



Universidad
Politécnica
de Cartagena



UNIVERSIDAD POLITÉCNICA DE CARTAGENA

TRABAJO FIN DE MÁSTER

Orquestación de servicios en redes ópticas multicapa

Autor:
José Juan Pedreño Manresa

Directores:
Pablo Pavón Mariño
José Luis Izquierdo Zaragoza

Autor:	José Juan Pedreño Manresa
E-mail del autor:	josej.pedreno@upct.es
Director:	Pablo Pavón Mariño
E-mail del director:	pablo.pavon@upct.es
Co-director:	José Luis Izquierdo Zaragoza
E-mail del codirector:	jose.l.izquierdo@ieee.org
Título del TFM:	Orquestación de servicios en redes ópticas multicapa.
Resumen:	
Desarrollo de una extensión para la herramienta Net2Plan, capaz de dar soporte a los protocolos BGP-LS y PCEP. Dicha extensión permitirá obtener emular de forma funcional un <i>PCE</i> multicapa capaz de determinar caminos para nuevos servicios (o aquellos servicios afectados por un corte de fibra) reutilizando algoritmos ya existentes en dicha herramienta, maximizando la interoperabilidad con equipamiento real.	
Titulación:	Máster en Ingeniería de Telecomunicación
Departamento:	Tecnologías de la Información y las Comunicaciones
Fecha de presentación:	Julio de 2019

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Plan de trabajo	2
1.4. Estructura de la memoria	3
2. Estado del arte	5
2.1. Software-Defined Networking	5
2.2. Path Computation Element y Path Computation Client	7
2.3. Application-Based Network Operations	8
3. Desarrollo de PCE, PCC y entorno de simulación	11
3.1. Comunicando Net2Plan con ABNO	11
3.2. Operación dinámica multicapa	13
3.3. Caso de Estudio	14
3.3.1. Workflow 1: Establecimiento de conexión PCC-PCE	14
3.3.2. Workflow 2: Aprovisionamiento de servicios	14
3.3.3. Workflow 3: Reacción en caso de fallo en enlace de fibra	15
4. Conclusiones	19
A. Dynamic Operation of an IP/MPLS-over-WDM Network Using an Open-Source Active Stateful BGP-LS-Enabled Multilayer PCE	21

Lista de acrónimos

ABNO	Application-Based Network Operations
ALTO	Application-Layer Traffic Optimization
BGP-LS	Border Gateway Protocol with Link-State extensions
CAPEX	CAPital EXpenditure
ERO	Explicit Route Object
IETF	Internet Engineering Task Force
IP	Internet Protocol
I2RS	Interface to the Routing System
LSP	Label-Switched Path
NMS	Network Managment System
NLRI	Network Layer Reachability Information
OAM	Operations, Administration, and Management
OPEX	Operational EXpenditure
OSPF-TE	Open Shortest Path First with Traffic Engineering extensions
OSS	Operational Support System
PCC	Path Computation Client
PCE	Path Computation Element
PCEP	Path Computation Element Protocol
RSVP-TE	Resource reSerVation Protocol with Traffic Engineering extensions
SDN	Software-Defined Networking
TE	Traffic Engineering
TED	Traffic Engineering Database
VNTM	Virtual Network Topology Manager

Capítulo 1

Introducción

1.1. Motivación

Existe una complejidad inherente en la planificación y operación de redes multicapa. Por un lado, las operadoras y proveedores de servicios poseen departamentos separados para la administración de la capa IP y la capa óptica; existiendo una mínima, o a veces incluso nula, cooperación entre dichos departamentos. Esta falta de comunicación ocasiona que nuevos servicios finales por parte de clientes puedan tardar días o incluso semanas en gestionarse y desplegarse, tal y como se muestra en la Figura 1.1 (imagen obtenida de [1]). Además, el aprovisionamiento de circuitos virtuales así como otros procesos requieren intervención humana, ocasionando todo ello una operación de red lenta y costosa.

Afortunadamente, el incremento en la demanda de ancho de banda, así como la dinamicidad y flexibilidad de nuevos servicios (servicios de *streaming*, *online gaming*...), están obligando a las operadoras a explorar soluciones basadas en la orquestación multicapa [2], las cuales se espera que ayuden a optimizar los flujos de trabajo en el aprovisionamiento de servicios, mejoren los costes de operación, y solventen los problemas ocasionados por la falta de interoperabilidad entre fabricantes.

Una de las soluciones aparecidas en los últimos años para la orquestación multicapa es el concepto de *Path Computation Element* (PCE) [3], una entidad software que permite agilizar el establecimiento de circuitos virtuales mediante el cómputo de la ruta más óptima.

1.2. Objetivos

El principal objetivo de este trabajo de Fin de Máster es el desarrollo de un PCE completamente funcional que posea las siguientes características:

- **Multicapa:** Debe ser capaz de operar tanto en la capa óptica como en la capa IP.
- **Stateful:** El PCE debe conocer el estado completo de la red, esto es, topología, recursos, y el estado de los mismos.
- **Activo:** De forma proactiva, el PCE debe ser capaz de sugerir cambios en la red (despliegue de circuitos ópticos o caminos MPLS) sin una petición expresa, por ejemplo en caso de fallo en un enlace de fibra.

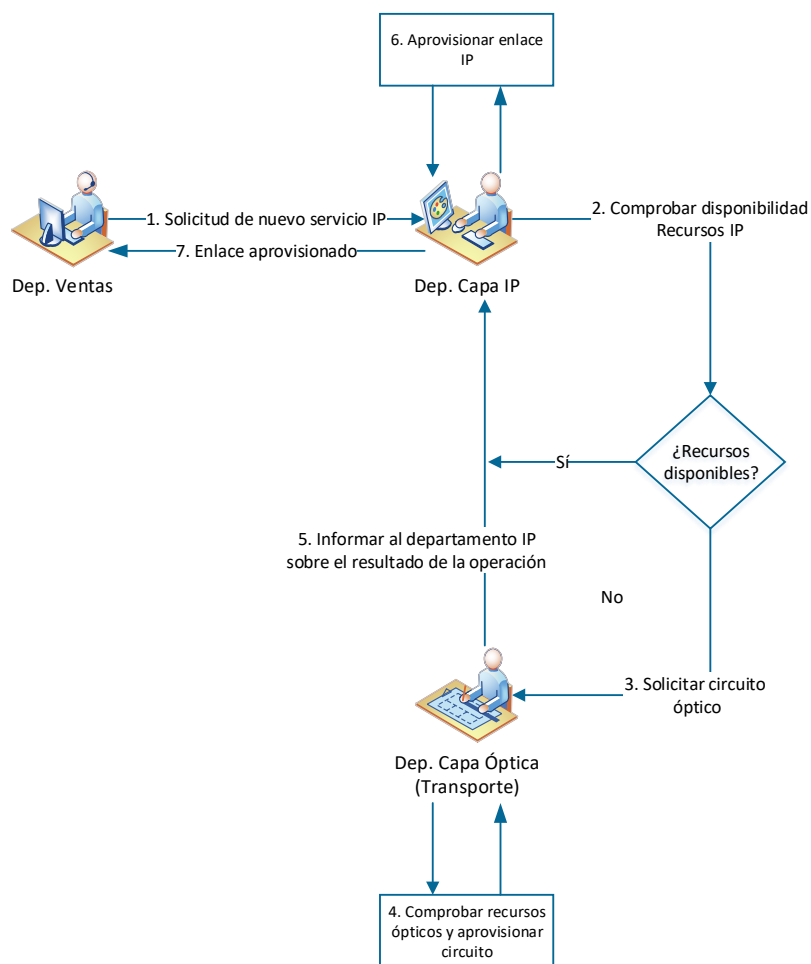


FIGURA 1.1: Flujo de trabajo para el aprovisionamiento de nuevos servicios

Además, para su validación, se creará un entorno de pruebas, incluyendo un *Path Computation Client* (PCC), que actuará como cliente del PCE, emulando una red óptica multicapa, proporcionando las herramientas para el control y gestión de la misma. Ambos elementos se programarán en *Java*, como extensiones de la herramienta *open-source* de planificación y simulación *Net2Plan* [4, 5].

1.3. Plan de trabajo

La ejecución de este trabajo se ha dividido en una serie de fases bien definidas:

1. Estudio y aprendizaje. En primer lugar se procederá al estudio y familiarización del estado del arte actual, incluyendo las necesidades del mercado y las tecnologías incipientes. También se estudiarán los diversos protocolos y estándares (algunos de ellos todavía en desarrollo), así como trabajos previo e investigaciones precedentes.
2. Diseño y modelado software. Se realizará el diseño y estructura de clases tanto del PCE como del PCC, incluyendo los mecanismos de comunicación entre ambos.

3. Programación del PCE y PCC (en ese orden).
4. Definición del caso de estudio. Una vez ambas extensiones hayan sido programados, se definirá y elaborará un caso de estudio que comprenda el uso de todas las funcionalidades descritas en los objetivos.
5. Pruebas y depurado. En base al caso de estudio anterior, se realizarán las pruebas que permiten comprobar y validar el correcto funcionamiento de los elementos programados, así como depurar fallos y mejorar funcionalidades.

1.4. Estructura de la memoria

El Capítulo 2 contiene un breve repaso de las tecnologías y estado relacionado con el trabajo realizado. Se detalla en el Capítulo 3 la arquitectura empleada, los módulos desarrollados así como el caso de estudio realizado para la evaluación de este trabajo. Por último, las conclusiones se presentan en el Capítulo 4.

Capítulo 2

Estado del arte

2.1. Software-Defined Networking

Desde un punto de vista funcional, las redes de comunicaciones pueden clasificarse en tres planos diferentes: datos, control y gestión (véase Figura 2.1, imagen obtenida de: [6]). El plano de datos (*data plane*) engloba los diferentes dispositivos de red, responsables de la transmisión y encaminamiento de datos. El plano de control (*control plane*) representa los protocolos utilizados para rellenar las tablas de encaminamiento de los dispositivos antes mencionados. Por último, el plano de gestión (*management plane*) incluye aquellos servicios y aplicaciones utilizados para monitorizar y configurar de forma remota la funcionalidad de control, por ejemplo, las herramientas basadas en SNMP [7]. En resumen, el plano de control responsable de definir las políticas de red, el plano de control de su ejecución y el plano de datos del encaminamiento de los datos acorde a dicha política.

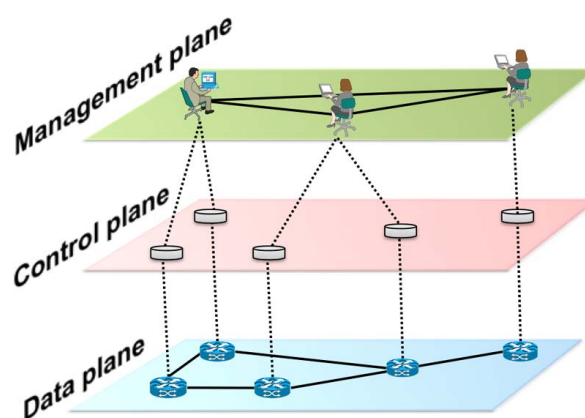


FIGURA 2.1: Planos funcionales de una red de comunicaciones

Tradicionalmente, en las redes basadas en tecnología Internet Protocol (IP), los planos de control y de datos han estado intrínsecamente acoplados e integrados de forma individual en cada uno de los dispositivos de la red, dando lugar a una estructura muy descentralizada. En su momento, dicho diseño era parte integral del concepto de “Internet”, y por tanto se consideró la mejor opción a la hora de garantizar resiliencia de red (capacidad de mantener la operatividad y funcionalidad ante fallos). No obstante, la consecuencia directa de este enfoque ha sido que con el paso de los años, y debido al crecimiento exponencial, las redes tradicionales son cada vez más rígidas y complejas de operar.

Para facilitar las tareas de gestión y operación, los fabricantes de equipamiento de red ofrecen soluciones especializadas de hardware, software, así como aplicaciones especiales de control. Sin embargo, por norma general, dichos productos solo son compatibles con los dispositivos del propio fabricante. Para conseguir una total “interoperabilidad” de sus dispositivos, las operadoras de red deben por tanto adquirir y mantener una multitud de soluciones/aplicaciones de gestión. Los costes de despliegue y operación (CAPEX/OPEX) asociados a una infraestructura de red son más que significativos, con tasas de retorno de inversión de décadas, lo cual dificulta la innovación así como el añadir nuevos servicios y característica (por ejemplo, control de acceso, *traffic engineering*...)

El concepto de *Software-Defined Networking (SDN)* está estrechamente ligado al de OpenFlow, y a los trabajos de investigación de la *Universidad de Stanford*. Haciendo uso de su definición original, las redes definidas por software definen un nuevo paradigma de red, donde el plano de datos y el de control están desacoplados, siendo el primero controlado por el segundo de forma remota. Con el paso de los años, no obstante, dicha definición se ha ido volviendo más laxa, y a día de hoy es ampliamente usada para definir cualquier situación que implique el uso de “software” para la gestión de una red. Una descripción canónica de SDN fue proporcionada por los autores de [6]:

La definición original se refiere a un paradigma de red, donde el estado de reenvío en el plano de datos está desacoplado del plano de control, que controla a distancia el primero. Sin embargo, muchas veces el término ha cambiado de la definición original al referirse a cualquier cosa que involucre “software” en la gestión de redes. Una definición más estricta, basada en cuatro pilares [6], otorga a SDN las siguientes características:

1. Desacoplo de los planos de datos y control. El control funcional se elimina de los dispositivos de red, y su única función pasa a ser el encaminamiento de datos.
2. Encaminamiento basado en flujos/destino y no en saltos. Puede definirse un flujo como un conjunto de campos y valores, haciendo la función de filtros, así como un conjunto de acciones a realizar basadas en las coincidencias de dicho filtrado. Esta abstracción permite normalizar el comportamiento de diferentes equipos de red (*routers, switches, firewalls, middleboxes*...) [8].
3. Control lógico externo. Puesto que el plano de control está desacoplado de los dispositivos, es necesario el uso de un controlador SDN, el cual es una entidad que proporciona la lógica de control necesaria para configurar cada dispositivo de red con las políticas de encaminamiento de paquetes.
4. Programabilidad *Off-platform*. La red debe ser configurable mediante aplicaciones que interactúen con el controlador SDN, sin tener conocimiento de los dispositivos subyacentes o la tecnología de los mismos. Esta quizás sea la característica más fundamental de SDN.

Por último, cabe destacar que la centralización del control de red ofrece ventajas adicionales a las antes mencionadas. El uso de lenguajes y descriptores de alto nivel, hacen que la definición de las políticas de red sea no solo más fácil, sino además menos propensa a errores, en comparación con la definición individual por dispositivo. Además, el programa de control puede actuar de forma proactiva ante fallos o comportamientos anómalos de la red, sin necesidad de intervención humana. Por último, un control centralizado, conociendo el

estado global de la red así como de los recursos que la componen, simplifica en gran medida el desarrollo de nuevos servicios y aplicaciones.

2.2. Path Computation Element y Path Computation Client

A la hora de aprovisionar Label-Switched Paths (LSPs), flujos de tráfico de extremo a extremo, el cálculo del camino a seguir es una tarea crucial si se quiere hacer una utilización mínima de los recursos así como aplicar técnicas de ingeniería de tráfico (Traffic Engineering (TE)). En dominios singulares, la búsqueda de dicho camino suele realizarlo el nodo de origen, mientras que la reserva de recursos y la señalización se lleva a cabo mediante protocolos distribuidos como Resource reSerVation Protocol with Traffic Engineering extensions (RSVP-TE). Que estos dos procesos estén separados puede ocasionar un uso subóptimo de los recursos de la red. Si dicho problema se aplica en redes multicapa y multidominio, la complejidad aumenta de forma sustancial. Cuando el origen y destino de un servicio pertenecen a operadoras distintas, la necesidad de confidencialidad impide la compartición de forma abierta de la información sobre los recursos existentes.

Para solventar esta problemática, el organismo Internet Engineering Task Force (IETF) propuso la creación de una entidad llamada Path Computation Element (PCE) [9]. La finalidad de un PCE es nutrirse del estado de la red para poder calcular caminos de la forma más eficiente posible; dichos cálculos son realizados a petición de los diversos nodos de la red, que pasan a convertirse en clientes, o Path Computation Clients (PCCs). La principal ventaja que aporta un PCE es el poseer una visión global de la red, un “ojo que todo lo ve”, que permita el uso más eficiente posible los recursos. Un ejemplo de esto se muestra en la Figura 2.2.

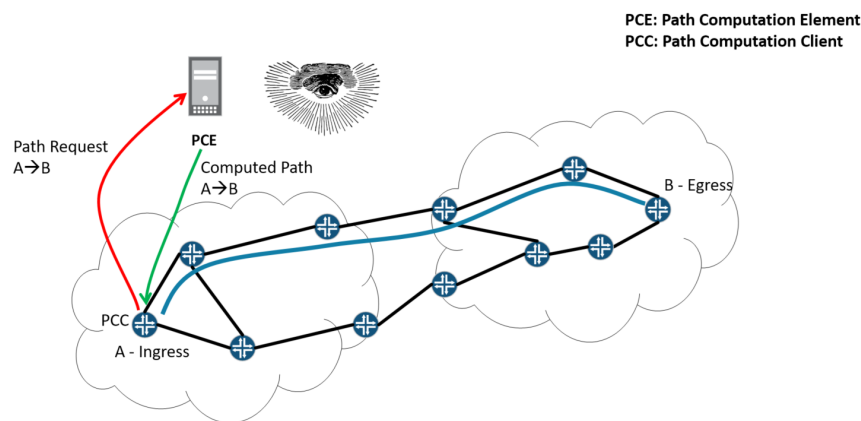


FIGURA 2.2: Ejemplo de petición de camino a un PCE por parte de un PCC

Un PCE puede implementarse de diferentes maneras, según la tecnología empleada, o las restricciones en su uso. Por ejemplo, basándose en el dominio del mismo, puede ser monocapa, limitándose al cálculo de caminos solamente en la capa IP o la capa óptica; en cambio un PCE multicapa puede establecer caminos sobre diferentes capas. También puede clasificarse como *stateful* o *stateless*. En el primer caso, el PCE tiene en todo momento un conocimiento global sobre la topología (nodos y enlaces), así como los recursos utilizados; sin embargo un PCE *stateless* no conoce el estado de la red, cuando recibe una petición del PCC, además de las restricciones normales (nodos de origen y destino y capacidad), e incluye además

un subconjunto de enlaces sobre los que hacer dicho cálculo. Por último, un PCE “pasivo” tan solo realiza cálculos ante peticiones explícitas de un PCC, mientras que si es “activo”, el PCE puede tomar decisiones de forma autónoma, reaccionando ante fallos o averías de forma inmediata sin esperar a recibir una petición.

2.3. Application-Based Network Operations

En los últimos años, el paradigma SDN se ha convertido en la opción dominante para la orquestación de redes, y como se ha visto en multitud de casos anteriormente, cualquier cambio de tecnología que parte desde cero es propenso a la proliferación de nuevos protocolos e interfaces; lo cual puede ocasionar una limitación en la evolución de dicho paradigma así como la interoperabilidad de dispositivos. Entre las posibles soluciones propuestas para solventar este problema surge Application-Based Network Operations (ABNO) [10]. ABNO puede definirse como una arquitectura modular desarrollada con el objetivo de proporcionar una operación y conectividad multidominio y multitecnología basándose en la reutilización de estándares y protocolos ya existentes.

Según el estándar, ABNO se compone de ocho bloques funcionales (véase Figura 2.3) además de las interfaces que los interconecta, sin embargo, es importante destacar que en un entorno de producción no todos los bloques son de obligada implementación. Uno de los componentes principales es el Controlador ABNO (*ABNO Controller*). Este elemento permite a diferentes aplicaciones, como Network Management System (NMS)/Operational Support System (OSS), dictar el encaminamiento de datos y servicios así como conocer el estado actual de la red. La solicitud de un nuevo encaminamiento al controlador hace que este compruebe el estado de la red y aprovisione los recursos necesarios para establecer dicho camino sin violar la política de red. Igualmente, el controlador también es responsable de reoptimizar la red en estado de operación, a la vez que atiende nuevas peticiones de encaminamiento, vigila el estado de la red y reacciona ante fallos. Otro elemento de la arquitectura ABNO es el *Operations, Administration, and Management (OAM) handler*, cuya responsabilidad reside en la recolección de errores de las diferentes capas de la red. *OAM handler* además monitoriza la red recopilando las avisos de error de diferentes dispositivos (usando protocolos como IPFIX o NETCONF), generando después reportes de alto nivel que serán enviados al controlador ABNO así como el NMS. Además, la forma de integrar en la arquitectura ABNO las diversas políticas de red es mediante un *cliente Interface to the Routing System (I2RS)* [11]. Como grupo de trabajo del organismo IETF, I2RS tiene como objetivo proporcionar una arquitectura para la interacción en tiempo real (con un sistema basado en eventos) con los sistemas de encaminamiento de los dispositivos de red.

El *PCE* proporciona búsqueda y selección de caminos, mientras que el *provisioning manager* está encargado de desplegar y configurar los caminos proporcionados por el PCE, usando protocolos del plano de control como OpenFlow o NETCONF. Es importante señalar que Asimismo, ABNO posee un elemento opcional llamado *Virtual Network Topology Manager (VNTM)*, el cual es capaz de proveer conectividad sobre la capa física, asociando enlaces virtuales de la capa superior con circuitos ópticos de la red WDM.

El descubrimiento de la topología y por tanto conocer el estado global de la red es fundamental para la labor del PCE; la arquitectura ABNO se nutre de diferentes bases de datos

para conocer dicho estado. La base de datos de ingeniería de tráfico, *Traffic Engineering Database (TED)*, es un elemento indispensable para cualquier implementación de ABNO, la capacidad de los enlaces, y su utilización. Dicha base de datos se nutre con las extensiones de ingeniería de tráfico de diversos protocolos como *Open Shortest Path First with Traffic Engineering extensions (OSPF-TE)*. Como añadido, se sugiere el soporte de bases de datos de *LSP* que contienen información de los caminos establecidos, así como la información relativa a los circuitos ópticos ocupados.

En elemento crítico para un despliegue de ABNO en producción es poseer un criterio unificado en cuanto al despliegue de nuevos caminos y servicios. La arquitectura define un elemento llamado *Policy Agent*, el cual es controlado directamente por el OSS/NMS. Este agente autentica las peticiones entrantes, así como mantiene la contabilidad e información de los recursos asignados, lo cual se ve reflejado en los algoritmos de selección de caminos. El *policy agent*, es una parte fundamental que interviene en todas las decisiones tomadas por el controlador ABNO, el PCE y el servidor Application-Layer Traffic Optimization (ALTO). Por último, el protocolo ALTO tiene como finalidad proveer de información detallada sobre la capacidad de los enlaces y su utilización, ayudando en los procesos de selección de caminos y monitorización de la red.

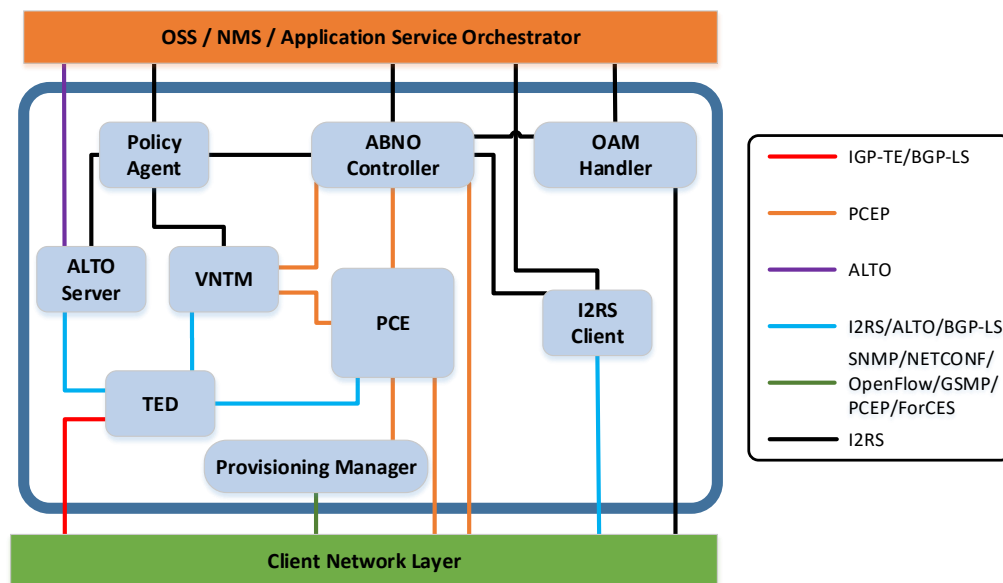


FIGURA 2.3: Arquitectura de ABNO

Capítulo 3

Desarrollo de PCE, PCC y entorno de simulación

Este capítulo está dedicado a presentar el grueso del trabajo realizado durante este TFM. Se desarrolló un PCE *stateful* activo y multicapa con el objetivo de ser ejecutado en una implementación de ABNO desarrollada por Telefónica I+D [12]. La comunicación y el intercambio de mensajes y respuestas del PCE con el resto de la arquitectura ABNO se realiza usando conocidos protocolos y estándares y borradores del IETF. En concreto, se hace especial uso de los protocolos *Path Computation Element Protocol (PCEP)* y *Border Gateway Protocol with Link-State extensions (BGP-LS)*, implementados en Java por Telefónica bajo la librería `netphony-network-protocols` [12, 13]. La característica “*stateful* activo” del PCE ofrece como ventaja que la entidad en cualquier momento dado tiene la información completa de la red multicapa (topología y recursos utilizados) y puede ofrecer recomendaciones de encaminamiento (ya sea como un nuevo LSP o circuito óptico en caso de fallo) sin una petición expresa del controlador (PCC). Tanto la implementación del PCE, PCC y entorno de pruebas se han realizado como extensiones de la herramienta de planificación *open-source* Net2Plan. Como forma de validación de los elementos desarrollados, se presenta también en este capítulo un caso de estudio completo que cubre los diferentes flujos de trabajo típicos de una operadora de red: establecimiento de servicios en una red multicapa y restauración dinámica en caso de fallo en la infraestructura física.

3.1. Comunicando Net2Plan con ABNO

La arquitectura de referencia usada para la integración del PCE desarrollado se muestra en la Figura 3.1. La red física se compone de routers IP/MPLS y equipamiento óptico. Cómo muestra la figura anterior, el entorno de pruebas utilizado hace uso de dos instancias de Net2Plan¹, con extensiones que habilitan su comportamiento como PCE y PCC respectivamente

El primer hito a conseguir en el entorno de pruebas es una red cuyo estado poder consultar y modificar. Para ello se implementó un PCC, emulador multicapa, como una nueva herramienta dentro de la interfaz gráfica de Net2Plan, cómo muestra la Figura 3.2a. Dicha herramienta además emula el comportamiento de un NMS/OSS, permitiendo la gestión y orquestación de la red, y por tanto el operador de la misma puede consultar el estado de

¹la versión 4.0.0 fue usada para este trabajo.

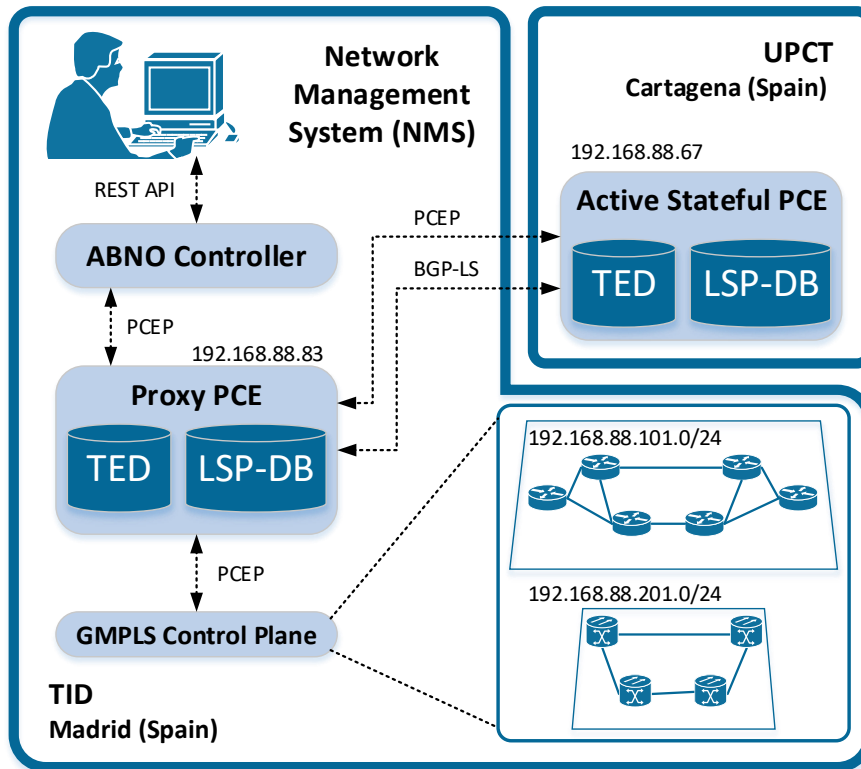


FIGURA 3.1: Arquitectura de la interconexión de Net2Plan y ABNO

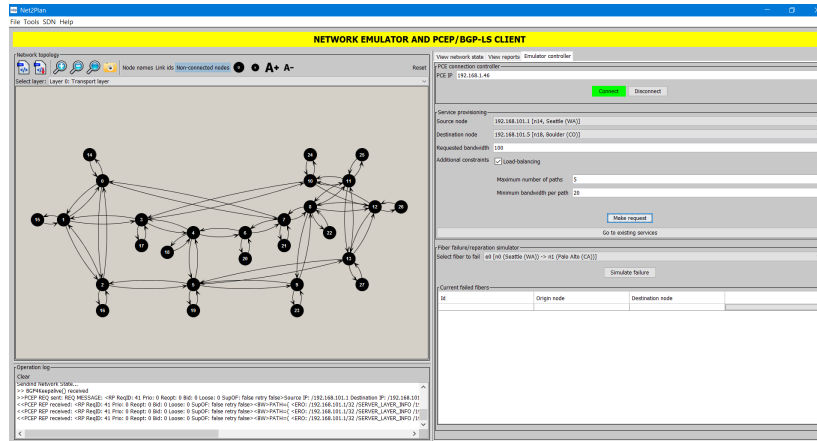
la misma y establecer nuevos servicios, ya sean (i) caminos MPLS-TE, o (ii) circuitos ópticos (*lightpaths*). Se asume que una vez establecido un camino MPLS, este es permanente. Además, la herramienta PCC permite emular eventos de fallos y reparaciones en enlaces de fibra.

En segundo lugar, el PCE multicapa (como elemento externo) permite recibir peticiones del PCC anteriormente mencionado, proporcionando las respuestas pertinentes. Es más, puesto que el PCE tiene la característica de “activo”, de forma proactiva reaccionará ante aquellos fallos que se produzcan en la red (los cuales son notificados mediante mensajes BGP-LS de tipo *Update*) sugiriendo acciones destinadas a evitar la interrupción del tráfico: reencaminamiento de servicios MPLS o establecimiento de nuevos *lightpaths*. El prototipo de PCE, corriendo como daemon, puede verse en la Figura 3.2b, como una modificación de la interfaz de comando de Net2Plan.

Para asegurar la autonomía e interoperabilidad de los diferentes módulos descritos en los párrafos anteriores, la comunicación entre PCC y PCE se realiza exclusivamente mediante los protocolos *PCEP* [3] y *BGP-LS* [14], de forma que cualquiera de los dos pueda ser sustituido por otra aplicación o equipamiento sin merma de rendimiento o funcionalidad.

Para iniciar la comunicación, el PCC establece dos *sockets* con el PCE. El primero de estos sockets permite una comunicación bidireccional mediante *PCEP*. Por otro lado, el segundo tiene como objetivo la interacción unidireccional del PCC con el PCE (en la otra dirección

solo se transmiten mensajes de tipo *Open* y *Keep-Alive*); el PCE recibe notificaciones acerca de la topología (nodos y enlaces), así como de eventos de fallo y reparación.



(A) Emulador de red y cliente PCEP/BGP-LS (PCC)

```

C:\WINDOWS\system32\cmd.exe - java -jar Net2Plan-cli.jar -mode pce

C:\Net2Plan-0.4.0>java -jar Net2Plan-cli.jar -mode pce

PCE MAIN SERVER
=====
Starting PCEP & BGP-LS servers...
PCE successfully started
=====
|  MULTILAYER AS-PCE  |
=====
| Options:            |
| 1. View current sessions |
| 2. Print TEDB and LSPDB  |
| 3. Exit              |
|=====|
Choose an option (1-3):

```

(B) PCE como interfaz de comando en línea

FIGURA 3.2: Capturas de pantalla de ambos módulos

3.2. Operación dinámica multicapa

El principal reto a la hora de desarrollar un algoritmo de operación multicapa consiste en poder establecer de forma eficiente nuevos LSP teniendo cuenta las restricciones impuestas por cada una de las capas. Dicho algoritmo debe ser capaz no solo de establecer un circuito óptico sobre la topología física, sino también acomodar una conexión extremo a extremo sobre la topología virtual generada en base a los *lightpaths* establecidos (red del cliente). Por último, y no por ello menos importante, también debe ser capaz de reaccionar de forma automática ante ciertos eventos, como fallos y reparaciones, informando al PCC de las acciones apropiadas en cada caso. Debido a sus características “activo y *stateful*”, el PCE puede aprovechar el campo “*delegate*” del protocolo PCEP [15] para que el PCC pueda delegar en

el PCE los lightpaths y LSP ya establecidos. En otras palabras, el PCE por tanto puede sugerir el aprovisionamiento de nuevos circuitos ópticos o el reencaminamiento de conexiones existentes sin una petición explícita del PCC.

3.3. Caso de Estudio

El resto de este capítulo está dedicado al caso de estudio inicialmente mencionado. Se describirán una serie de flujos de trabajo (*workflows*), mostrando como interactúan PCC, PCE y el algoritmo multicapa desarrollado, en cada uno de estos casos, incluyendo capturas de pantalla así como el intercambio de mensajes. Se asume que el operador de la red posee el control tanto del emulador de NMS/OSS como del PCE (es irrelevante si su ejecución se realiza en la misma máquina o en dispositivos separados). Para ejecutar el PCC, debe descargarse e instalarse Net2Plan [4] y posteriormente añadir la extensión desarrollada². Una vez abierta la herramienta, se selecciona la opción “Network Emulator and PCEP/BGP-LS Client” bajo el menú SDN (ver Figura 3.2a). Para este caso de estudio, se consideró como topología de red una versión modificada de NSFNet [16], asumiendo que cada nodo posee tanto un router IP como equipamiento óptico. Dicha red puede observarse en la parte izquierda de la Figura 3.2a.

El analizador de paquetes Wireshark [17] fue usado para comprobar la correcta comunicación entre PCC y PCE, usando los filtros `pcep` | | `bgp` para ocultar aquellos paquetes no relacionados con este experimento. Por último, se asume que cada enlace de fibra da soporte a 80 canales WDM con una velocidad de línea de 40 Gbps por circuito.

3.3.1. Workflow 1: Establecimiento de conexión PCC-PCE

Para que el PCC pueda establecer una comunicación con el PCE, debe introducirse la dirección IP de este último en el campo apropiado y pulsar el botón “Connect” (ver parte superior derecha de la Figura 3.2a). Tal y como se definen los documentos RFC del IETF [18], la comunicación debe ser iniciada por el PCC. A continuación, se produce un intercambio de mensajes `PCEOpen`, `PCEPKeepalive`, `BGPOpen` y `BGPKeepalive` (tramas 187-202 en la Figura 3.3a).

Una vez establecida la comunicación, el PCC envía toda la topología al PCE. Una serie de mensajes `BGPUUpdate` con el campo Network Layer Reachability Information (NLRI) [19] son enviados (tramas 203-2015 en la Figura 3.3a). Puede apreciarse que diversos de estos paquetes aparecen con el mensaje “Malformed Packet” (paquete malformado); en el momento de realizar este caso de estudio, el disector de paquetes de Wireshark no estaba actualizado con las últimas especificaciones en cuanto a diseminación de topologías [14]. En concreto, la librería Netphony implementaba un borrador prematuro de la mencionada especificación³.

3.3.2. Workflow 2: Aprovisionamiento de servicios

La herramienta de emulación de NMS/OSS permite aprovisionar dos tipos de servicios: (i) circuitos ópticos (cuando la opción “Transport layer” está seleccionada, véase parte superior

²Código fuente disponible en: <https://github.com/jjpedreno/PCE-and-PCC-Net2Plan-Plugin>

³BGP Link-State Information Distribution Implementation Report, available at <https://tools.ietf.org/id/draft-ietf-idr-ls-distribution-impl-04.txt>

izquierda de la Figura 3.2a), y (ii), servicios MPLS, desplegando caminos extremo a extremo sobre la topología virtual creada mediante *lightpaths* (seleccionando la opción “Client layer”). El algoritmo multicapa sigue una estrategia *top-down*, estableciendo primero nuevos *lightpaths* en caso de que no exista capacidad suficiente en la capa superior, evitando así el bloqueo de servicios en las peticiones de LSP. Cabe destacar que para el aprovisionamiento de servicios MPLS, la conexión puede repartirse en diferentes caminos en caso de no existir un camino único capaz de alojar todo el tráfico solicitado. Esto es posible debido al uso del objeto `LOAD_BALANCING` (balanceo de carga), como parte del paquete `PCEPRequest`. Este flujo de trabajo comprende las tramas (276-280 de la Figura 3.3a):

1. El PCC solicita un nuevo servicio MPLS usando la opción “Service provisioning” (véase la sección central de la parte derecha de la Figura 3.2a), para ello se envía un mensaje `PCEPRequest` (Figura 3.3b). La capacidad solicitada puede estar comprendida entre 10 y 100 Gbps (para este experimento, se seleccionó el máximo valor). De forma opcional, puede incluirse la antes mencionada opción de `LOAD_BALANCING`, dando así la opción de desplegar el servicio a lo largo de diferentes caminos (como muestra la Figura 3.3b).
2. Una vez recibida la solicitud, el PCE intenta encontrar una secuencia de *lightpaths* (o varias secuencias, dependiendo de si la opción LB está activada), usando un algoritmo de *shortest-path* que tenga en cuenta la capacidad de los enlaces. En caso de que no haya ningún camino disponible por el cual aprovisionar el servicio, el PCE sugerirá el despliegue de cuantos nuevos *lightpaths* sean necesarios para acomodar el servicio. El algoritmo usa una técnica de *shortest-path* y *first-fit* a la hora de asignar la frecuencia de cada nuevo circuito óptico.
3. Si se diere el caso de que ningún camino se encuentra disponible (ni pueden desplegarse nuevos *lightpaths*), el PCE devuelve una respuesta de tipo `NO_PATH`, y la solicitud se considera rechazada. En caso contrario se procede al siguiente paso de la lista.
4. El PCE envía una respuesta en forma de mensaje `PCEPReply` (Figura 3.3c) con las acciones a llevar a cabo para el aprovisionamiento del servicio. En caso de estar activada la opción LB y se deban notificar múltiples caminos, el mensaje `PCEPReply` contendrá diferentes objetos Explicit Route Objects (EROs). Si deben desplegarse nuevos *lightpaths*, esto se notificará en el ERO mediante el objeto `SERVER_LAYER`.
5. A continuación, el PCC actualizará su estado de red, y remitiendo un mensaje `PCEPReport` que contiene el campo `delegate` con el valor 1. A partir de ese momento, el PCE toma el control de forma activa sobre aquellos *lightpaths* y servicios MPLS aprovisionados en la petición original. En caso de que se produzca un fallo, el PCE podrá sugerir nuevas acciones sin necesidad de una petición explícita por parte del PCC.

3.3.3. Workflow 3: Reacción en caso de fallo en enlace de fibra

Desde la herramienta NMS/OSS, el PCC puede emular fallos y roturas de fibra, con el fin de evaluar la reacción del PCE y su algoritmo multicapa, haciendo uso de la opción “Simulate failure” (parte inferior derecha de la Figura 3.2a). Debe señalarse que el PCC envía las actualizaciones sobre el estado de la red mediante BGP-LS, y puesto que todos los servicios y circuitos aprovisionados han sido “delegados” al PCE (cómo se explica en el Apartado 3.3.2),

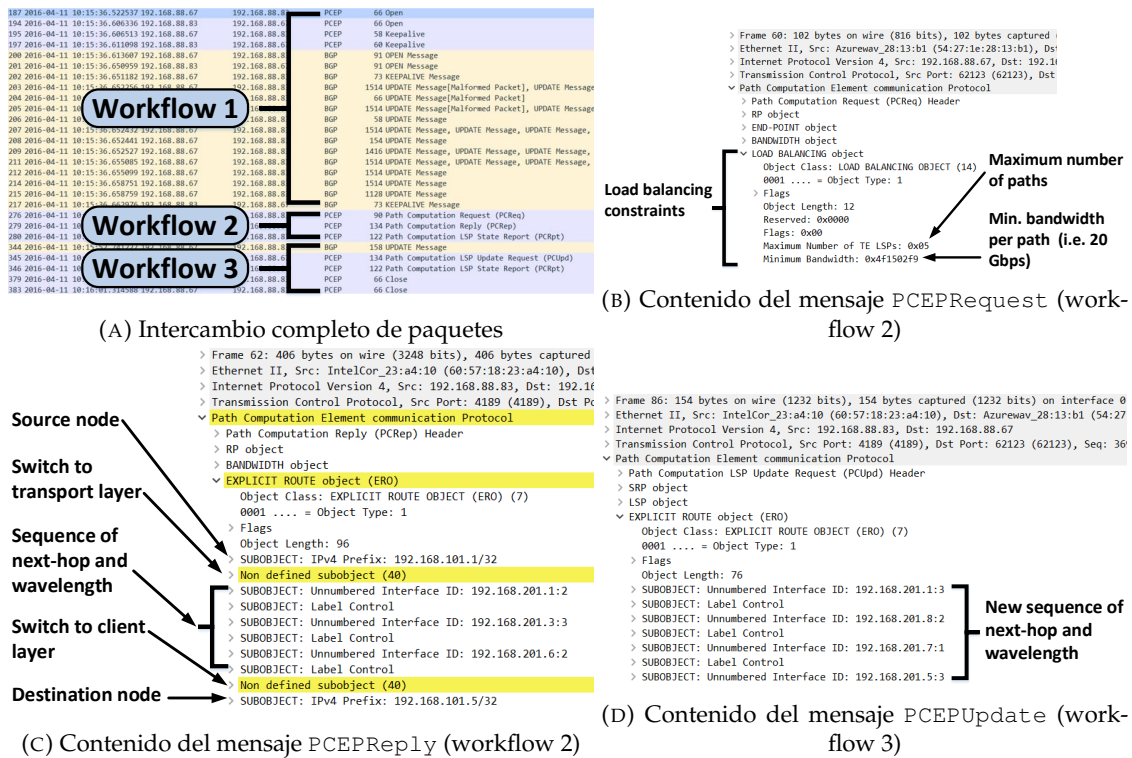


FIGURA 3.3: Trazas obtenidas con Wireshark para los diferentes flujos de trabajo

este último puede reaccionar sin necesidad de un mensaje PCERequest por parte del PCC. El algoritmo de restauración multicapa sigue una estrategia *bottom-up*, es decir, en primer lugar intenta restablecer por caminos alternativos todos aquellos *lighpaths* que se han visto afectados por el corte de fibra, y si no fuere posible, restablecer los servicios MPLS afectados sobre la topología virtual existente. El flujo de trabajo se describe a continuación (tramas 344-346 de la Figura 3.3a):

1. El PCC informa del fallo ocurrido mediante un mensaje BGPUpdate que contiene el atributo MP_UNREACH, relativo al enlace averiado (cómo describe la especificación NL-RI).
2. El PCE intenta reencaminar sobre la topología física aquellos *lighpaths* afectados. En caso de no poder reencaminar un circuito, el PCE intenta restablecer los servicios MPLS (correspondientes a dicho *lighpath*) sobre la topología virtual existente (suponiendo que exista capacidad suficiente para llevarlo a cabo).
3. El PCE devuelve al PCC una lista ordenada de acciones a llevar a cabo, mediante un mensaje PCEUpdate (ver la Figura 3.3d).
4. El PCC confirma la recepción de dicho mensaje devolviendo un paquete de tipo PCE-Report, con el campo `delegate` con valor 1.

La Figura 3.4 muestra las consecuencias de un corte de fibra y su posterior reencaminamiento. En este ejemplo, un corte en la fibra entre los nodos 2 y 5 (línea roja discontinua) ocasiona

el reencaminamiento del circuito óptico (camino original y reencaminado se muestran con líneas azules discontinuas y continuas respectivamente).

Por último, en este caso de estudio no se contempla ningún mecanismo de “vuelta a la normalidad”, en el cual los *lightpaths* reencaminados en caso de fallo vuelvan a su camino original una vez reparada la avería. Esta decisión se toma para minimizar la interrupción de tráfico, ya que dicha migración de caminos ocasionaría una situación de bloqueo [20]. Puede simularse un evento de reparación de un enlace dañado usando la opción “Simulate reparation”, la cual envía un mensaje BGP-LS (BGPUpdate con el atributo MP_REACH), pero tras su recepción, el PCE no tomará acción alguna.

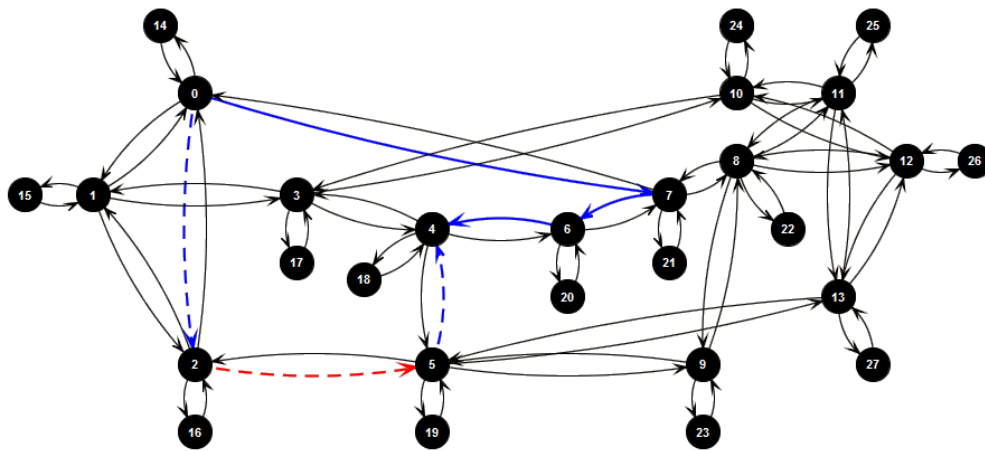


FIGURA 3.4: Reencaminamiento de un *lightpath* debido a fallo en enlace de fibra

Capítulo 4

Conclusiones

A lo largo de esta memoria de Trabajo de Fin de Máster se han presentado dos extensiones de Net2Plan: (i) un PCE activo, *stateful*, multicapa; y (ii) un emulador de red multicapa así como herramienta NMS/OSS, que realiza la labor de PCC. Se ha logrado la comunicación de ambos elementos a través de los protocolos BGP-LS y PCEP, y mediante un caso de estudio se ha evaluado su rendimiento. Gracias al uso de los estándares ya mencionados, dicho PCE debería ser capaz de interactuar con equipamiento real.

Parte del trabajo asume que el equipamiento óptico soporta dichos estándares, pero a decir verdad, estos dispositivos suelen configurarse y operarse con interfaces cerradas y propietarias. No obstante, a día de hoy los esfuerzos de la industria se centran en llevar el paradigma SDN a los dispositivos ópticos mediante proyectos como Transport API [21], con la finalidad de lograr un control completo extremo a extremo sobre redes ópticas de transporte “disgregadas” [22]

Por otro lado, plataformas como ONOS [23] u OpenDaylight [24] han demostrado que los controladores SDN han evolucionado más allá de proporcionar una interfaz común para la configuración de dispositivos. Dentro del contexto de ABNO, este tipo de aplicaciones pueden asumirlo no solo el rol de PCE, sino también de base de datos TED/LSP, VNTM, etc. Dando lugar a una plataforma que en si misma establece su propia arquitectura SDN.

Dicho esto, futuros esfuerzos deben centrarse en seguir sacando provecho de las capacidades de Net2Plan como valor añadido al paradigma de las redes definidas por software, sea cual sea su arquitectura. Dichos esfuerzos deben incluir mantener las capacidades multicapa, activa y *stateful*, introduciendo además elementos como la reoptimización proactiva, a fin de alcanzar el concepto de “*in-operation network planning*” [25].

En último lugar, destacar que el trabajo realizado fue presentado (y posteriormente publicado) por el alumno en el 18º Congreso Internacional de Redes Ópticas Transparentes (International Conference on Transparent Optical Communications, ICTON), en el año 2016. Dicho artículo puede consultarse en el Apéndice A.

Apéndice A

Dynamic Operation of an IP/MPLS-over-WDM Network Using an Open-Source Active Stateful BGP-LS-Enabled Multilayer PCE

Dynamic Operation of an IP/MPLS-over-WDM Network Using an Open-Source Active Stateful BGP-LS-Enabled Multilayer PCE

Jose-Luis Izquierdo-Zaragoza¹, Jose-Juan Pedreno-Manresa¹, Pablo Pavon-Marino¹, Oscar Gonzalez de Dios², Victor Lopez²

¹ Universidad Politécnica de Cartagena, Cuartel de Antiguones, Plaza del Hospital 1, 30202 Cartagena, Spain

² Telefónica I+D, C/ Don Ramón de la Cruz 82-84, 28006 Madrid, Spain

e-mail: {josel.izquierdo, josej.pedreno, pablo.pavon}@upct.es, {ogondio, vlopez}@tid.es

ABSTRACT

Leveraging on reference implementations of PCEP and BGP-LS protocols, we have developed an active stateful multilayer PCE extension for the open-source Net2Plan planning tool to orchestrate multilayer service provisioning and restoration in IP/MPLS-over-WDM networks.

Keywords: PCE, BGP-LS, multilayer dynamic operation, IP-over-WDM, multilayer restoration, Net2Plan.

1. INTRODUCTION

Considering the operational complexity of multilayer networks, operators typically have two separated departments to manage the IP layer and the optical layer, with minimal interaction between them. Moreover, many processes require manual intervention, thus leading to a slow and very costly operation [1]. Fortunately, the increasing bandwidth demand along with dynamic (and flexible) service offerings are forcing carriers to explore solutions based on multilayer network orchestration [2], which are expected to optimize workflows for service provisioning, improve costs and to get rid from vendor lock-in.

In this work, we present an active stateful multilayer Path Computation Element (PCE) [3][4], where a centralized entity has the complete multilayer information to compute paths according to the network state (*stateful* feature), and it is able to issue network recommendations (i.e., lightpath rerouting upon failures) without any request (*active* feature) from the Path Computation Client (PCC). Exchange of requests and replies is based on IETF standards and drafts related to PCEP (PCE protocol) and BGP-LS [5], which are emerging as the de-facto solution in carrier-grade networks, using the `netphony-network-protocols` library [6]. Thanks to BGP-LS, we consider a real-world mechanism to automatically discover and update network state in the PCE.

The PCE is implemented as an external plugin of the open-source Net2Plan tool [7][8]. We test our implementation using an emulated multilayer network scenario (also developed as a plugin for Net2Plan). Our case study is based on typical multilayer workflows of operators, that is, service provisioning across several layers, and dynamic restoration under failures in the infrastructure. To the best of our knowledge, this is the first implementation of an active stateful PCE with multilayer capabilities.

The rest of the paper is organized as follows. Our reference testbed and multilayer scenario are described in Sections 2 and 3, respectively. Section 4 reports a case study on NSFNet network. Finally, we conclude in Section 5.

2. TESTBED SETUP

Fig. 1 depicts the reference architecture for the integrated multilayer PCE scenario. The physical network consists of IP/MPLS routers and optical equipment. As shown in Fig. 1, our testbed is based on the interplay of two Net2Plan (version 0.3.1) instances.

First, we implemented a basic multilayer network emulator able to maintain network state, as a tool inside the graphical user interface of Net2Plan (see Fig. 2a). This network is orchestrated by a controller managed from a Network Management System/Operations Support System (NMS/OSS), also emulated therein, from which the operator is able to monitor network state and establish new services, either (i) MPLS-TE paths (LSPs) or (ii) optical lightpaths. In this study, established MPLS services are assumed to be permanent. Also, we can create fiber failure/reparation events from the emulator.

Second, the external PCE is in charge of receiving requests from the network emulator (acting as PCC), and giving the corresponding responses. In addition, it will be able to proactively react to failure/reparation notifications from the PCC, based on BGP-LS updates, suggesting changes (i.e., setup new lightpaths, rerouting MPLS services, and so on). This preliminary version runs as a daemon on top of Net2Plan command-line interface (see Fig. 2b).

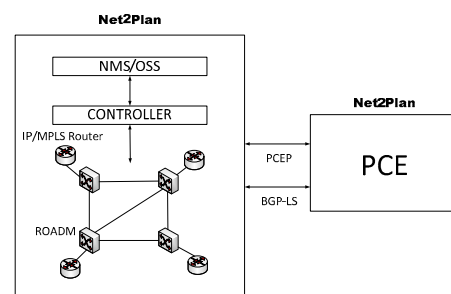
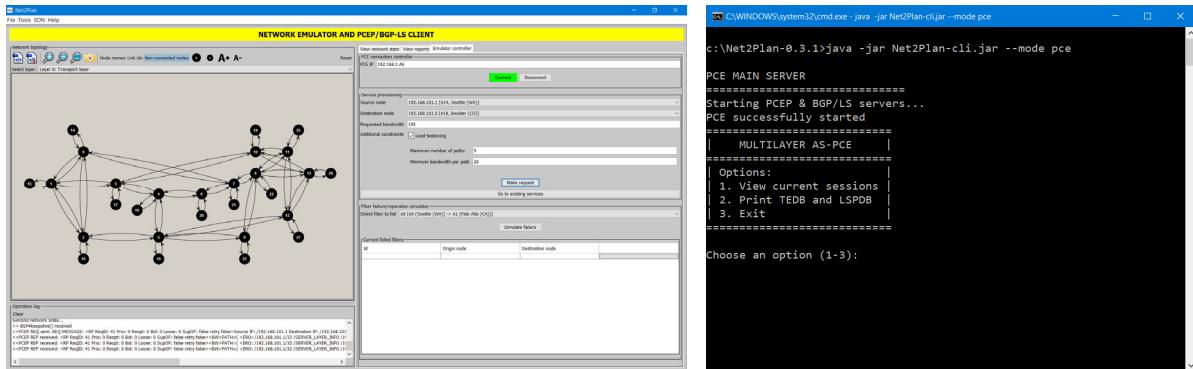


Figure 1. Reference architecture.

Communication between PCC and PCE is uniquely based on PCEP and BGP/LS. To do this, the PCC establishes two sockets with the PCE. The first one is used for bidirectional PCEP communication [3]. In contrast, the second one is used for unidirectional BGP-LS interaction [5] (excluding Open and Keep-Alive messages) for BGP Updates, either for topology reporting or for failure/reparation notification, from PCC to PCE.



(a) Network emulator and PCEP/BGP-LS client (PCC) (b) PCE daemon and hypervisor
 Figure 2. Snapshot of the implemented modules.

3. MULTILAYER DYNAMIC OPERATION

The primary challenge of active multilayer operation algorithms is to setup new LSPs in different layers while taking into account of different layer-specific constraints. These algorithms should be able to: (i) route lightpaths over the physical topology; and (ii) route connections in the lightpath-based client network. Last, but not least, they should be able to react to failure/reparation events, and to inform to the PCC of the corresponding actions to be performed. Thanks to the active and stateful nature of our multilayer PCE, and the “delegate” field [3] to support LSP delegation from the PCC to the PCE, the latter has full control over lightpaths and MPLS services. Namely, the PCE is able to suggest the establishment of new lightpaths and rerouting existing connections without an explicit request from the PCC. We consider a bottom-up strategy, that is, first we try to reroute affected lightpaths over the surviving physical topology, and then still-affected MPLS traffic is restored over the virtual topology.

4. CASE STUDY

In this section, we describe a reduced set of workflows, including message exchanges, from the implemented multilayer operation. We consider a multilayer version of the NSFNet reference topology. Optical fibers support up to 80 WDM channels working at 40 Gbps. We consider two instances of Net2Plan are running according to Section 2, whereas Wireshark packet-analyzer is used to monitor the PCC-PCE communication. This experiment can be replicated through a Net2Plan installation with the corresponding plugins in [9].

4.1 Workflow 1: Establishing PCC-PCE connection

To establish the connection with the PCE, the PCC just needs to put its IP address in the tool, and click over “Connect” (see upper right side in Fig. 2a). Then, a sequence of PCEP/BGP Open and Keep-Alive messages is exchanged (frames 187-202, 217 in Fig. 3a), as defined by IETF documents. Upon connection, the PCC sends the whole topology information to the PCE via BGP-LS update messages using the Network Layer Reachability Information (NLRI) field [10] (frames 203-215 in Fig. 3a). Note that several BGPUpdate messages appear as “Malformed Packet”. The reason is that Wireshark’s BGP packet dissector does not reflect the latest BGP specification, at the time of writing, for topology information dissemination [5]. In any case, the used reference implementation of PCEP and BGP-LS protocols [6] has been reported to work properly [11], even interoperating with other implementations from other players, and validation processes using real-world equipment are being undergone.

4.2 Workflow 2: Service provisioning

Our NMS/OSS provides two different provisioning services to customers: (i) MPLS services (when “Client layer” is selected), where LSP provisioning is performed on top of the virtual topology, including automatic setup of new lightpaths to avoid blocking of LSP requests; and (ii) lightpath services (when “Transport layer” is selected), in order to allow setup of single lightpaths.

Focusing on the MPLS service provisioning, the connection might be established over multiple paths, if no one is able to carry the full connection. This is reflected by the LOAD_BALANCING (LB) object in the PCEPRequest. The workflow comprises the following steps (frames 276-280 in Fig. 3a):

1. The PCC issues a `PCEPRequest` (see Fig. 3b) for the MPLS service using the “Service provisioning” option (see middle section on the right side in Fig. 2a). Any requested bandwidth between 10 and 100 Gbps is valid (100 Gbps in the example). Optionally, it may include a LB object relaxing the routing policy to serve the LSP over several paths (see Fig. 3b).
2. The PCE looks for a sequence of lightpaths, or a set of them (if LB is allowed), with enough capacity to setup the LSP (using a capacitated shortest path algorithm). If fails, try to establish an end-to-end, or several (if LB is allowed) lightpath to carry the whole connection. The algorithm uses a shortest path routing with first-fit for the wavelength assignment.
3. If a path (or more than one, if LB is allowed) could not be found, the PCE issues a `NO_PATH` reply and the request is withdrawn. Otherwise, go to 4.
4. The PCE issues a `PCEPReply` (see Fig. 3c) with the allocation found. If multiple paths exist (and LB is allowed), the `PCEPReply` will include multiple Explicit Route Objects (EROs). In case a path requires a new lightpath, this information is handled between a pair of `SERVER_LAYER` objects in the ERO.
5. The PCC then updates the network state, and issues a `PCEPReport` message with “delegate” option. Hence, from now on the PCE has control over lightpaths and MPLS services (i.e. failures will be proactively handled by the PCE).

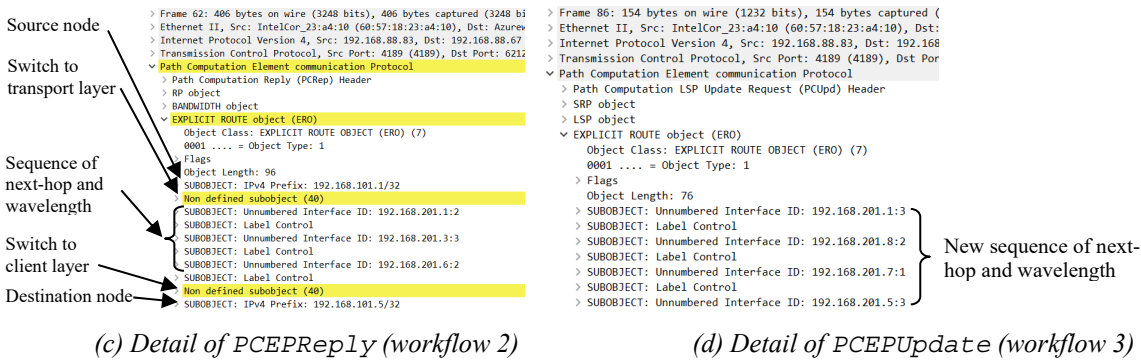
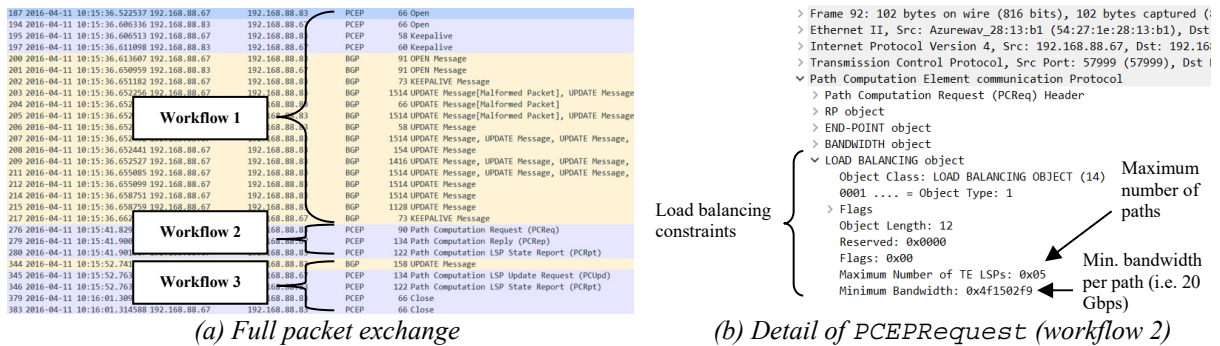


Figure 3. Traces from Wireshark.

4.3 Workflow 3: Reaction to fiber link failure

From the emulated NMS/OSS, the PCC can emulate a fiber link failure to evaluate the reaction of our multilayer PCE algorithm using the “Simulate failure” option (see lower right side in Fig. 2a). We would like to remark that the PCC sends topology updates via BGP-LS and the PCE automatically reacts in case of failure, without the need for an explicit `PCEPRequest` message.

The workflow is as follows (frames 344-346 in Fig. 3a):

1. The PCC issues the failure via a `BGPUpdate` message with the `MP_UNREACH` path attribute corresponding to the failed fiber indicated by the NLRI.
2. The PCE tries to reroute affected lightpaths over the surviving topology. In case some lightpath could not be rerouted, then we try to restore MPLS services looking for a sequence of lightpaths (or set of them) with enough capacity to carry the affected routes.
3. The PCE sends back to the PCC a list of ordered actions in a `PCEPUpdate` (see Fig. 3d) message to be performed, with “delegate” option.
4. The PCC acknowledges with `PCEPReport` messages, again with “delegate” option.

In Fig. 4 we represent the effect of lightpath rerouting upon fiber failure. In our example, a failure in fiber 2-5 (red dashed line) triggers rerouting of the selected lightpath from 0 to 5 (original and rerouted fiber paths are highlighted with thick solid and dashed lines, respectively).

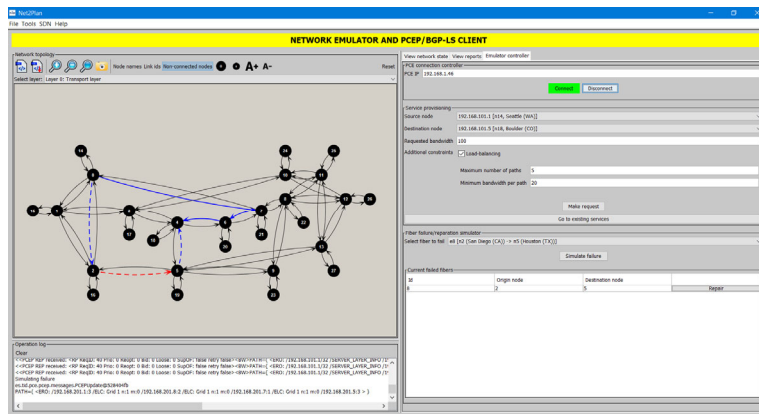


Figure 4. Lightpath rerouting upon fiber failure.

Finally, in our case study we do not consider any “reverting back to normal” mechanism, where lightpaths rerouted during failure states are reverted to their original state upon reparation, in order to avoid traffic outage during lightpath migration [12]. We can actually simulate the reparation of the failed fiber link using the “Simulate reparation” option, which sends a BGP-LS reparation event (`BGPUpdate` with `MP_REACH` attribute), but no further action will be performed by the PCE.

5. CONCLUSIONS AND FURTHER WORK

This work demonstrates an active stateful multilayer PCE with full topological information, based on the open-source Net2Plan tool, able to intercommunicate with the network using PCEP and BGP-LS protocols. Thanks to the usage of standard communication protocols, we should be able to substitute our emulator with real equipment. We also plan to extend our work following the concept of “in-operation” network planning [13]. In other words, we will employ capabilities of our active stateful multilayer PCE to explore i.e. lightpath reoptimization mechanisms.

ACKNOWLEDGEMENTS

This work was partially supported by the FPU fellowship program of the Spanish Ministry of Education, Culture and Sports (ref. no FPU14/04227) and by the Spanish project grants TEC2014-53071-C3-1-P (ONOFRE) and TEC2015-71932-REDT (ELASTIC).

REFERENCES

- [1] J.-P. Fernandez-Palacios, *et al.*: IP and Optical Convergence: Use Cases and Technical Requirements, White Paper, Jan. 2014.
- [2] O. Gerstel, V. Lopez: The Need for SDN in Orchestration of IP over Optical Multi-Vendor Networks, in *Proc. ECOC 2015*, Valencia, Spain, Sep.-Oct. 2015.
- [3] J.-P. Vasseur, J.-L. Le Roux: Path Computation Element (PCE) Communication Protocol (PCEP), RFC 5440, Mar. 2009.
- [4] E. Crabbe, *et al.*: PCEP Extensions for Stateful PCE, draft-ietf-pce-stateful-pce-14, Mar. 2016
- [5] H. Gredler, *et al.*: North-Bound Distribution of Link-State and Traffic Engineering (TE) Information using BGP, RFC 7752, Mar. 2016.
- [6] Java Library of Networking Protocols. [Online] Available: <https://github.com/telefonicaid/netphony-network-protocols> [Last accessed: Apr. 2016]
- [7] Net2Plan – The Open-Source Network Planner. [Online] Available: <http://www.net2plan.com> [Last accessed: Apr. 2016]
- [8] P. Pavon-Marino, J.L. Izquierdo-Zaragoza: Net2Plan: An open-source network planning tool for bridging the gap between academia and industry, *IEEE Network*, vol. 29, no. 5, pp. 90-96, Sep.-Oct. 2015.
- [9] Repository of Net2Plan Plugins for Active Stateful Multilayer PCE, Multilayer Network Emulator and PCEP/BGP-LS Client. [Online] Available: <https://github.com/girtel/n2p-plugins-activePCE> [Last accessed: Apr. 2016]
- [10] T. Bates, *et al.*: Multiprotocol Extensions for BGP-4, RFC 4760, Jan. 2007.
- [11] H. Gredler, *et al.*: BGP Link-State Information Distribution Implementation Report, draft-ietf-idr-ls-distribution-impl-04, May 2015.
- [12] A. Mayoral, *et al.*: Minimizing Resource Protection in IP Over WDM Networks: Multi-layer Shared Backup Router [Invited], *IEEE/OSA JOCN*, vol. 7, no. 3, pp. A440-A446, Mar. 2015.
- [13] L. Velasco, *et al.*: In-Operation Network Planning, *IEEE Com. Mag.*, vol. 52, no. 1, pp. 52-60, Jan. 2014.

Bibliografía

- [1] J.-P. Fernandez-Palacios, F. J. Rios, G. Cauchie, N. Lemonnier, R. Davey *et al.*, “IP and Optical Convergence: Use Cases and Technical Requirements,” Telefónica I+D, AXTEL Mexico, Bouygues Telecom, BT, China Unicom, Colt, Deutsche Telekom, KDDI, Korea Telecom, Orange and Telecom Italia, White Paper, Jan. 2014. [Online]. Available: http://www.tid.es/sites/526e527928a32d6a7400007f/assets/532c7ce328a32d4d710006bd/White_paper_IP_Optical_Convergence__1_.pdf
- [2] O. Gerstel and V. Lopez, “The need for sdn in orchestration of IP over optical multi-vendor networks,” in *Proceedings of the 41st European Conference on Optical Communication (ECOC 2015)*, Sep. 2015. [Online]. Available: <http://dx.doi.org/10.1109/ECOC.2015.7341833>
- [3] J. Vasseur and J. L. Roux, “Path Computation Element (PCE) Communication Protocol (PCEP),” IETF, RFC 5440, Mar. 2009. [Online]. Available: <http://tools.ietf.org/rfc/rfc5440.txt>
- [4] “Net2Plan - The open-source network planner,” [Last accessed: July 2019]. [Online]. Available: <http://www.net2plan.com>
- [5] P. Pavon-Marino and J.-L. Izquierdo-Zaragoza, “Net2Plan: An Open Source Network Planning Tool for Bridging the Gap between Academia and Industry,” *IEEE Network*, vol. 29, no. 5, pp. 90–96, Sep.–Oct. 2015. [Online]. Available: <http://dx.doi.org/10.1109/MNET.2015.7293311>
- [6] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015. [Online]. Available: <http://doi.org/10.1109/JPROC.2014.2371999>
- [7] R. Presuhn, “Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP),” IETF, RFC 3416, Dec. 2002. [Online]. Available: <http://tools.ietf.org/rfc/rfc3416.txt>
- [8] H. Jamjoom, D. Williams, and U. Sharma, “Don’t call them middleboxes, call them middlepipes,” in *Proceedings of the 3rd Workshop on Hot topics in software defined networking*. ACM, 2014, pp. 19–24. [Online]. Available: <http://doi.org/10.1145/2620728.2620760>
- [9] A. Farrel, J.-P. Vasseur, and J. Ash, “A Path Computation Element (PCE)-Based Architecture,” IETF, RFC 4655, Aug. 2006. [Online]. Available: <http://tools.ietf.org/rfc/rfc4655.txt>

- [10] D. King and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations," IETF, RFC 7491, Mar. 2015. [Online]. Available: <http://tools.ietf.org/rfc/rfc7491.txt>
- [11] A. Atlas, T. Nadeau, and D. Ward, "Problem Statement for the Interface to the Routing System," IETF, RFC 7920, Jun. 2016. [Online]. Available: <http://tools.ietf.org/rfc/rfc7920.txt>
- [12] "Netphony - ABNO," [Last accessed: July 2019]. [Online]. Available: <https://github.com/telefonicaid/netphony-abno/>
- [13] V. López, R. Jimenez, O. G. de Dios, L. M. Contreras, and J. F. Palacios, "Open Source Netphony suite: Enabling multi-layer network programmability," in *Proceedings of the 21st International Conference on Optical Network Design and Modeling (ONDM)*, 2017. [Online]. Available: <https://doi.org/10.23919/ONDM.2017.7958538>
- [14] H. Gredler, J. Medved, S. Previdi, A. Farrel, and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP," IETF, RFC 7752, Mar. 2016. [Online]. Available: <http://tools.ietf.org/rfc/rfc7752.txt>
- [15] E. Crabbe, I. Minei, J. Medved, and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE," IETF, RFC 8231, Sep. 2017. [Online]. Available: <http://tools.ietf.org/rfc/rfc8231.txt>
- [16] "SNDlib," [Last accessed: July 2019]. [Online]. Available: <http://sndlib.zib.de>
- [17] "Wireshark packet analyzer," [Last accessed: July 2019]. [Online]. Available: <https://www.wireshark.org/>
- [18] "IETF - Internet Engineering Task Force," [Last accessed: July 2019]. [Online]. Available: <https://www.ietf.org/>
- [19] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, "Multiprotocol Extensions for BGP-4," IETF, RFC 4760, Jan. 2007. [Online]. Available: <http://tools.ietf.org/rfc/rfc4760.txt>
- [20] A. Mayoral, V. Lopez, O. Gerstel, E. Palkopoulou, O. Gonzalez de Dios, and J.-P. Fernandez-Palacios, "Minimizing Resource Protection in IP Over WDM Networks: Multi-layer Shared Backup Router [Invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 3, pp. A440–A446, Mar. 2015. [Online]. Available: <http://dx.doi.org/10.1364/JOCN.7.00A440>
- [21] "ONF Open Transport API," [Last accessed: July 2019]. [Online]. Available: <https://github.com/OpenNetworkingFoundation/TAPI>
- [22] V. López, I. Maor, K. Sethuraman, A. Mayoral, L. Ong *et al.*, "E2E transport API demonstration in hierarchical scenarios," in *Proceedings of the Optical Fiber Communications Conference and Exhibition (OFC)*, Mar. 2017. [Online]. Available: <https://doi.org/10.1364/OFC.2017.Tu3L.4>
- [23] "ONOS: A new carrier-grade SDN network operating system," [Last accessed: July 2019]. [Online]. Available: <https://onosproject.org/>

-
- [24] "The OpenDaylight Platform," [Last accessed: July 2019]. [Online]. Available: <https://www.opendaylight.org/>
- [25] L. Velasco, A. Castro, D. King, O. Gerstel, R. Casellas, and V. Lopez, "In-Operation Network Planning," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 52–60, Jan. 2014. [Online]. Available: <http://dx.doi.org/10.1109/MCOM.2014.6710064>