolutivo del sector de las comunicaciones hacen que las redes y el volumen de datos que estas manejan hayan aumentado considerablemente. Satisfacer las necesidades actuales de servicios es prácticamente imposible con las tecnologías de red tradicionales por lo que las empresas y proveedores de servicios de red tienen que invertir gran cantidad de recursos en aumentar las capacidades de sus sistemas, intentando aprovecharlos al máximo para cubrir requerimientos de sus usuarios, tales como el incremento del ancho de banda y la movilidad.

En nuestro país los sistemas de información en los últimos años han aumentado sus requerimientos debido al proceso de informatización de la sociedad cubana, que se describe como la utilización ordenada y masiva de las TIC para satisfacer las necesidades de información y conocimiento de todas las personas y esferas de la sociedad (6). Como se evidencia en la Tabla 1.1, que representa los indicadores físicos de las TIC en el período de 2009 a 2014, la cantidad de usuarios y recursos que integran la infraestructura informática cubana se ha incrementado de manera gradual, haciendo de la gestión de la misma una tarea cada vez más compleja y costosa.

TABLA 1.1: INDICADORES FÍSICOS DE LAS TIC EN CUBA (6)

Concepto	UM	2009	2010	2011	2012	2013	2014
Cantidad de computadoras existentes	M	700,0	724,0	783,0	834,0	1014,4	1067,4
De ellas en red:	M	455,0	434,4	469,8	500,4	515,4	533,9
Cantidad de usuarios de servicios de internet	M	1600,0	1790,0	2610,0	2871,0	2923,0	3048,5
Computadoras personales por 1 000 habitantes	U	62	64	70	74	90	95
Usuarios de internet por 1 000 habitantes	U	142	159	232	257	261	271
Dominios registrados bajo.cu (a)	U	2331	2255	2285	2345	4839	6698
Total de abonados del sistema celular	M	752,8	1127,9	1431,5	1792,3	2104,6	2636,7

Abonados móviles del sistema celular	M	621,2	1003,0	1315,1	1681,6	1995,7	2530,8
Abonados fijos del sistema celular	M	131,6	124,9	116,4	110,7	108,9	105,9
Cobertura de la población de celular móvil	%	79,3	82,4	83,7	85,3	85,3	85,3

Las tecnologías de la información en Cuba se basan en protocolos y redes convencionales que solventan en la actualidad las necesidades de información de usuarios e instituciones. Sin embargo, las redes tradicionales presentan una serie de limitaciones respecto a nuevas tecnologías. Entre estas limitaciones de las redes actuales se incluyen (4):

- **Complejidad:** La tecnología de red, hasta la fecha, posee un conjunto discreto de protocolos diseñados, fundamentalmente, para conectar dispositivos, pero estos tienden a ser definidos de forma tal que cada uno brinda una solución para un problema específico por lo que el administrador de una red debe realizar tareas y gestionar dispositivos de manera individual y sin el beneficio de una acción conjunta que le permita la administración centralizada. Debido a esta complejidad las redes actuales tratan de minimizar el riesgo de interrupción de los servicios y, por tanto, no pueden adaptarse dinámicamente a las demandas de las aplicaciones y usuarios.
- **Dificultad para aplicar políticas:** Aplicar políticas puede ser extremadamente complejo en redes de gran tamaño debido a que los administradores de red deben configurar miles de dispositivos y mecanismos que abarquen a la red completamente. Esto se evidencia, por ejemplo, cuando se crea una nueva máquina virtual, el administrador de red puede tardar horas y hasta días, para volver a configurar Listas de Control de Acceso o ACL (*Access Control List*, por sus siglas en inglés) en la red, lo que afecta el acceso y la seguridad de los recursos y la Calidad de Servicio o QoS (*Quality of Service*, por sus siglas en inglés).

- **Dificultad para la innovación:** En las redes actuales los fabricantes de equipamiento proporcionan implementaciones propietarias que no permiten acceder al código del software del dispositivo impidiendo el desarrollo de innovaciones en la red. Además, los investigadores no pueden realizar sus experimentos en escenarios reales, viéndose enfrentados a restricciones de políticas de seguridad, a la hora de probar nuevos diseños y protocolos, pudiendo transcurrir hasta 10 años para su estandarización.
- **Dependencia de los proveedores:** Las empresas buscan desplegar servicios y capacidades rápidamente en respuesta a las necesidades cambiantes de su negocio y demandas de los usuarios. Sin embargo, su capacidad de respuesta se ve limitada por los ciclos de desarrollo de los vendedores de productos, que pueden abarcar hasta tres años o más.
- **Incapacidad para escalar:** Al igual que la demanda hacia los centros de datos crece rápidamente, lo hace también de manera proporcional la infraestructura de las redes. En la actualidad, al existir un incremento considerable de la cantidad de dispositivos interconectados que deben ser gestionados y configurados, la gestión adecuada de los mismos se convierte en una tarea cada vez más compleja.

Estos elementos hacen que la necesidad de implementar y desplegar una nueva arquitectura de red que permita mantener la calidad de los servicios informáticos de las redes de datos sea indispensable. La falta de correspondencia entre las necesidades reales del mercado, las capacidades de la red y los altos costes de la gestión en las redes tradicionales ha llevado a la industria a un punto de inflexión, en el cual surge como solución más promisoria SDN.

1.3 RED DEFINIDA POR SOFTWARE

La Fundación de Redes Abiertas u ONF (*Open Networking Foundation,* por sus siglas en inglés) (7) es una organización sin ánimos de lucro impulsada por los usuarios dedicada al desarrollo, estandarización y comercialización de SDN. ONF ha propuesto la definición más explícita y mejor recibida de SDN, la cual se describe a continuación:

Redes Definidas por Software (SDN) es una arquitectura de red emergente donde plano de control de red es desacoplado del plano de datos siendo directamente programable (4).

Con la separación del plano de control del plano de datos no solo se perfecciona la gestión y control sobre una red, sino también se obtiene libertad para definir su comportamiento sin afectar el flujo de datos. Los dispositivos de red convencionales tienen una estrecha integración entre los planos de control y de datos, que realiza depuración de problemas de configuración o control del comportamiento, lo que representa una tarea muy complicada, por lo cual las empresas de equipos de hardware comenzaron a implementar la lógica de reenvío de paquetes en hardware (plano de datos), separada del plano de control. Esto facilita con el surgimiento de SDN que la inteligencia y el estado de la red estuvieran lógicamente centralizados, y la infraestructura de red subyacente se abstrajera de las aplicaciones de negocio. Como resultado, las empresas y las compañías ganan la automatización y el control de la red, lo que les permite construir redes altamente escalables y flexibles que se adaptan fácilmente a las cambiantes necesidades del negocio, a los usuarios finales y al mercado (4).

1.3.1 Arquitectura de SDN

La figura 1.1 representa una vista lógica de la arquitectura de SDN, en la que se muestran las tres capas por la que está compuesta: Infraestructura, Control y Aplicaciones.

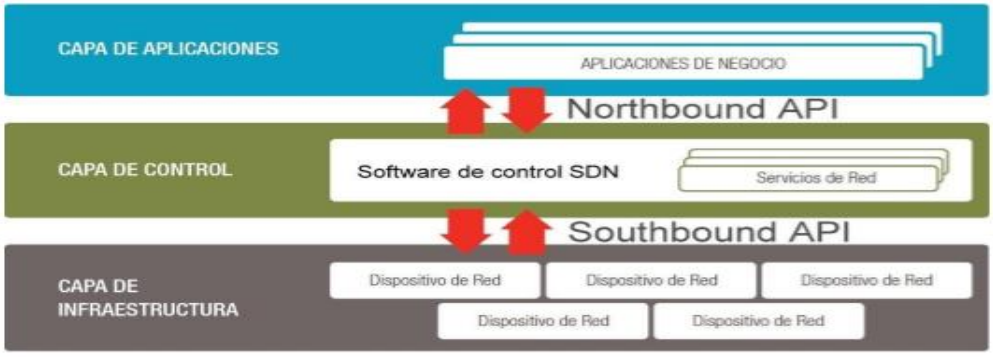

Figura 1.1 - Arquitectura de SDN (8)

La inteligencia de la red, en SDN, se encuentra (lógicamente) centralizada en controladores basados en software, que mantienen una visión global de la misma. Como resultado, la red aparece frente a las aplicaciones y a las decisiones de política como un conmutador lógico y único.

1.3.1.1 Capa de Infraestructura

La capa Infraestructura está compuesta por los elementos o dispositivos de red, los que exponen sus capacidades a través de las Interfaces de Programación de Aplicaciones o API (*Application Programming Interface*, por sus siglas en inglés). La interfaz de control del plano de datos, también denominada *Southbound*[1] API, permite la comunicación entre el controlador SDN y los dispositivos de red, al realizar cambios dinámicos de acuerdo a las demandas en tiempo real y las necesidades. Dicha interfaz de control del plano de datos se formaliza a través del protocolo OpenFlow (9), que se ha convertido en el protocolo oficial a utilizar para la conexión remota entre el controlador SDN y los dispositivos de red, además de otros muy utilizados como Cisco OpFlex (10).

1.3.1.2 Controlador SDN

En el centro de la arquitectura de las SDN se encuentra la capa Control con el controlador SDN, que es quien gestiona los flujos de datos. El controlador equivale al sistema operativo de la red que controla todas las comunicaciones entre las aplicaciones y los dispositivos. El controlador SDN se encarga de traducir las necesidades o requisitos de la capa Aplicación a los elementos de red, y de proporcionar información relevante a las aplicaciones SDN, pudiendo incluir estadísticas y eventos (4).

Por tanto, en la arquitectura SDN, es el controlador central el que dicta el comportamiento general de la red a partir de los requerimientos de las aplicaciones. Ejemplos de algunos controladores de código abierto existentes son: Ryu (11), Maestro (12), Trema (13), Beacon (14) y Floodlight (15).

Los dispositivos de red delegan su lógica al controlador y pasan a ser simples unidades de conmutación de tráfico. Es en el sistema operativo del controlador donde se configura el enrutamiento y estará abierto a la implementación de las nuevas funcionalidades hacia las aplicaciones, también denominadas *Northbound*[2] APIs (16). Estas interfaces pueden ser usadas para facilitar la innovación y permitir la organización y

[1] Southbound API se refiere a API en dirección sur o hacia abajo. Toda la bibliografía encontrada en español se refiere a este tipo de API en inglés, por lo que en esta investigación se referirán de la misma forma.

[2] Northbound API se refiere a API en dirección norte o hacia arriba. Toda la bibliografía encontrada en español se refiere a este tipo de API en inglés, por lo que en esta investigación se referirán de la misma forma.

automatización de la red para suplir las necesidades de las diferentes aplicaciones a través de la configurabilidad (17).

1.3.1.3 Capa de Aplicaciones

La capa Aplicaciones de la arquitectura SDN, permite comunicar al controlador SDN a través de las APIs sus necesidades y el comportamiento que desean de la red. La interfaz de los controladores SDN hacia las aplicaciones es un conjunto de interfaces ya que la definición de aplicaciones SDN es muy amplia, cubriendo desde servicios de red, como QoS, a aplicaciones de negocio.

SDN consta de un plano de control centralizado con *Southbound* APIs para la comunicación con la infraestructura de hardware y cuenta con *Northbound* APIs para comunicarse con las aplicaciones de red.

1.3.1.4 Southbound APIs

Southbound APIs facilitan el control en la red, permitiendo al controlador realizar cambios dinámicos de acuerdo a las demandas en tiempo real y las necesidades. La principal *Southbound* API existente hoy en día es OpenFlow, que está estandarizado por la ONF (16). Algunos investigadores argumentan que la gran cantidad de protocolos de gestión y de control existentes anteriormente, tales como XMPP (*Extensible Messaging and Presence Protocol,* por sus siglas en inglés), SNMP (*Simple Network Management Protocol,* por sus siglas en inglés), NETCONF, pueden convertirse en *Southbound* APIs (10).

OpenFlow

El protocolo OpenFlow es el primer estándar definido para la *Southbound* API y se implementa tanto en los dispositivos de red (switches OpenFlow), como en el controlador. En el Anexo 1 se resumen las principales características de las versiones de este protocolo.

En OpenFlow se especifican las primitivas básicas que pueden ser usadas por una aplicación SDN para programar el plano de conmutación de los dispositivos de red, ya sean físicos o virtuales, análogo al conjunto de instrucciones que le permiten a la Unidad Central de Procesamiento o CPU (*Central Processing Unit*, por sus siglas en inglés) programar el sistema de la computadora. OpenFlow utiliza el concepto de flujos para identificar el tráfico de la red e instaurar reglas de conmutación en los switches OpenFlow, las cuales pueden

ser estáticas o dinámicas. Logra un control granular debido a la programación de la red basada en dichos flujos, lo que permite una respuesta rápida a los cambios en tiempo real de las aplicaciones, usuarios y servicios (4).

1.3.1.5 Northbound APIs

Las *Northbound* APIs en cambio, pueden ser usadas para facilitar la innovación y permitir la organización y automatización de la red para suplir las necesidades de las diferentes aplicaciones a través de la programabilidad de la red SDN. Se podría decir que estas interfaces son las más críticas en un entorno SDN, debido a que soportan una gran variedad de aplicaciones y servicios por encima y por lo tanto con algunas de ellas no funciona correctamente. Hay una amplia variedad de posibles interfaces de este tipo situadas en diferentes lugares de la pila para controlar los diferentes tipos de aplicaciones a través del controlador SDN. Sin embargo, estas interfaces son el componente más indeterminado de todo el entorno SDN, por lo cual no han sido estandarizadas. Cada controlador puede tener una interfaz de programación diferente. Hasta que estas APIs sean estandarizadas, el desarrollo de aplicaciones de red para SDN, estará limitado (18).

1.3.2 Ventajas de SDN

SDN como arquitectura de red emergente provee las ventajas que se describen a continuación (5) (8):

- **Innovación mejorada**: Al poder controlar los dispositivos y los servicios a través del controlador, sin necesidad de configurar el equipamiento específico de la red, SDN permite la innovación en nuevos protocolos, limitado solo esto por las capacidades de los administradores de red, lo que permite crear nuevos tipos de aplicaciones y modelos de negocio por parte de las empresas, que las beneficia y aumenta el valor de sus redes.
- **Gestión dinámica de errores**: Los errores que ocurren en la red, ya sean en enlaces o dispositivos, se gestionan más eficientemente. Los sistemas convergen más rápido hacia el objetivo óptimo y su comportamiento es predecible.
- **Alta utilización**: Una ingeniería de tráfico centralizada proporciona una visión global de la red, tanto en la oferta como en la demanda de recursos en la red. Al gestionar a través de esta visión global los caminos de extremo a extremo logran un alto aprovechamiento de los enlaces.

- **Entorno de pruebas eficiente**: Las funcionalidades de la red pueden ser completamente emuladas mediante software, lo que no solo ayuda en la comprobación y verificación de los parámetros de control, sino también a la gestión de posibles escenarios que se puedan dar en la red.
- **Actualizaciones dinámicas**: La separación del plano de control respecto al plano de datos, permite llevar a cabo actualizaciones de software en la red sin pérdida de información o degradación de la capacidad misma, posibilitando la configuración de los dispositivos en tiempo real.
- **Control granular**: La arquitectura SDN posibilita la aplicación de una amplia variedad de políticas en la red a diferentes niveles: sesiones, usuarios, dispositivos y aplicaciones.
- **Enfoque centralizado**: SDN permite tener una perspectiva centralizada de la estructura de la red simplificando la gestión y el aprovisionamiento de los recursos y servicios que en la misma se brindan.
- **Reduce gastos operativos**: SDN permite el control algorítmico de la red y de su equipamiento que cada vez es más programable, haciendo más sencilla la configuración y gestión. Esto permite una reducción del tiempo de gestión por parte de los administradores, lo que reduce la probabilidad de error humano.
- **Reduce las inversiones en infraestructura**: Mediante la posibilidad de reutilizar el hardware existente, SDN limita la necesidad de invertir en hardware nuevo.

1.3.4 Seguridad en entornos SDN

La arquitectura SDN ofrece una plataforma ventajosa para centralizar, fusionar y verificar políticas de seguridad para hacer que la implementación de las mismas brinde el grado de protección que permita identificar brechas de seguridad en una red de manera proactiva y reactiva. Entre las posibilidades y beneficios que brinda para proteger la disponibilidad, integridad y privacidad de la información están los siguientes (19):

- **Automatización**: La capacidad de implementar las políticas de seguridad de la red de manera automática, en lugar de tener que realizar los cambios manuales en cada equipo es fundamental. Esto significa que los dispositivos necesitan APIs bien definidas que permitan activar desde el controlador, de forma automática los mecanismos de seguridad.

- **Programación**: Hoy en día, las políticas de seguridad están definidas para zonas de seguridad que son estáticas y ligadas a las interfaces físicas. En SDN la seguridad en la red se define mediante programación al definirse las configuraciones de manera centralizada (20).
- **Simplificación de los dispositivos de seguridad**: SDN proporciona una visión única de la red por lo que es fácil definir la ubicación de los cortafuegos y dispositivos de Sistema de Prevención de Intrusos o IDS (*Intrusion Detection System*, por sus siglas en inglés). En el controlador SDN se puede garantizar que todo el tráfico sospechoso se redirija a los cortafuegos / IDS sin tener que preocuparse por múltiples puntos vulnerables que deban ser asegurados, sobre todo, de los extremos de la red que deben configurarse de manera uniforme para las políticas de seguridad.
- **Simplifica el cumplimiento de políticas**: Cada flujo está acoplado a una política establecida por el operador, no siendo necesario instalar configuraciones y políticas en múltiples dispositivos. Las actualizaciones a los programas contra los softwares maliciosos pueden ser mucho más fáciles y más completas.

Sin embargo, aunque existen perspectivas competentes sobre la seguridad en la arquitectura SDN, estas todavía no están bien definidas, ya que no convergen en una idea común. La seguridad en SDN es un aspecto de vital importancia para proteger los recursos, operaciones y servicios en la red, debido a las características de gestión centralizada de esta arquitectura y los riesgos y vulnerabilidades que presenta. Entre las consideraciones de seguridad que hay que tener en cuenta y los elementos que deben ser protegidos están los siguientes (19):

- **Seguridad del controlador**: Esto surge del hecho fundamental de que el controlador es la inteligencia y quien gestiona el flujo de la red, lo que significa que la vulnerabilidad del controlador es la vulnerabilidad de toda la red, al contrario de las redes tradicionales en las que cuando un dispositivo de conmutación se encuentra en peligro, la vulnerabilidad está localizada. Por lo tanto, se hace importante elegir algoritmos y métodos que puedan garantizar un alto nivel de seguridad para el controlador a través, por ejemplo, de una definición cuidadosa de las políticas para el tráfico.
- **Protección de las comunicaciones**: Además de asegurar el controlador es imprescindible proteger las aplicaciones que carga y los dispositivos de red que gestiona mediante políticas y protocolos de seguridad como TLS (*Transport Layer Security*, por sus siglas en inglés).

- **Alta disponibilidad en el controlador**: La arquitectura SDN provee alta disponibilidad para la red debido a que esta pueden centrarse en el controlador en lugar de tener que diseñar una alta disponibilidad en todos los componentes de la infraestructura de red. Esto puede ser una gran ventaja en la simplificación de la red y la optimización de los costos. Sin embargo, esta alta disponibilidad en el controlador también se puede convertir en un problema, ya que toda la red puede ser derribada por un corte en el controlador (19).

1.3.5 Implementaciones de SDN

A continuación, se describen dos ejemplos de implementaciones de la arquitectura SDN. El primero, en el campus de la Universidad de Stanford, en California, Estados Unidos, y el segundo en el centro de datos de Google.

1.3.5.1 Universidad de Stanford

Una parte de la red del campus de la Universidad de Stanford migró en el 2010 a la tecnología SDN mediante la implementación del protocolo OpenFlow. Originalmente, el hardware de red no soportaba el protocolo OpenFlow y la red legada era gestionada por el software de código abierto Zenoss, junto con configuraciones basadas en interfaz de línea de comandos o CLI (*Command Line Interface*, por sus siglas en inglés). La migración fue dirigida hacia los usuarios de la red inalámbrica, luego se amplió hacia los usuarios cableados de dos edificios: William Gates y el Paul Allen.

La migración fue planeada para ofrecer una mejor visibilidad del tráfico de red y permitir la experimentación sobre la misma. La meta era, por lo tanto, migrar una Red de Área Local Virtual o VLAN (*Virtual Local Area Network*, por sus siglas en inglés*)* y los usuarios seleccionados hacia el control de OpenFlow. Es decir, el tráfico OpenFlow fue asignado a una VLAN pre-especificada y gestionada por el controlador OpenFlow, mientras que el resto del tráfico (no OpenFlow) fue asignado a otras VLANs. Este enfoque donde coexisten VLANs OpenFlow y VLANs no OpenFlow en un switch, constituye un modelo híbrido y a largo plazo permitió la implementación de OpenFlow junto a redes heredadas en la misma infraestructura física.

Los principales objetivos del despliegue de OpenFlow en Stanford estaban enfocados en motivar la necesidad de SDN, comprender y verificar la nueva tecnología SDN y contribuir a la especificación y comunidad de OpenFlow.

Para lograr dichos objetivos, fue necesaria la creación de una sección de red para ser utilizada con el propósito de la experimentación. Esto se logró mediante la introducción de una capa de virtualización de red en el plano de control que permite crear redes virtuales llamadas secciones; cada sección se maneja normalmente por un controlador OpenFlow diferente. Esta separación implica que las acciones en una de las secciones no afectan a otras secciones de la red. El enfoque adoptado para la migración a las redes SDN fue mover gradualmente los usuarios individuales y luego cada VLAN al control basado en OpenFlow.

Es importante hacer notar que la VLAN cableada abarcó tanto la parte de la red OpenFlow como la no OpenFlow, mientras que la VLAN inalámbrica se gestionó exclusivamente mediante OpenFlow. El despliegue de la red OpenFlow en la Universidad de Stanford operó, inicialmente, con los controladores NOX (21), SNAC (22) y Trema (13).

Las herramientas que Stanford utiliza para la monitorización son estándares como ping, tcpdump, y wget. En casos específicos, fueron utilizadas algunas herramientas especiales para realizar la monitorización basada en la disponibilidad de sondas y de APIs (23).

Por ejemplo, el controlador NOX utilizado en la implementación no expone una API que permita consultar las estadísticas de plano de control. En ese caso, se realizó un tcpdump para archivar la comunicación del controlador y la captura de paquetes fue analizada utilizando la herramienta Oftrace para revelar las estadísticas necesarias. Stanford, además, utiliza varias herramientas adicionales como (23):

- Wireshark: Herramienta utilizada para capturar el tráfico OpenFlow.
- Mininet: Emulador de red que crea una red de hosts virtuales, conmutadores y controladores.
- Ofrewind: Reproducción de eventos de la red, mediante la reproducción del tráfico del plano de control y del plano de datos.
- Hassel y NetPlumber: Comprobación y depuración de las directivas de red en tiempo real.
- ATPG (*Automatic Test Packet Generation*): Generación automática de paquetes de prueba para la depuración de la red.

Estas herramientas en sentido general ofrecieron resultados positivos a partir de métricas que fueron monitoreadas como tiempo de establecimiento de flujo, uso del CPU y entradas en la tabla del switch, lo que permitió considerar la migración de la red en la Universidad de Stanford a la tecnología SDN como una mejora de infraestructura y calidad en los servicios. Sin embargo, algunos de los resultados asociados con las métricas que fueron monitoreadas en la implementación deben ser tomados en cuenta para futuras soluciones, entre los que se encuentran (23):

- El controlador SDN no pudo descubrir los conmutadores sin protocolo OpenFlow en la topología.
- Los controladores utilizados no apoyaron el protocolo STP (*Spanning Tree Protocol*, por sus siglas en inglés).
- El sistema SDN no tenía visibilidad completa tanto de los flujos y como de los usuarios que se extendieron en segmentos OpenFlow y no OpenFlow de la red.
- Los conmutadores utilizados no funcionan bien con la agregación LACP (*Link Aggregation Control Protocol*, por sus siglas en inglés).

1.3.5.2 Centro de datos de Google

Google ofrece muchos servicios (Ej. Motor de búsqueda, Google+, Gmail, YouTube, Google Maps) a usuarios globales, lo que requiere que una gran cantidad de datos sean trasladados de una región del mundo a otra, haciendo que estas aplicaciones o servicios sean la carga fundamental de la red WAN (Red de Área Amplia o *Wide Area Network*, por sus siglas en inglés). Google, analizó este panorama, y concluyó que la prestación de tales servicios no sería escalable con las tecnologías actuales, debido a la complejidad exponencial en la gestión y configuración, por lo cual Google decidió implantar la tecnología SDN para la gestión de su infraestructura WAN.

La WAN del centro de datos de Google se organiza en dos columnas vertebrales (ver Figura 1.2), una red (I-escala) orientada a Internet que lleva el tráfico de usuarios y una red interna (G-escala) que transporta el tráfico entre los centros de datos. Estas dos columnas vertebrales tienen necesidades y características del tráfico muy diferentes. Es en la red G-escala en la que Google ha desplegado una solución OpenFlow empleando las SDN (24).

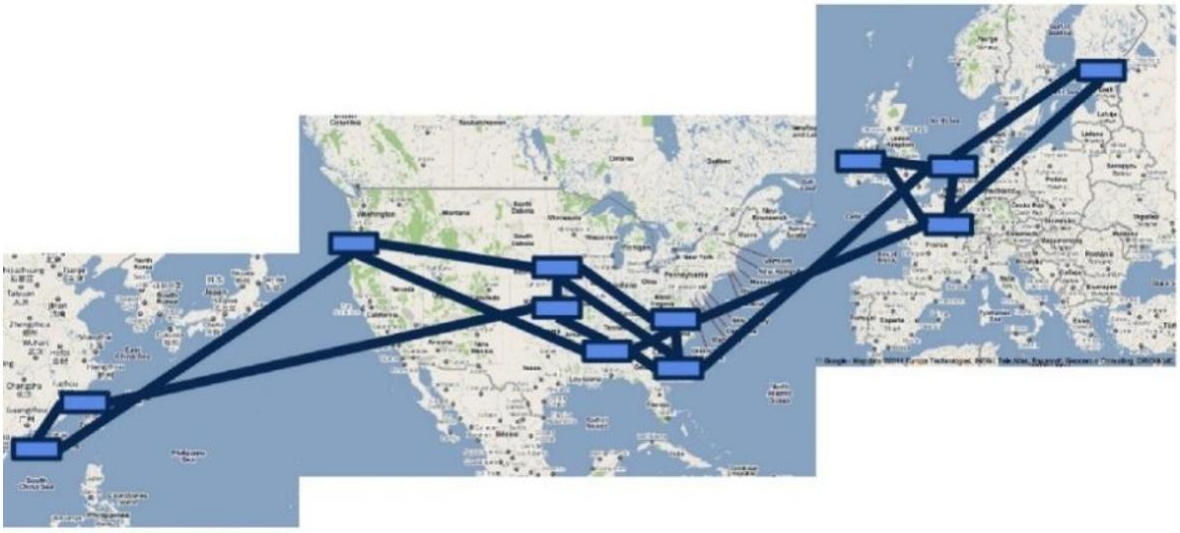

Figura 1.2 - WAN del centro de datos de Google (24)

Google adoptó la arquitectura de las SDN en la interconexión WAN de su centro de datos lo que le permitió desplegar los protocolos de enrutamiento e ingeniería de tráfico para sus necesidades únicas. Cuando comenzó la migración no había ningún dispositivo disponible de red que soportara OpenFlow y pudiera cumplir con los requisitos de Google, por lo que Google construyó su propio switch de silicio con tablas de enrutamiento de código abierto con soporte OpenFlow.

En cada centro de datos se encuentran múltiples racks de conmutadores para proporcionar escalabilidad (múltiples terabits de ancho de banda) y tolerancia a fallos. Los centros de datos están conectados entre sí y varios controladores OpenFlow se comunican con los conmutadores utilizando OpenFlow. Múltiples controladores aseguran que no exista un único punto de fallo.

Google en su WAN creó un servicio centralizado de ingeniería de tráfico (TE). Este servicio recopila datos de la topología de la red subyacente, así como la demanda de ancho de banda de las aplicaciones / servicios. Con estos datos, calculan las asignaciones de ruta de los flujos de tráfico y luego programan los conmutadores utilizando OpenFlow. En caso de acontecimientos que cambien la demanda o que provoquen algún cambio en la red, el servicio centralizado TE recalcula las asignaciones de ruta y reprograma los conmutadores (24).

Google ejecuta en la actualidad satisfactoriamente la red WAN de sus centros de datos sobre una arquitectura SDN y OpenFlow. Esta es la red más grande en funcionamiento dónde se utiliza esta tecnología y ha permitido mejorar la operación, el rendimiento, la utilización y la eficiencia de costos. Sin embargo, existen recomendaciones que según Google deben ser tomadas en cuenta para un correcto despliegue de la tecnología SDN (24).

- Deben existir varios controladores OpenFlow distribuidos para proporcionar tolerancia a fallos.
- La programación de flujos individuales para grandes redes puede demorar mucho tiempo.
- El protocolo OpenFlow es estable y soporta muchas aplicaciones de red, pero se sigue desarrollando.

1.4 RED DEFINIDA POR SOFTWARE HÍBRIDA

Una SDN híbrida es una red donde las tecnologías de red tradicionales y los protocolos SDN operan en el mismo entorno. SDN híbrida permite a los administradores de red introducir nuevas tecnologías de SDN como OpenFlow a entornos heredados sin una completa visión de la arquitectura de la red.

En una SDN híbrida, los ingenieros pueden correr tecnologías SDN y protocolos estándares simultáneamente en el hardware físico. Un administrador de red puede configurar el controlador SDN para descubrir y controlar el flujo de tráfico, mientras que la tradicional continúa dirigiendo el resto del tráfico de la red (25).

1.5 GESTIÓN DE RED BASADA EN POLÍTICAS

Uno de los elementos que forma parte del modelo híbrido que se desarrolla en la investigación es la gestión de red basada en políticas. La gestión de red basada en políticas o PBNM (*Policy Based Network Management*, por sus siglas en inglés) es una de las tareas de gestión que evita a los administradores de red la configuración y manipulación de los elementos de la red manualmente a través de interfaces de control, proceso propenso a errores y que consume tiempo. PBNM en una organización permite la automatización de los recursos en la red para ofrecer un servicio rápido y con calidad, lo cual significa una mejor posición en el mercado y, por tanto, un mayor volumen de negocio que los competidores (26).

1.5.1 Funcionamiento de PBNM

Las políticas son reglas independientes de la tecnología que las utiliza que tienen como objetivo mejorar la funcionalidad no modificable de dispositivos gestionados mediante la introducción de la lógica interpretada que se puede cambiar de forma dinámica sin necesidad de modificar la aplicación subyacente. Esto permite un cierto grado de capacidad de programación, sin necesidad de interrumpir el funcionamiento del sistema gestionado o del propio sistema de gestión.

La Figura 1.3 describe una arquitectura general para un sistema PBNM (26) donde interactúan cuatro componentes principales:

- Herramienta de gestión de políticas, la cual permite la gestión de las políticas de red que están almacenadas en un repositorio de políticas.
- Sitio de ejecución de políticas o PEP (Policy *Enforcement Point*, por sus siglas en inglés), entidad en donde se ejecutan las políticas.
- Sitio de decisión de políticas o PDP (*Policy Decisión Point*, por sus siglas en inglés), responsable de obtener las políticas del repositorio de políticas y generar las decisiones acordes con las peticiones de los PEP.
- Sitio local de decisión de políticas LPDP (*Local* PDP), el cuál es opcional y está localizado en el PEP.

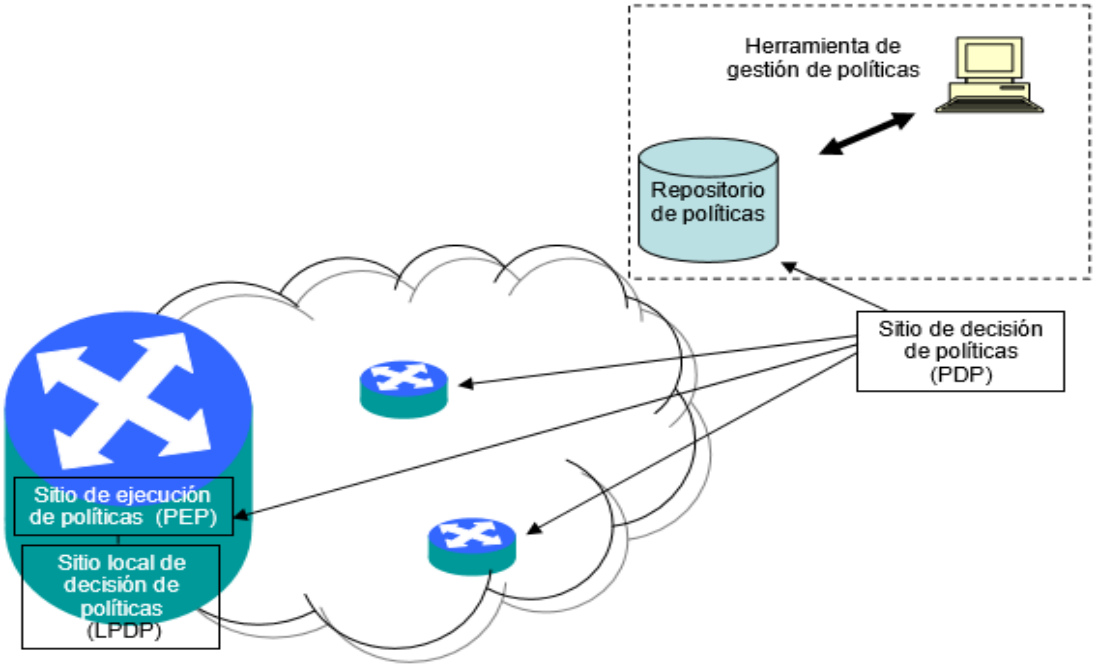

Figura 1.3 - Arquitectura PBNM (26)

1.5.2 Estándares empleados en PBNM

En la implementación de la PBNM se utilizan algunos estándares entre los cuales están:

- SNMP (*Simple Network Management Protocol*).
- PCIM (*Policy Core Information Model*).
- COPS (*Common Object Policy Service*).

Para el cumplir el objetivo de la investigación solo se utiliza SNMP como protocolo de gestión de la configuración como parte del modelo propuesto, por lo cual es el único que se describe en este estudio.

SNMP

SNMP es un protocolo para la gestión de red para proveer una gestión estándar, simplificada, y extensible de una red que tiene diferentes equipos de interconexión. El SNMP fue diseñado para facilitar la gestión de la red sin hacerla muy compleja.

La arquitectura del SNMP se basa en la interrelación de tres componentes básicos: un gestor, un agente, y una base de información gestionada datos o MIB (*Management Information Base*, por sus siglas en inglés). El gestor SNMP representa un programa como, por ejemplo, el HP OpenView de la plataforma de gestión de Hewlett-Packard. El agente constituye un software residente en los dispositivos gestionados de la red, tales como un conmutador, un enrutador, o una computadora. Cada agente almacena datos de gestión y responde a las preguntas del gestor SNMP. El tercer elemento, la base de datos, es referenciada como la Base de Información de Gestión y contiene los objetos gestionados (26).

Hasta el momento SNMP cuenta con tres versiones, pero las más utilizadas son SNMP versión 1 (SNMPv1) y SNMP versión 2 (SNMPv2). La última versión del protocolo (SNMPv3) posee cambios significativos con relación a sus predecesores, sobre todo en aspectos de seguridad, sin embargo, no ha sido aun mayoritariamente implementada (27).

1.3.3 Comparación entre SDN y redes tradicionales

La arquitectura SDN con la separación del plano de control del plano de datos ofrece gran control sobre los recursos, operaciones y servicios de la red a los administradores, que, respecto a tecnologías de red tradicionales, obtienen beneficios en cuanto a configuración, desempeño e innovación, tal y como se resume en la Tabla 1.2.

TABLA 1.2: COMPARACIÓN ENTRE SDN Y REDES TRADICIONALES (17)

	SDN	REDES TRADICIONALES
Características	Separación del plano de control del plano de datos, programabilidad.	Un nuevo protocolo por problema, control de la red complejo.
Configuración	Configuración automatizada y validación centralizada.	Errores de configuración manual.
Desempeño	Control global dinámico	Información limitada y configuración relativamente estática.
Innovación	Implementación fácil de nuevas ideas de software, entorno de pruebas con aislamiento, rápido despliegue usando actualización de software.	Dificultad de implementación de nuevas ideas de software, entorno de pruebas limitado, largo proceso de estandarización.

1.6 HERRAMIENTAS DE MONITORIZACIÓN

Para efectuar una correcta gestión de la información en una red de datos se hace necesario realizar los dos modos de actuación de la gestión, el control y la monitorización (28). Como parte del control en esta investigación se utiliza PBNM para facilitar las tareas administración de la configuración en la arquitectura SDN híbrida de manera automatizada. Por su parte, para la monitorización se evalúan un conjunto de herramientas dedicadas a la gestión de recursos que permiten la aplicación y evaluación de la propuesta. A continuación, se muestra un resumen con las principales características de dicho conjunto de herramientas.

Nagios

Nagios es un sistema de código abierto de monitorización de equipos y de servicios de red desarrollado en C y publicado bajo la licencia GPL. Esta herramienta fue originalmente diseñada para ser ejecutada en GNU/Linux, pero también se ejecuta en variantes de Unix (29).

Nagios se caracteriza por la monitorización en varios sistemas operativos de servicios de red como SMTP, POP3, HTTP, ICMP, SNMP, FTP y SSH y recursos de *host* como carga del procesador, uso de disco, los

registros del sistema, uso de memoria, entre otros. El diseño de *plugins* es simple y además pueden ser escritos en varios lenguajes (Bash, C++, Perl, Ruby, Python, PHP, C#, Java, entre otros). Nagios está formado por 2 módulos diferenciados: el núcleo, llamado Nagios Core, que contiene los componentes fundamentales del *software*, y los *plugins*, donde cada *plugin* monitoriza una serie de recursos o de servicios. Nagios permite la visualización del estado de la red en tiempo real al proveer una interfaz web para visualizar el estado actual de la red con la posibilidad de generar informes y gráficas (29).

Nagios constituye un sistema de monitorización de redes ampliamente utilizado, pero a pesar de sus grandes ventajas una de sus principales desventajas constituye el hecho de que los usuarios que no están familiarizados con la herramienta deben aprender el funcionamiento de un sistema complejo que no dispone de una herramienta intuitiva de configuración. Es significa que, a pesar de ser un software gratuito, supone un gasto importante para las empresas ya que necesita personal calificado dedicado a su configuración y mantenimiento. Además, cualquier modificación en la configuración requiere un reinicio completo del sistema, ya que, por ejemplo, no es capaz de auto descubrir nodos nuevos que se incluyan al mismo. Otro de los inconvenientes de esta herramienta es que posee una interfaz web que solo sirve para visualizar los eventos, mientras que cualquier cambio debe realizarse manualmente desde el servidor de Nagios.

Zabbix

Zabbix es un sistema código abierto de monitorización de redes programado en PHP y C y distribuido bajo la licencia GPL. Esta herramienta está diseñada para monitorear y registrar el estado de varios servicios de red al ofrecer un control centralizado de sus parámetros (30).

Zabbix permite la monitorización de los recursos que influyen en el rendimiento de la red como procesos de carga, actividad en la red, actividad en disco, parámetros del sistema operativo y la disponibilidad y capacidad de respuesta de servicios estándar como SMTP o HTTP. Incluye una API mediante la cual se puede mantener una comunicación con la herramienta brindando la posibilidad de administrar y configurar la información que ella gestiona. Para almacenar los datos de monitoreo de la red Zabbix ofrece soporte de base de datos en MySQL, PostgreSQL, Oracle o SQLite. Además, es capaz de realizar la auto detección de dispositivos y servicios monitorizados. Esta herramienta posee gran cantidad de información sobre su instalación, uso y soporte y cuenta con una amplia comunidad que desarrolla continuamente *plugins* y aplicaciones que facilitan su uso y gestión (30).

Sin embargo, a pesar de todas las ventajas que posee Zabbix, un aspecto a tener en cuenta a la hora de seleccionar esta herramienta es el tema del almacenamiento en la base de datos. Si el número de parámetros monitorizados crece y su intervalo de actualización es muy reducido, es posible que, de no haber ajustado convenientemente el rendimiento de la base de datos, ocurra un cuello de botella que resultará en muchos datos de monitorización encolados y por consecuente un deterioro de la funcionalidad del sistema.

Munin

Munin es una aplicación libre y de código abierto de monitorización de los recursos de red diseñada en el lenguaje de programación Perl y publicada bajo la licencia GPL. Esta herramienta permite monitorizar el uso de recursos de hardware como disco duro, red, uso de CPU y RAM, aunque también es capaz de realizar la monitorización de procesos de Apache, Squid, consultas de MySQL, entre otros (31).

Munin tiene una arquitectura servidor/agente en la que el servidor se conecta a todos los equipos agentes para interrogarlos, recopilar información y en caso de ser necesario actualizar los gráficos en la interfaz web. El servidor corre sobre Linux y el agente puede correr sobre Linux y Windows. Esta herramienta brinda la posibilidad de configurar umbrales de alerta para los estados de advertencia y crítico y a través de la recolección de datos, puede mostrar tendencias que pueden ayudar a predecir cuellos de botella (31).

Munin constituye una aplicación de monitorización potente, pero tiene como desventajas que su interfaz web sólo es para visualizar los resultados sin proporcionar un control sobre la aplicación que obliga a que toda configuración se deba realizar de forma manual por línea de comandos. Además, recibe la información del servidor central sin autenticación y en texto plano, por lo cual es vulnerable a posibles violaciones de seguridad.

Cacti

Cacti es una herramienta multiplataforma para la monitorización de desempeño y utilización de los recursos de red desarrollada en PHP y distribuida bajo la licencia GPL. Está basada en la generación de gráficos y provee un interfaz ágil, plantillas de gráficos avanzadas, múltiples métodos para la recopilación de datos y manejo de usuarios (32).

Cacti tiene una interfaz de usuario fácil de usar, que resulta conveniente para instalaciones del tamaño de una LAN, así como también para redes complejas con cientos de dispositivos. Soporta el protocolo SNMP. Incluye plantillas para reutilizar definiciones de gráficos, datos y fuentes de dispositivos. Cuenta con una funcionalidad para garantizar su seguridad, que permite administrar los usuarios local o remotamente y asignar niveles detallados de autorización basados en usuarios o grupos. Es capaz de recopilar la utilización del canal en las interfaces de sus equipos, así como los registros de errores. Permite la configuración de alertas y notificaciones basada en umbrales (32).

Cacti, entre sus principales inconvenientes tiene que la configuración de las interfaces de monitoreo suele ser trabajosa y al igual que la creación de *plugins* por lo que hacer actualizaciones de su sistema puede tornarse complejo. Sin embargo, cuenta con una comunidad grande y activa en torno a sus foros que proporciona *scripts*, plantillas y consejos sobre la gestión de la herramienta.

Zenoss

Zenoss es una herramienta informática de código abierto desarrollada en Python y publicada bajo la licencia GPL para el monitoreo de disponibilidad, configuración, desempeño y eventos en una red (33).

Zenoss a través de una interfaz o consola web muestra la visualización y el control total de la aplicación y permite el manejo del estado y situación de la infraestructura al ofrecer monitorización de dispositivos y servicios en la red mediante protocolos como SNMP, HTTP, POP3, entre otros. Zenoss puede detectar automáticamente nuevos recursos y cambios en la configuración, que se almacenan en una Base de datos de la Gestión de Configuración (CMBD), en la que se guardan detalles relevantes de cada elemento y con la cual se puede llevar a cabo acciones correctivas de fallos, notificaciones y alertas. Además, posee una comunidad que dispone de un repositorio de *plugins* llamado *ZenPacks*, con los cuales los miembros de dicha comunidad pueden extender las funcionalidades de Zenoss (33).

Zenoss, sin embargo, no utiliza agentes para la gestión de sus clientes por lo que se requiere una configuración previa a su instalación del protocolo SNMP en cada uno de las máquinas a monitorizar siguiendo un procedimiento distinto según la versión o tipo de sistema operativo. Además, para funcionar en sistemas operativos de *Microsoft Windows* necesita de la aplicación externa *VMplayer.*

1.6.1 Evaluación de las herramientas de monitorización

Como parte del estudio de las herramientas asociadas al proceso de gestión de la configuración para desarrollar la presente investigación se evalúan las herramientas de monitoreo descritas anteriormente, teniendo en cuenta un conjunto de indicadores que permiten seleccionar una de ellas en base a cumplir los objetivos propuestos. Estos indicadores para la selección de la herramienta más adecuada son:

- **Soberanía Tecnológica**: Este indicador hace referencia a la capacidad de poseer el poder absoluto sobre determinada tecnología, contemplándose esto como la posibilidad de conocer, modificar y distribuir la tecnología. La evaluación de este parámetro se realiza sobre la base de la existencia o no en las herramientas de este indicador, identificándose con las siglas **ST**.
- **Soporte:** Este indicador hace referencia a la facilidad que posee la herramienta de extenderse con la adición de nuevas funcionalidades, el desarrollo de nuevas mejoras y revisiones para la corrección de errores, la disponibilidad de un foro y wiki para preguntas y resolución de problemas o peticiones de usuarios y la cantidad de documentación e idiomas en la que está disponible. Este indicador es evaluado teniendo en cuenta la presencia de las características antes mencionadas en la herramienta a analizar en **Alto**, cuando posee tres o más de estas características, y **Bajo** en caso contrario. Para identificarlo se utilizará la sigla **S**.
- **API**: Este indicador muestra la capacidad que tiene la herramienta de brindar servicios para realizar una conexión a ella a través de un API de programación. La evaluación de este parámetro se realiza sobre la base de la existencia o no en la herramienta de este indicador, identificándose con las siglas **API**.
- **Usabilidad**: Este indicador refleja cuan intuitiva es para el usuario la herramienta analizada contemplándose principalmente la facilidad de uso. La evaluación de este parámetro se realiza sobre la base de la existencia o no en la herramienta de este indicador, identificándose con la sigla **U.**
- **Licencia**: En este indicador se mostrará el tipo de licencia bajo la cual fue creada la herramienta analizada. Para identificarlo se utilizará la sigla **L**.
- **Funcionalidad**: Este indicador hace referencia a la capacidad que posee la herramienta analizada por si sola (sin la utilización de *plugins*) de llevar a cabo los dos modos de actuación de la gestión de red (monitorización y control). Este indicador es evaluado teniendo en cuenta la monitorización

de servicios, hardware y sistema operativo, la compatibilidad en el cliente, la generación de gráficas, informes y estadísticas, y el envío de alarmas y notificaciones en la herramienta a analizar, contemplándose en **Alta**, cuando posee tres o más de estas características, y **Baja** en caso contrario. Para identificarlo se utilizará la sigla **F**.

- **Tecnología de gestión**: Este indicador mostrará el protocolo que utiliza la herramienta analizada para llevar a cabo las funciones de gestión. Para identificarlo se utilizará la sigla **TE**.

TABLA 1.3: EVALUACIÓN DE HERRAMIENTAS DE MONITORIZACIÓN (34)

HERRAMIENTAS	ST	S	API	U	L	F	TE
Nagios	Si	Alto	No	No	GPL	Alta	SNMP
Zabbix	Si	Alto	Si	Si	GPL	Alta	SNMP
Munin	Si	Bajo	No	Si	GPL	Alta	SNMP
Cacti	Si	Bajo	No	Si	GPL	Baja	SNMP
Zenoss	Si	Alto	No	Si	GPL	Alta	SNMP

Concluida la evaluación se decide elegir Zabbix. Entre las principales características que justifican esta elección se encuentran el hecho de que Zabbix lleva a cabo estas acciones sin la adición de *plugins*, permitiendo que esta herramienta contenga todas las funcionalidades que permiten evaluar la propuesta de arquitectura híbrida. Además, Zabbix brinda la posibilidad de crear gráficos automáticamente con los que apoyar la presentación de los datos en los que se observa la evolución del parámetro en el tiempo. Junto a esos gráficos por defecto, el usuario puede crear sus propios gráficos personalizados para mostrar los valores de monitorización que estime oportunos. Otros elementos que sirven como basamento de esta elección son la alta facilidad de extensión que brinda la herramienta y la posibilidad de encontrar mucha información y documentación asociada a la misma.

1.7 CONTROLADORES SDN

En la actualidad existen muchos controladores SDN de código abierto que permiten el despliegue de la tecnología SDN, así como el soporte de gran cantidad de aplicaciones. Para el desarrollo de esta investigación se realiza la evaluación de cinco controladores que actualmente se utilizan en implementaciones de SDN. A continuación, se muestra un resumen con las principales características de dicho conjunto de herramientas:

NOX

Es un controlador SDN de código abierto que fue desarrollado por Nicira y donado a la comunidad de investigación e 2008, por lo que se utiliza a menudo en investigaciones en las redes académicas para desarrollar aplicaciones SDN y protocolos de red. NOX constituye un controlador esencial y un *framework* basado en componentes para el desarrollo de aplicaciones SDN. Proporciona módulos de soporte específicos para OpenFlow. El núcleo NOX ofrece métodos auxiliares y APIs para interactuar con los conmutadores de OpenFlow, incluyendo un motor de eventos (21).

Reciente el uso académico de NOX se ha generalizado ya que su código está disponible para la emulación de un conmutador de aprendizaje y un conmutador lógico de toda la red, que puede ser utilizado como código de arranque para varios proyectos de programación y experimentación (35).

POX

Es un controlador SDN de código abierto heredado del controlador NOX basado en Python. Este controlador, puede considerarse una plataforma para el rápido desarrollo y creación de prototipos de aplicaciones de red. El objetivo principal de POX es la investigación y se utiliza para explorar la depuración de las SDN, la virtualización de red, diseño de controlador y modelos de programación (36).

POX respecto a su predecesor NOX incorpora componentes reutilizables para la selección de rutas y el descubrimiento de topología, tiene un mejor rendimiento que las aplicaciones de NOX escritas en Python, posee una interfaz OpenFlow denominada *Pythonic*, que le permite el desarrollo en Python y puede correr en cualquier plataforma: Linux, Windows, Mac OS y otras, ya que se puede combinar con *PyPy*, entorno de ejecución que permite soportar la ejecución de programas escritos en Python (36).

Ryu

Es un controlador SDN de código abierto desarrollado en Python basado en componentes. Contiene un conjunto de componentes predefinidos los cuales pueden ser modificados, extendidos y diseñados para crear aplicaciones personalizadas. Cualquier lenguaje de programación puede ser utilizado para crear dichos componentes (11).

Ryu proporciona un controlador de lógica centralizada y una API bien definida que facilita la creación de nuevas aplicaciones de gestión de red a los operadores. Además, es compatible con varios protocolos para gestionar los dispositivos de red, como son OpenFlow (1.0, 1.2, 1.3 y extensiones Nicira), NETCONF y OF-CONFIG, entre otros. El objetivo de Ryu es el desarrollo de un sistema operativo para las SDN que tenga la suficiente calidad para su uso en un entorno de trabajo de gran tamaño (11).

Trema

Trema es un *framework* OpenFlow programable para el desarrollo de controladores OpenFlow. Fue creado originalmente por NEC con aportes posteriores de código abierto bajo la licencia GPL. A diferencia de los controladores OpenFlow más convencionales que lo precedieron, Trema proporciona servicios de infraestructura como parte de sus módulos básicos que apoyan el desarrollo de módulos de usuario (aplicaciones Trema). Los desarrolladores pueden crear sus módulos de usuario en Ruby o C, este último se recomienda cuando la velocidad de ejecución se convierte en una preocupación), por lo que entre sus objetivos de diseño más importantes son código de fácil escritura y el rendimiento (13).

Trema constituye un controlador SDN rápido, modular y sencillo, con un entorno de desarrollo que produce resultados con una base de código más pequeña (13).

Floodlight

Floodlight es un controlador SDN de código abierto desarrollado en Java, el cual soporta conmutadores OpenFlow físicos y virtuales. Nacido de Beacon, el controlador Floodlight consta de un conjunto de módulos, donde cada módulo provee un servicio a otros módulos y a la lógica de control de aplicación a través de interfaces REST o Java. El controlador puede ejecutarse sobre Linux, Mac y Windows (15).

El controlador Floodlight tiene la intención de ser una plataforma para una amplia variedad de aplicaciones de red. Floodlight se puede ejecutar como un *plugin* de red para OpenStack, herramienta de computación en la nube para desarrollo de aplicaciones (37). Una vez que el controlador está integrado en OpenStack, los administradores pueden proporcionar dinámicamente los recursos de red junto a otros recursos informáticos físicos y virtuales al usuario. Esto mejora la flexibilidad y el rendimiento general (15).

1.7.1 Evaluación de controladores SDN

Para realizar la propuesta de arquitectura se comparan los controladores SDN antes descritos teniendo en cuenta: soporte de protocolos OpenFlow (OpenFlow y OF-CONFIG), virtualización, interfaz gráfica de usuario o GUI (*Graphic User Interface*, por sus siglas en inglés), soporte de APIs REST, documentación existente sobre los mismos, lenguaje de programación, sistema operativo y soporte de versiones del protocolo OpenFlow y OpenStack. Los resultados de la comparación se muestran en la Tabla 1.3 (38).

TABLA 1.4: EVALUACIÓN DE CONTROLADORES SDN (38)

Característica	NOX	POX	Ryu	Trema	Floodlight
Protocolos OpenFlow	OpenFlow	OpenFlow	OpenFlow OF-CONFIG	OpenFlow	OpenFlow
Virtualización	Mininet y Open v*Switch*	Mininet y Open v*Switch*	Mininet y Open v*Switch*	Herramienta virtual de simulación incorporada	Mininet y Open v*Switch*
Interfaz Gráfica	Sí	Sí	Sí	No	Web (usando REST)
REST API	No	No	Si	No	Si
Documentación	Media	Pobre	Media	Media	Buena
Lenguaje en que está desarrollado	C++	Python	Python	Ruby/C	Java y cualquier lenguaje que utilice REST
Soporte de Sistemas Operativos	Linux	Linux, Mac OS y Windows	Mayormente soportado por Linux	Linux	Linux, Mac OS y Windows
Soporte OpenFlow	OF v1.0	OF v1.0	OF v1.0, v1.2, v1.3	OF v1.0	OF v1.0
Soporte OpenStack	No	No	Fuerte	Débil	Medio

En base a las características descritas de los controladores estudiados se define utilizar en la investigación el controlador Floodlight ya que este cuenta con una documentación buena, con soporte multiplataforma y con una interfaz gráfica web que facilita la visualización de los elementos que conforman la topología de red SDN.

1.8 HERRAMIENTAS DE SIMULACIÓN DE RED

Para aplicar y evaluar la propuesta de arquitectura SDN híbrida se emplea un software de simulación de redes que a través de una interfaz gráfica de usuario permite la creación de una topología simulada de red física que permite configurar y gestionar los dispositivos, todo esto sin necesidad de hardware de red dedicado, como routers y switches. Para la selección de esta herramienta de simulación se tienen en cuenta un conjunto de soluciones de este tipo, cuyas características y funcionalidades se describen a continuación.

Packet Tracer

Packet Tracer es una herramienta gráfica de simulación de red desarrollada por Cisco que ofrece un entorno de simulación basado en el aprendizaje para diseñar y configurar soluciones sobre equipamiento de Cisco como parte del plan de capacitación CCNA (*Cisco Certified Network Associate*, por sus siglas en inglés). (39).

Packet Tracer permite a los usuarios crear topologías de red, configurar dispositivos, insertar paquetes y simular una red con múltiples representaciones visuales. Soporta los protocolos HTTP, HTTPS, DHCP, DHCPv6, Telnet, SSH, TFTP, DNS, TCP, UDP, IPv4, IPv6, ICMP, ICMPv6, ARP, IPv6 ND, FTP, SMTP, POP3, entre otros (39).

Esta herramienta ofrece una interfaz de usuario muy fácil de manejar con dos espacios de trabajo donde el usuario puede realizar las soluciones: el lógico y el físico. El espacio de trabajo lógico permite la creación de la topología de red, la configuración de los diferentes dispositivos de red, la interconexión de los mismo mediante interfaces de varios tipos y el soporte de redes remotas multiusuario (Figura 1.4). Por su parte, el espacio de trabajo físico proporciona una visión general de la topología de red creada, la distribución del equipamiento físicamente, la estructura del cableado y la administración de cobertura inalámbrica. Además, Packet Tracer incluye un modo de simulación que permite el control y el análisis del tráfico de los paquetes en la red, así como de sus campos y los valores que contienen (39).

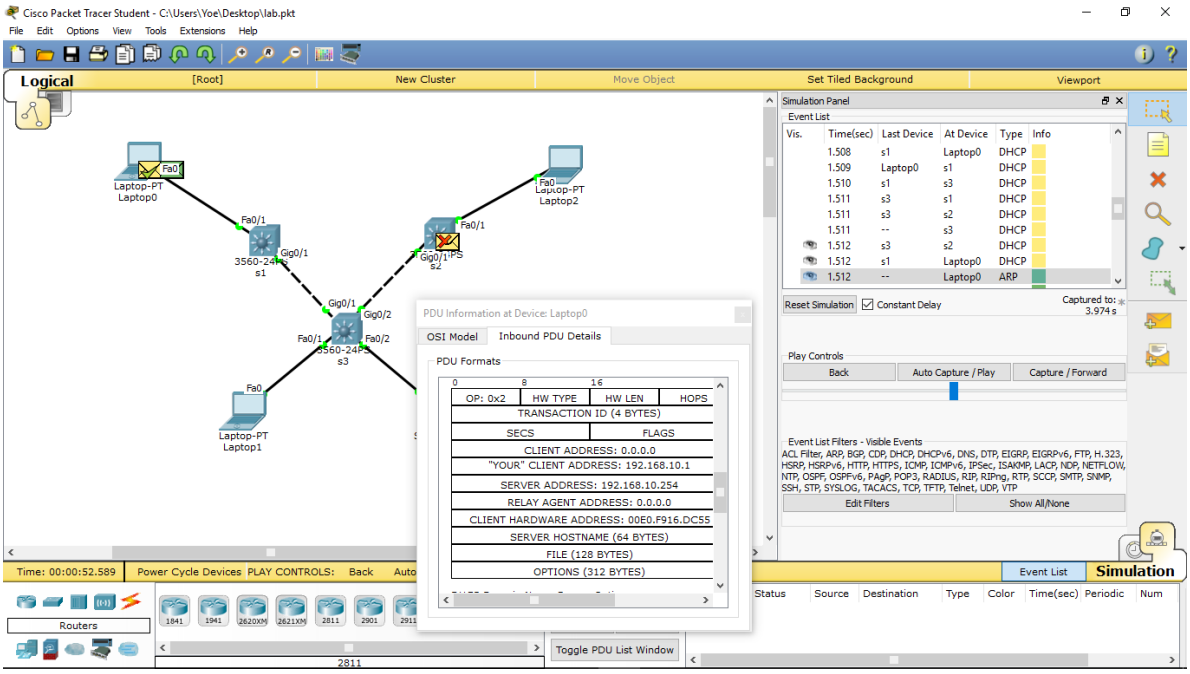

Figura 1.4 - Espacio de trabajo lógico en Packet Tracer (elaboración propia)

Este simulador en general, constituye una herramienta educativa que permite la creación de soluciones de red de mediana complejidad y el análisis de la estructura y el funcionamiento de una red, sin embargo, es propietario y por ende se debe pagar una licencia para instalarlo. Además, la simulación de una red de más complejidad se ve limitada ya que no se pueden desplegar aplicaciones de terceros en el equipamiento lo que no permite la ejecución de funcionalidades agregadas en los dispositivos y el despliegue de servicios que no soporta Packet Tracer de manera nativa.

eNSP

eNSP (*Enterprise Network Simulation Platform*) es una herramienta de simulación de red libre, extensible y gráfica desarrollada por Huawei. Esta aplicación permite el despliegue de soluciones sobre equipamiento de Huawei en diferentes escenarios de red (40).

eNSP proporciona una interfaz de usuario gráfica conveniente para simplificar las operaciones en una gestión de redes compleja (Figura 1.5) y les permite a los usuarios ver los modelos de dispositivo y obtener ayuda en línea sobre configuración del equipamiento y documentación de dispositivos (40).

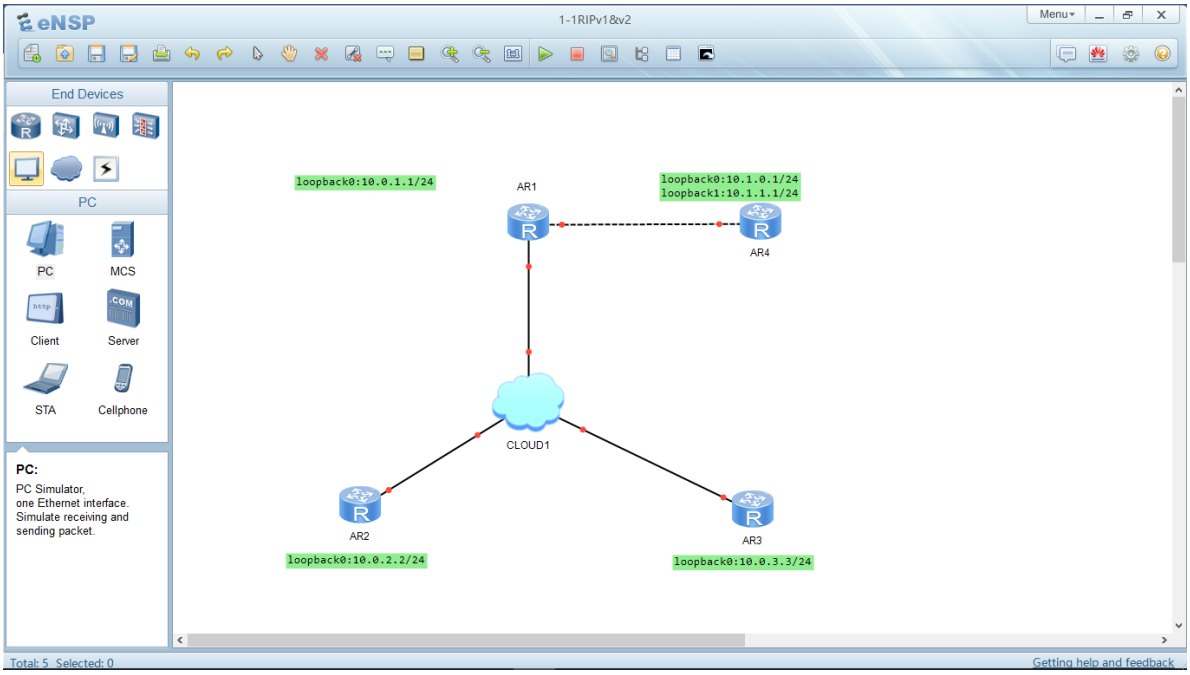

Figura 1.5 - Interfaz gráfica de eNSP (elaboración propia)

eNSP puede enlazar los adaptadores en red físicos para llevar a cabo la conexión entre los dispositivos simulados y los dispositivos reales, permitiendo una gestión de redes flexible. Cuenta también con soporte para VirtualBox lo que permite implementar nuevos servicios a través de nodos virtuales. Además del despliegue de un solo nodo, el eNSP soporta también el despliegue distribuido. En el modo del despliegue distribuido, el servidor del eNSP se despliega en los múltiples servidores, formando una red compleja, y se asignan recursos en los servidores automáticamente (40).

eNSP en general, proporciona una plataforma de la simulación basada en redes que es fácil de usar y apoya las interfaces gráficas del usuario o GUI (*Graphic User Interface,* por sus siglas en inglés) extensibles

GNS3

GNS3 es un software de código abierto de simulación de red que permite diseñar topologías de redes complejas y poner en marcha simulaciones sobre ellas (41).

GNS3 tiene una interfaz gráfica intuitiva (Figura 1.6) que permite la gestión de dispositivos a través de un motor de emulación denominado Dynamips que permite a los usuarios ejecutar imágenes binarias del IOS de Cisco Systems. Dynamips permite probar y experimentar las capacidades del IOS de Cisco, revisar configuraciones rápidas para luego utilizarlas en routers reales, y especifica configuraciones de hardware para routers virtuales específicos (41).

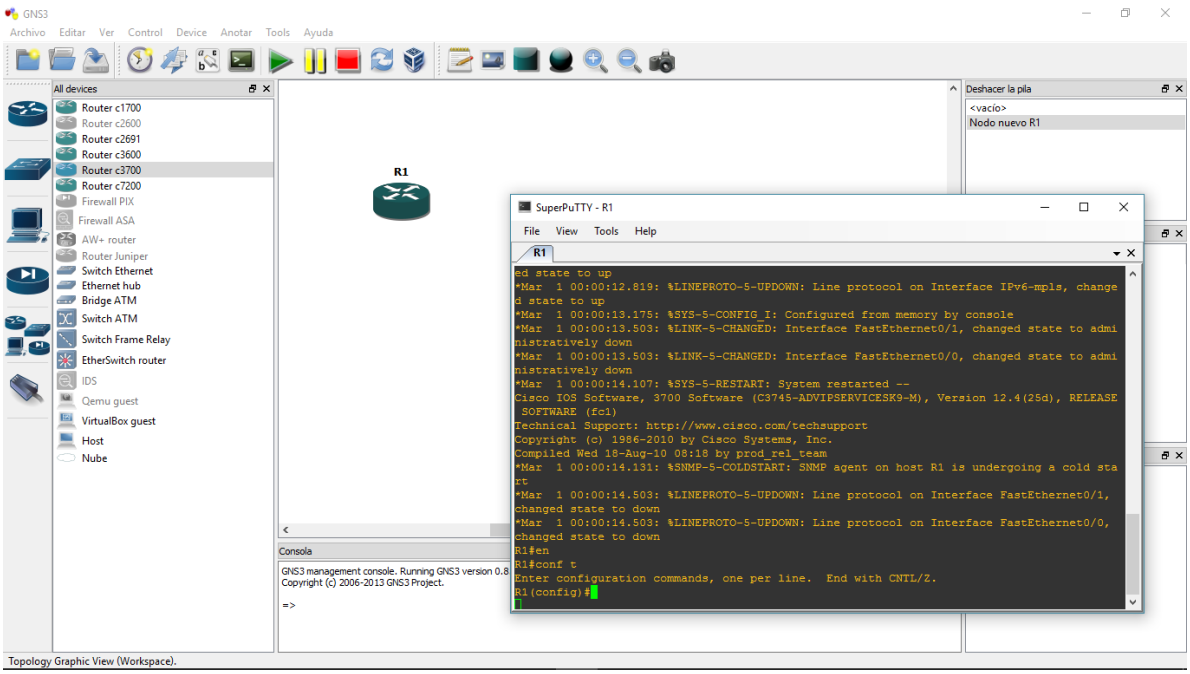

Figura 1.6 - Interfaz gráfica de GNS3 (elaboración propia)

GNS3 se puede utilizar para experimentar con características o para comprobar configuraciones que necesitan ser desplegadas más adelante en dispositivos reales e incluye funcionalidades que permiten, por ejemplo, la conexión de la red virtual con redes reales y capturas de paquetes utilizando aplicación Wireshark (42). Además, provee administración vía comandos para listar los dispositivos, iniciar, detener, recargar, suspender, resumir, y conectar a las consolas de los routers virtuales (41).

Entre algunas de las funcionalidades de GNS3 están las siguientes (41):

- Diseño de alta calidad de topologías de red complejas.
- Emulación de varias plataformas de Cisco IOS en routers, switch, firewalls PIX, entre otros.

- Simulación conectores simple Ethernet, ATM y Frame Relay.
- Captura de paquetes.
- Vinculación con aplicaciones de máquinas virtuales.

OMNeT++

OMNeT++ es un software simulador de red enfocado al área académica y orientado a modelar y simular eventos discretos en redes de comunicaciones a través de la recreación de dichos eventos discretos por módulos orientados a objetos. OMNeT++ es una versión libre, de la versión comercial OMNEST desarrollado por Omnest Global, Inc. Esta herramienta se puede ejecutar perfectamente sobre sistemas operativos Windows y sobre algunas versiones de UNIX y Linux, usando varios compiladores de C++ (43).

Este simulador, utiliza el lenguaje de programación NED, que se basa en el lenguaje C++; como herramienta para modelar topologías de red; este lenguaje facilita la descripción modular de una red, es decir, un modelo en OMNeT++ se construye con módulos jerárquicos mediante el lenguaje NED, dichos módulos pueden contener estructuras complejas de datos y tienen sus propios parámetros usados para personalizar el envío de paquetes a los destinos a través de rutas, compuertas y conexiones. componentes y especificaciones de la descripción de una red de comunicaciones.

Con el fin de facilitar el diseño de redes y la simulación de eventos sobre las mismas, OMNeT++, permite al usuario trabajar gráficamente empleando el editor del lenguaje NED (GNED). Este editor es la interfaz gráfica que permite crear, programar, configurar y simular redes de comunicaciones, sin necesidad de hacerlo utilizando la codificación del lenguaje NED (Figura 1.7). Las simulaciones en OMNeT++ pueden utilizar varias interfaces de usuario, dependiendo del propósito. La interfaz más avanzada permite visualizar el modelo, controlar la ejecución de la simulación y cambiar variables/objetos del modelo. Esto facilita la demostración del funcionamiento de un modelo.

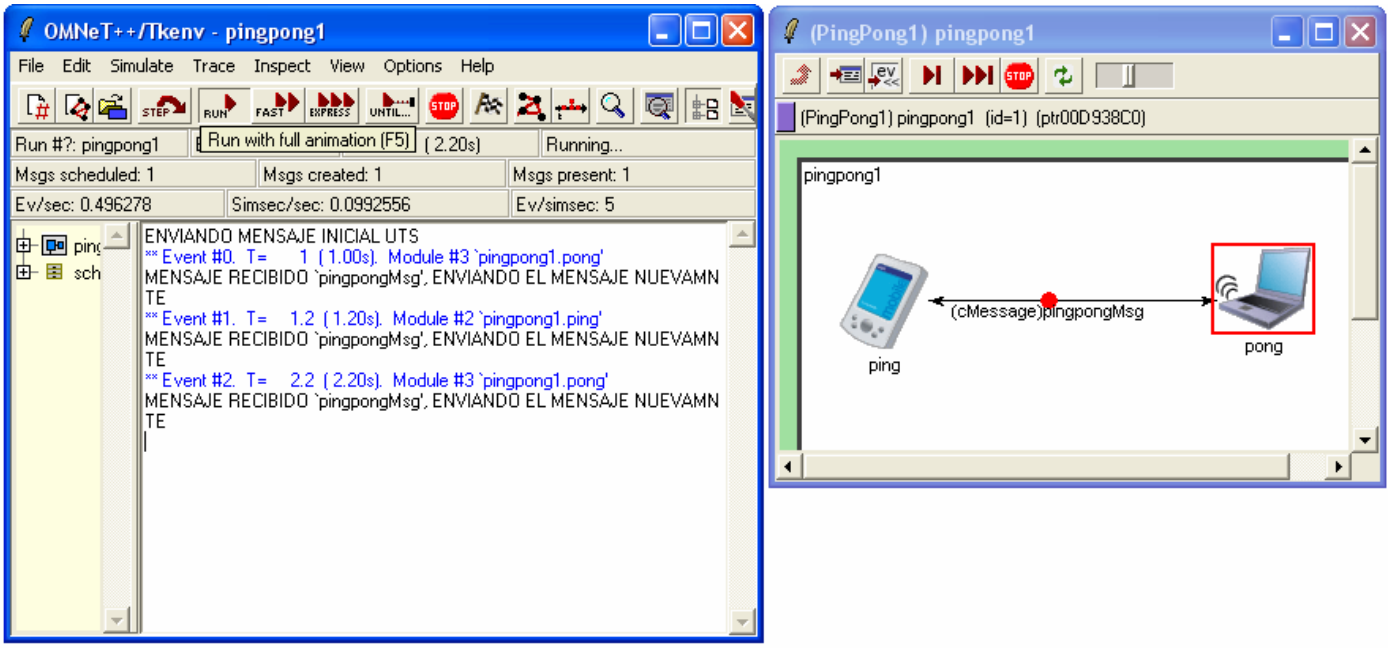

Figura 1.7 - Interfaz gráfica de OMNeT++ (elaboración propia)

OMNeT++ es una herramienta multiplataforma gratuita solamente para propósitos académicos, lo que facilita su utilización en universidades y grupos de investigación. Sin embargo, para su uso en estos fines, tiene un alto grado de complejidad en su manejo ya que es necesario saber programar en lenguaje NED, pues el trabajo con el editor gráfico no permite acceder a todas las posibilidades de configuración.

1.8.1 Evaluación de las herramientas de simulación de red

A partir del análisis de estas herramientas se determinó utilizar el simulador gráfico de red GNS3 debido a que este permite implementar funcionalidades de gestión de red avanzadas para soluciones de red de gran complejidad. GNS3, además de incluir soporte para una gran cantidad de dispositivos, permite el despliegue de aplicaciones de terceros que brindan la posibilidad agregar y probar nuevas características a la simulación que luego pueden ser implementadas en un escenario real.

1.9 CONCLUSIONES PARCIALES

Luego de haber analizado el marco teórico de la investigación se ha podido arribar a las siguientes conclusiones:

- En el análisis del contexto actual de las Tecnologías de la Información y las Comunicaciones se comprueba que existen deficiencias en las redes de telecomunicaciones actuales, que hacen necesaria la implementación de una nueva tecnología de red que satisfaga las necesidades reales de usuarios y negocios.
- El análisis de SDN evidencia que esta tecnología permite mejorar significativamente tanto la capacidad de gestión y control, como la escalabilidad y agilidad de una red de información.
- El estudio de la gestión de red basada en políticas o BPNM demuestra que esta permite en redes tradicionales la automatización de los recursos en la red reduciendo tiempo, costos, y problemas asociados con la configuración de los dispositivos.

CAPÍTULO 2 : Propuesta de Arquitectura para la Gestión de SDN Híbrida

2.1 INTRODUCCIÓN

La tecnología SDN ha surgido como un nuevo paradigma que promete una transformación de las arquitecturas de red y de la gestión de las redes como conocemos en la actualidad a través de una mayor flexibilidad, capacidad de programación, gestión y rentabilidad. Un esquema híbrido, por su parte, permite desplegar tecnologías de SDN como OpenFlow y protocolos estándares de comunicación simultáneamente en el hardware físico. Para lograr los objetivos propuestos de la presente investigación se hace necesario profundizar en algunos de los elementos de la gestión basada en políticas, a fin de tener un mejor entendimiento de su funcionamiento y de cómo se fusionan en un modelo híbrido SDN y las tecnologías de redes tradicionales.

En este capítulo primeramente se describen cada uno de los elementos que conforman un sistema de operación centralizada de red basado en políticas, como elemento base del modelo SDN híbrido y se introduce la propuesta de arquitectura para la gestión de SDN híbrida a la que se le realiza un análisis de los elementos que la componen y la relación que existe entre los mismos. Además, se desarrolla un esquema metodológico para la aplicación de la arquitectura propuesta y se describen escenarios de aplicación de la misma.

2.2 ARQUITECTURA DE UN SISTEMA DE OPERACIÓN CENTRALIZADA DE RED

Un sistema es un conjunto de elementos, relacionados entre sí que contribuyen para alcanzar un fin. Los sistemas de redes, en específico, están compuestos por toda una serie de componentes que deben permitir mantener el control centralizado y la automatización de los activos de la red y el conocimiento de su estructura. A través de la gestión de red basada en políticas se puede lograr esto mediante la creación de políticas que permiten elevar la calidad de los servicios y el funcionamiento óptimo del equipamiento activo. Para lograr esto, conforme ilustra la Figura 2.1, el sistema tiene como elementos los módulos:

- Base de datos de gestión de configuraciones o CMDB (*Configuration Management Data Base, por sus siglas en inglés*): Se encarga de mantener el control de los activos de la red y conocer la estructura de la misma.

- Cliente de configuraciones o NMS.
- PEP: Son los equipos de red en los cuales se ejecutan las políticas.
- Gestor de políticas: Es el responsable de permitir la creación de políticas para lograr el funcionamiento deseado en la red.

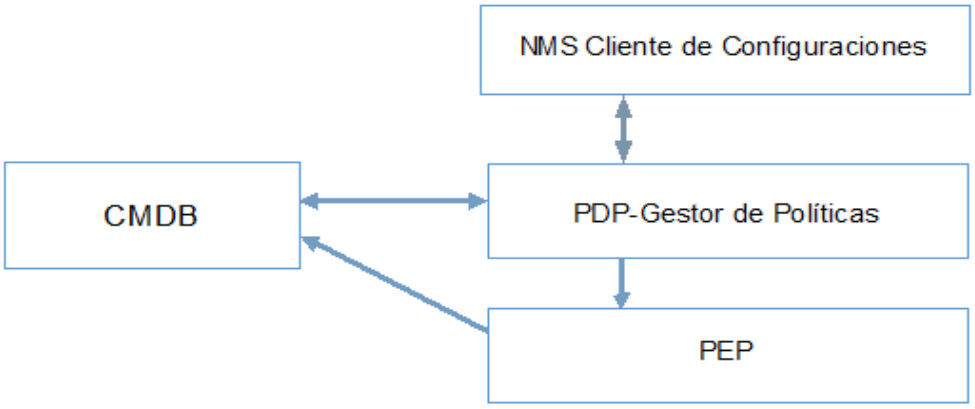

Figura 2.1 - Flujo de la información entre los componentes del sistema (elaboración propia)

Estos componentes están organizados de tal forma que la comunicación entre ellos permita garantizar la operación centralizada de la red y la automatización de las tareas de control para lograr el funcionamiento deseado del equipamiento activo.

2.2.1 Componente de gestión basada en políticas

El componente gestor de políticas o PDP se encarga de la automatización de las tareas de control; permite las definiciones de políticas teniendo en cuenta los parámetros establecidos para la operación de cada uno de los dispositivos que forman parte de la red. La Figura 2.2 es referente al funcionamiento de este componente.

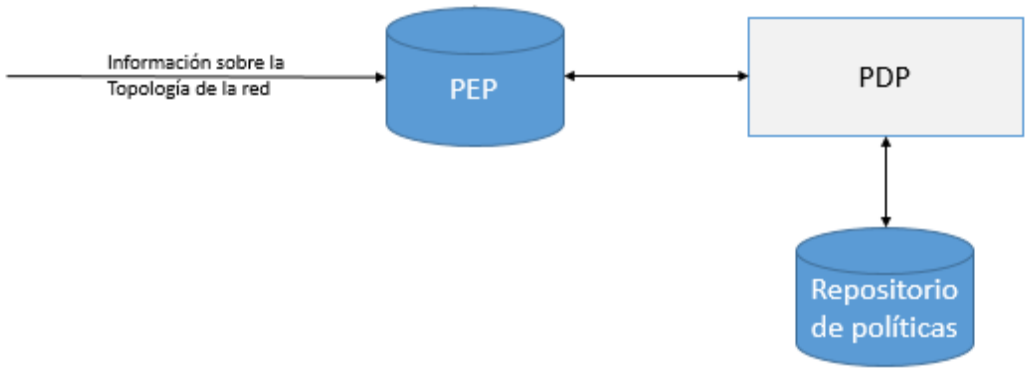

Figura 2.2 - Componte de gestión de políticas (elaboración propia)

El PDP se comunica con el componente de gestión de configuraciones solicitando información de los dispositivos que componen a la red, los cuales corresponden a los puntos de ejecución de políticas. La aplicación de políticas sobre los PEP depende del estado deseado de operación de cada uno de los dispositivos que forman parte de la red, las cuales reflejan el estado de los PEP.

2.2.2 Componente de base de datos de gestión de configuraciones

El componente base de datos de gestión de configuraciones o CMDB es el elemento base del sistema, sobre su estructura operan los demás módulos. La inclusión del módulo CMDB en el sistema tiene como finalidad resolver el aspecto relacionado a la inexistencia de un control de inventario (hardware y software), facilitando la localización física de equipos y servicios como parte de la operación de la red.

No siendo suficiente tener una lista de los activos de la red, el control inventario se extiende hasta el mantenimiento del registro actualizado de todos los activos y sus características, las interrelaciones entre ellos, los diferentes estados de operación deseados, los cambios efectuados en la red y la representación de la topología. A tal efecto, el módulo CMDB provee funcionalidades de los procesos gestión de configuraciones, de riesgos y de cambios conforme ilustra la Figura 2.3.

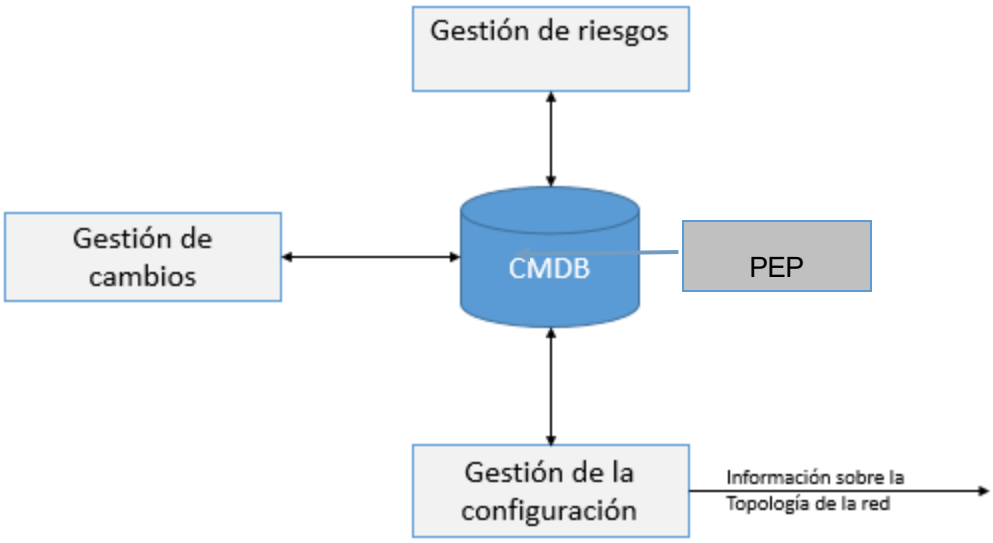

Figura 2.3 - Funcionalidades del componente CMDB (elaboración propia)

Cumpliendo estas funcionalidades, el módulo CMDB debe permitir verificar el cumplimiento de las políticas para llevar a cabo los controles de activos de la red. Sobre los activos controlados por este módulo opera el gestor de políticas.

2.3 PROPUESTA DE ARQUITECTURA SDN-PBNM

Para realizar la propuesta del modelo híbrido de red SDN que se proyecta en esta investigación hay que remitirse a la arquitectura planteada para PBNM como base dicho modelo por el Grupo de Trabajo de Ingeniería de Internet o IETF (*Internet Engineering Task Force*, por sus siglas en inglés) (44), en la Figura 2.4. IETF es una organización internacional abierta sin fines de lucro de normalización y estandarización que regula las propuestas de estándares de Internet conocidos como RFC (*Request for Comments*, por sus siglas en inglés) (45).

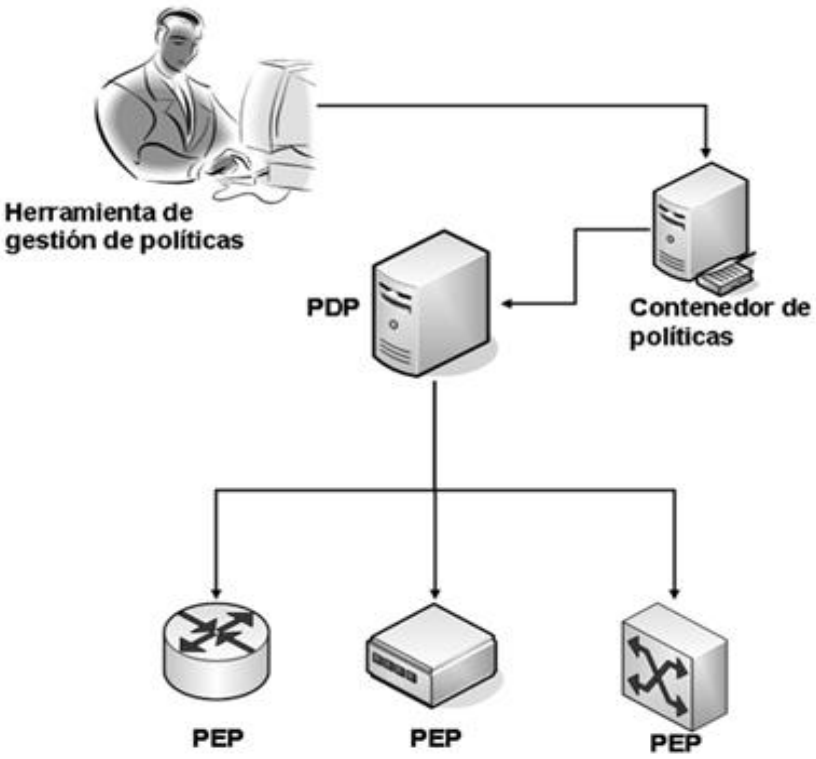

Figura 2.4 - Arquitectura propuesta por IETF de BPNM (44)

En esta arquitectura se identifican los elementos siguientes (26):

- Herramienta de gestión de políticas: Herramienta que ayuda a los administradores a crear políticas que se van a ejecutar en las redes.
- Contenedor de políticas: Entidad que almacena las políticas, las cuales pueden ser almacenadas utilizando un modelo estándar, un repositorio de políticas, un directorio LDAP o una base de datos relacional dependiendo de la implementación.
- PEP: Entidad donde las políticas son aplicadas.
- PDP: Entidad en donde se evalúan las políticas que posteriormente se ejecutan en el PEP.

A partir de este modelo de BPNM se presenta en la Figura 2.5 la arquitectura de SDN-PBNM.

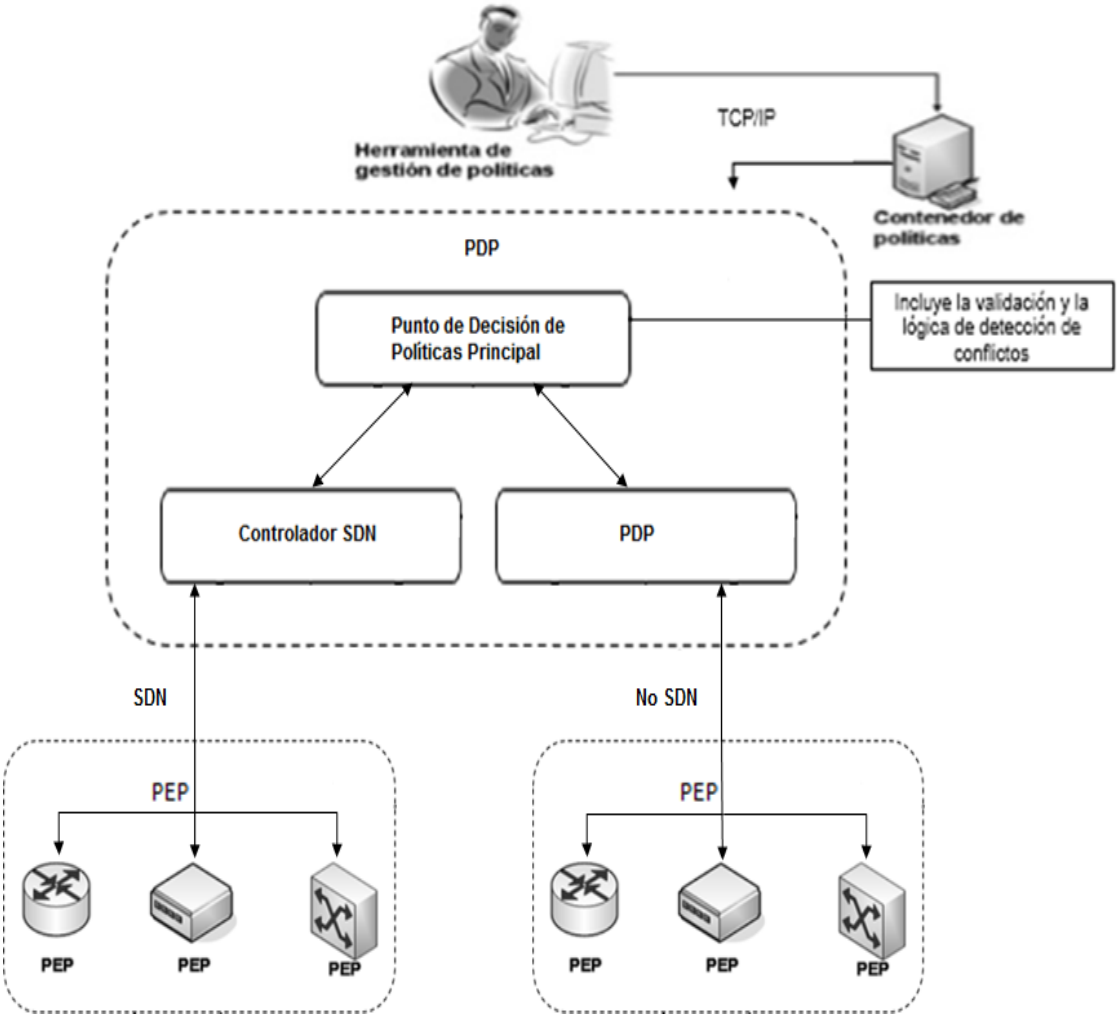

Figura 2.5 - Arquitectura SDN-PBNM (elaboración propia)

2.3.1 Funcionamiento de propuesta de arquitectura SDN-PBNM

En esta arquitectura SDN híbrida se define una abstracción dentro de PDP que constituye el componente híbrido que permitirá la interacción entre los diferentes tipos de redes.

Además de los elementos que conforman la arquitectura de PBNM se incluye, un controlador SDN, que como componente central de la arquitectura SDN, gestiona el comportamiento general de la red no tradicional a partir de los requerimientos de los dispositivos de la misma, un PDP secundario que controla

la ejecución de políticas en la red tradicional y un PDP principal que gestiona la ejecución de políticas entre las entidades de manejan el tráfico de la red SDN y no SDN (controlador SDN y PDP secundario respectivamente).

2.3.1.1 Controlador SDN

El controlador SDN es la entidad que gestiona los flujos de datos de la red SDN. En el mismo se manejan las peticiones entre los dispositivos activos de red, que constituyen los PEP, y las aplicaciones SDN que proveen los servicios de la red disponibles. El controlador SDN a su vez se encuentra subordinado a el PDP principal.

Es el controlador SDN quien implementa las políticas de gestión validadas por el PDP principal en el equipamiento activo de la red SDN. El controlador debe ser cualquiera de código abierto como NOX, POX, Beacon, Ryu, Trema, Floodlight, etc., ya que a estos controladores dada su implementación libre se le pueden realizar modificaciones con el fin de que además de realizar sus funciones básicas, permitan la comunicación con la entidad que evalúa la aplicación de políticas.

Actualmente, no hay funciones definidas para la gestión del controlador ni se ha propuesto alguna interfaz para su gestión. No obstante, en la arquitectura propuesta esta comunicación se realiza a través de Transferencia de Estado Representacional o REST (*Representational State Transfer*, por sus siglas en inglés), tal y como se describe en la Figura 2.6.

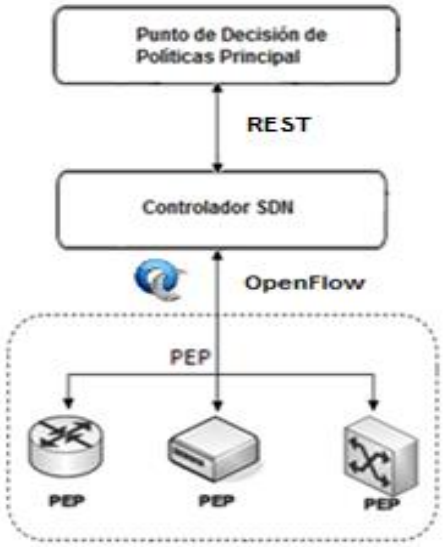

Figura 2.6 - Comunicación del controlador SDN (elaboración propia)

REST describe la interfaz que pueden utilizar dos entidades para comunicarse por medio de intercambio de HTTP para obtener datos o indicar la ejecución de operaciones sobre los mismos en cualquier formato, independientemente de la tecnología, el sistema operativo y la programación que implementen. La utilización de REST para intercambio de mensajes basados en XML (*eXtensible Markup Language*, por sus siglas en inglés) permite invocar procedimientos remotos de muchos lenguajes a través del protocolo de transporte TCP, por lo que se puede programar, interrogar y configurar numerosas funciones del controlador entre las que se encuentran la programación del reenvío de flujos y la topología. Esto permite, por lo tanto, una gran interoperabilidad entre el controlador SDN, que implementa protocolos de gestión propios de SDN como OpenFlow, y la entidad de aplicación de políticas principal.

Las aplicaciones que se implementan en la red SDN pueden ser desplegadas en el controlador SDN o fuera del mismo haciendo uso de REST. El controlador SDN será el encargado de traducir los requisitos de estas aplicaciones a los elementos de red y de suministrar información relevante como estadísticas, eventos, y la forma en que están conectados los recursos, a las aplicaciones SDN para que estas puedan realizar sus funciones de gestión.

OpenFlow, por su parte, dirige la comunicación entre el controlador y los PEP que lo soportan utilizando el concepto de flujos para identificar el tráfico de la red e instaurar reglas de conmutación, las cuales pueden ser estáticas o dinámicas. Esta comunicación entre el controlador y los dispositivos OpenFlow se efectúa de forma segura empleando el protocolo TLS para garantizar el cifrado del canal y, por tanto, parte de la seguridad de la red. El controlador SDN logra un control granular debido a la programación de la red basada en dichos flujos, lo que permite una respuesta rápida a los cambios en tiempo real de las aplicaciones, usuarios y servicios.

2.3.1.2 Punto de Decisión de Políticas Principal

El PDP principal es el componente de la arquitectura encargado de gestionar la ejecución de políticas que permiten la comunicación entre los diferentes tipos de redes. El PDP principal evalúa y aplica en las entidades pertinentes las políticas almacenadas en la herramienta de gestión de políticas que permiten la interoperabilidad entre la red SDN y la tradicional, tal y como se describe en la Figura 2.7.

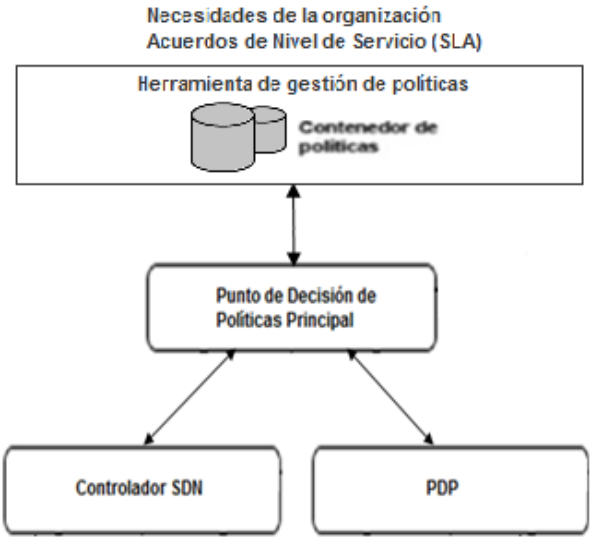

Figura 2.7 - Comunicación del PDP principal (elaboración propia)

El PDP principal obtiene estas reglas a través de consultas al contenedor de políticas. Estas políticas implementadas representan tanto las necesidades de la organización como las del cliente definidas mediante Acuerdos de Nivel de Servicio o SLA (*Service Level Agreement*, por sus siglas en inglés) y

constituyen las reglas que controlan el tráfico de información y la interacción entre el equipamiento de red con soporte OpenFlow y el equipamiento de red tradicional.

Este componente constituye la entidad que incluye la validación y la lógica de detección de conflictos de las políticas del PDP secundario y las que se aplican en el controlador SDN. Este puede interactuar con el PDP secundario mediante protocolos como SNMP y COPS y con el controlador SDN a través de REST. El PDP principal resuelve el punto único de falla en el controlador SDN cuando este no está disponible permitiendo legar el tráfico de la red SDN al PDP secundario. Además, permite un control centralizado sobre los dispositivos que no soportan SDN.

2.3.1.3 Punto de Decisión de Políticas Secundario

El PDP secundario, por su parte, constituye la entidad que de forma nativa gestiona la ejecución de políticas en el equipamiento de red tradicional que no cuenta con soporte para SDN. Este componente interactúa con el PDP principal y con los PEP que no soportan el protocolo OpenFlow a través de protocolos de gestión como SNMP y COPS, tal y como se describe en la Figura 2.8.

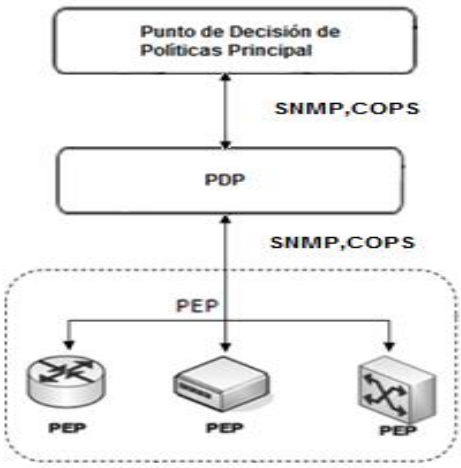

Figura 2.8 - Comunicación del PDP secundario (elaboración propia)

El PDP secundario evalúa y aplica las políticas en los PEP de la red tradicional, que, a través de un agente, recogen y almacenan información de administración, la cual es puesta a disposición de una herramienta de

monitorización que la supervisa y controla. Esta información es la que permite la validación de las políticas almacenadas en el repositorio de políticas.

Además, en el caso de una falla en el controlador SDN la autoridad encargada de aplicar políticas sobre la red SDN, el PDP principal, redirige el control de la red hacia el PDP secundario que asume la ejecución de políticas sobre los PEP de la red SDN, convirtiéndose de manera temporal en el PDP que centraliza la gestión de las dos redes, mientras tanto no exista disponibilidad del controlador SDN.

2.3.2 Gestión de propuesta de arquitectura SDN-PBNM

La gestión de la propuesta de arquitectura SDN-PBNM debe recopilar información que permita determinar el rendimiento percibido por los usuarios finales para identificar anomalías y fallos en el desempeño y estabilidad de la red. En la propuesta realizada, basado en la experiencia del empleo de la gestión de la red SDN en la Universidad de Stanford, se realiza la monitorización de las CPUs de los *switches* habilitados con OpenFlow debido a que es una métrica determinante para el desempeño de la red SDN, así como la tasa de transferencia para medir la rendimiento y estabilidad de la red tradicional.

Debido a que la amplia mayoría de los *routers* y *switches* tradicionales soportan un agente SNMP, se realiza la obtención de estas métricas a través de un gestor SNMP en el sistema de monitoreo y control del administrador de red.

Se propone, además, utilizar la herramienta Iperf para medir la calidad de los enlaces de red. De esta forma se puede medir el ancho de banda, la tasa de pérdidas de datagramas y la latencia entre dos pares de nodos en un enlace.

2.4 APLICACIÓN DE LA PROPUESTA DE ARQUITECTURA SDN-PBNM

Para aplicar la arquitectura SDN-PBNM propuesta se deben seguir una serie de procedimientos para integrar los elementos de la tecnología SDN en las redes tradicionales. Dichos procedimientos se describen en la Figura 2.9 a continuación:

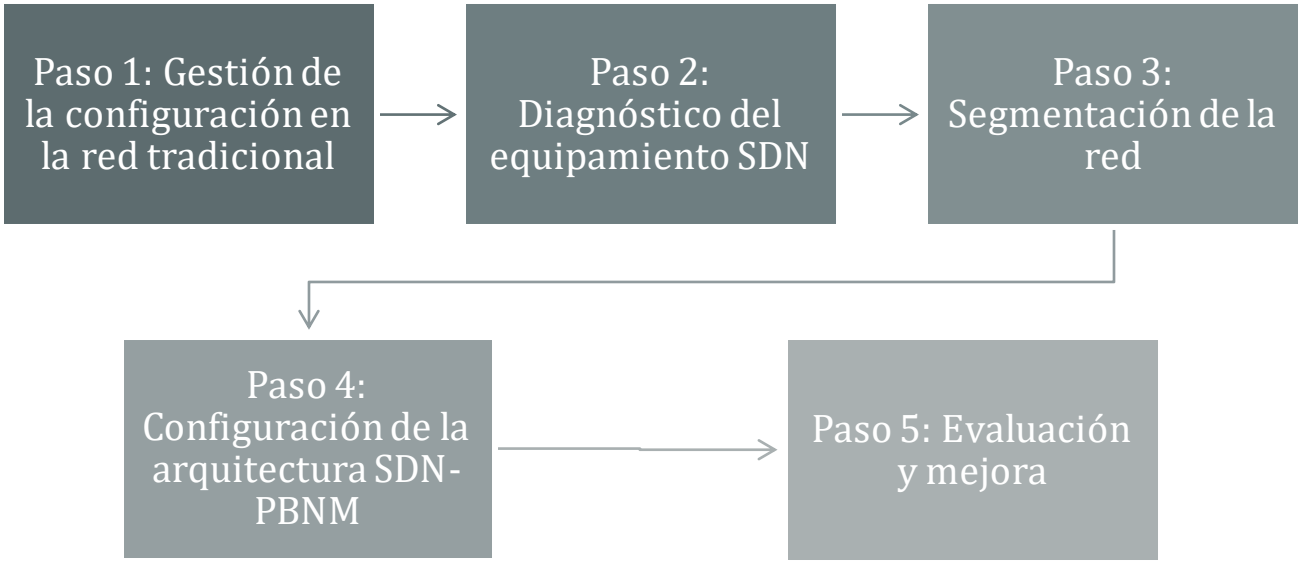

Figura 2.9 - Procedimiento para despliegue de arquitectura SDN-PBNM (elaboración propia)

Gestión de la configuración en la red tradicional

En este paso se configuran las herramientas de gestión de la configuración que mantienen un registro de los tiempos de uso de la red, de los servidores, del procesador, de las impresoras y de otros dispositivos. Esta información se guarda en una base de datos utilizando software de inventario con el que el administrador puede generar informes a efectos de planificación. La mayoría de los sistemas de red actuales disponen de aplicaciones de control de inventario que puede utilizarse para obtener la información necesaria para resolver determinados problemas en la red.

Diagnóstico del equipamiento SDN

Para realizar un correcto despliegue de la arquitectura SDN-PBNM se necesita verificar que el equipamiento activo de red con el que se dispone cuente con soporte para la tecnología SDN. Para ello el equipamiento debe tener soporte para el protocolo OpenFlow o una actualización de firmware para proporcionar soporte a este. El Anexo 3 especifica una lista de hardware compatible con el mismo.

Segmentación de la red

La segmentación de la red se realiza a través de la creación de redes lógicas independientes dentro de una la red física, también denominadas VLANs, que permiten la administración de diferentes segmentos lógicos de una red de área local y reducen el tamaño del dominio de difusión. Esto permite verificar la funcionalidad del protocolo OpenFlow mediante la adición de una VLAN experimental en la cual se despliega la red SDN que será gestionada por un controlador.

Configuración de la arquitectura SDN-PBNM

La configuración de la arquitectura SDN-PBNM se lleva a cabo a partir de la configuración del controlador SDN y de las políticas que van a gestionar el tráfico de la red. Para ello se deben implementar dichas políticas en el PDP principal, que gestiona la comunicación entre las redes SDN y no SDN, y el PDP secundario que controla el tráfico de la red tradicional.

Evaluación y mejora

Para la evaluación y la mejora de la arquitectura SDN-PBNM se deben tener en cuenta aspectos como la disponibilidad, que se determina a partir de los tiempos de funcionamiento y detección y corrección de errores o fallas en el equipamiento, el rendimiento que se comprueba a partir de la fijación de umbrales dentro de los indicadores que se monitorizan en función de las necesidades de la organización y de la infraestructura, los cuáles se convierten en condiciones a evaluar en el PDP que permiten identificar y corregir anomalías y comportamientos incorrectos, y la estabilidad que se establece midiendo dichas estadísticas durante un largo período de tiempo para verificar fallas en el desempeño y la salud de la red.

2.5 CONCLUSIONES PARCIALES

Una vez estudiado los elementos asociados a la gestión basada en políticas, como elemento base del modelo SDN híbrido, y a la propuesta de arquitectura SDN híbrida se puede arribar a las siguientes conclusiones:

- El análisis de un sistema de operación centralizada de red basado en políticas demuestra que gestión de red basada en políticas constituye el mecanismo idóneo para conformar la propuesta de arquitectura SDN híbrida debido a que resuelve mediante la automatización del punto de falla de la red SDN cuando el controlador deja de estar disponible.
- La modelación de la propuesta de arquitectura SDN híbrida permite que en una misma red coexista la tecnología SDN con el equipamiento de red tradicional.
- La descripción de un procedimiento para implementar la arquitectura SDN-PBNM constituye una metodología para aplicar la arquitectura propuesta en un escenario de red real.

CAPÍTULO 3 : Evaluación de Propuesta de Arquitectura para la Gestión de SDN Híbrida

3.1 INTRODUCCIÓN

Para cumplir los objetivos de esta investigación en este capítulo la propuesta de arquitectura SDN híbrida se evalúa en un entorno de trabajo simulado, el cual permite una mejor visualización del equipamiento de red, así como de sus configuraciones y comunicación con el resto de los elementos que conforman el modelo híbrido. Para realizar la simulación se utiliza la herramienta de simulación GNS3 en una computadora de escritorio ASUS, que cuenta con las siguientes características:

- Procesador Intel Core i3 4100.
- Sistema Operativo Windows 10 Pro de 64 bits.
- Memoria RAM de 4GB.

3.2 ESCENARIO DE EVALUACIÓN DE LA PROPUESTA

Para un mejor entendimiento de la propuesta de arquitectura híbrida SDN-PBNM se describen sus funcionalidades y características a través de un escenario[3] de simulación (Figura 3.1). La configuración de dicho escenario en la herramienta de simulación GNS3 se detalla en el Anexo 3.

[3] En el resto de la investigación se considera el término escenario como el entorno de simulación en donde se evalúa la arquitectura SDN-PBNM.

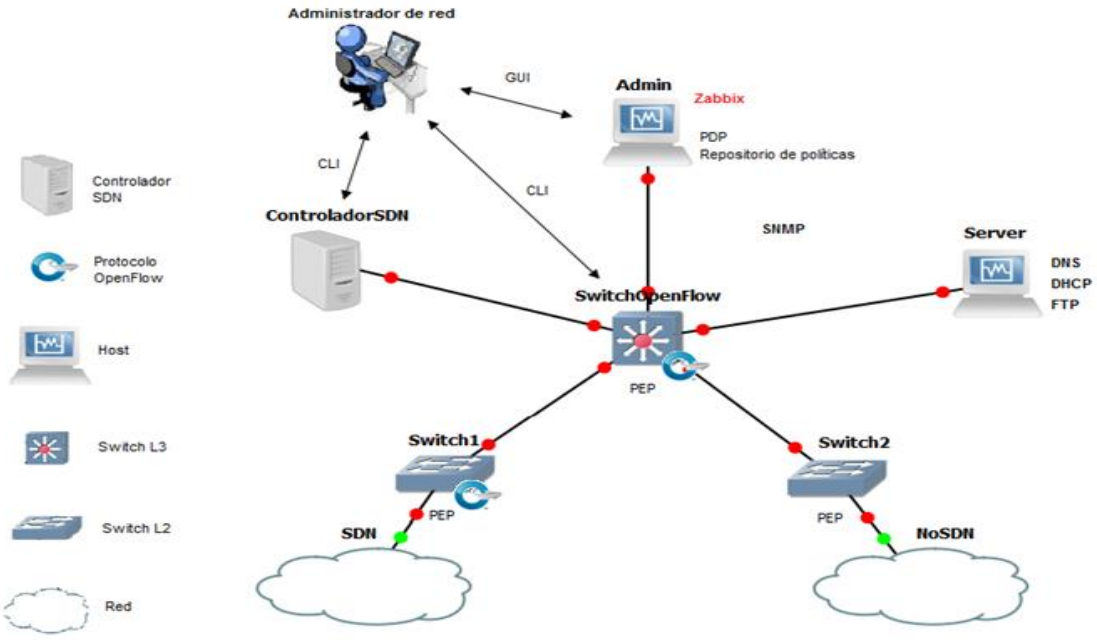

Figura 3.1 - Escenario general de la arquitectura SDN-PBNM (elaboración propia)

El escenario representa una red LAN en la que se implementa un controlador que gestiona la red SDN a través del equipamiento híbrido identificado por el protocolo OpenFlow. Se visualizan, además, un host denominado **Admin** sobre el que se implementan el repositorio de políticas y la aplicación de monitoreo y control de red Zabbix, y un host **Server**, el cual presta servicios como DNS, DHCP y FTP a la infraestructura de red tradicional.

En este escenario se identifican los elementos que componen la arquitectura SDN-PBNM. La Tabla 3.1 describe cada uno de los dispositivos que contiene el escenario y la función que realizan como parte la arquitectura propuesta.

Tal y como se muestra en la figura, el administrador de red realiza la configuración sobre los hosts a través de GUI y en el controlador y los switches de red mediante CLI. Todos los dispositivos se interconectan utilizando enlaces Fast Ethernet.

TABLA 3.1: DISPOSITIVOS DEL ESCENARIO GENERAL DE LA ARQUITECTURA SDN-PBNM (ELABORACIÓN PROPIA)

NOMBRE	FUNCIÓN	DESCRIPCIÓN
Server	-	Host que va a alojar servidores DNS, DHCP y FTP
Admin	Herramienta de gestión de políticas,Contenedor de políticas,PDP principal,PDP secundario	Host que va a alojar la herramienda de monitoreo y control Zabbix, en la que se almacenan y gestionan las políticas de la red híbrida.
ControladorSDN	Controlador SDN	Host que va a alojar el controlador SDN y los aplicaciones que se implementan sobre la red SDN.
SwitchOpenFlow	PEP	Switch de capa 3 que se va encargar de gestionar el tráfico entre los diferentes dispositivos y el enrutamiento de paquetes.
Switch1	PEP	Switch de capa 2 que va a conmutar el tráfico de la red SDN hacia SwitchOpenFlow y viceversa.
Switch2	PEP	Switch de capa 2 que va a conmutar el tráfico de la red no SDN hacia SwitchOpenFlow y viceversa.
SDN	-	Red física SDN.
NoSND	-	Red física no SDN.

3.2.1 Configuración básica de dispositivos

Teniendo el escenario de red definido se procede a realizar la configuración que permita lograr la comunicación entre los diferentes dispositivos de red que componen la simulación partiendo de la Tabla 3.2 que representa el direccionamiento de los dispositivos de red y la Tabla 3.3 en la que se describen las asignaciones iniciales de los puertos. Dicha configuración básica se detalla en el Anexo 4.

TABLA 3.2: DIRECCIONAMIENTO (ELABORACIÓN PROPIA)

NOMBRE	INTERFAZ	DIRECCIÓN IP	MÁSCARA DE SUBRED	GATEWAY
Server	NIC	192.168.1.9	255.255.255.0	192.168.1.3
Admin	NIC	192.168.1.10	255.255.255.0	192.168.1.3
ControladorSDN	NIC	192.168.10.100	255.255.255.0	192.168.10.254
SwitchOpenFlow	VLAN 1	192.168.1.3	255.255.255.0	-
Switch1	VLAN 1	192.168.1.1	255.255.255.0	-
Switch2	VLAN 1	192.168.1.2	255.255.255.0	-
SDN	NIC	192.168.10.1	255.255.255.0	192.168.10.254
NoSND	NIC	192.168.20.1	255.255.255.0	192.168.20.254

TABLA 3.3: ASIGNACIONES INICIALES DE LOS PUERTOS SWITCH 1 Y 2 (ELABORACIÓN PROPIA)

PUERTOS	ASIGNACIÓN	RED
f1/0	Enlaces troncales 802.1q (VLAN 1 nativa)	192.168.1.0/24
f1/1 – 1/15 (switch1)	VLAN 10: sdn	192.168.10.0/24
f1/1 – 1/15 (switch2)	VLAN 20: nosdn	192.168.20.0/24

3.2.2 Gestión de la configuración de la red tradicional

Para la gestión de la configuración en la red tradicional del escenario se utiliza la herramienta de monitorización Zabbix que dispone de funcionalidades para realizar inventarios de equipamiento con el que el administrador puede generar informes a efectos de planificación. Zabbix utiliza MySQL como base de datos en las que almacena información asociada a diferentes estadísticas como carga de CPU, utilización de la red, espacio de disco duro, etc., tal y como se represen en la Figura 3.2.

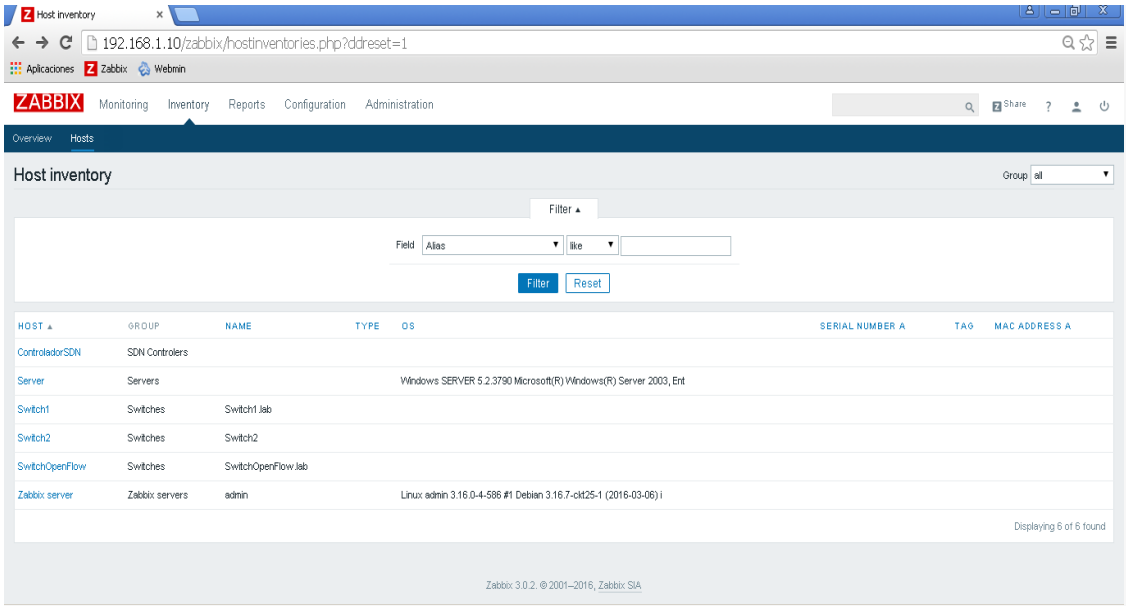

Figura 3.2 - Gestión de configuración de Zabbix (elaboración propia)

Mediante Zabbix, además, se realiza la gestión del equipamiento activo a través del protocolo de gestión SNMP que permite un monitoreo y control estándar, simplificado y extensible de una red que tiene diversos equipos de interconexión de diferentes fabricantes, lo cual permite tener cierto grado de genericidad en la gestión de equipamiento en el escenario de evaluación de la propuesta.

3.2.3 Diagnóstico del equipamiento SDN

El equipamiento activo de red del escenario con que cuenta la herramienta de simulación GNS3 son switches serie c3745 del fabricante Cisco de que no cuentan con soporte para SDN. Sin embargo, para los efectos de la investigación la validación del funcionamiento de la tecnología SDN como parte del modelo híbrido se realiza mediante un caso de uso en donde el controlador SDN, sufre un deterioro de su disponibilidad.

3.2.4 Segmentación de la red

En el escenario de simulación se realiza una segmentación lógica de la red partir de la separación de las redes en VLANs diferentes, lo que permite mejorar la gestión de cada segmento de red y la separación de los dominios de difusión, lo que mejora la seguridad y el rendimiento y posibilita la migración paulatina del

equipamiento tradicional hacia SDN. En el caso de estudio, la VLAN 10 corresponde al segmento de red SDN gestionado por el controlador a través de OpenFlow y la red tradicional por su parte, opera en la VLAN 20. La VLAN 1 se reserva para la administración.

En el escenario las VLANs tienen asignación por puertos. Los equipos que se conectan al **Switch1** se le asignan automáticamente direcciones que pertenecen a la subred 192.168.10.0/24 definida por la VLAN SDN y los equipos que se conectan al **Switch2** se le asignan direcciones que pertenecen a la subred 192.168.20.0/24 definida por la VLAN NoSDN.

3.2.5 Configuración de la arquitectura SDN-PBNM

Para la configuración de la arquitectura SDN-PBNM se identifican primeramente las entidades dentro del escenario que van a alojar a cada elemento que compone la arquitectura híbrida. El PDP principal, PDP secundario y la herramienta de gestión de políticas con el contenedor que aloja las mismas van a estar implementados en host **Admin** como parte de la herramienta Zabbix, que constituye la entidad principal del modelo híbrido. El controlador SDN, por su parte, se va a alojar en un host independiente en el que se implementa el sistema base requerido para el funcionamiento de controlador OpenFlow. A continuación, se describe la configuración realizada en estos dos elementos de la arquitectura propuesta.

3.2.5.1 Configuración de Controlador SDN

En la investigación realizada, como parte del marco teórico se escogió el controlador SDN Floodlight para implementar en el escenario de estudio debido fundamentalmente a que es un controlador multiplataforma muy popular, que brinda total soporte para el descubrimiento de los dispositivos (OpenFlow o no OpenFlow), gestión de los mismos, cálculo de rutas, acceso web, etc. Además, cuenta con una amplia lista de fabricantes de *hardware* de *switches* compatibles que lo amparan. Floodlight incluye aplicaciones de ejemplo básicas como: *hub*, *switch*, flujo de red virtual, cortafuegos, entre otras, que permiten virtualizar varios dispositivos para la experimentación.

Floodlight se despliega en el escenario en el host **ControladorSDN** a partir de una máquina virtual que cuenta, además del controlador, con varias herramientas que permiten la monitoreo y el control de los

recursos de red que este maneja. Entre estas herramientas están Mininet, Open vSwitch y Wireshark w/OpenFlow disector.

Luego de ejecutar el host **ControladorSDN** se procede a inicializar Floodlight, tal y como se aprecia en la Figura 3.3 a través del comando **java –jar target/Floodlight.jar**

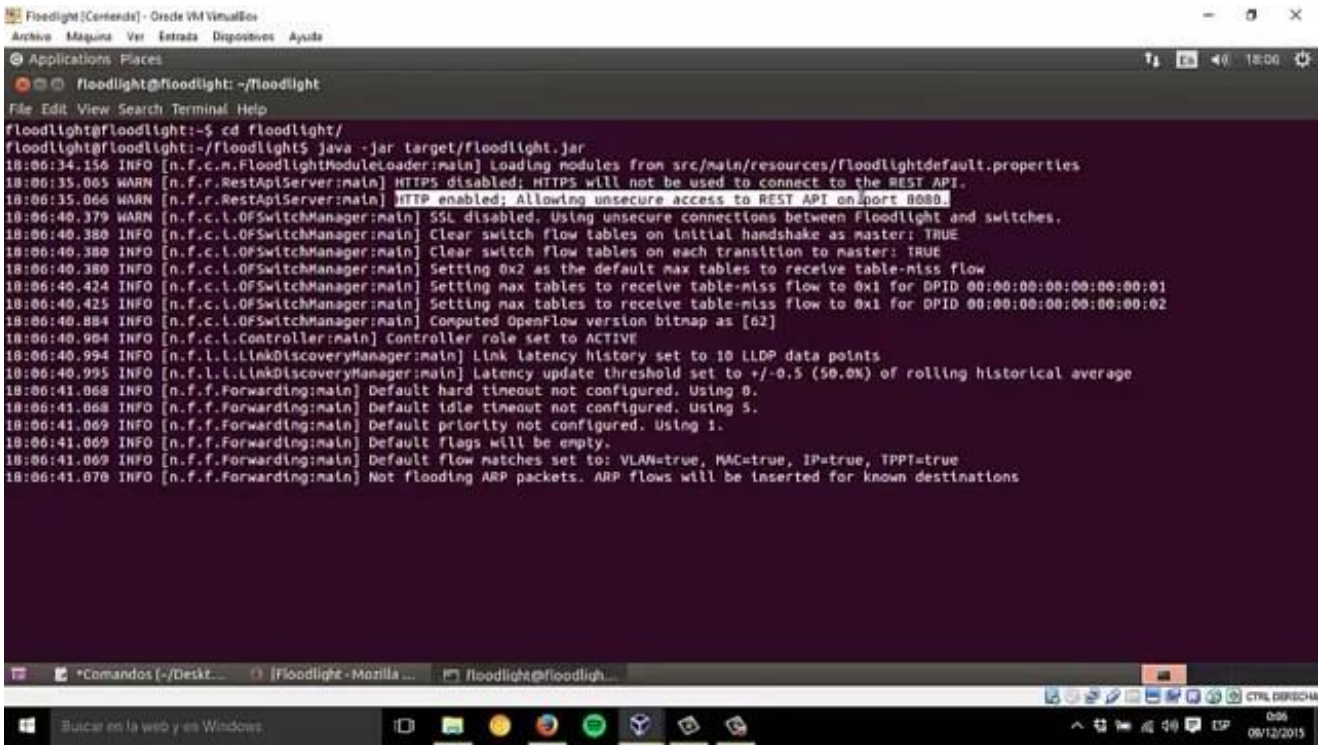

Figura 3.3 - Ejecución del controlador Floodlight (elaboración propia)

Luego de ejecutarse el proceso de inicialización para comprobar el correcto funcionamiento del controlador SDN se debe crear una topología que permitan identificar el intercambio de datos entre el controlador Floodlight y los switches virtuales. Para ello se ejecuta el comando **sudo mn --topo single,3 --controller=remote,ip=127.0.0.1,port=6653 --switch ovsk,protocols=OpenFlow13**.

Para verificar el funcionamiento del controlador se ejecuta la herramienta Wireshark, en la cual se define primeramente la interfaz Loopback y luego se inicia la captura de paquetes. Al realizar una solicitud ICMP desde el nodo h1 a h2 mediante el comando **h1 ping h2** se puede apreciar cómo se establece la

comunicación entre los dispositivos especificados mediante el protocolo OpenFlow, tal y como se aprecia en la Figura 3.4 en la captura de paquetes. Con esto se comprueba el correcto funcionamiento de Floodlight.

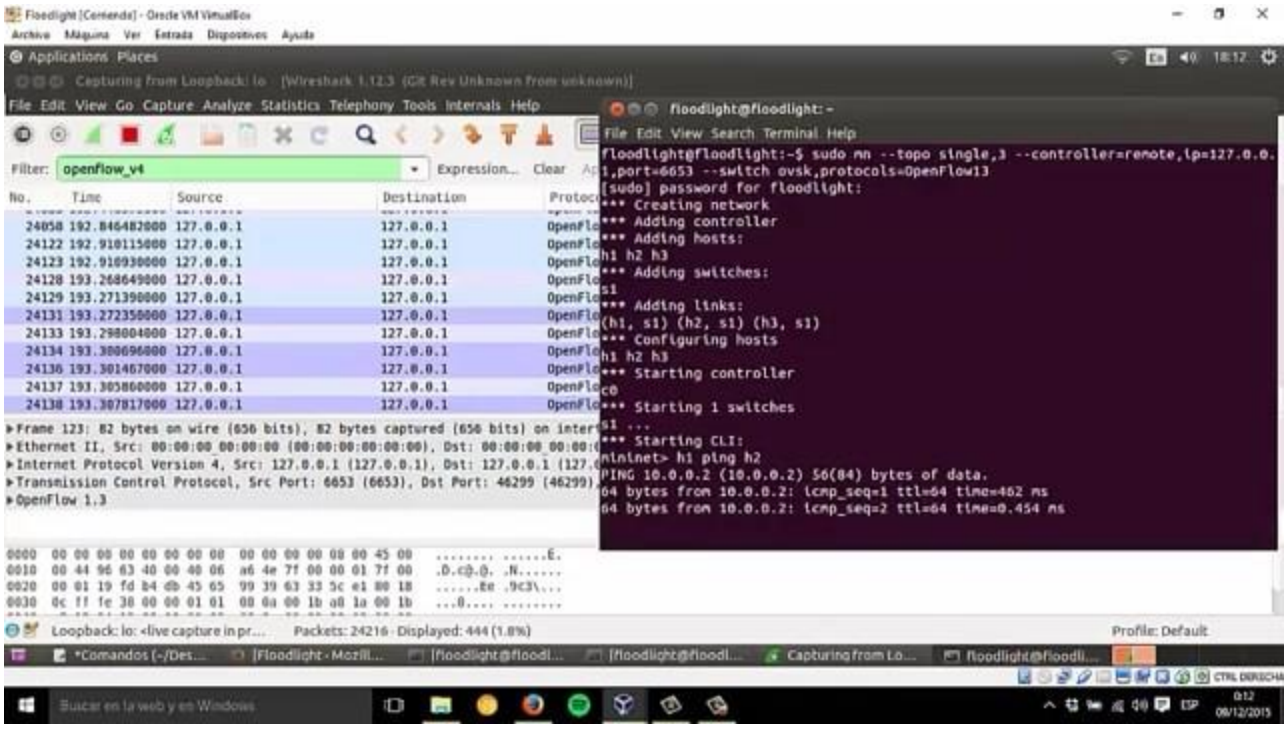

Figura 3.4 - Funcionamiento del controlador Floodlight (elaboración propia)

3.2.5.2 Configuración del PDP

Los PDP que componen la arquitectura SDN-PBNM van a ser implementados en el escenario de estudio a través de la herramienta Zabbix y las políticas que evalúan se almacenan en igualmente en dicha herramienta.

Zabbix define un conjunto de funcionalidades que permiten el monitoreo de la configuración de los dispositivos que tiene registrados en un inventario de activos de red disponibles. Estas funcionalidades denominadas Disparadores o *Triggers* identifican cambios de estado en los dispositivos, a partir de reglas de discriminación que permiten generar alertas. Estas reglas evalúan condiciones que el usuario define y que determinan el comportamiento que tomará el sistema. Conforme se generan las alertas, el sistema puede ejecutar acciones automáticamente en base a las restricciones que fueron predefinidas, que pueden

ser para enviar un mensaje o ejecutar un comando remoto a través de SSH, Telnet, IPMI o script basado en bash. Estas acciones que se evalúan y ejecutan constituyen las políticas que precisan el comportamiento de la red en la arquitectura SDN-PBNM propuesta en el escenario.

La Figura 3.5 representa una de las políticas que se implementan en el PDP para restaurar la configuración tradicional del equipamiento de red híbrido en caso de una falla en la disponibilidad del controlador SDN.

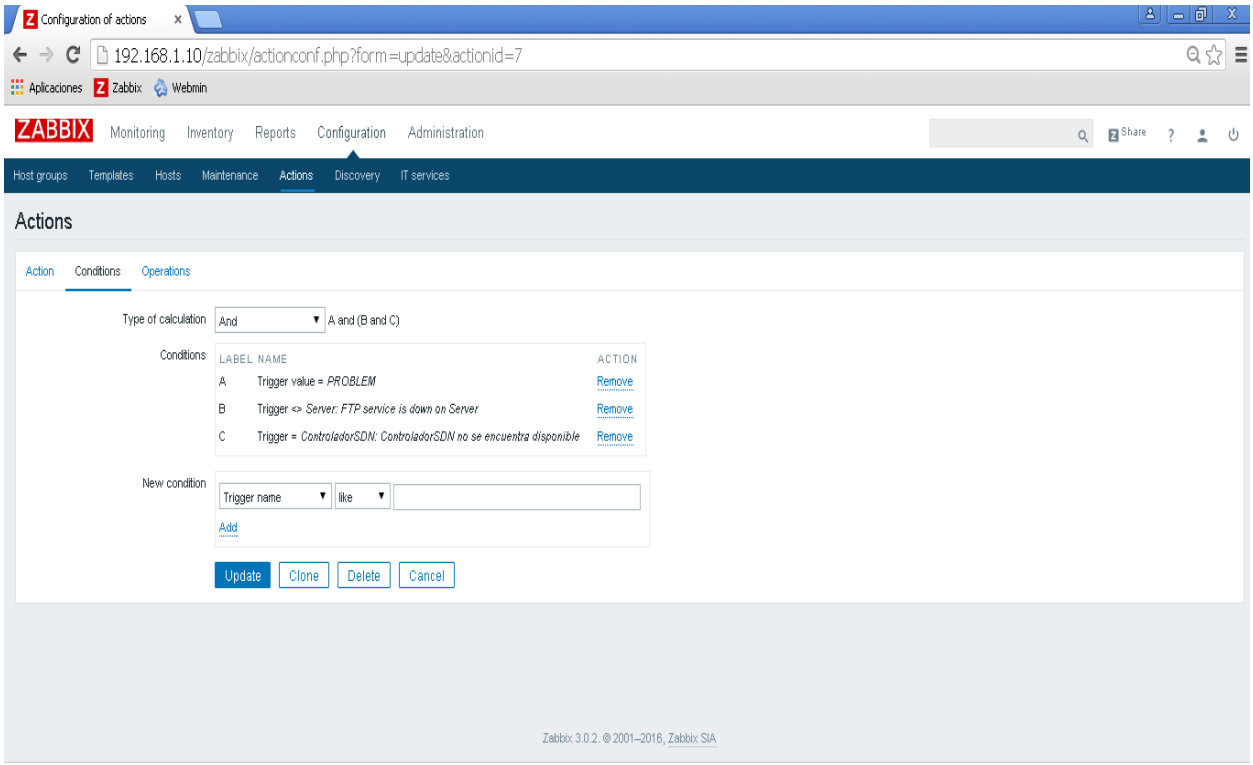

Figura 3.5 - Política definida para resolver fallo de disponibilidad en el controlador SDN (elaboración propia)

En este caso, para ejecutar la política de Restaurar configuración Switches después de una falla en el controlador se evalúan los disparadores siguientes:

A. Trigger value = *PROBLEM* (El estado del disparador es un problema)
B. Trigger <> *Server: FTP Service is down on server* (El servicio FTP alojado en el host **Server** está caído)

C. Trigger = *ControladorSDN: ControladorSDN no se encuentra disponible* (El host **ControladorSDN** no se encuentra disponible)

Cuando la expresión lógica definida por (A and (not B and C)) es verdadera entonces se aplica la acción de ejecutar un script basado en bash almacenado en el host **Admin** que restaura la configuración de los dispositivos, la cual se encuentra almacenada en el servidor FTP del host **Server**.

En este escenario se implementan otras políticas específicas como parte de la evaluación de la propuesta de arquitectura para la gestión de SDN híbrida. Todas estas políticas se ejecutan a través de scripts basados en bash que realizan procedimientos remotos en los PEP. Estas políticas son:

- QoS: Evalúa valores obtenidos mediante el protocolo SNMP de los PEP y aplica sobre los mismos configuraciones de balanceo de carga.
- Salva centralizada: Realiza la operación de salva del estado funcional actual de los PEP en un servidor FTP.

3.2.6 Evaluación y mejora

El escenario de evaluación de la propuesta de arquitectura SDN-PBNM debe recopilar información que permita determinar si los indicadores de rendimiento del equipamiento activo se encuentran dentro de los umbrales establecidos como condiciones en el PDP. En la propuesta se realiza la monitorización de las CPUs de los *switches* habilitados con OpenFlow debido a que es una métrica determinante para el desempeño de la red SDN, así como la tasa de transferencia y ancho de banda para medir la rendimiento y estabilidad de la red tradicional.

Para la monitorización de los PEP se definen intervalos de monitoreo en la herramienta Zabbix, los cuales permiten la detección de cada uno de los parámetros que son evaluados en las políticas implementadas en el PDP. El valor de dichos intervalos impacta en el tiempo de detección de problemas en el equipamiento, por lo que valores demasiado altos pueden incidir, por ejemplo, en el tiempo de respuesta ante fallas de disponibilidad del controlador SDN. Por su parte, valores demasiado bajos pueden perjudicar el rendimiento de la red ya que el PDP realiza peticiones a los dispositivos monitoreados de manera más frecuente, lo cual en instantes tiempo en donde los recursos de la red se encuentran en alta demanda puede afectar la

estabilidad los mismos. Teniendo en cuenta esto, se definen intervalos de monitoreo entre 30 segundos y 1 minuto.

Se utiliza la herramienta Iperf para medir la calidad de los enlaces de red a través del ancho de banda, la tasa de pérdidas de datagramas y la latencia entre dos pares de nodos en un enlace. A continuación, se describen algunas métricas en las que se evidencia el aumento de rendimiento después de aplicadas las políticas sobre los PEP.

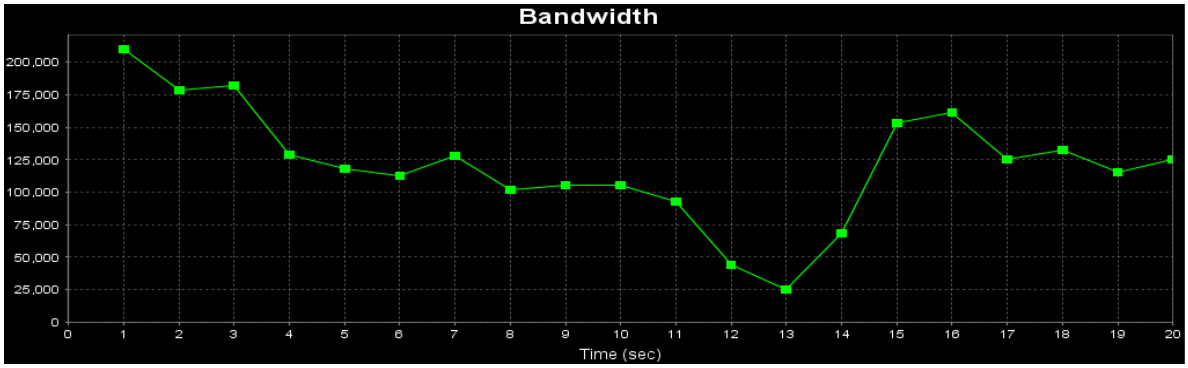

Figura 3.6 - Ancho de banda de SwitchOpenFlow en alta demanda antes de aplicar políticas de balanceo de carga (elaboración propia)

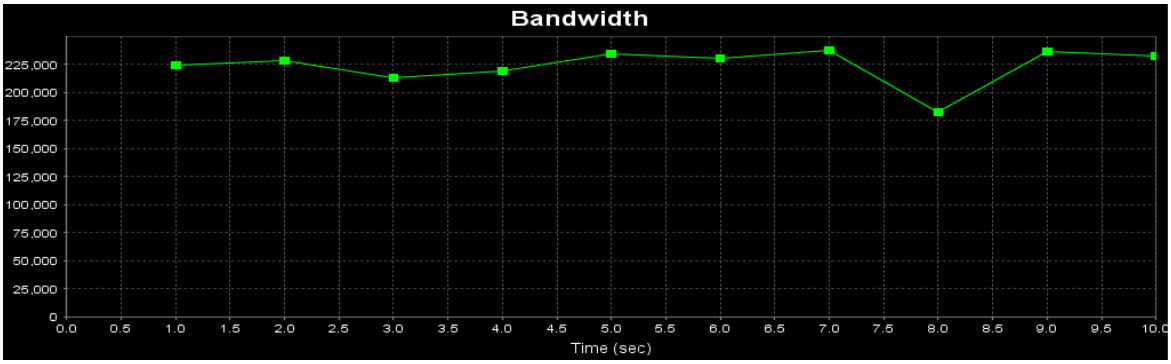

Figura 3.7 - Ancho de banda de SwitchOpenFlow en alta demanda después de aplicar políticas de balanceo de carga (elaboración propia)

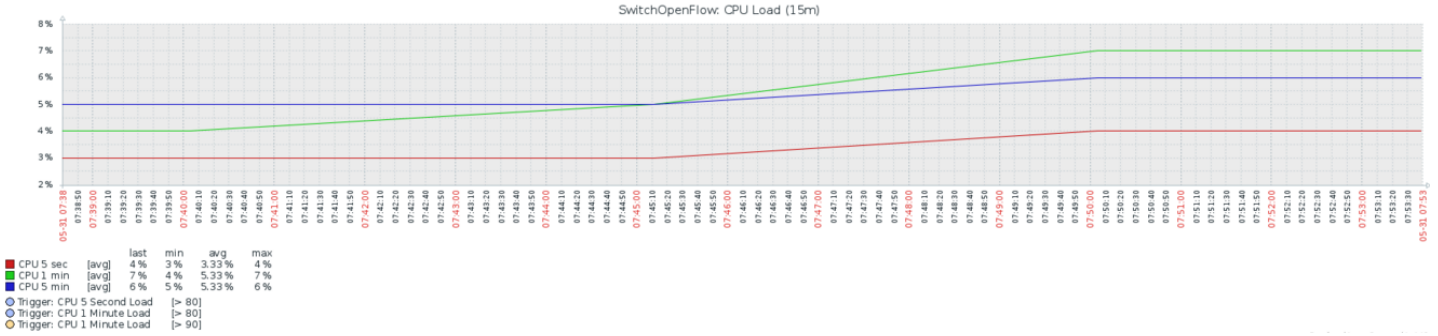

Figura 3.8 - Uso de CPU en el SwitchOpenFlow en alta demanda antes de aplicar políticas de balanceo de carga (elaboración propia)

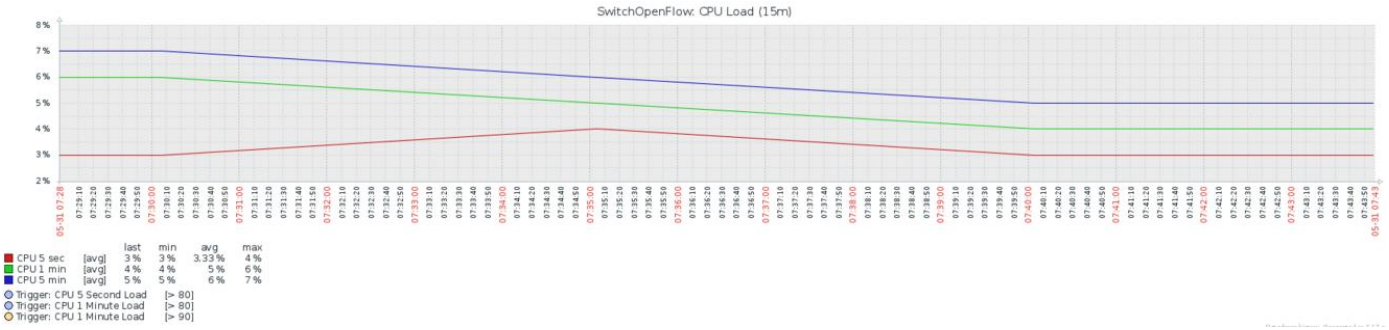

Figura 3.9 - Uso de CPU en el SwitchOpenFlow en alta demanda después de aplicar políticas de balanceo de carga (elaboración propia)

3.3 CONCLUSIONES PARCIALES

Una vez aplicada y evaluada la propuesta de arquitectura para la gestión de SDN híbrida un entorno de trabajo simulado a modo de conclusiones se pueden plantear las siguientes ideas:

- La propuesta de arquitectura SDN-PBNM realizada se ha validado y se han considerado, para ello, elementos de software y hardware que están actualmente en desarrollo.
- La validación de la arquitectura demuestra que SDN se puede incorporar en las redes cubanas permitiendo a las entidades mejorar su gestión.

CONCLUSIONES GENERALES

Con la realización de la presente investigación y teniendo en cuenta los resultados alcanzados y el cumplimiento de los objetivos específicos se puede concluir:

- El estudio de las redes tradicionales, la arquitectura SDN y la gestión basada en políticas demostró que la gestión de SDN híbridas es de gran complejidad por lo que existe la necesidad de diseñar una arquitectura que permita la integración de la gestión basada en políticas de las redes tradicionales con el paradigma de gestión SDN.
- La modelación de la arquitectura híbrida SDN-PBNM constituye un esquema para migrar la tecnología SDN hacia un entorno de red tradicional heterogéneo utilizando la gestión basada en políticas como elemento que permite la interoperabilidad entre las tecnologías.
- La elaboración de un procedimiento para implementar la arquitectura SDN-PBNM constituye una guía para aplicar la arquitectura propuesta en un escenario de red real.
- La evaluación de la arquitectura SDN-PBNM demostró que SDN se puede incorporar en las redes cubanas permitiendo a las entidades mejorar su gestión de recursos informáticos al construir redes altamente escalables y flexibles.

RECOMENDACIONES

La arquitectura para la gestión de SDN híbrida propuesta en esta investigación constituye un modelo de implementación de la tecnología SDN, en este caso, utilizando la gestión de red basada en políticas como mecanismo de control de la información entre la red SDN y la red tradicional. Este modelo representa un paso de avance en el estudio en nuestro país de una tecnología de vanguardia en redes de comunicación y de cómo integrarla a la infraestructura telemática cubana. Sin embargo, no todos los elementos que conforman SDN se analizaron con la profundidad requerida por lo que al término de la presente investigación se recomiendan las siguientes ideas con el objetivo de poderle dar continuidad.

- Desplegar y evaluar la propuesta de arquitectura SDN híbrida en un escenario real.
- Validar la arquitectura propuesta en otros escenarios que incluyan soporte de infraestructura para plataforma móvil.
- Incorporar elementos asociados a la Inteligencia Artificial en el PDP para automatizar las acciones de control en el equipamiento de red.
- Integrar la arquitectura al sistema integral de gestión de red de manera tal que se integre la gestión de los servidores y los servicios.
- Incorporar un Sistema de Soporte a la Toma de Decisiones que permita mejorar el proceso de implementación de la arquitectura.
- Realizar trabajos de Investigación, Desarrollo e Innovación (I+D+i) con el objetivo de profundizar en cada uno de los elementos que la conforman.

GLOSARIO DE TÉRMINOS

Bash: Es un programa informático basado en el shell de UNIX y un lenguaje de programación de consola cuya función consiste en interpretar y ejecutar órdenes.

Framework: Es una estructura conceptual y tecnológica de soporte definido, normalmente con objetos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

GitHub: Es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

Iperf: Es una herramienta que se utiliza para hacer pruebas en redes informáticas a partir de la creación de flujos de datos TCP y UDP para medir el rendimiento de la red.

Mininet: Es un emulador de red que crea una red de máquinas virtuales, conmutadores, reguladores, y enlaces. Los anfitriones Mininet ejecutan software de red estándar de Linux, y sus conmutadores son compatibles con OpenFlow para enrutamiento personalizado y altamente flexible.

NETCONF: Es un protocolo de gestión de redes desarrollado y estandarizado por IETF que provee mecanismos para instalar, manipular, y eliminar la configuración de dispositivos de redes.

Plugins: Son pequeños programas auxiliares o dispositivos de hardware que permiten a sistemas mayores extender sus capacidades normales o aportar una función, generalmente muy específica, de manera que no se afecten las funciones ya existentes ni se complique el desarrollo del programa principal.

OpenStack: Es un proyecto global para crear una plataforma de computación en la Nube de código abierto, que cumpla con las necesidades de los proveedores de servicios en la Nube, tanto públicas como privadas, independientemente de su tamaño, que sea fácil de implementar y masivamente escalable. El proyecto fue

fundado en octubre de 2010 por la empresa Rackspace Cloud y por la agencia espacial estadounidense, NASA.

Tremashark: Proporciona un *plugin* a Wireshark y un puente entre el mundo Trema y Wireshark para mostrar varios eventos en Wireshark. Usted puede monitorear cualquier evento IPC entre módulos Trema, paquetes de interfaces de red enlaces, o los mensajes de registro en tiempo real.

Wireshark: Antes conocido como Ethereal, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para el desarrollo de software y protocolos, y como una herramienta didáctica.

Nicira: Fue una empresa enfocada en la creación de redes definida por software (SDN) y la virtualización de red. Fue fundada en 2007 por Martín Casado, Nick McKeown y Scott Shenker. Nicira ha creado sus propias versiones propietarias de OpenFlow, Open vSwitch, y los proyectos de redes OpenStack.

Python: Es un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

REFERENCIAS BIBLIOGRÁFICAS

1. **CISCO SYSTEMS, INC.** Cisco visual networking index: Forecast and methodology, 2011–2016. [En línea] Mayo de 2012. [Citado el: 25 de Noviembre de 2015.] http://ec.europa.eu/newsroom/dae/document.cfm?doc_id=2037.

2. **ZHANG, L.** *Named Data Networking (ndn) project.* Palo Alto : XeroxPaloAltoResearchCenter-PARC, 2010.

3. **CAMPBELL, A. T.** *A survey of programmable networks.* 2, Abril de 1999, Vol. 29, págs. 7–23.

4. **ONF.** Software-defined networking: The new norm for networks. *Open Networking Foundation (ONF).* [En línea] Abril de 2012. [Citado el: 25 de Noviembre de 2015.] https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf.

5. **SDNCENTRAL, LLC.** SDN Central What's Software-Defined Networking (SDN)? *Sdx Central.* [En línea] [Citado el: 26 de Diciembre de 2015.] https://www.sdncentral.com/resources/sdn/ what-the-definition-of-software-defined-networking-sdn/.

6. **ONEI.** *Anuario Estadístico de Cuba 2014.* s.l. : Oficina Nacional de Estadística e Información (ONEI), 2015.

7. **ONF.** Sitio Oficial Open Networking Foundation. *Open Networking Foundation (ONF).* [En línea] https://www.opennetworking.org/.

8. **SPERA, C.** Software Defined Network: el futuro de las arquitecturas de red. *DATA CENTER.* [En línea] 2013. [Citado el: 2 de Diciembre de 2015.] http://www.la.logicalis.com/globalassets/latin-america/logicalisnow/revista-20/lnow20-nota-42-45.pdf.

9. **MCKEOWN, N.** *OpenFlow: Enabling innovation in campus networks.* 2, s.l. : SIGCOMM Comput, Marzo de 2008, Vol. 38, págs. 69–74.

10. **SDNCENTRAL, LLC.** SDN Central What are SDN Southbound APIs? *SDx Central.* [En línea] [Citado el: 26 de Diciembre de 2015.] https://www.sdncentral.com/resources/sdn/ southbound-interface-api/.

11. **RYU SDN FRAMEWORK COMMUNITY.** *Ryu.* [En línea] 29 de Diciembre de 2015. http://osrg.github.com/ryu/.

12. **CAI, Z.** Maestro: Achieving scalability and coordination in centralized network control plane. *Ph.D. dissertation.* Houston, TX, USA : s.n., 2011.

13. **TREMA COMMUNITY.** Sitio Oficial Trema. *Trema.* [En línea] [Citado el: 4 de Febrero de 2016.] http://trema.github.com/trema/.

14. **ATLASSIAN CONFLUENCE.** *Beacon.* [En línea] [Citado el: 2 de Febrero de 2016.] https://openflow.stanford.edu/display/Beacon/Home.

15. **PROJECT FLOODLIGHT.** Sitio Oficial Floodlight. *Floodlight.* [En línea] [Citado el: 5 de Febrero de 2016.] http://www.projectfloodlight.org/.

16. **CENTENO, A. G., VERGEL, C. M. R., CALDERÓN, C. A., BONDARENKO, F. C. C.** *Controladores SDN, elementos para su selección y evaluación.* 18 de Noviembre de 2014, Revista Telem@tica, Vol. 3, págs. 10-20.

17. **XIA, WENFENG, y otros.** *A Survey on Software-Defined Networking.* 1, 2015, IEEE COMMUNICATION SURVEYS & TUTORIALS, Vol. 17.

18. **SDNCENTRAL, LLC.** SDN Central What are SDN Northbound APIs? *SDx Central.* [En línea] [Citado el: 26 de Diciembre de 2015.] https://www.sdncentral.com/resources/sdn/ north-bound-interfaces-api/.

19. **KUMAR, R.** *Software Defined Networking - a definitive guide.* [En línea] 2013. [Citado el: 28 de Enero de 2016.] http://uploaded.net/file/nhcvroti/Software_Defined_Networking_SDN__a_definitive_guide.rar.

20. **NOLLE, T.** *Software-Defined Networking: Top Architecture and Security Considerations.* [En línea] 2012. [Citado el: 1 de Febrero de 2016.]

http://docs.media.bitpipe.com/io_11x/io_110278/item_706616/Juniper_sSDN_IO%23110278_EGuide_060513_21.pdf.

21. **TOOTOONCHIAN, A, y otros.** *On controller performance in software-defined networks.* 2012. pág. 10.

22. **ROMERO DE TEJADA MUNTANER, G.** *Evaluation of OpenFlow Controllers.* 2012.

23. **ASADULLAH, S, y otros.** Migration Use Cases and Methods. *Open Networking Foundation (ONF).* [En línea] https://www.opennetworking.org/images/stories/downloads/sdn-resources/use-cases/Migration-WG-Use-Cases.pdf.

24. **AGARWAL, A.** *Inter-Datacenter WAN with centralized TE using SDN and OpenFlow.* 2012.

25. **SEARCHSDN.** *SearchSDN What is hybrid SDN?* [En línea] [Citado el: 14 de Febrero de 2016.] http://searchsdn.techtarget.com/definition/hybrid-SDN.

26. **ANÍAS CALDERÓN, C., VILLARIÑO BOLAÑO, L. A. y PRECIADO VELASCO, J. E.** Gestión de Red Basada en Políticas. [En línea] 2014. [Citado el: 16 de Febrero de 2016.] http://www.bibliociencias.cu/gsdl/collect/eventos/import/Gestion_Red_politicas.pdf.

27. **IETF.** *RFC1157.* [En línea] 24 de Febrero de 2016. http://tools.ietf.org/html/rfc1157.

28. **ANÍAS CALDERON, C.** *Funcionalidades de los Sistemas de Gestión de Redes.* 2014.

29. **NAGIOS ENTERPRISES, LLC.** Sitio Oficial Nagios. *Nagios.* [En línea] [Citado el: 1 de Marzo de 2016.] http://www.nagios.org.

30. **ZABBIX SIA.** Sitio Oficial Zabbix. *Zabbix.* [En línea] [Citado el: 1 de Marzo de 2016.] http://www.zabbix.com.

31. **MUNIN COMMUNITY.** Sitio Oficial Munin. *Munin.* [En línea] [Citado el: 1 de Marzo de 2016.] http://munin-monitoring.org.

32. **THE CACTI GROUP, INC.** Sitio Oficial Cacti. *Cacti.* [En línea] 1 de Marzo de 2016. http://www.cacti.net.

33. **ZENOSS INC.** Sitio Oficial Zenoss. *Zenoss.* [En línea] [Citado el: 1 de Marzo de 2016.] http://zenoss.com/.

34. **CABALLERO CRUZ, L.** *Sistema de monitorización.* Universidad de Sevilla. 2012.

35. **NOX COMMUNITY.** *POX.* [En línea] [Citado el: 14 de Marzo de 2016.] http://www.noxrepo.org/pox/about-pox/.

36. —. Sitio Oficial NOX. *NOX.* [En línea] [Citado el: 14 de Marzo de 2016.] http://www.noxrepo.org/.

37. **OPENSTACK FOUNDATION.** Sitio Oficial OpenStack. *OpenStack.* [En línea] http://www.openstack.org.

38. **KHONDOKER, R, y otros.** *Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers.*

39. **CISCO SYSTEMS, INC.** Sitio Oficial de Cisco. *Cisco.* [En línea] [Citado el: 24 de Marzo de 2016.] http://www.cisco.com.

40. **HUAWEI TECHNOLOGIES CO.** Sitio Oficial eNSP. *eNSP.* [En línea] [Citado el: 21 de Marzo de 2016.] http://enterprise.huawei.com/cn/.

41. **GNS3 TECHNOLOGIES INC.** Sitio Oficial GNS3. *GNS3.* [En línea] [Citado el: 21 de Marzo de 2016.] http://www.gns3.net/.

42. **WIRESHARK FOUNDATION.** Sitio Oficial Wireshark. *Wireshark.* [En línea] [Citado el: 24 de Febrero de 2016.] http://www.wireshark.org.

43. **OPENSIM LTD.** Sitio Oficial OMNeT++. *OMNeT++.* [En línea] [Citado el: 21 de Marzo de 2016.] http://www.omnetpp.org.

44. **IETF.** Sitio oficial IETF. *Internet Engineering Task Force.* [En línea] http://www.ietf.org.

45. **ASSOCIATION MANAGEMENT SOLUTIONS, LLC.** *RFC-Editor.* [En línea] [Citado el: 2 de Abril de 2016.] http://www.rfc-editor.org.

46. **AL-SOMAIDAI, MOHAMMED BASHEER.** *Survey of software components to emulate OpenFlow protocol as an SDN implementation.* 16 de Diciembre de 2014, American Journal of Software Engineering and Applications.

47. **ONF.** SDN Product Directory. *Open Networking Foundation (ONF).* [En línea] [Citado el: 24 de Abril de 2016.] https://www.opennetworking.org/products-listing.

ANEXOS

ANEXO 1: Versiones del protocolo OpenFlow

Las especificaciones OpenFlow describen las funcionalidades y rasgos que deben poseer los switches OpenFlow, específicamente las características funcionales y las versiones de dicho protocolo. Como se puede observar la Tabla A1.1 muestra un resumen de las principales características de las versiones del protocolo OpenFlow, en las cuales se aprecia la integración y soporte de nuevas funcionalidades en versiones más recientes (46).

TABLA A1.1: CARACTERÍSTICAS DE LAS VERSIONES DE OPENFLOW (46)

Versión / Rasgos	0.8.9	1.0	1.1	1.2	1.3	1.4	1.5
Fecha de publicación	2/12/2008	31/12/2009	28/2/2011	5/12/2011	25/6/2012	15/9/2013	19/12/2014
Desarrollo	Medio	Amplio	Medio	Medio	Amplio	Medio	Medio
Tabla de flujos	Una	Una	Múltiples	Múltiples	Múltiples	Múltiples	Múltiples
Grupo de tablas	No	No	Sí	Sí	Sí	Sí	Sí
Contadores	No	No	No	No	Sí	Sí	Múltiples
Etiquetas VLAN y MPLS	No	No	Sí	Sí	Sí	Sí	Sí
Compatibilidad con banderas TCP	No	No	No	No	No	No	Compatible (SYN,ACK,FIN,etc)
Fallas en conexiones al controlador	Tablas de emergencia	Tablas de emergencia	Modo seguro/Conmutación Ethernet	Modo seguro/Conmutación Ethernet	Modo seguro/Conmutación Ethernet	Modo seguro/Conmutación Ethernet	Modo seguro/Conmutación Ethernet
Soporte IPv6	No	No	No	Sí	Sí	Sí	Sí
Múltiples controladores	No	No	No	Sí	Sí	Sí	Sí
Soporte a protocolos	IPv4,TCP/UDP	IPv4,TCP/UDP	IPv4,TCP/UDP,MPLS	IPv4,IPv6,TCP/UDP,MPLS,ICMPv6	IPv4,IPv6,TCP/UDP,MPLS,ICMPv6	IPv4,IPv6,TCP/UDP,MPLS,ICMPv6	IPv4,IPv6,TCP/UDP,MPLS,ICMPv6

ANEXO 2: HARDWARE COMPATIBLE CON EL PROTOCOLO OPENFLOW

La Tabla A2.1 y A2.2 representan un listado de switches y routers compatibles con el protocolo OpenFlow respectivamente.

TABLA A2.1: LISTA DE SWITCHES COMPATIBLE CON OPENFLOW (47)

NOMBRE DE COMPAÑÍA	NOMBRE DE PRODUCTO
Accton Technology Corporation	AS4610-54T
Accton Technology Corporation	AS5712-54X
Accton Technology Corporation	Wedge-16X
Brocade Communication Systems	Brocade ADX Series with Application Resource Broker (ARB)
Brocade Communication Systems	Brocade NetIron CES 2000 Series
Centec Networks	Centec CTC5162/5163 (GreatBelt)
Centec Networks	Centec Switch Boost OpenStack Network Virtualization
Centec Networks	Centec V330 Series Switch
Centec Networks	Centec V350 Series Switch
Cisco Systems	Cisco 2500 Series Connected Grid Switches
Cisco Systems	Cisco Embedded Service 2020 Series Switches
Cisco Systems	Cisco HyperFlex HX-Series
Cisco Systems	Cisco Industrial Ethernet 2000 Series Switches
Cisco Systems	Cisco Industrial Ethernet 2000U Series Switches

Cisco Systems	Cisco Industrial Ethernet 3000 Series Switches
Cisco Systems	Cisco Industrial Ethernet 3010 Series Switches
Cisco Systems	Cisco Industrial Ethernet 4000 Series Switches
Cisco Systems	Cisco Industrial Ethernet 5000 Series Switches
Cisco Systems	Cisco Nexus 3000 Series Switches
Cisco Systems	Cisco Nexus 9000 Series Switches
Corsa Technology	Corsa 10G/100G SDN Switches
Dell	Dell Force10 S Series S4810 High-Performance 10/40 GbE Top-of-Rack Switch
Dell	Dell Force10 Z9000 Core Switch
Dell	Dell Networking MXL 10/40 GbE Blade Switch
Dell	Dell Networking S5000
Extreme Networks	Extreme Networks Summit X440 Series
Hewlett Packard Enterprise (HPE)	HP 10500 Switch Series
Hewlett Packard Enterprise (HPE)	HP 12500 Switch Series
Hewlett Packard Enterprise (HPE)	HP 2920 Switch Series
Hewlett Packard Enterprise (HPE)	HP 3500 and 3500 yl Switch Series
Hewlett Packard Enterprise (HPE)	HP 3800 Switch Series
Hewlett Packard Enterprise (HPE)	HP 5400R zl2 Switch Series
Hewlett Packard Enterprise (HPE)	HP 5400 zl Switch Series

Hewlett Packard Enterprise (HPE)	HP 5900 Switch Series
Hewlett Packard Enterprise (HPE)	HP 5920 Switch Series
Hewlett Packard Enterprise (HPE)	HP 8200 zl Switch Series
Hewlett Packard Enterprise (HPE)	HPE Hyper-Converged 250
Hewlett Packard Enterprise (HPE)	HP FlexFabric 11900 Switch Series
Hewlett Packard Enterprise (HPE)	HP FlexFabric 12900 Switch Series
Hewlett Packard Enterprise (HPE)	HP FlexFabric 5930 Switch Series
Juniper Networks	Juniper EX9200 Programmable Switch
Juniper Networks	Juniper EX-Series Ethernet Switches
Juniper Networks	Juniper QFX Series Switches
Lenovo	Lenovo RackSwitch Family
Mellanox Technologies	Mellanox MSX1410-OCP
Mellanox Technologies	Mellanox MSX1710-OCP
NEC	NEC ProgrammableFlow PF5240 Switch
NEC	NEC ProgrammableFlow PF5248 Switch
Northbound Networks	Zodiac FX
NoviFlow Inc.	NoviFlow NoviSwitch 1132
NoviFlow Inc.	NoviFlow NoviSwitch 1248

TABLA A2.2: LISTA DE ROUTERS COMPATIBLES CON OPENFLOW (47)

NOMBRE DE COMPAÑÍA	NOMBRE DE PRODUCTO
Brocade Communication Systems	Brocade MLX Series
Brocade Communication Systems	Brocade NetIron XMR Series
Cisco Systems	Cisco 1000 Series Connected Grid Routers
Cisco Systems	Cisco 2000 Series Connected Grid Router
Cisco Systems	Cisco 500 Series WPAN Industrial Routers
Cisco Systems	Cisco 5915 Embedded Service Router
Cisco Systems	Cisco 5921 Embedded Services Router
Cisco Systems	Cisco 5940 Embedded Services Router
Cisco Systems	Cisco 809 Industrial Integrated Services Routers
Cisco Systems	Cisco 819 Integrated Services Routers
Cisco Systems	Cisco 829 Industrial Integrated Services Routers
Cisco Systems	Cisco Aironet 1530 Series

Cisco Systems	Cisco Aironet 1550 Series
Cisco Systems	Cisco Aironet 1570 Series
Cisco Systems	Cisco ASR 900 Series Aggregation Services Routers
Cisco Systems	Cisco Industrial Wireless 3700 Series
Cisco Systems	Cisco Mobile IP Gateway 2450
Cisco Systems	Cisco Unified Fabric
Ericsson	Ericsson SSR (Smart Services Router) 8000 Family
Extreme Networks	Extreme Networks BlackDiamond X8
Fiberhome Technologies	CiTRANS R8000-03
Fiberhome Technologies	CiTRANS R8000-05
Fiberhome Technologies	CiTRANS R8000-10
Fiberhome Technologies	CiTRANS R810
Fiberhome Technologies	CiTRANS R820

Huawei	Huawei ATN910 Series Cell Site Router (CSR)
Huawei	Huawei CX600 Series Aggregation Router
Huawei	Huawei S12700 Series Agile Switches
Juniper Networks	ACX Series (Universal Access Routers)
Juniper Networks	Juniper MX Series 3D Universal Edge Router
Juniper Networks	Juniper PTX Series (Packet Transport Routers)

ANEXO 3: INSTALACIÓN Y CONFIGURACIÓN DE GNS3

Para realizar una simulación en la herramienta GNS3 primeramente se debe configurar las imágenes binarias del IOS Cisco correspondiente de los dispositivos de interconexión que puede emular Dynamips.

Posteriormente a estos pasos se puede proceder a realizar el diseño de la topología del escenario que se va a simular como parte de la evaluación del modelo híbrido. Para ello primeramente estando en la interfaz principal se define un nuevo proyecto en el menú *Archivo*, *Proyecto nuevo en blanco*. Luego se define el nombre del proyecto nuevo, tal y como se aprecia en la Figura A3.1 y se da clic en *Aceptar.*

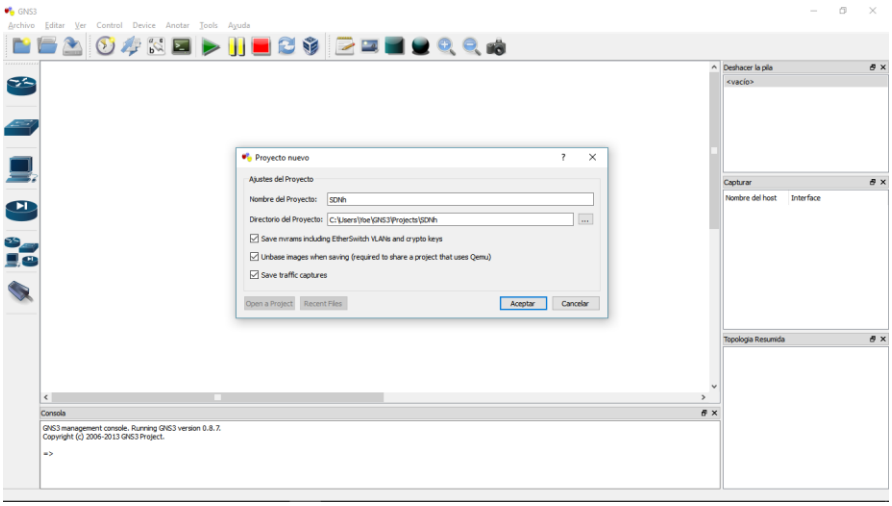

Figura A3.1 - Crear nuevo proyecto (elaboración propia)

Teniendo el proyecto creado el siguiente paso es definir el equipamiento que a conformar el escenario a simular. Para ello en el panel de *Dispositivos* en la parte izquierda de la pantalla se selecciona la opción *Buscar todos los dispositivos,* desplegando una lista con el equipamiento disponible (Figura A3.2).

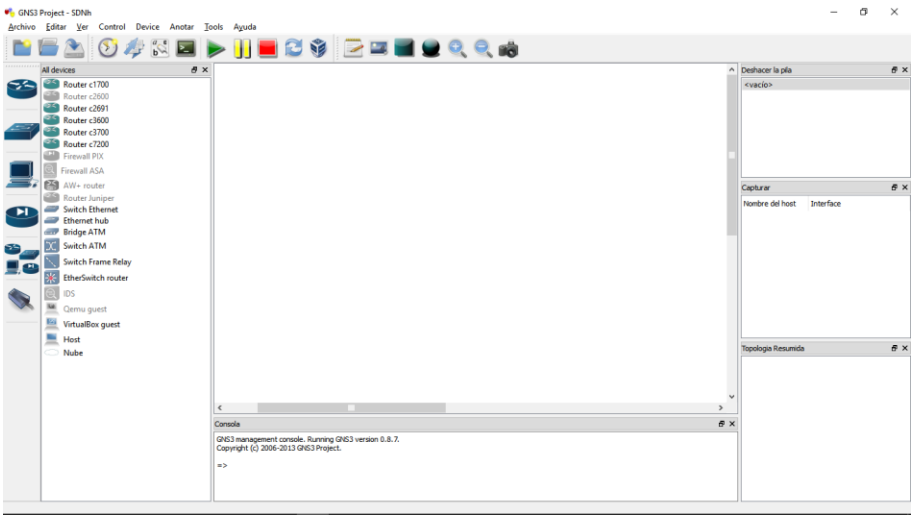

Figura A3.2 - Dispositivos disponibles (elaboración propia)

A partir de la arquitectura propuesta se definen entonces cada uno de los elementos que se van a agregar en la topología de red. El diseño de dicha topología de red se representa en la Figura A3.3 con los dispositivos que la componen

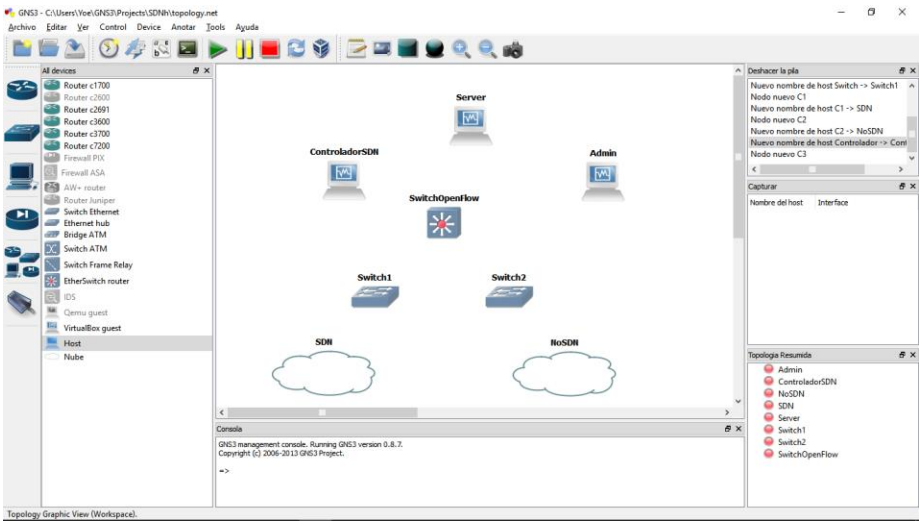

Figura A3.3 – Dispositivos que componen la topología de red (elaboración propia)

Finalmente, en el panel de *Dispositivos* en la parte izquierda de la pantalla se selecciona la opción *Agregar vínculo* para realizar las conexiones necesarias entre los dispositivos a través de sus interfaces de red. La Figura A3.4 muestra diseño final del diagrama de topología para la red.

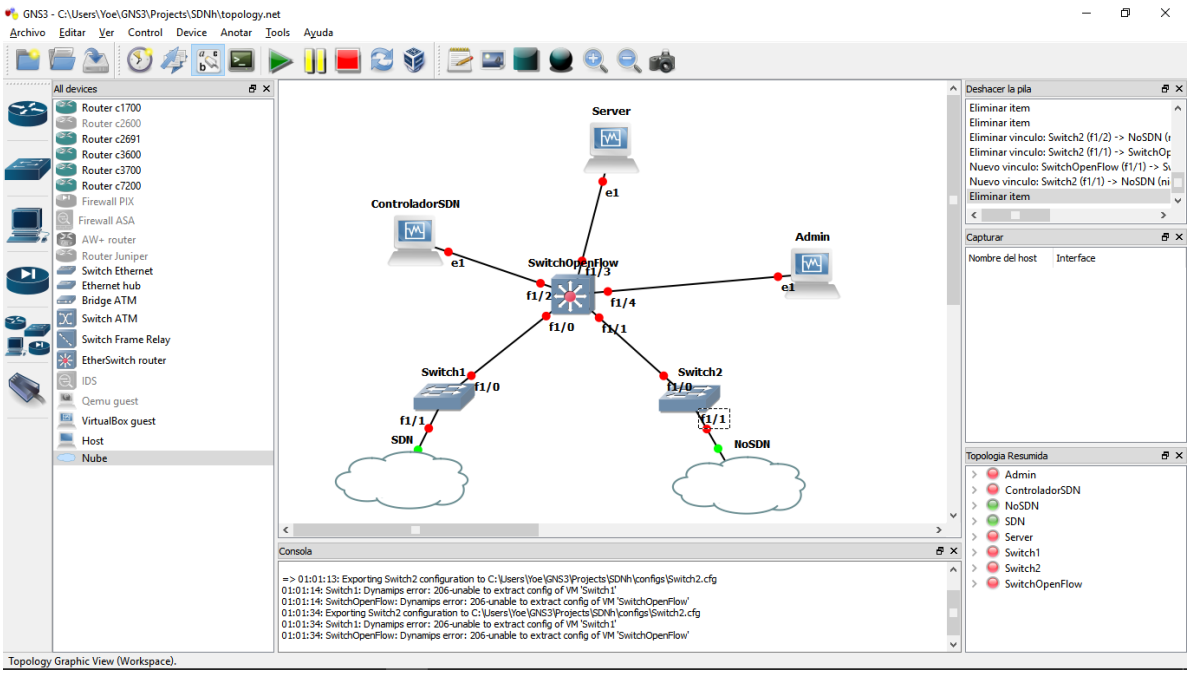

Figura A3.4 - Diagrama de topología de red (elaboración propia)

ANEXO 4: CONFIGURACIÓN BÁSICA DE DISPOSITIVOS

Procedimiento 1: Realizar las configuraciones básicas de switches.

Las configuraciones básicas se realizan en los tres switches de la topología. Los comandos para estas configuraciones se especifican en la Tabla A4.1.

TABLA A4.1: COMANDOS BÁSICOS EN SWITCHES (ELABORACIÓN PROPIA)

COMANDOS	SIGNIFICADO
hostname *dispositivo*	Configurar el nombre de host
enable secret *contraseña*	Configurar contraseña para el modo de configuración global
line *línea*	Configurar el modo línea (console, aux, and VTY (Telnet))
password *contraseña*	Define una contraseña para el modo línea
login	Cuando la línea es configurada para usar una contraseña el comando login debe definirse

Se procede a realizar la configuración básica de los switches de la siguiente forma.

```
Switch(config)#hostname SwitchOpenFlow
SwitchOpenFlow(config)#enable secret lab
SwitchOpenFlow(config)#line 0
SwitchOpenFlow(config-line)#password lab
SwitchOpenFlow(config-line)#login
SwitchOpenFlow(config-line)#line vty 0 4
SwitchOpenFlow(config-line)#password lab
SwitchOpenFlow(config-line)#login
```

Procedimiento 2: Configurar acceso por SSH a los switches

En lugar de Telnet, es posible usar el intérprete de ordenes seguras o SSH (*Secure Shell*, por sus siglas en inglés), que permite crear sesiones de consola más seguras que Telnet para la comunicación remota hacia

los dispositivos mediante la utilización de llaves encriptadas para el intercambio de datos. Los comandos para su configuración se especifican en la Tabla A4.2.

TABLA A4.2: COMANDOS DE SSH EN SWITCHES (ELABORACIÓN PROPIA)

COMANDOS	SIGNIFICADO
ip domain-name *dispositivo.dominio*	Configurar el nombre de dominio del dispositivo
crypto key generate rsa general-keys export	Genera la llave para las conexiones cifradas
ip ssh time-out	Establece el tiempo de espera de conexiones
ip ssh authentication-retries *número*	Establece la cantidad de intentos de conexión
line vty *primera-línea última-línea*	Especifica las líneas a configurar
transport input ssh telnet	Establece el tipo de conexión remota que se establecerá en el dispositivo

Se procede a realizar la configuración de SSH en los switches de la siguiente forma

```
Switch1(config)#ip domain-name Switch1.lab
Switch1(config)#crypto key generate rsa general-keys export 1024
The name for the keys will be: Switch1.lab
% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys
[OK]
Switch1(config)#ip ssh time-out 60
Switch1(config)#ssh authentication-retries 2
Switch1(config)#line vty 0 1180
Switch1(config-line)#transport input ssh telnet
```

Procedimiento 3: Configurar las VLAN de switches.

La configuración de las VLAN en los switches de la topología se realiza de manera individual en cada dispositivo mediante los comandos que se describen en la Tabla A4.3.

TABLA A4.3: COMANDOS DE VLAN EN SWITCHES (ELABORACIÓN PROPIA)

COMANDOS	SIGNIFICADO
vlan database	Accede al modo de configuración de VLAN
vlan *id* name *nombre*	Crea una VLAN
show vlan-switch brief	Verifica la configuración de las VLANs creadas
interface *tipo id*	Accede al modo de configuración de la interfaz
interface range *tipo id*	Accede al modo de configuración de la interfaz por rango
switchport mode *modo-de-acceso*	Asigna el modo Acceso/Trunk a la interfaz espeficicada
switchport *modo-de-acceso* vlan *id*	Asigna el modo Acceso/Trunk en la interfaz espeficificada a la VLAN correspondiente
show vlan-switch id	Verifica la configuración de la VLAN especificada
copy running-config startup-config	Guarda la configuración en ejecución a la NVRAM del dispositivo

A continuación, se define la configuración en **Switch1**.

```
Switch1#vlan database
Switch1(vlan)#vlan 10 name sdn
VLAN 10 added:
   Name: sdn
Switch1(vlan)#exit
APPLY completed.
Exiting....
```

Se puede utilizar el comando **show vlan-switch brief** para verificar la VLAN se haya creado.

```
Switch1#show vlan-switch brief

VLAN Name                         Status  Ports
---- -------------------------------- --------- -------------------------------
1    default                      active  Fa1/0, Fa1/1, Fa1/2, Fa1/3
                                          Fa1/4, Fa1/5, Fa1/6, Fa1/7
                                          Fa1/8, Fa1/9, Fa1/10, Fa1/11
                                          Fa1/12, Fa1/13, Fa1/14, Fa1/15
```

```
10   sdn                          active
1002 fddi-default                 active
1003 token-ring-default           active
1004 fddinet-default              active
1005 trnet-default                active
```

A continuación, se asignan los puertos del **Switch1** a la VLAN 10 creada a partir de la distribución descrita en la Tabla 3.2.

```
Switch1(config)#interface range FastEthernet 1/1-15
Switch1(config-if)#switchport mode Access
Switch1(config-if)#switchport access vlan 10
Switch1(config-if)#no shutdown
```

Se puede utilizar el comando **show vlan-switch id** para verificar los cambios.

```
Switch1#show vlan-switch id 10
VLAN Name                        Status    Ports
---- ------------------------------- --------- -------------------------------
10   sdn                         active    Fa1/1, Fa1/2, Fa1/3, Fa1/4
                                           Fa1/5, Fa1/6, Fa1/7, Fa1/8
                                           Fa1/9, Fa1/10, Fa1/11, Fa1/12
                                           Fa1/13, Fa1/14, Fa1/15

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
10   enet  100010     1500  -      -      -        -    -        0      0
```

A través del comando **copy running-config startup-config** se procede a guardar la configuración realizada en el dispositivo.

```
Switch1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

Después de configurar la VLAN en el **Switch1** se procede a realizar la configuración del **Switch2** de manera similar utilizando la VLAN 20 que identifica la red no SDN. La configuración realizada queda de la siguiente manera.

```
Switch2#show vlan-switch brief
VLAN Name                        Status    Ports
---- ------------------------------- --------- -------------------------------
1    default                     active    Fa1/0
20   nosdn                       active    Fa1/1, Fa1/2, Fa1/3, Fa1/4
                                           Fa1/5, Fa1/6, Fa1/7, Fa1/8
                                           Fa1/9, Fa1/10, Fa1/11, Fa1/12
                                           Fa1/13, Fa1/14, Fa1/15
1002 fddi-default                active
1003 token-ring-default          active
1004 fddinet-default             active
1005 trnet-default               active
```

Configurados el **Switch1** y **Switch2** se procede a realizar la configuración de las VLAN en el SwitchOpenFlow.

```
SwitchOpenFlow#vlan database
SwitchOpenFlow#vlan 10 name sdn
VLAN 10 added:
   Name: sdn
SwitchOpenFlow#vlan 20 name nosdn
VLAN 20 added:
   Name: nosdn
SwitchOpenFlow#exit
APPLY completed.
Exiting....
```

Posteriormente se procede a realizar la configuración de las interfaces del **SwitchOpenFlow** de la siguiente manera.

```
SwitchOpenFlow(config)#int f1/2
SwitchOpenFlow(config-if)#switchport access vlan 10
SwitchOpenFlow (config-if)#no shutdown
SwitchOpenFlow(config)#int vlan 10
SwitchOpenFlow(config-if)#ip address 192.168.10.254 255.255.255.0
SwitchOpenFlow(config-if)#no shutdown
SwitchOpenFlow(config)#int vlan 20
SwitchOpenFlow(config-if)#ip address 192.168.20.254 255.255.255.0
SwitchOpenFlow(config-if)#no shutdown
```

Procedimiento 4: Configurar las VLAN de administración de switches.

Una VLAN de administración es cualquier VLAN que se configura para acceder a las capacidades administrativas de un switch. La VLAN 1 funciona por defecto como VLAN de administración si no ha definido específicamente otra VLAN. La asignación de una dirección de administración permite la comunicación IP entre switches y permite también que cualquier host conectado a un puerto asignado a la VLAN de administración se conecte a los switches. En el escenario de estudio se utiliza la VLAN 1 por defecto y se identifica con la subred 192.168.1.0/24. En la Tabla A4.4 se describen los comandos asociados a la asignación de VLAN de administración.

TABLA A4.4: COMANDOS PARA CONFIGURAR VLAN DE ADMINISTRACIÓN EN SWITCHES (ELABORACIÓN PROPIA)

COMANDOS	SIGNIFICADO
interface *tipo id*	Accede al modo de configuración de la interfaz
ip address *dirección-ip mascara-subred*	Asigna una dirección IP a la interfaz

La configuración de las VLAN de administración en los switches se describe a continuación.

```
Switch1(config)#interface vlan 1
Switch1(config-if)#ip address 192.168.1.1 255.255.255.0
Switch1(config-if)#no shutdown

Switch2(config)#interface vlan 1
Switch2(config-if)#ip address 192.168.1.2 255.255.255.0
Switch2(config-if)#no shutdown

SwitchOpenFlow(config)#interface vlan 1
SwitchOpenFlow(config-if)#ip address 192.168.1.3 255.255.255.0
SwitchOpenFlow(config-if)#no shutdown
```

Procedimiento 5: Configurar enlaces troncales y VLAN nativa de switches.

Los enlaces troncales son conexiones entre los switches que permiten a los mismos intercambiar información para todas las VLAN. De manera predeterminada, un puerto troncal pertenece a todas las VLAN, a diferencia del puerto de acceso que sólo puede pertenecer a una sola VLAN.

Un enlace troncal admite tráfico de varias VLAN (tráfico etiquetado) así como el tráfico que no proviene de una VLAN (tráfico sin etiquetar). El puerto de enlace troncal con encapsulamiento 802.1Q coloca el tráfico sin etiquetar en la VLAN nativa. El tráfico sin etiquetar se genera con una computadora conectada a un puerto del switch que se configura con la VLAN nativa. En la Tabla A4.5 se describen los comandos asociados a la configuración de enlaces troncales y VLANs nativas.

TABLA A4.5: COMANDOS PARA CONFIGURAR ENLACES TRONCALES EN SWITCHES (ELABORACIÓN PROPIA)

COMANDOS	SIGNIFICADO
interface *tipo id*	Accede al modo de configuración de la interfaz
interface range *tipo id*	Accede al modo de configuración de la interfaz por rango
switchport mode trunk	Asigna el modo Trunk a la interfaz espeficicada
switchport trunk native vlan *id*	Asigna la VLAN nativa en la interfaz espeficificada
show interface trunk	Verifica la configuración de las interfaces en modo Trunk
copy running-config startup-config	Guarda la configuración en ejecución a la NVRAM del dispositivo

La configuración de las interfaces que poseen enlaces troncales se define a continuación.

```
Switch1(config)#interface f1/0
Switch1(config-if)#switchport mode trunk
Switch1(config-if)#switchport trunk native vlan 1
Switch1(config-if)#no shutdown

Switch2(config)#interface f1/0
Switch2(config-if)#switchport mode trunk
Switch2(config-if)#switchport trunk native vlan 1
Switch2(config-if)#no shutdown

SwitchOpenFlow(config)#interface range f1/0-1
SwitchOpenFlow (config-if-range)#switchport mode trunk
SwitchOpenFlow (config-if-range)#switchport trunk native vlan 1
SwitchOpenFlow (config-if-range)#no shutdown
```

A través del comando **show interface trunk** se puede verificar la configuración de los enlaces troncales realizada en el dispositivo.

```
SwitchOpenFlow#show interface trunk

Port     Mode       Encapsulation  Status       Native vlan
Fa1/0    on         802.1q         trunking     1
Fa1/1    on         802.1q         trunking     1

Port     Vlans allowed on trunk
Fa1/0    1-1005
Fa1/1    1-1005

Port     Vlans allowed and active in management domain
Fa1/0    1,10,20
Fa1/1    1,10,20

Port     Vlans in spanning tree forwarding state and not pruned
Fa1/0    1,10,20
Fa1/1    1,10,20
```

Finalmente se procede a guardar la configuración realizada en cada dispositivo.

Procedimiento 6: Configurar enrutamiento de switches.

Para la correcta comunicación entre los dispositivos que conforman los segmentos de la red delimitados por VLAN es necesario configurar el enrutamiento en un dispositivo de capa 3, debido a que estos hosts se encuentran en diferentes subredes. En este caso, **SwitchOpenFlow** se va a encargar de gestionar la ruta entre las subredes separadas a través del protocolo de enrutamiento RIP (*Routing Information Protocol*, por sus siglas en inglés). En la Tabla A4.6 se describen los comandos asociados al enrutamiento de los switches.

TABLA A4.6: COMANDOS DE ENRUTAMIENTO EN SWITCHES (ELABORACIÓN PROPIA)

COMANDOS	SIGNIFICADO
ip routing	Habilita el enrutamiento sobre IP
router rip	Accede a la configuración de RIP
version *version*	Especifica la versión de RIP
network *dirección-subred*	Especifica la dirección de la red
show ip route	Verifica la configuración de enrutamiento del dispositivo
copy running-config startup-config	Guarda la configuración en ejecución a la NVRAM del dispositivo

Para ello primeramente es necesario habilitar en el switch mediante el comando **ip routing** el enrutamiento sobre IP.

```
SwitchOpenFlow(config)#ip routing
```

Posteriormente se define el protocolo RIP, la versión y las redes hacia las cuales el dispositivo de capa 3 encaminará los paquetes.

```
SwitchOpenFlow(config)#router rip
SwitchOpenFlow(config)#version 2
SwitchOpenFlow(config)#network 192.168.1.0
SwitchOpenFlow(config)#network 192.168.10.0
SwitchOpenFlow(config)#network 192.168.20.0
```

A través del comando **show ip route** se puede verificar la configuración de enrutamiento realizada en el dispositivo mediante el listado de redes asociadas al mismo.

```
SwitchOpenFlow#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
```

```
Gateway of last resort is not set

C   192.168.10.0/24 is directly connected, Vlan10
C   192.168.20.0/24 is directly connected, Vlan20
C   192.168.1.0/24 is directly connected, Vlan1
```

Finalmente se procede a guardar la configuración realizada en el dispositivo.

Procedimiento 7: Verificar la comunicación entre los dispositivos.

Para verificar la configuración realizada y la funcionalidad de comunicación entre los dispositivos se pueden utilizar los comandos **ping** y **tracert** o **traceroute**. El comando **ping** es una utilidad que diagnostica el estado de comunicación entre el host y el resto de los dispositivos de una red basada en TCP/IP a través del envío de paquetes ICMP (*Internet Control Message Protocol*, por sus siglas en inglés) que determinan mediante el envío de mensajes de solicitud y respuesta si un host remoto está disponible. El comando **tracert** o **traceroute** por su parte, permite mostrar el camino que realizan los paquetes ICMP y el retardo que tienen hacia el destino. A continuación, en la Tabla A4.7 se describe el uso de estos comandos en los switches para comprobar la comunicación entre los mismos.

TABLA A4.7: COMANDOS DE COMUNICACIÓN EN SWITCHES (ELABORACIÓN PROPIA)

COMANDOS	SIGNIFICADO
ping *host*	Envía mensajes de solicitud y respuesta ICMP para determinar si el host especificado está disponible
traceroute *host*	Envía mensajes de solicitud y respuesta ICMP y muestra el camino que recorren estos hacia el host especificado

```
SwitchOpenFlow#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 92/203/312 ms


SwitchOpenFlow#ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/73/152 ms

Switch1#ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/213/540 ms


Switch2#traceroute 192.168.1.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

 1 192.168.1.1 460 msec 608 msec 116 msec
```

En estos casos, la comunicación entre SwitchOpenFlow-Switch1, SwitchOpenFlow-Switch2 y Switch1-Switch2 fue satisfactoria.

En la Figura A4.1 se describe el uso de estos comandos en los hosts Server y ControladorSDN para comprobar la comunicación entre los equipos de las diferentes subredes. Téngase en cuenta que la asignación de direcciones IP en los hosts del escenario se realizó automáticamente en el host **Server** mediante el protocolo DHCP (*Dynamic Host Configuration Protocol*, por sus siglas en inglés) que posibilita la asignación de direcciones y parámetros de configuración en los equipos automáticamente y de manera centralizada,

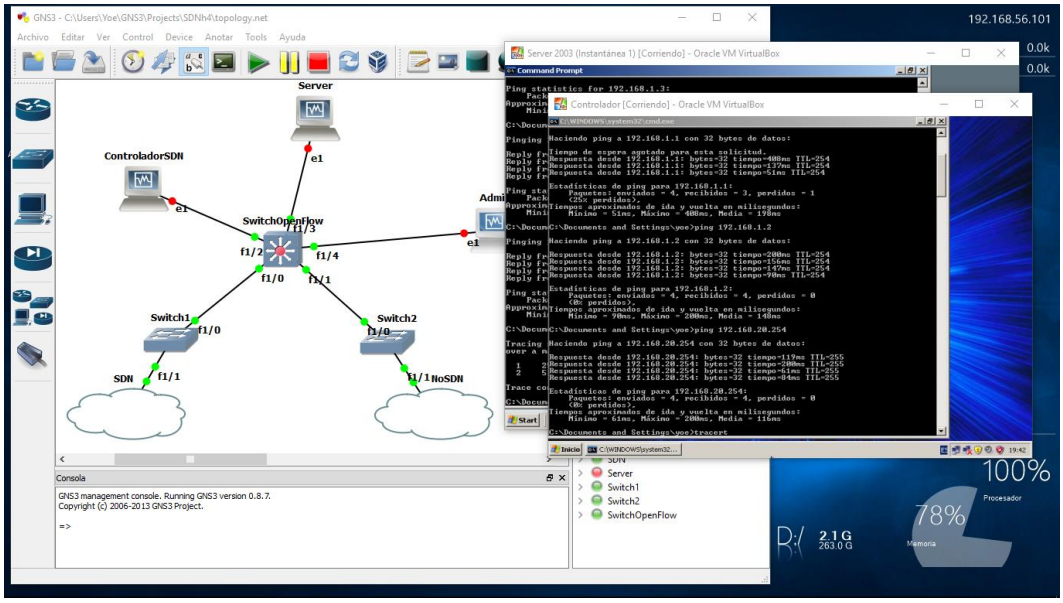

Figura A4.1 - Comprobación de conectividad en ControladorSDN (elaboración propia)

La comprobación de conectividad de los hosts fue satisfactoria en todos los casos ya que se logró obtener respuesta a las solicitudes ICMP enviadas desde cada dispositivo en la topología de red.