Entrega: Proyecto de Investigación

Redes Definidas por Software

Alumnos:

Alejandro Vargas Mena

20160110239

Carlos Aguirre Arroyo

(2010021096)

Universidad Latina de Costa Rica

San Pedro

Sistemas Operativos 2 – BIT-28

Profesor: Ing. Carlos Andrés Méndez

20-4-2024

**Tabla de Contenido**

[**Introducción** 2](#_Toc164100082)

[**Objetivo General** 3](#_Toc164100083)

[**Objetivos Específicos** 4](#_Toc164100084)

[**¿Qué es SDN?** 4](#_Toc164100085)

[**¿Como se compone una estructura de un SDN?** 5](#_Toc164100086)

[**¿Porque SDN?** 6](#_Toc164100087)

[**Beneficios** 9](#_Toc164100088)

[**¿Qué tipo de lenguaje de programación se planea investigar?** 9](#_Toc164100089)

[**Pyretic** 10](#_Toc164100090)

[**Características de Pyretic:** 10](#_Toc164100091)

[**Instalación de Pyretic:** 11](#_Toc164100092)

[**¿Qué protocolos se pueden implementar?** 13](#_Toc164100093)

[**¿Qué tipos de redes se pueden implementar en SDN?** 14](#_Toc164100094)

[**¿Con que funciones se puede acoplar las redes definidas por software?** 15](#_Toc164100095)

[*Infraestructura de datos* 15](#_Toc164100096)

[*Control* 15](#_Toc164100097)

[*Aplicación* 16](#_Toc164100098)

[*OpenFlow* 16](#_Toc164100099)

[**¿Qué es Cisco ACE?** 16](#_Toc164100100)

[**¿Cómo se involucra el software definido en redes en la arquitectura 5G?** 17](#_Toc164100101)

[**¿Qué ejemplo es el mejor para la implementación de las redes definidas por software en una arquitectura de red?** 17](#_Toc164100102)

[**Características** 18](#_Toc164100103)

[**Ventajas y desventajas** 19](#_Toc164100104)

[**Conclusiones** 19](#_Toc164100105)

[**Bibliografía** 20](#_Toc164100106)

[**Resultados** 21](#_Toc164100107)

[**Discusión** 21](#_Toc164100108)

[**Notas** 22](#_Toc164100109)

[**Tablas** 22](#_Toc164100110)

[**Figuras** 22](#_Toc164100111)

# **Introducción**

Durante esta investigación ahondaremos en el ámbito de las redes definidas por software, es importante en primera instancia mencionar que podemos entender por estas redes.

Las redes definidas por software (SDN por sus siglas en inglés) son una tecnología de redes que utiliza API y controladores basados en software para gestionar, automatizar, aprovisionar y programar los recursos de redes, para un mejor control y gestión de las redes.

La configuración de redes en las empresas medianas y grandes ha sido de mucha importancia para la segmentación, virtualización y envío de tráfico a la red global utilizando medios de traducción y distribución para el eficiente manejo y trayectorias más cortas.

Estas se pueden acceder durante el desplazamiento de las tramas a través de las redes mundiales con el propósito de enlazar y comunicar dispositivos, sean de almacenamiento, presentación de información mediante navegadores o aplicaciones disponibles en los dispositivos sean teléfonos celulares computadoras, con o sin interfaz gráfica, con la disponibilidad de ejecutar programas configurados para funciones específicas.

En las redes se implementan practicas o pruebas de configuración de dispositivos encargados de dirigir el tráfico de red usando programación para simular diversos dispositivos de una red y fluir tráfico de red por medio de ellos para generar tratamientos especializados de calidad, enrutamiento entrevarias funciones, para beneficio de todos.

Hoy en día el software basado en redes toma una gran importancia, en Costa Rica el pasado Enero en el día 5, se inauguró la primer torre 5G en el campus de la Universidad de Costa Rica liderada por equipos del ICE, este tipo de tecnología involucra a un volumen más amplio la el software basado en redes para el monitoreo de inmensa cantidad de dispositivos, motivo por el cual está diseñado la red 5G.

# **Objetivo General**

Investigar acerca de las transformaciones de la infraestructura de red tradicional mediante redes definidas por software en una infraestructura de red.

# **Objetivos Específicos**

* Investigar sobre las redes definidas por software y su funcionamiento.
* Identificar el lenguaje de programación utilizado en las redes definidas por software y sus beneficios.
* Analizar las ventajas y desventajas de las redes definidas por software.

# **¿Qué es SDN?**

Las redes definidas por software son un conjunto de técnicas relacionadas con el área de redes computacionales, cuyo objetivo es facilitar la implementación e implantación de servicios de red de manera determinista, dinámica y escalable. Todo esto se consigue mediante la separación de los planos control y de datos.

Utilizan un enfoque pensado para el sector empresarial que pueda optimizar los recursos disponibles y mejore las comunicaciones al momento de enrutar el tráfico, esto mediante el uso de software para priorizar y ordenar el tráfico de la red, que de forma general envía el uso de ciertas aplicaciones a través de determinadas conexiones, considerando métricas como la velocidad, latencia y consumo que demandan estas aplicaciones.

El flujo de la información y la tecnología SD-WAN es la autoridad encargada de poner orden asegurando un flujo adecuado ahorrando recursos.

# **¿Como se compone una estructura de un SDN?**

Aunque la ONF está continuamente modificando la terminología, los términos más comunes para los componentes de esta arquitectura son los siguientes:

* *Aplicación SDN (SDN App):* las aplicaciones SDN son programas que directamente comunican las necesidades y los comportamientos deseados de su red al controlador SDN a través de los NBI. Están formadas por una lógica de aplicación y uno o más NBI.
* *Controlador SDN (SDN Controller):* entidad lógica de control encargada de traducir las peticiones de la aplicación SDN a las rutas de datos más abajo, dando a la capa de aplicación una visión abstracta de la red mediante estadísticas y posibles eventos. Un controlador SDN consiste en uno o más NBIs, la lógica de control SDN y el CDPI driver.
* *Ruta de datos SDN (SDN Datapath):* componente lógico que expone visibilidad y control sobre sus capacidades de reenvío y procesamiento. La representación lógica, por lo tanto, puede abarcar todos o un subconjunto de los recursos físicos.

Está formado por un agente CDPI, un conjunto de motores de reenvío y de funciones de procesamiento, que incluyen simples reenvíos entre interfaces externas de esta y procesamiento interno del tráfico. Las rutas de datos pueden contenerse en un único elemento de red (físico).

* *Interfaz SDN del plano de control al plano de datos (SDN Control total Data-Plane Interface, CDPI):* interfaz entre el controlador y la ruta de datos, que provee programabilidad a la hora del reenvío, anuncio de capacidades, reporte estadístico y notificación de eventos.
* *Interfaces hacia el norte SDN (NBI):* son interfaces entre las aplicaciones SDN y los controladores que proveen vistas abstractas del comportamiento de la red y sus requerimientos.
* *Conductores y agentes de interfaz (Interface Drivers & Agents):* cada interfaz es implementada por un par de este tipo, que representa el fondo (relacionado con la infraestructura) y la cima (relacionada con la aplicación).
* *Gestión y administración (Management & Admin):* el plano de gestión cubre tareas estáticas, manejadas mejor fuera de los planos de aplicación, control y datos, como la asignación de recursos a los clientes, la configuración de equipos físicos y la concordancia entre alcanzabilidad y credenciales entre entidades físicas y lógicas.

# **¿Porque SDN?**

*Limitaciones de las tecnologías actuales*

Hacer frente a los requerimientos de telecomunicaciones por parte del público mundial, es imposible con las redes tradicionales. Los departamentos de IT de infinidad de empresas y proveedores de servicios de redes, están intentando aprovechar al máximo sus redes, invirtiendo gran parte de sus beneficios en esta tarea.

Sin embargo, esta es una solución temporal, ya que de ninguna manera las arquitecturas de red actuales están diseñadas para conocer los requerimientos actuales de usuarios, empresas y proveedores.

Las limitaciones de las redes actuales incluyen:

* Complejidad: para acomodar las redes a las necesidades de sus usuarios en general, la industria ha mejorado los protocolos de red para ser más seguros y eficientes.

Los protocolos tienden a ser definidos en aislamiento, sin embargo, con cada uno resolviendo un problema específico y sin el beneficio de una acción conjunta (abstracciones).

* Políticas incoherentes: para implementar una política que abarque a la red completamente, los administradores de red, deben configurar miles de mecanismos y aparatos.

**Por ejemplo**, todas las veces que una nueva máquina virtual se introduce en la red, puede llevar (en el peor de los casos) horas, hasta que el administrador encargado reconfigura las listas de acceso (ACL) en toda la red.

* Imposibilidad de escalabilidad: a la vez que las demandas de centros de datos aumentan rápidamente, la red debe crecer de la misma forma. Sin embargo, la red se vuelve más compleja con la suma de cientos de miles de aparatos de red que deben ser configurados y gestionados.
* Dependencia del vendedor: las nuevas capacidades y servicios perseguidas por proveedores y empresas en respuesta rápida a las necesidades dinámicas de negocios y demanda de clientes, se ven frenadas por los ciclos de producción de los equipamientos por parte de los vendedores, que pueden abarcar hasta más de tres años.

*Necesidad de una nueva red arquitectónica*

Los servicios de red surgidos en los últimos tiempos, están llevando a las redes tradicionales a sus límites. Algunos de los elementos que están llevando a la necesidad de una nueva arquitectura de red son los siguientes:

* Heterogeneidad en los patrones de tráfico: en contraposición con las aplicaciones cliente-servidor en las que la gran parte de la comunicación ocurre entre un cliente y un servidor, las aplicaciones modernas crean tráfico máquina a máquina mediante accesos a bases de datos y servidores, antes del retorno de los datos al usuario final. Además, los usuarios están cambiando los patrones de tráfico, al poder conectarse desde cualquier punto en cualquier momento en determinadas redes inalámbricas.
* Aumento de carga de trabajo considerable de los administradores de red: los administradores se ven bajo presión a la hora de proteger los datos de los usuarios y mantener la seguridad en las redes, mientras nuevos dispositivos móviles, como teléfonos inteligentes y tabletas, acceden a la red.
* El aumento de los servicios basados en la nube: este aumento, principalmente debido a la gran acogida por parte de las empresas en el plano público y privado, junto a la necesidad actual de aumentar la agilidad de acceso a aplicaciones, infraestructuras y otros recursos de telecomunicaciones y junto a la complejidad de mejorar la seguridad de estos servicios, requieren una escalabilidad dinámica de la capacidad de cómputo, almacenamiento y recursos de red.
* Big data y el aumento de ancho de banda requerido: manejar un big data actualmente requiere procesamiento masivo por parte de miles de servidores y su aumento demanda ancho de banda a los ISP.

# **Beneficios**

Además de ofrecer redes centralizadas programables que pueden atender dinámicamente las necesidades de las empresas, SDN provee los siguientes beneficios:

* **Reduce el Capex (Capital Expenditures):** mediante la posibilidad de reutilizar el hardware existente, SDN limita la necesidad de invertir en hardware nuevo.
* **Reduce el Opex (Operating Expense):** SDN permite control algorítmico de la red, en enrutadores y puentes (hardware y software) que cada vez son más programables, haciendo más sencillo la configuración y gestión de las redes. Además, esto permite una reducción del tiempo de gestión por parte de los administradores, lo que reduce la probabilidad de error humano.
* **Agilidad y flexibilidad**: SDN permite a las organizaciones desplegar aplicaciones, servicios e infraestructuras rápidamente para alcanzar los objetivos propuestos por empresas en el menor tiempo posible.
* **Permite innovación**: 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.

# **¿Qué tipo de lenguaje de programación se planea investigar?**

La programación en SDN se da mediante los lenguajes de alto nivel puede emplear usando diversos tipos de lenguajes, C++, Python, C, Haskell, OCaml, Java, entre otros tipos de lenguajes, la elección de cada uno de ellos se da por medio de las necesidades que tiene el cliente y las mejores disponibilidades que da cada uno de estos lenguajes, ya que varían y es de mucha importancia el definir los controladores para poder definir el tipo de lenguaje de programación.

Cada lenguaje posee sus propias variables, funciones, librerías, sintaxis, métodos, semántica, Entornos de Desarrollo Integrado, plataformas, soporte, documentación y otras características que conducen a los programadores a decidirse por uno u otro.

Controladores SDN para programación en Python

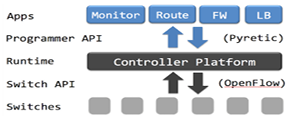
# **Pyretic**

Es un lenguaje SDN y tiempo de ejecución que implementan algunos operadores de composición. El lenguaje es una manera de expresión de estas políticas de alto nivel y la ejecución en tiempo provee las funciones de compilación estas políticas de reglas OpenFlow que eventualmente se incorporan en el switch.

Pyretic es un programa libre que ofrece el estilo de licenciamiento BSD compatible con la necesidad de desarrollo comercial y búsqueda. Provee un sistema de tiempo real que disponibiliza a programadores a normas de redes específicas a alto nivel de abstracción, se compone junto a variedad de vías y ejecuta en tipologías abstractas.

Pyretic es un nuevo lenguaje y sistema que permite la programación modular por:

* Operadores de composición definidos y librerías de políticas de revisión y consulta de tráfico.
* Permite a cada política operar en una topología abstracta que restringe implícitamente que puede ver y hacer el módulo.
* Provee un modelo de paquete de abstracción que disponibiliza a programadores a extender paquetes en ambientes virtuales que puede ser usado para asociar paquetes con alto nivel de meta datos.



Plataforma y lenguaje de Pyretic

# **Características de Pyretic:**

* Disponible de tomar un paquete como entrada y después devolver paquetes en distintas ubicaciones de la red.
* Tiene la noción de secuencias booleanas. Aunque las normas de OpenFlow la cual no permiten expresiones booleanas. Pyretic brinda las normas de términos de estas secuencias.
* Provee la noción de los campos de encabezados de paquetes virtuales.
* Las normas son funciones que mapean paquetes para otros paquetes.
* Pyretic enlaza salidas de funciones incluso paquetes o ninguna dependencia en el cual se cumple el requisito.
* El paquete no es más que un diccionario que mapea un amplio nombre como una dirección IP de destino a un valor. Los campos nombrados pueden corresponder a campos en un encabezado de paquete. Pero puede incluso ser virtual.

# **Instalación de Pyretic:**

Se puede instalar de dos maneras:

1. Directamente en el Sistema Operativo Ubuntu.

Comandos de instalación:

*sudo apt-get install python-dev python-pip python-netaddr screen hping3 ml-lpt graphviz ruby1.9.1-dev libboost-dev libboost-test-dev libboost-program-options-dev libevent-dev automake libtool flex bison pkg-config g++ libssl-dev python-all python-all-dev python-all-dbg*

*sudo pip install networkx bitarray netaddr ipaddr pytest ipdb sphinx pyparsing==1.5.7 yappi*

* En caso de que no se cuente con Pyretic instalado siga estos comandos:

*wget* [*https://raw.github.com/freneticlang/pyretic/master/pyretic/backend/patch/asynchat.py*](https://raw.github.com/freneticlang/pyretic/master/pyretic/backend/patch/asynchat.py)

*sudo mv asynchat.py /usr/lib/python2.7/*

*sudo chown root:root /usr/lib/python2.7/asynchat.py*

* Instalacion del arbol git

*git clone* [*https://github.com/git/git.git*](https://github.com/git/git.git)

*pushd git/contrib/subtree/*

*make*

*mv git-subtree.sh git-subtree*

*sudo install -m 755 git-subtree /usr/lib/git-core*

*popd*

*rm -rf git*

Clonación de repositorio al mi directorio principal

*cd~*

*git clone* [*http://github.com/frenetic-lang/pyretic.git*](http://github.com/frenetic-lang/pyretic.git)

Ejecutar Pyretic

**\*\*Esta ejecución deberá ser paso por paso en un ordenador ubicando las capturas de pantallas.**

Step 1: Open a new terminal: Ctrl + Alt + t (Above command will open a new terminal)

Step 2: Go to Pyretic folder via terminal cd pyretic/

Step 3: Update and install few additional packages sudo apt-get update

Step 4: Try to run the controller using below command ./pyretic.py –m p0 (When you first run the Pyretic for the first time, you may encounter a problem like the following figure)

Step 5: Install the following command to solve the module error sudo pip install yappi (Above command will install the yappi packages)

**After the installation, we can run again and check the controller run. We get the following information on our screen. The p0 is the proactive mode where packets are pushed to switches based on Pyretic policy. Generally, the highest performant mode is made currently available.**

Step 6: Run the following example to run the controller in a proper manner. ./pyretic.py –m p0 pyretic.modules.mac\_learner

Ejecutando Pyretic en Mininet

This section will help you to understand how to configure your Mininet tool to work with Pyretic controller instead of the default controller of Mininet.

Step 1: Open a new terminal. Ctrl + Alt + t (above command will open a new terminal)

Step2: Type the topology command to run the mininet Sudo mn –topo linear,3 –controller=remote (The above command will create a linear topology remotely)

Please do find out your port and ip address using the command netstat –a and further run the command in mininet as per below figure.

Step 3: Ping the hosts to calculate the packet transmission using the command h1 ping –c 3 h3 (above command will change directory path to floodlight )

1. Descargando el archivo OVA (Open Virtual Appliance) en el link: <http://sdnhub.org/tutorials/sdn-tutorial-vm/>.

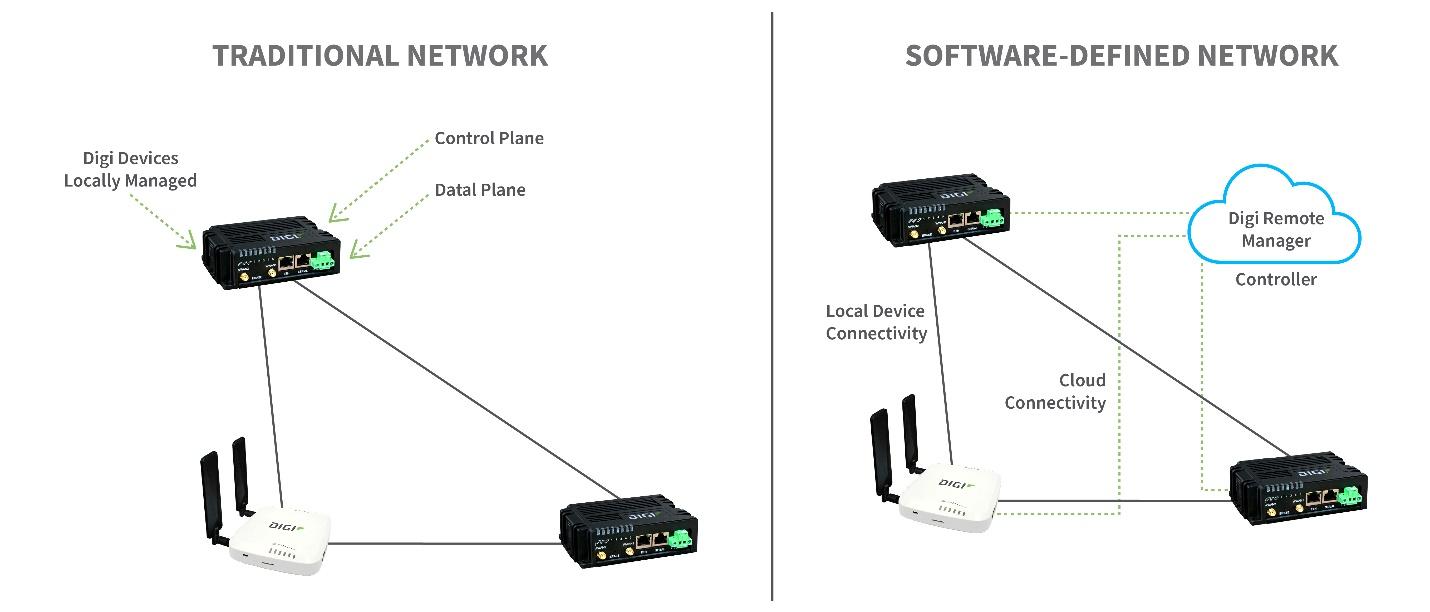
**Generando grafos de rendimiento de TCP y UDP en Mininet.**

**Grafos de trazado (GNUPLOT) con script gráfico.**

# **¿Qué protocolos se pueden implementar?**

***En qué se diferencia una red tradicional a una SDN.***

En las redes tradicionales, las tres capas (aplicación, control y red) residen en el mismo dispositivo de red. Con las SDN, la funcionalidad de la red existe en tres capas diferentes. Cada una gestiona una parte del funcionamiento de la red. En una red tradicional, cada dispositivo debe gestionarse y controlarse individualmente. Pero en SDN, las capas separadas operan a través de una interfaz unificada.



He aquí un resumen de una SDN frente a las redes tradicionales:

Las SDN separan las funciones de red en tres capas independientes. Las redes tradicionales combinan todas las funciones en un solo dispositivo.

Los dispositivos de red tradicionales se gestionan individualmente. Los dispositivos SDN IoT pueden gestionarse desde una plataforma de software central.

Las SDN utilizan software dinámico para redirigir el tráfico en caso de fallo, mientras que las redes tradicionales necesitan hardware adicional para la redundancia.

# **¿Qué tipos de redes se pueden implementar en SDN?**

Las soluciones de redes definidas por software gestionar dinámicamente el tráfico de red. Utilizan la inteligencia para optimizar el rendimiento. Como resultado, reducen significativamente los costes. Hay 4 arquitecturas SDN disponibles. Cada una funciona de forma diferente y proporciona ventajas distintas.

* **SDN abierta:**La SDN abierta utiliza protocolos de software de código abierto como OpenFlow para controlar y enrutar el tráfico de red.
* **API SDN:**API SDN utiliza algo llamado APIs southbound, para controlar el flujo de datos de cada dispositivo.
* **Modelo de superposición SDN:**Modelo de superposición SDN crea túneles para ejecutar múltiples redes separadas sobre la red existente.
* **Modelo híbrido SDN:**Un modelo híbrido SDN combina SDN y redes tradicionales como un paso hacia SDN, lo que significa que permite una transición gradual.

# **¿Con que funciones se puede acoplar las redes definidas por software?**

## *Infraestructura de datos*

Esta es la primera capa en la arquitectura SDN y se utiliza para el reenvío de un conjunto de paquetes basado en dispositivos de red que componen la infraestructura de red, cuyas funciones principales sirven para hacer cumplir las acciones de reenvío de paquetes de flujo de acuerdo con las instrucciones correspondientes proporcionadas por el controlador y para informar el estado de 6 la red cuando lo soliciten las aplicaciones de red.

## *Control*

El controlador de red es el componente más importante de la arquitectura SDN pues tiene una vista centralizada de la red y gestiona tanto la capa de aplicación como la capa de datos o infraestructura. El control de una configuración SDN consiste en uno o más controladores SDN que usan APIs

abiertas para ejercer control sobre los switches de la red. Además, impulsa las reglas de reenvío a los switches, los controladores también vigilan el medio, de esta manera los controladores tienen la capacidad de tomar decisiones de reenvío integradas con la gestión del tráfico en tiempo real.

## *Aplicación*

Reside sobre la capa de control. En esta capa se encuentran las aplicaciones de negocio y el servicio que se encarga de dirigirlas. Estas aplicaciones SDN comunican sus requisitos a la red a través de una API (Northbound) que es la que conecta con la capa de control, y están diseñadas para satisfacer las necesidades de los usuarios.

## *OpenFlow*

Fue desarrollado para estandarizar la comunicación entre el conmutador OpenFlow y el controlador basado en software en la arquitectura SDN, permitiendo así la programación de las tablas de flujos por parte de las aplicaciones software.

La arquitectura de OpenFlow comprende tres componentes principales;

1. Los interruptores compatibles con OpenFlow constituyen el plano de datos.

2. El plano de control tiene uno o más controladores OpenFlow.

3. El plano de control está conectado con los interruptores a través de un canal de control seguro, es decir la interfaz OpenFlow.

# **¿Qué es Cisco ACE?**

Es un servicio de virtualizacion robusta para las capa(4-7) con la finalidad de satisfacer las crecientes necesidades de los centros de datos actuales.

El módulo Cisco ACE permite a las empresas lograr cuatro objetivos principales de TI para la entrega de aplicaciones:

* *Aumentar la disponibilidad de las aplicaciones*
* *Acelerar el rendimiento de las aplicaciones*
* *Proteger el centro de datos y las aplicaciones empresariales críticas*
* *Facilitar la consolidación del centro de datos mediante el uso de menos servidores, balanceadores de carga y firewalls.*

Un elemento de diseño crucial de Cisco ACE, y un diferenciador entre la solución de Cisco y otras soluciones en el mercado, es su capacidad para utilizar la arquitectura virtualizada y las capacidades de administración basadas en roles que agilizan y reducen el coste de las operaciones involucradas en el despliegue, escalado, aceleración y protección de aplicaciones.

El módulo Cisco ACE proporciona la mejor escalabilidad y rendimiento del sector para gestionar el tráfico de aplicaciones, hasta 16 Gbps en un solo módulo; se pueden ejecutar hasta cuatro módulos en un solo chasis Cisco Catalyst 6500 Series, actualizables mediante licencias de software o nuevas adiciones de módulos, lo que proporciona a TI protección de la inversión y escalabilidad a largo plazo.

# **¿Cómo se involucra el software definido en redes en la arquitectura 5G?**

Las exigencias combinadas de recursos del tráfico de red 5G e Internet de las cosas requerirán un enfoque arquitectónico diferente. En primer lugar, se debe adaptar las redes para recibir el enorme crecimiento de contenido, como los videos y el gaming en la nube, en un gran ancho de banda.

Además, se deberá manejar una cantidad en aumento de casos prácticos de Internet de las cosas, en los que los sensores y otros dispositivos recolectan pequeños paquetes de datos que se deben analizar de forma continua. Por último, las redes deben ser fáciles de actualizar y administrar para que los CoSPs puedan seguir el ritmo de las exigencias en constante evolución.

# **¿Qué ejemplo es el mejor para la implementación de las redes definidas por software en una arquitectura de red?**

En un entorno de centro de datos, la capacidad de gestionar y controlar el tráfico de red de manera centralizada y programática es crucial para optimizar el rendimiento, la seguridad y la eficiencia operativa.

Imagina un centro de datos con miles de servidores interconectados que necesitan comunicarse entre sí para ejecutar diversas aplicaciones y servicios. Con SDN, en lugar de depender de la configuración manual de cada dispositivo de red individual, el controlador SDN centralizado puede tomar decisiones inteligentes sobre cómo dirigir el tráfico en función de las necesidades de las aplicaciones y las políticas de la red.

Por ejemplo, si una aplicación en particular requiere un ancho de banda adicional o una priorización de tráfico, el controlador SDN puede asignar dinámicamente recursos de red adicionales para satisfacer esas demandas, todo ello de manera automatizada y basada en políticas predefinidas.

Además, SDN facilita la implementación de características avanzadas de seguridad, como la segmentación de red basada en políticas y la detección de amenazas, ya que el control centralizado permite una visibilidad y un control más granulares sobre el tráfico de red.

En resumen, los centros de datos son un ejemplo destacado de cómo la implementación de SDN puede mejorar la agilidad, la flexibilidad y la seguridad de una arquitectura de red al permitir la gestión centralizada y programática del tráfico de red.

# **Características**

Características de la red definida por software. Hay 4 características únicas que definen las redes definidas por software:

* *Ágil*. A medida que cambian las necesidades empresariales y de aplicaciones, los administradores pueden ajustar la configuración de la red según sea necesario.
* *Gestión centralizada.* La SDN consolida la inteligencia de red, que proporciona una visión holística de la configuración y la actividad de la red.
* *Programable*. Capacidad para programar directamente las características de la red y de configurar los recursos de red de forma fácil y rápida a través de servicios automatizados de SDN.
* *Conectividad abierta.* La SDN se basa en estándares abiertos y se implementa siguiéndolos. En consecuencia, la SDN agiliza el diseño de la red y proporciona redes consistentes en una arquitectura independiente del proveedor.

# **Ventajas y desventajas**

* *Conmutación por error de WAN:* cuando un único router SDN gestiona muchas conexiones a Internet y falla una conexión, las SDN redirigen automáticamente el tráfico a otro router celular.
* *Mantener la conectividad:* las SDN automatizan la validación de la conectividad y pueden redirigir a conexiones de Internet de reserva. Con esta función, un único router puede admitir varios operadores para reducir significativamente el tiempo de inactividad.
* *Supervisión del estado de la red:* enruta de forma inteligente el tráfico de red para reducir los excesos de uso de datos y garantizar una conexión más rápida.
* *Personalizable:* el software inteligente SDN controla hardware de muchos proveedores diferentes a través de una interfaz común. Esto permite a las organizaciones supervisar sus equipos de la forma que tenga más sentido para los dispositivos.

# **Conclusiones**

Según el análisis realizado durante esta investigación, se ha determinado que las SDN son un conjunto de técnicas relacionadas con el área de redes computacionales, utilizando un enfoque pensado para el sector empresarial que pueda optimizar los recursos disponibles y mejore las comunicaciones al momento de enrutar el tráfico, esto mediante el uso de software para priorizar y ordenar el tráfico de la red.

Estas redes definidas por software surgen debido a la incapacidad de las redes convencionales de permitir cambios en los patrones de tráfico de forma dinámica, mediante la adición, eliminación o modificación de reglas de flujo en los dispositivos de interworking que soporte OpenFlow.

Y según la investigación podemos determinar que las principales características de la SDN son: Agil, Gestion centralizada, Programable y Conectividad abierta. De igual forma, logramos observar que el lenguaje utilizado en las SDN es Pyretic el cual es un lenguaje de las SDN y tiempo de ejecución que implementan algunos operadores de composición.

# **Bibliografía**

“Redes definidas por software” Obtenido de: <https://es.wikipedia.org/wiki/Redes_definidas_por_software>

Pleasant, Nate.2023 “Redes definidas por software (SDN): Por qué su organización la necesita” Obtenido de: <https://es.digi.com/blog/post/software-defined-networking>

Cisco. “Cisco ACE Application Control Engine Module for Cisco Catalyst 6500 Series Switches and Cisco 7600 Series Routers” 2013. Obtenido de: <https://www.cisco.com/c/en/us/products/collateral/interfaces-modules/ace-application-control-engine-module/product_data_sheet0900aecd8045861b.html>

“La Transformación de Red Habilitará una Nueva 5G” Obtenido de: <https://www.intel.la/content/www/xl/es/wireless-network/5g-network-architecture.html>

Núñez Ramires, Alex Vladimir. UNIVERSIDAD TÉCNICA DE AMBATO. 2015. “RED DEFINIDA POR SOFTWARE (SDN) EN BASE A UNA INFRAESTRUCTURA

DE SOFTWARE DE LIBRE DISTRIBUCIÓN” Obtenido de: https://repositorio.uta.edu.ec/bitstream/123456789/10587/1/Tesis\_982ec.pdf

Ccoyllo Sulca, Ingrid. 2018. “Redes definidas por Software (SDN)” Obtenido de: <https://informatica.ucm.es/data/cont/media/www/pag-103596/transparencias/redes-por-software-SDN.pdf>

Cáceres Guevar, Johan Eduardo y Casilimas Fajardo, Carlos Alexis. Arquitectura y “Funcionamiento de redes definidas por software (SDN)” Obtenido de: <https://repository.udistrital.edu.co/bitstream/handle/11349/29727/CasilimasFajardoCarlosAlexis2022.pdf?sequence=2&isAllowed=y>

# **Resultados**

# **Discusión**

# **Notas**

# **Tablas**

# **Figuras**