

Redes definidas por software (agosto 2020)

Michael Sebastian Preciado Garzon, Estudiante de la Escuela Colombiana de Ingeniería Julio Garavito

Abstract—Debido a que las redes tradicionales se vieron limitadas a la demanda de los usuarios hacia las aplicaciones actuales, al flujo masivo de información actual y demás virtudes del internet en los últimos años, se vio la necesidad de una nueva arquitectura que permitiera esta escalabilidad y flexibilidad a la hora de implementar una red, en 2008 se empezaron los primeros avances que luego se construiría en lo que hoy conocemos como redes definidas por software, pues en ese año se concreto un proyecto que buscaba separa el trafico de una red para poder usar la mis infraestructura pero una dedicada a investigación en la universidad de Stanford, la idea es separar la metodología de las tablas de flujo que usualmente manejaba cada fabricante en sus switches, de esta forma se abstraía el enrutamiento de la información y se dejaba en términos de envío en la arquitectura, logrando así, que se pueda programar el camino del flujo de información a través de los switches, todo esto con la comunicación de un controlador y el canal de los switches hablado por el protocolo OpenFlow, luego a raíz del éxito de esta arquitectura, se añadió una abstracción para que sea automático este redireccionamiento dados unos parámetros de una capa de aplicación que se conecta directamente a este controlador, permitiendo así que la implementación de una red así como su configuración sean más flexibles

Index Terms— API, CONTROLADOR, ENRUTAMIENTO, INFRAESTRUCTURA, OPENFLOW, RED, SDN

I. INTRODUCCIÓN

Para empezar a hablar de como se consolido las redes definidas por software o como sus siglas en ingles SDN, tenemos que empezar por definir cual fue la necesidad que surgió, que hacia que las redes convencionales no eran suficientes.

Y es que el enorme crecimiento de internet junto con los datos que transporta, incremento de igual forma a nivel de dispositivos, esto presento un problema para los administradores de red pues no solo implica asegurar la red contra atacantes si no que es necesario una infraestructura que sea capaz de responder a la demanda de solicitudes, con ello la solución mas inmediata es comprar mas equipos que soporten esta carga, sin embargo esto es poco recomendable debido a lo ya se mencionó, ante un crecimiento tan grande en la demanda una inversión a nivel de hardware podrá salir cada vez más costosa, esto entre otros factores implica que en una empresa con políticas de acceso definidas, el ingreso de un equipo a la red puede demorar horas para que sea configurado efectivamente, por lo que lo hace inviable en términos de escalabilidad

Otro de los factores principales a la hora de abordar esta carga en una red viene dada por la complejidad de las aplicaciones actuales, pues no solo implica la conexión entre cliente-servidor, sino que además el servidor tiene que conectarse a varios equipos a la vez para procesar la información del cliente antes de poderla retornar, tales conexiones como bases de datos o en casos donde se tenga una arquitectura de microservicios es necesario poder conectar esos servidores sin que exista una demora en dicha conexión.

En términos de seguridad, es un problema para los administradores de la red pues tienen que asegurar que la información que se desplaza dentro de la red empresarial, este protegida.

Una arquitectura dedicada a manejar grandes volúmenes de datos para su procesamiento tal es el caso de arquitecturas dedicadas a Big Data pues requiere de una transferencia de información veloz por lo que es preciso una infraestructura adecuada para esta demanda

II. OPENFLOW

Conociendo la necesidad de una mejor arquitectura en el 2008 a raíz de un proyecto en la universidad de Stanford “OpenFlow: Enabling Innovation in Campus Networks”, la

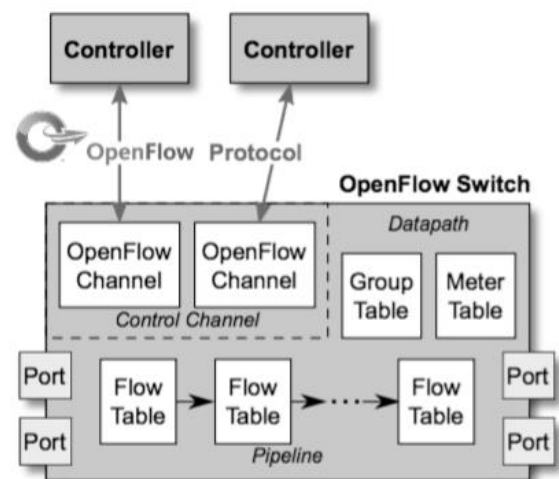


Fig. 2. Componentes principales de un switch que usa OpenFlow 1.5

propuesta es sencilla, a diferencia de los switch convencionales donde el camino de un paquete y el envío del mismo se hacen el mismo dispositivo, el objetivo es separar estas acciones abstrayendo la capa de encaminamiento en un controlador y

delegando la comunicación entre este controlador y el dispositivo que hace el envío, existe una API definida para esta comunicación, que también lleva el nombre de OpenFlow.

Al observar las tablas de flujo de los switch tradicionales se dieron cuenta que cada fabricante maneja las tablas de flujo a su manera, es por ello que vieron potencial en crear un protocolo que pudiera manejar estas tablas de flujo y no solo manejarlas, si no que programarlas dejando así que el administrador de la red tenga la flexibilidad suficiente para poder asignar los caminos que llevan a servidores en particular logrando así un escalabilidad al nivel de poder de decidir que tipo de servidor y a que ip va cierto flujo de información

Los switch que pueden usar el protocolo de OpenFlow tienen ciertos componentes que logran esta abstracción:

- El canal que se conecta con el controlador, este está asegurado con SSL además define como estará encapsulada la información y demás parámetros del protocolo
- El flujo que pasa a través del switch esta monitoreado y antes de ser agregado a la tabla, puede ser descartado, reenviado o almacenado, para tener un control sobre el paquete
- La tabla de grupos se usa para identificar que tipo de acciones se hacen sobre un flujo, con el objetivo de generalizar ese tipo de acciones sobre diferentes flujos
- La Meter Table se usa como su nombre lo indica para proveer mediciones sobre el trafico y tomar acciones, ya sea colocar con prioridad ese flujo o en un caso de DDOS que sea posible identificarlo y detenerlo

III. SOFTWARE DEFINED NETWORKING (SDN)

Las redes definidas por software, son arquitecturas dinámicas que se adaptan a la naturaleza fluctuante de las aplicaciones actuales, se ayuda de la existencia del protocolo OpenFlow pues desacopla el control del enrutamiento de la información y de su envío, logrando que ese control sea programable por lo que es mas sencillo poder definir los flujos anteriormente mencionados, sin embargo para implementar una red SDN no basta con implementar un sistema de OpenFlow existe mas de su arquitectura que a grandes rasgos se puede simplificar como un intermediario entre las aplicaciones y su flujo de información a través de la red, los diferentes componentes de un sistema SDN:

- Aplicación SDN: Se encargan principalmente de definir como es y como se comporta su red y lo hace saber al controlador por medio de un NBI driver
- Controlador SDN: Es el componente mas importante de esta arquitectura pues es el que

traduce los requerimientos de la red proveída por el NBI driver al NBI agent

- Control to Data-Plane Interface (CDPI): Esta interfaz es la que provee la comunicación entre el controlador y el paquete que contiene la ruta de datos de uno o varios equipos, se puede interpretar como la representación lógica del recurso físico, entonces se define la comunicación, entre un elemento de red lógico y el controlador, la representación lógica del recurso es a su vez la que se comunicara con el recurso físico real.
- Northbound Interface (NBI): se define de manera abstracta la representación de una red, y será así transmitida al controlador, con una serie de

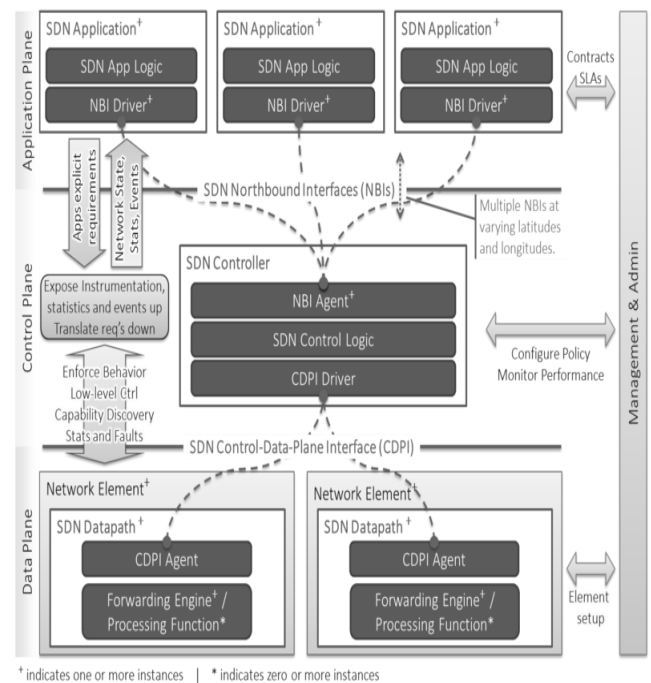


Fig. 1. Arquitectura a grandes rasgos de una SDN. Fuente: https://es.wikipedia.org/wiki/Redes_definidas_por_software#/media/Archivo:SDN-architecture-overview-transparent.png

parámetros y empaquetado, definido en acuerdo entre el driver y el agent

IV. SDN VS REDES TRADICIONALES

La principal diferencia es que la implementación de redes tradicionales es principalmente en hardware, mientras que un SDN esta implementado sobre software, debido a esto es más fácil administrar una red explícitamente porque no se depende de fabricantes y sus políticas, además del costo que acarrea el mismo.

Las conexiones que se hacen sobre las redes tradicionales dependen de switches, routers y demás infraestructura física mientras que, gracias a la API superior de la infraestructura de un SDN, es mas sencillo para los desarrolladores implementar una red, ya que de la otra forma estaría delegado al equipo de

infraestructura

Las redes definidas por software permiten mayor personalización a demanda que las redes tradicionales, es decir, los SDN permiten redirigir el tráfico de una solicitud a un servidor específico aun si ese servidor está temporalmente o es trasladado virtualmente a otra ubicación, para el usuario será imperceptible este cambio y permitirá mayor flexibilidad

REFERENCIAS

- [1] Sdn.systemsapproach.org. n.d. Chapter 1: Introduction — Software-Defined Networks: A Systems Approach Version 0.3-Dev Documentation. [online] Available at: <<https://sdn.systemsapproach.org/intro.html>> [Accessed 29 August 2020]
- [2] Sdn.systemsapproach.org. n.d. Chapter 3: Basic Architecture — Software-Defined Networks: A Systems Approach Version 0.3-Dev Documentation. [online] Available at: <<https://sdn.systemsapproach.org/arch.html>> [Accessed 29 August 2020].
- [3] Es.wikipedia.org. n.d. Openflow. [online] Available at: <<https://es.wikipedia.org/wiki/OpenFlow>> [Accessed 29 August 2020].
- [4] Es.wikipedia.org. n.d. Redes Definidas Por Software. [online] Available at: <https://es.wikipedia.org/wiki/Redes_definidas_por_software> [Accessed 29 August 2020].
- [5] Ibm.com. n.d. SDN Versus Traditional Networking Explained. [online] Available at: <<https://www.ibm.com/services/network/sdn-versus-traditional-networking>> [Accessed 29 August 2020].
- [6] Open Networking Foundation. n.d. Software-Defined Networking (SDN) Definition - Open Networking Foundation. [online] Available at: <<https://www.opennetworking.org/sdn-definition/>> [Accessed 29 August 2020].