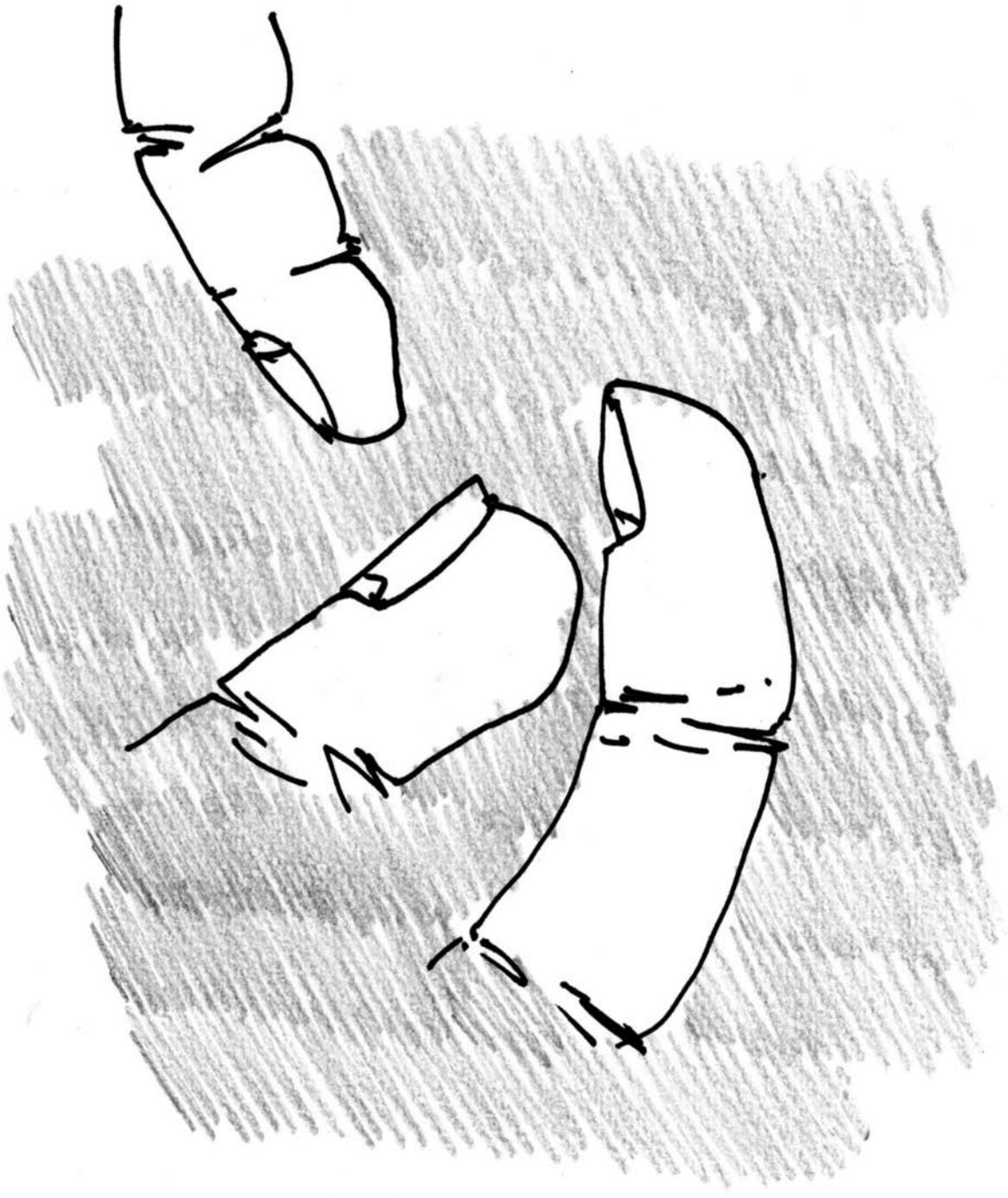

OPENFLOW: EL PROTOCOLO DEL FUTURO*

Openflow: The future protocol

*Daniel Felipe Blandón Gómez***

* Este artículo hace parte del trabajo final presentado para optar por el título de Especialista en Telecomunicaciones, Universidad de Manizales en el año 2013.

** Ingeniero de Sistemas y Telecomunicaciones, Universidad Católica de Pereira. Especialista en Telecomunicaciones. Contacto: daniel.blandon@ucp.edu.co

SINTESIS:
Con la aparición de las redes definidas por *software Defined Networking* (SDN), se ha dado apertura a nuevos proyectos que buscan optimizar y controlar el tráfico que corre por las redes de comunicaciones. El estándar *Openflow* es propuesto como uno de estos proyectos. Este artículo pretende explicar el protocolo en un lenguaje sencillo. Se parte de los referentes de proyectos a nivel mundial y nacional, para entender su usabilidad y alcance; asimismo, se busca conocer cómo las NGN (*Next Generation Networks*) trascienden a un plano donde el usuario tiene menos interacción con los dispositivos. Se proponen nuevas posibilidades de programación de rutas alternativas que permiten obtener los mismos resultados, sin causar traumatismos en los flujos de tráfico previamente diseñados.

DESCRIPTORES:
Software Defined Networking; Next Generation Networks; Openflow; NOX; Field-Programmable Gate Array; NetFPGA.

ABSTRACT:
With the emergence of networks defined by software Defined Networking (SDN), new projects that seek to optimize and control the traffic that runs through the communications networks have been opening. Openflow standard is proposed as one of these projects. This article aims to explain the Protocol in a simple language. It starts by referencing projects at global and national levels, to understand its usability and scope; it also seeks to know how the NGN (Next Generation Networks) transcend to a level where the user has less interaction with devices. New programming possibilities of alternative routes which can obtain the same results are proposed, without causing trauma to the previously designed traffic flows.

DESCRIPTORS:
Software Defined Networking; Next Generation Networks; Openflow; NOX; Field-Programmable Gate Array; NetFPGA.

OPENFLOW: EL PROTOCOLO DEL FUTURO

Para citar este artículo: Blandón Gómez, Daniel F. "Openflow: el protocolo del futuro". En: Revista Académica e Institucional, Páginas de la UCP, Nº 93, (Ene. - Jun. 2013): p. 61 - 72.

Primera versión recibida el 23 de abril de 2013. Versión final aprobada el 27 de marzo de 2014

Las redes de nueva generación, NGN, actualmente son punto de convergencia de tecnologías que por muchos años han ofrecido servicios de telecomunicaciones y se han propagado por el mundo. Redes que fueron pensadas para perdurar en el tiempo e imponerse como estándares, hoy son poco usadas, tal es el caso de X.25, ATM, *Frame Relay*, entre otras.

Un motor preponderante en la innovación y futuro de las redes de telecomunicaciones son las universidades y grupos líderes en investigación. Ellos están explorando nuevas estrategias para hacer las LAN[1] y WAN[2] del mañana más fáciles de manejar, más seguras y potentes, capaces de operar sobre diferentes tecnologías bajo el concepto de convergencia y buscando cambiar el modo de controlarlas; una de estas nuevas formas son las ya nombradas SDN.

Las SDN han despertado la creación de proyectos como *Openflow*. Este nuevo protocolo de comunicaciones (Brocade, 2013) se abre paso dentro de los protocolos tradicionales de red y transporte, y utiliza algunos de ellos para comunicación.

De otro lado, así como el *hardware* requiere un *firmware* para funcionar, en las redes definidas por *software* pasa igual. Un proyecto adicional es NOX[3] (*Network Operating System*), el cerebro (controlador) de una red implementada con *Openflow*; este componente completa el ecosistema SDN.

Bajo esta arquitectura se están proyectando los grandes fabricantes de *hardware* (Standford University, 2012). De ahí el interés por conocer acerca de estas tecnologías que seguramente llegarán a ser revolucionarias en la región, como lo están siendo ya en el mundo.

SDN (*Software Defined Networking*)

Esta nueva arquitectura de redes asistidas por *software*, define una nueva relación entre los dispositivos de la red y el *software* que los controla. En otras palabras, lo que busca SDN es cambiar el modo de operación para el reenvío de flujos de paquetes, y debido a que los fabricantes se reservan el derecho de divulgar cómo operan sus máquinas (*Firmware*), sobrepone un sistema operativo (Controlador), capaz de programar todo lo relacionado con la red (Brocade, 2013) (Figura 1).

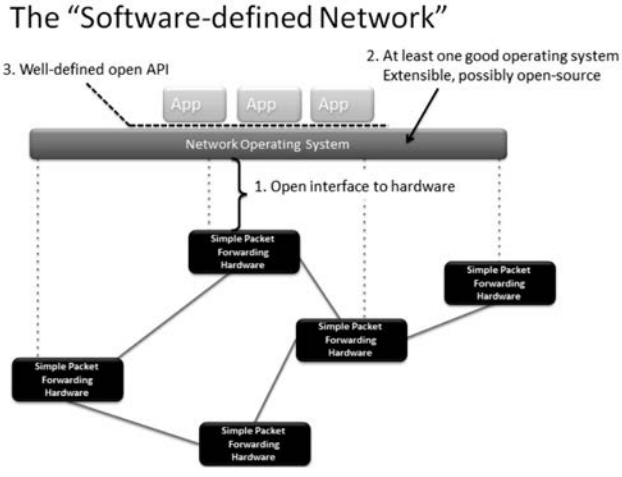

Figura 1. Arquitectura SDN (Heller, 2011)

1 Redes de área local
2 Redes de área extensa
3 Sistema operativo de red

Lo innovador de SDN reside en la posibilidad de controlar toda la red desde un único punto. Actualmente, la gestión de redes puede dividir en cinco partes (Ávila, 2007):

- Gestión de fallos
- Gestión de configuración
- Gestión contable
- Gestión del rendimiento
- Gestión de seguridad

No obstante, para aplicar cada uno de ellos hacen falta sistemas que ofrezcan los resultados esperados para alcanzar una gestión eficiente. Las SDN y *Openflow* aparecen como tecnologías prometedoras en el campo.

Las SDN se aplicarán en muchos campos. Existen algunos proyectos en versión *demo* que ya están siendo probados; por ejemplo: Virtualización de redes (*FlowVisor*), creación de prototipos de hardware (*OpenPipes*), balanceo de carga (*PlugNServe*), ahorro de energía (*ElasticTree*), movilidad (*MobileVMs*), ingeniería de tráfico (*Aggregation*) y video inalámbrico (*OpenRoads*) (Heller, 2011). Todos ellos implementados con *Openflow*, como componente en compañía de NOX.

NOX

El principal componente que tiene una red definida por *software* es el controlador o cerebro. Este se encarga de tomar decisiones sobre las mejores rutas que se encuentran disponibles para que un flujo de paquetes viaje entre un origen y un destino, almacena las políticas, registra las estadísticas y demás información relacionada con la red.

La debilidad generalizada en las redes actuales radica en que no existen protocolos ampliamente utilizados para mantener esta información consistente. Con la centralización (lógica) es fácil mantener estos componentes, porque se pueden unir mientras se mueven alrededor de la red. Los cambios de estado de la red simplemente requieren la actualización de los enlaces en el Controlador (Casado, Freedman, Pettit, Luo, McKeown & Shenker, 2007)

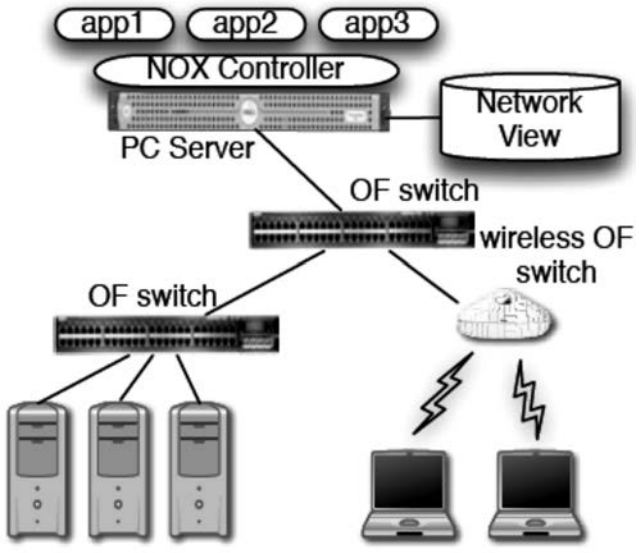

Figura 2. Componentes de una red basada en NOX: OpenFlow switches, un servidor con NOX para procesos de control y una base de datos con la red vista (Gude y otros, 2008)

El NOX es uno de los primeros controladores con el que se realizaron pruebas en redes SDN y que ha obtenido mayor rendimiento. Sin embargo, NOX no es el único controlador (Beacon, Maestro, Trema, Helios, BigSwitch, basado en Beacon); SNAC (basado en NOX 4.0, son otros) se toma como referencia por ser el más destacado (Tootoonchian, Gorbunov, Ganjali, Casado & Sherwood, 2010).

Las primeras pruebas realizadas con NOX mostraron que respondía a 30Krps (Miles de solicitudes por segundo). No obstante, las mediciones que se tenían de un tráfico promedio en una red con 100 *Switches* mostraban que este controlador era incapaz de administrar una red de estas dimensiones.

Debido a esas limitaciones, se evoluciona al NOX-MT, un controlador que trabaja con multi-hilos, donde la relación entre números de hilos y solicitudes atendidas es directamente proporcional. El NOX-MT utiliza diferentes técnicas de optimización; una de estas es la Entrada/Salida de procesos por lotes, lo que le permite un mayor rendimiento (Tootoonchian et al., 2010)

Según las pruebas realizadas, esta evolución conllevó a un rendimiento 6 veces superior al NOX; esto en una CPU de un solo núcleo. En una CPU de 8 núcleos se alcanzó un rendimiento 33 veces superior (Figura 3).

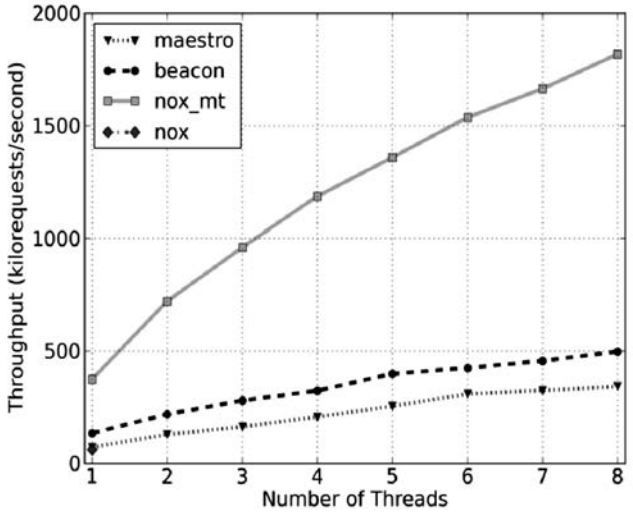

Figura 3. Análisis de controladores (Tootoonchian et al., 2010)

En la Figura 3 se muestran los resultados obtenidos con diferentes controladores. Puede verse cómo NOX-MT es superior en la relación número de hilos contra Kilo solicitudes por segundo. La medición que se realizó incluyó la comparación de varios controladores.

Entendiendo esto, se puede deducir que la función de un controlador radica en procesar el mayor número de solicitudes en el menor tiempo. Ahora bien, dependiendo del tamaño de la red puede aplicarse control distribuido o centralizado, como se muestra en la Figura 4.

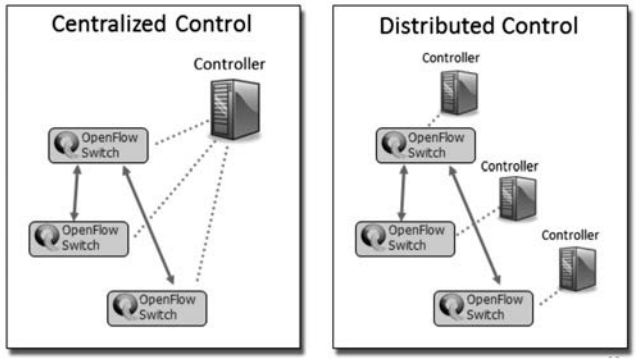

Figura 4. Control Centralizado frente a Control Distribuido (Heller, 2011)

Este artículo no busca concluir cuál es el mejor controlador. Se hace referencia a ellos para explicar en qué consiste su función y cómo pueden utilizarse de acuerdo con las necesidades que tenga cada red.

Openflow

Es uno de los más recientes hallazgos en lo que a innovación de protocolos de comunicación se refiere. Tuvo un predecesor llamado *Ethane,* que dio las pautas para lo que sería más adelante su desarrollo.

Ethane fue presentado como una nueva arquitectura de red para las empresas. Permite a los administradores definir una única política para toda la red, y luego se aplica directamente (Casado et al., 2007). En la Figura 5 se explica cómo el sistema opera con una pareja de conmutadores *Ethane* (*switch* 1 y 2), manejados con un sistema controlador centralizado (*Controller*), que gestiona la admisión y el enrutamiento de los flujos y las políticas de la red. Aunque se pensó que era algo radical, el diseño es compatible con los *hosts* y *switches* que actualmente se conocen.

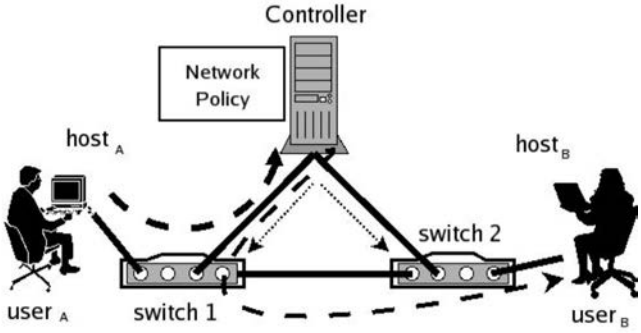

Figura 5. Ejemplo de comunicación en una red de *Ethane*. Configuración de ruta mostrada por líneas de puntos, el camino tomado por el primer flujo de paquetes que se muestra por líneas de trazos (Casado et al., 2007)

Desde el diseño de *Ethane* los ingenieros de Standford y Berkeley pensaron que era necesario abstraer el control de datos de los equipos de conmutación, debido a que la configuración de esos equipos se hace de manera individual. Por tal razón, implementaron un controlador el cerebro de la red, quien se encarga de realizar las tareas que se programen y luego replicarla en todos los dispositivos.

Ethane tiene dos características que dificultan poner en práctica las técnicas tradicionales de gestión de red:

- Se requiere conocimiento de los principios de la red (por ejemplo: usuarios y nodos).
- Requiere el control granular del enrutamiento en una tupla de 7 campos (usuario de origen, *host* de origen, *switch* de primer salto, protocolo, usuario destino, *host* destino, *switch* de último salto) (Gude et al., 2008).

Ahora bien, *Openflow* surgió luego de *Ethane*, con un método mucho más estructurado. Este protocolo "permite acceder directamente y manipular el plano de redireccionamiento de dispositivos de red como conmutadores y enrutadores, ya sean físicos o virtuales (basados en hipervisor)" (Hewlett Packard, 2008), es decir, facilita el acceso a dispositivos de red mediante una interfaz estándar. La facilidad a la hora de programar permite configurar una capa de control para poder centralizar la inteligencia de la red y brinda la capacidad de programarla tal como lo enuncia la tecnología SDN.

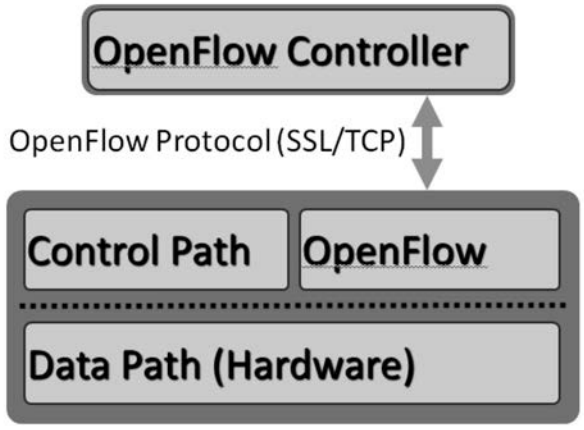

Figura 6. Arquitectura de comunicación Openflow (Heller, 2011)

Openflow utiliza protocolos (TCP/SSL) capa 4 y 5 del modelo OSI para la comunicación del plano de control y el controlador. Esta investigación se originó en los laboratorios de la Universidad de Standford y fue implementado como prueba a nivel del campus universitario. Ahora está siendo utilizado en redes de área amplía (WAN) y promete grandes logros para los *Carrier's*[4] del mundo entero.

¿Qué es *Openflow*?

Según sus creadores, *Openflow* es un protocolo para operar redes SDN. Fue desarrollado con base en *switches Ethernet*; es un standard abierto de comunicación entre un controlador (elemento principal de las SDN) y dispositivos de conmutación (McKeown et al., 2008). Al igual que TCP, su estructura está diseñada por mensajes, que establecen una comunicación y generan las acciones correspondientes.

4 Operador de Telefonía que proporciona conexión a Internet a alto nivel.

Openflow versión 1.1.0 (Pfa, y otros, 2011) detalla cómo se componen estos mensajes, al igual que los tipos y valores que los componen. Algunos de estos mensajes son:

- Header (Encabezado de todos los paquetes)
- Type (Tipo de mensaje), pueden tener valores como:
  - Mensajes inmutables
  - Mensajes de configuración
  - Mensajes asíncronos
  - Mensajes con comandos al controlador
  - Mensajes estadísticos
  - Mensajes de barrera
  - Mensajes de configuración de colas

Asimismo, define la estructura de puertos: su descripción, convenciones, características. También la estructura de colas: descripción, propiedades etc., toda la documentación que sustenta:

- Entradas de flujos, el respectivo "wildcard" para identificación de puerto
- Vlan
- Tipo de trama *Ethernet*
- Tipo de servicio
- Protocolos de red o transporte como TCP, UDP, IP, MPLS, y todo lo relacionado con el flujo de instrucciones y acciones para cada uno de ellos.

En últimas, un protocolo es un lenguaje; por tanto, la comunicación sólo es posible establecerse con un emisor y un receptor. *Openflow* es el lenguaje de comunicación entre el controlador y los *switches Openflow*.

Funcionamiento de *Openflow*

El funcionamiento de este protocolo está dado porque al separar el plano de datos del plano de control, se puede tener un mejor control de la red, y por tanto, mayor eficiencia.

Los dispositivos actuales tienen sus propios *Firmware*. En ellos se define cómo deben ser tratados los paquetes de acuerdo con la configuración realizada; además son propietarios, por tanto, difíciles de integrar.

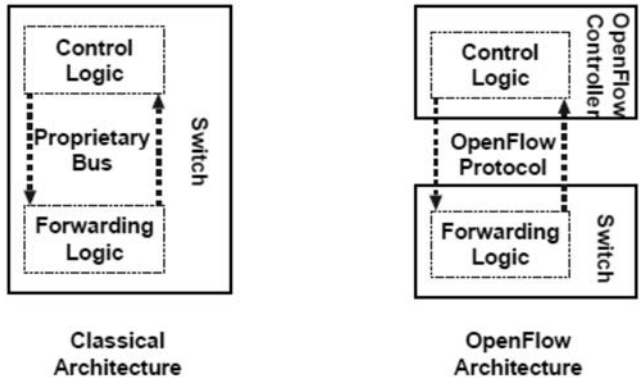

Figura 7. Relativo al hardware tradicional, el Protocolo OpenFlow mueve la ruta de control a un controlador externo (Sherwood et al., 2009)

La disociación del control de datos supone delegar esto a un controlador externo, es decir, se puede programar fuera del dispositivo la manera cómo van a ser procesados los flujos de paquetes. Estos últimos serán inventariados dentro de una tabla de flujos, que consignará la información del flujo de paquetes (MAC's, IP, puertos, etc.); así habrá un tipo de "Learning"[5] que permitirá conocer qué orígenes y destinos han pasado por el dispositivo (Pfa et al., 2011).

Con el transcurrir del tiempo se ha hecho evidente la necesidad de una tecnología de transporte que funcione con un plano de control centralizado. Además, que facilite la administración de todos los elementos de la red, para analizar y procesar el tráfico que fluye a través de ella.

Iniciativas como la gestión de redes busca, de alguna forma, utilizar aplicaciones que permitan entender el tráfico que transportan las redes. No

5 Una de las 5 funciones de un switch convencional: aprendizaje, inundación, reenvío, envejecimiento y filtrado (Learning, Flooding, Aging, Forwardind and Filtering).

obstante, sólo permiten obtener datos estadísticos; no hay programación que pueda reparar fallos o desviar rutas, por ejemplo.

El funcionamiento de *Openflow* contempla esta necesidad. El protocolo es el medio por el cual un dispositivo opera según la configuración que se haga en el controlador. En la Figura 5 se ilustra el *Openflow Switch* (OFS) y sus componentes.

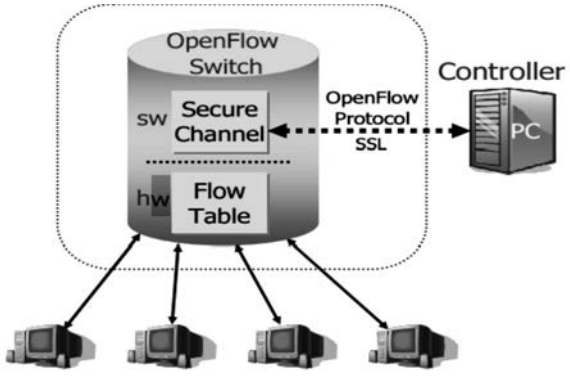

Figura 8. Arquitectura *Openflow Switch* (McKeown et al., 2008)

El *Openflow Switch* muestra claramente cómo se ubica el protocolo entre el *switch* y el controlador. También a nivel de *software* el canal de comunicación que establece con el protocolo SSL, y a nivel de hardware, la tabla de flujos. Estos 3 componentes necesarios para el funcionamiento, realizan:

1) Tabla de flujos: Acción asociada a cada entrada, que determina cómo el *switch* debe procesar el flujo.
2) SSL (*Secure Sockets Layer*): Capa de conexión segura, protocolo de conexión usado para el controlador y los dispositivos de conmutación.
3) OFP (*Openflow Protocol*): Un estándar abierto de comunicación entre el controlador y los dispositivos.

Las acciones que puede realizar el OFS son:

1) *Forwarding*: Reenvío de flujo de paquetes por un(os) puerto(s) específicos.
2) *Encrypting*: Encapsular y cifrar flujos de paquetes de datos para un controlador.
3) *Drop*: Borrado de paquetes por seguridad para frenar ataques de DDoS[6].

Estas acciones son consignadas en la tabla de flujos, que tiene 3 campos especiales, donde queda almacenada la información:

1) *Header*: Encabezado del paquete, define el flujo.
2) *Action*: La acción. Cómo se procesará el flujo.
3) *Statistics*: Estadísticas de procesamiento, número de paquetes y *bytes* por cada flujo.

En este orden, el *datapath* (camino de datos) de un *Switch* con *Openflow* asocia una acción a cada flujo de entrada. De acuerdo con la política (establecida desde el controlador), dicha acción es registrada, luego le indica al dispositivo cómo debe procesarla.

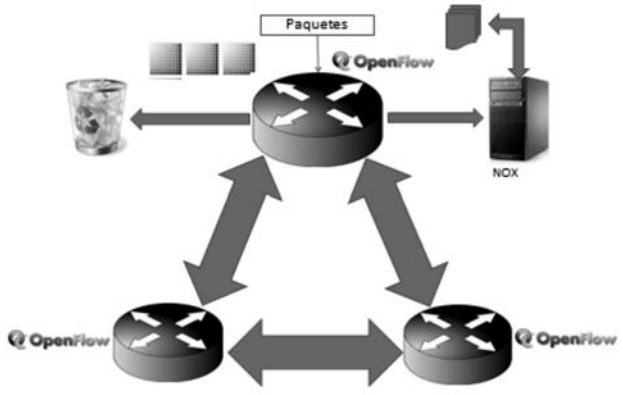

Figura 9. Ecosistema SDN/Openflow

Ahora bien, los fabricantes de tecnología (específicamente de *hardware* para redes) han sido por muchos años, dueños del *Firmware* para sus máquinas. Ese "secreto industrial" ha sido en

6 Denial of Service (Denegación de servicio).

gran medida lo que ha posibilitado que se busquen estándares para que exista interoperabilidad entre ellos.

De otro lado, ha venido creciendo una nueva propuesta llamada "FPGA" (*Field-Programmable Gate Array*) o Matriz de puertas (lógicas) programables en campo. Estas tarjetas provistas de "circuitos electrónicos reconfigurables que permiten crear nuevos circuitos que se comportan como nosotros queramos" (FPGA, 2010), han venido a ser complemento en el desarrollo de *Openflow,* ya que a través de estas placas se han formulado proyectos, como NetFPGA.

NetFPGA

El proyecto NetFPGA consiste en una plataforma reconfigurable de bajo costo, con *hardware* especial para configurar redes de alta velocidad. NetFPGA incluye los recursos de un dispositivo normal, memoria e interfaces *Gigabit Ethernet*, necesarios para construir un *router* y/o un dispositivo de seguridad (*Firewall*).

Debido a que el camino de datos (*dataphat*) se implementa en el *hardware*, el sistema puede manejar diferente tipo de tráfico (NetFPGA, 2011).

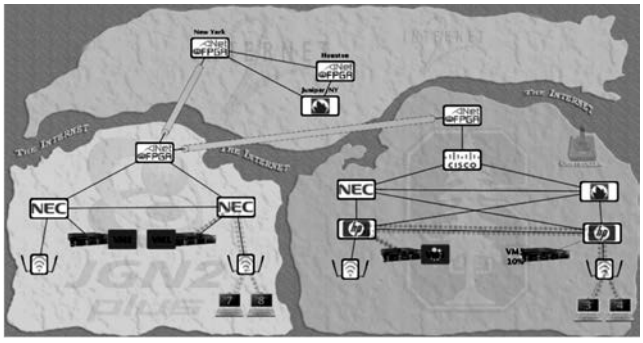

Figura 10. Interconexión de redes académicas con NetFPGA (Heller, 2011)

Esta tecnología ha permitido la interconexión de redes académicas y de investigación que utilizan dispositivos propietarios al interior de la red y como equipos de borde tarjetas NetFPGA (Heller, 2011). En la figura 10, se muestra la conexión con NetFPGA entre el campus de Standford University, JGN2[7] e Internet2[8]. Interconexiones de este tipo generarán la apertura de grandes desarrollos en países de Latinoamérica, donde también comienzan a utilizarse.

En Colombia, por ejemplo, hay un proyecto avalado por Colciencias donde se utilizó la tecnología NetFPGA con *Openflow* para experimentar el tratamiento de paquetes en un *router* remoto, a través de la Red Naciónal Académica de Tecnología Avanzada, RENATA (GITEL & TIC's, 2011).

Sería muy interesante que estos laboratorios trascendieran y llegaran a un entorno comercial, donde pudiera explotarse mucho más el uso de la tecnología. De ese modo lo hizo Google, que ya tiene en su red WAN, nodos implementados con *Openflow* (Dix, 2012).

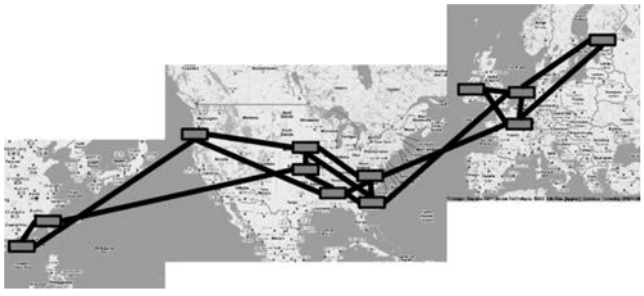

Figura 11. WAN *Openflow* Google (Levy, 2012)

7 Proyecto de investigación Asiático, más información en http://www.jgn.nict.go.jp
8 Proyecto de investigación Americano, más información en http://www.internet2.org/

Conclusiones

Las SDN comienzan a tomar fuerza en un momento donde el *software* y el *hardware* se están desagregando. Un precedente es la virtualización en la computación, donde grandes *mainframes* son utilizadas para procesamiento de datos con fines especiales.

Openflow se cataloga como uno de los protocolos del futuro. Su desarrollo y gran usabilidad han despertado en fabricantes como generadores de contenidos un gran interés; a esto se suman las exitosas implementaciones logradas por Google, como referente.

Las redes académicas de alta velocidad se han ido propagando a lo largo y ancho de todos los continentes. *OpenFlow* ofrece una excelente solución para generar valor agregado en estas redes, QoS, ToS, ingeniería de tráfico, entre otros, podrían ser de gran ayuda en la nueva generación de contenidos para estas redes.

Desarrollos de alta tecnología, como GRID, podrían valerse de la programabilidad de *Openflow*, centros de Bioinformática con nodos distribuidos a lo largo de la región y el país, establecerían una gran autopista controlada autónomamente.

NetFPGA, aunque limitada por el número de puertos, se plantea como solución de *hardware* libre para instituciones académicas y empresas que requieran manejar tráfico. Esto tendría un alto impacto en la economía, ya que podrían formularse desarrollos capaces de dar solución a necesidades puntuales dentro de estas instituciones.

Referencias

Anwer, M. B., Motiwala, M., Tariq, M. B., & Feamster, N. (10 de 2010). *ACM SIGCOMM*. Recuperado de sitio web de ACM SIGCOMM, http://www.sigcomm.org/sites/default/files/ccr/papers/2010/October/1851275-1851206.pdf

Ávila, R. A. (2007). "Gestión y administración de redes como eje temático de investigación " *Revista Avances - Investigación en Ingeniería.* Recuperado de sitio web de Avances, Investigación en Ingeniería, http://www.revistaavances.co/45

Brocade Corporation (2013). Recuperado de sitio web de Brocade, http://www.brocade.com/solutions-technology/technology/software-defined-networking/overview.page

Casado, M., Freedman, M. J., Pettit, J., Luo, J., McKeown, N., & Shenker, S. (2007). *ACM SIGCOMM*. Recuperado de sitio web de ACM SIGCOMM,, http://www.sigcomm.org/ccr/papers/2007/October/1282427.1282382

Dix, J. (2012). *Network World* . Recuperado de sitio web de Network World, http://www.networkworld.com/news/2012/060712-google-openflow-vahdat-259965.html?page=3

FPGA (2010). *FPGA LIBRE*. Recuperado de sitio web de FPGA LIBRE, http://fpgalibre.sourceforge.net/intro.html#tp2

GITEL, U. P., & TIC's, U. C. (2011). *U2 Route - Universal Universitary Router*. Recuperado de sitio web de U2 Route, http://u2route.ucp.edu.co/Proyecto2/u2route/

Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., et al. (2008). *ACM SIGCOMM*. Recuperado, de sitio web de ACM SIGCOMM, http://www.sigcomm.org/sites/default/files/ccr/papers/2008/July/1384609-1384625.pdf

Heller, B. (2011). *Openflow*. Recuperado de sitio web de Openflow, http://archive.openflow.org/wk/index.php/OpenFlow_Tutorial

Hewlett Packard (2008). *HP Networking*. Recuperado de sitio web de Hewlett Packard, http://h17007.www1.hp.com/co/es/solutions/technology/openflow/index.aspx

Levy, S. (2012). *Wired*. Recuperado de sitio web de Wired Enterprise, http://www.wired.com/wiredenterprise/2012/04/going-with-the-flow-google/

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., et al. (2008). *ACM SIGCOMM*. Recuperado de sitio web de ACM SIGCOMM, http://www.sigcomm.org/sites/default/files/ccr/papers/2008/April/1355734-1355746.pdf

NetFPGA. (2011). *NetFPGA*. Recuperado de sitio web de NetFPGA, http://netfpga.org/index.html

Open Networking Foundation (2012). *Open Networking Foundation*. Recuperado de sitio web de Open Networking Foundation, https://www.opennetworking.org/

Pfa, B., Lantz, B., Heller, B., Barker, C., Cohn, D., Talayco, D., et al. (2011). *Openflow*. Recuperado de sitio web de Openflow, http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf

Shenker, S., Rexford, J., Stoica, I., & Paxson, V. (2011). *EE122 Fall 2013* . Recuperado de sitio web de EECS, http://inst.eecs.berkeley.edu/~ee122/fa13/

Sherwood, R., Gibby, G., Yap, K.-K., Appenzellery, G., Casado, M., McKeowny, N. et al. (2009). *Openflow*. Recuperado de sitio web de Openflow, http://archive.openflow.org/wp/documents/

Standford University (2012). *Standford University School of Engineering*. Recuperado de sitio web de Standford University, http://www.stanford.edu/class/ee204/2012/Nicira%20Networks%20-DRAFT.pdf

Tootoonchian, A., Gorbunov, S., Ganjali, Y., Casado, M., & Sherwood, R. (2010). *Department of Computer Science, Standford University*. Recuperado de sitio web de The Mackeown Group, http://yuba.stanford.edu/~casado/sdnperf.pdf