Uso de técnicas de virtualización para la experimentación en redes ópticas basadas en GMPLS y DWDM

Fermín Galán, Raül Muñoz, Ricardo Martínez

Centre Tecnològic de Telecomunicacions de Catalunya
Parc Mediterrani de la Tecnologia
Av. Canal Olímpic, s/n, 08860 Castelldefels (Barcelona)
Teléfono: 93 645 29 12; Fax: 93 645 29 01
Correo Electrónico: fermin.galan@cttc.es

Resumen

*En la actualidad, la aplicación de técnicas de virtualización (VMware, Xen, UML, etc.) a las tecnologías de la información y las comunicaciones está despertando gran interés. Este tipo de técnicas permiten construir infraestructuras de máquinas y redes virtuales que emulan a máquinas y redes físicas convencionales proporcionando gran transparencia en funcionalidad y rendimiento. El presente artículo describe una aplicación de la virtualización al testbed ADRENALINE (*All-optical Dynamic REliable Network hAndLINg IP/Ethernet Gigabit traffic with QoS*) desarrollado por el Centre Tecnològic de Telecomunicacions de Catalunya (CTTC). En concreto, se describe el uso de la tecnología VNUML (*Virtual Network User Mode Linux*) para proporcionar una extensión virtual del plano de control del testbed, lo que permite aumentar considerablemente las capacidades de ADRENALINE para crear escenarios de red complejos sin incurrir en grandes costes de infraestructura. Se describe la tecnología de virtualización implicada y su utilización concreta para la implementación de la extensión.*

1. Introducción

Las técnicas de virtualización, surgidas en el ámbito de los *mainframes* durante los años 70 para la optimización de los, por aquel entonces, muy escasos recursos informáticos, están experimentando un resurgimiento en la actualizad motivado principalmente por la situación actual del hardware, que es cada vez más potente (en términos de CPU, memoria y espacio de almacenamiento) y a la vez más barato. No en vano, la virtualización está considerada una de las tecnologías estratégicas actuales, con un mayor impacto en los próximos años [1].

Pruebas del creciente interés del mercado por la virtualización son los diversos productos software existentes que hacen uso de este tipo de técnicas empleando distintas aproximaciones (VMware, Xen, etc.), o el hecho de que los principales fabricantes de microprocesadores (Intel y AMD) estén incluyendo funcionalidades de virtualización a nivel hardware en sus productos.

El presente trabajo trata el uso de técnicas de virtualización en el contexto de las redes ópticas inteligentes. Concretamente, se describe su empleo para la extensión de las capacidades del testbed de experimentación ADRENALINE (*All-optical Dynamic REliable Network hAndLINg IP/Ethernet Gigabit traffic with QoS*) [2], desarrollado íntegramente en el laboratorio de I+D de Redes Ópticas del Centre Tecnològic de Telecomunicacions de Catalunya (CTTC).

El testbed ADRENALINE (cuya descripción detallada se proporciona en la sección 2) implementa un plano de control GMPLS, formado por 14 nodos (denominados OCC, *Optical Connection Controller*) los cuales se configuran de forma flexible permitiendo cualquier topología lógica y diferentes escenarios de experimentación. Los OCC son los responsables de la gestión dinámica de los recursos hardware ópticos del nodo, mediante protocolos de señalización (aprovisionamiento de conexiones ópticas) y encaminamiento (diseminación de la topología y recursos de la red óptica), que cada OCC establece con los OCCs vecinos. No obstante, ciertos experimentos (por ejemplo, los relacionados con algoritmos de encaminamiento sobre redes ópticas inteligentes) necesitan de escenarios más grandes y complejos que utilizan un número elevado de nodos y enlaces, especialmente cuando se modela la topología de redes reales como RedIris [3] o NSFNet [4].

Con el fin de expandir las capacidades del testbed, se propone el uso de técnicas de virtualización que permitan la provisión de "OCCs virtuales" como alternativa a la ampliación física tradicional, siempre costosa en recursos y poco escalable. Estos OCCs virtuales se integran de forma homogénea en ADRENALINE, ejecutando los mismos procesos que sus contrapartidas convencionales.

Dentro del amplio espectro de técnicas de virtualización existentes, se ha hecho uso de la

tecnología VNUML (*Virtual Network User Mode Linux*) [5]. La ventaja que proporciona con respecto a otras técnicas de virtualización es que la construcción de escenarios virtuales es más sencilla, ya que se basa en modelos abstractos de red centrados en el usuario y en técnicas de despliegue automáticas, por lo que no se requiere de conocimientos específicos (muchas veces complejos) del sistema de virtualización subyacente.

El resto del artículo se estructura de la siguiente manera. En la sección 2 se realiza una breve introducción al testbed ADRENALINE, considerando aspectos de arquitectura funcional, física y analizando cuales son sus capacidades actuales. A continuación, la sección 3 describe la extensión virtual del testbed, prestando especial interés a los componentes de virtualización concretos utilizados (UML y a VNUML). La implementación concreta se detalla en la sección 4. Finalmente, la sección 5 expone las conclusiones y líneas de trabajo futuro. Un Apéndice con la especificación VNUML completa utilizada para la implementación y las Referencias cierran el artículo.

2. ADRENALINE Testbed

El testbed ADRENALINE [2] implementa una red inteligente de transporte óptica mediante la combinación de dos principales logros tecnológicos. En primer lugar, la tecnología de transporte DWDM (*Dense Wavelength Division Multiplexing*) que permite transportar decenas de canales ópticos en una misma fibra multiplexados en frecuencia, sustentada por dispositivos como los R-OADM (*Reconfigurable Optical Add-Drop Multiplexer*) que permiten insertar y extraer óptimamente canales ópticos de una fibra y los OXC (*Optical Cross Connect*) que permiten conmutar ópticamente canales ópticos. En segundo lugar, la inteligencia del testbed ADRENALINE es proporcionada por el plano de control, basado en la arquitectura ITU-T ASON (*Automatic Switched Optical Network)* y el conjunto de protocolos GMPLS (*Generalized Multi-Protocol Label Switching*).

A continuación, se describe la arquitectura funcional y física así como las actuales funcionalidades en el plano de control, las cuales se pretenden expandir haciendo uso de técnicas de virtualización.

2.1. Arquitectura funcional

Desde un punto de vista funcional, ADRENALINE está compuesto por tres planos: transporte, control y gestión (Fig. 1).

El plano de transporte está basado en equipos de conmutación todo-óptico (es decir la señal que atraviesa un nodo permanece constantemente en el dominio óptico), como son R-OADM y OXC, que se interconectan entre ellos mediante enlaces de fibra de longitud 35 km. Actualmente, el plano de transporte está compuesto por tres nodos ópticos en topología de anillo, si bien se está desarrollando un cuarto nodo cuya finalización está prevista para mediados del 2007. El transporte de datos dentro de la red se realiza mediante tecnología DWDM, multiplexando un máximo de 8 longitudes de onda en tercera ventana (1550 nm), además de una longitud de onda en segunda ventana (1310 nm) para el intercambio de información de control entre los OCCs. Adicionalmente, el plano de transporte también incluye equipos clientes comerciales (broadband testers, etc.) empleados tanto para experimentación acerca del establecimiento de conexiones ópticas de forma dinámica, así como monitorización de parámetros físicos de la señal óptica y generación de alarmas SNMP (*Simple Network Management Protocol*) enviadas y

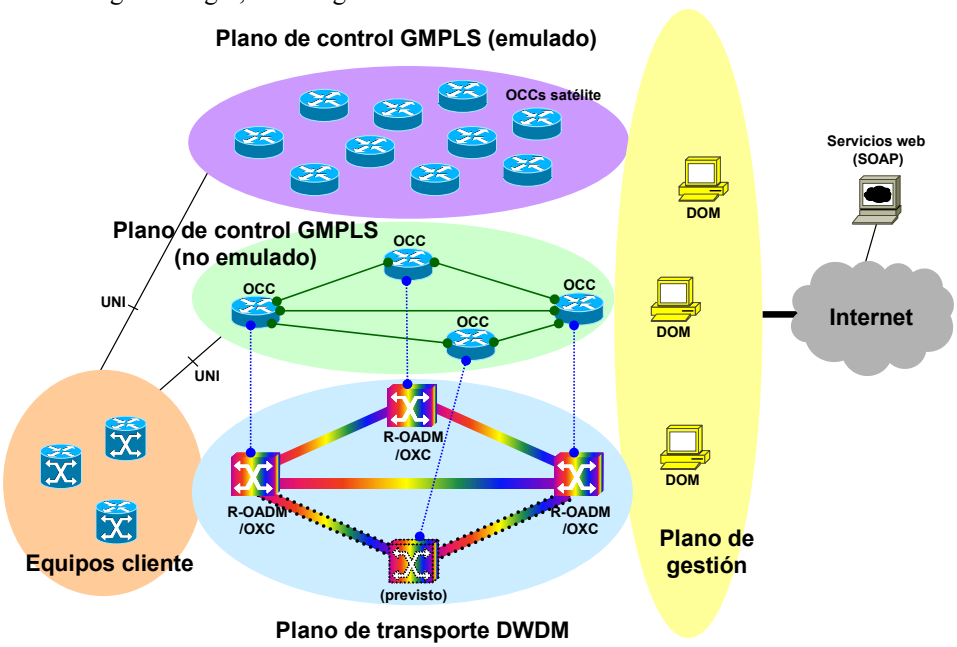

Figura 1. Arquitectura funcional de ADRENALINE

procesadas convenientemente entre los planos de gestión y control.

El plano de gestión de ADRENALINE se implementa de forma distribuida mediante un conjunto de nodos denominados DOM (*Distributed Optical Manager*). Cada nodo DOM implementa interfaces de comunicación con otros DOMs, y realiza funciones de gestión asociadas con los planos de transporte y control (por ejemplo, petición de establecimiento de canales ópticos entre dos puntos de la red, procesamiento de alarmas SNMP de los equipos de monitorización óptica, etc.), con los que implementa sendos interfaces. Adicionalmente, el plano de gestión puede integrarse en una arquitectura de servicios web mediante la cual clientes SOAP (*Simple Object Access Protocol*) pueden solicitar conexiones ópticas de forma dinámica a través de una conexión a Internet.

El plano de control distribuido está compuesto por un conjunto de controladores ópticos u OCC (*Optical Connection Controller*). Está basado en la arquitecta GMPLS, y representa la columna vertebral de ADRENALINE. Dicha arquitectura GMPLS [6] define un conjunto de protocolos, originalmente desarrollados para IP, que proporcionan mecanismos automáticos para la provisión de conexiones ópticas extremo a extremo de forma dinámica. Concretamente, se utiliza OSPF-TE (*Open Shortest Path First – Traffic Engineering*) [7] como protocolo de encaminamiento distribuido para la difusión de la topología y estado de los recursos de la red óptica, y RSVP-TE (*ReSerVation Protocol – Traffic Engineering*) [8] como protocolo de señalización distribuido para el establecimiento, mantenimiento y eliminación de conexiones ópticas. Asimismo, el plano de control es capaz de gestionar mecanismos de protección y/o restauración de conexiones en tiempo real en caso de fallos (por ejemplo, corte de una fibra). Finalmente, el plano de control acepta peticiones de equipos clientes conectados a la red mediante interfaces de control UNI (*User-Network Interface*), efectuadas mediante RSVP-TE. Este tipo de interfaz, también conocido como interfaz máquina-red, permite a equipos clientes (Routers IP, etc.) solicitar de forma dinámica y autónoma canales ópticos de la red, sin ninguna intervención humana.

Existen dos tipos de OCCs. Por una parte, OCCs que disponen de hardware óptico asociado (es decir, R-OADM/OXC), interactuando con él en el instante de establecer conexiones ópticas reales. Por otra parte, OCCs *satélite*, que no disponen de hardware óptico asociado, pero que son funcionalmente equivalentes a los anteriores (emulando el hardware) y resultan necesarios a la hora de crear topologías complejas. Los OCCs satélite configuran en cierto modo un plano de control *emulado*. Los enlaces entre OCCs dentro del plano control emulado pueden hacer uso de un servidor de enlaces (*linkserver*) que permite configurar características de transmisión específicas como son retardo, pérdida de paquetes, ancho de banda, etc.

La implementación de los OCCs está basada en equipos PC (Intel Xeon 3GHz, 1 GB RAM), que ejecutan el sistema operativo Debian GNU/Linux. Cada OCC consiste en uno de dichos PCs, y ejecuta una serie de procesos software que implementan las funcionalidades del plano de control (Fig. 2): controlador OSPF, controlador RSVP, controlador de recursos de enlace (LRM, *Link Resource Manager*), controlador SNMP y, en aquellos OCCs con nodo óptico asociado, un controlador de hardware óptico. Los procesos interactúan entre ellos utilizando interfaces internos y con otras entidades (en otros OCCs o en el plano de transporte y gestión) utilizando interfaces externos. Existe una red de control, paralela a la red de transporte de datos, que interconecta los OCCs entre sí.

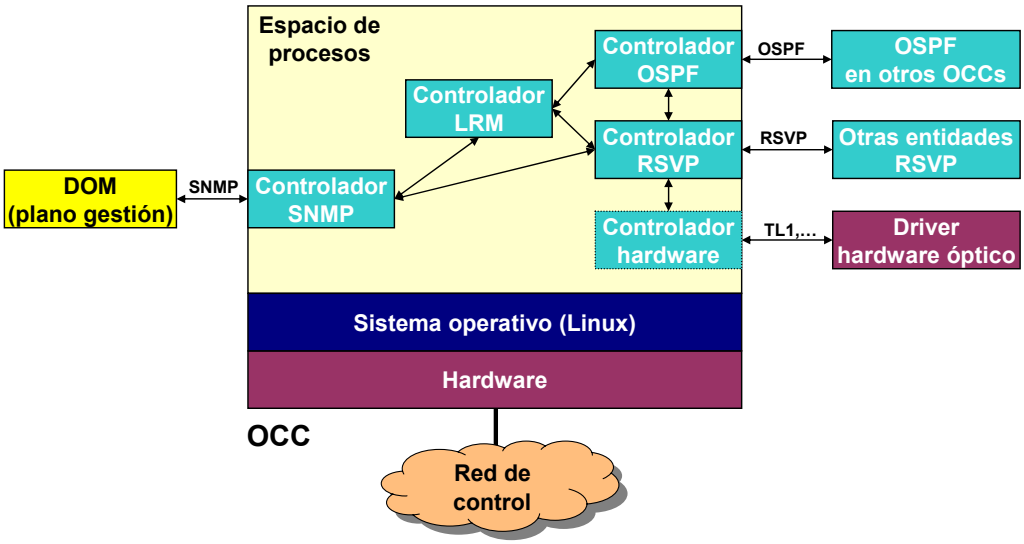

Figura 2. Arquitectura de OCC

Los detalles de los planos de transporte y gestión quedan fuera del alcance de este artículo. Si se desea obtener más información sobre ellos puede consultarse las referencias [9] [10] respectivamente. En lo sucesivo, nos centraremos en los OCCs que implementan el plano de control distribuido.

2.2. Arquitectura física

Una simplificación de la arquitectura física de ADRENALINE (centrada en el plano de control) se muestra en la parte superior de la Fig. 3. Cada OCC dispone de una tarjeta de red GbE (Gigabit Ethernet) mediante la cual se conecta físicamente a una infraestructura de conmutadores (*switches*), con el fin de interconectarse con otros OCCs. Los equipos cliente y los controladores del hardware óptico también utilizan esta infraestructura de conmutadores.

Una de las características principales de ADRENALINE es la reconfigurabilidad del plano de control: distintas topologías (por ejemplo, anillos y mallas) y con distintas características de transmisión (ancho de banda canal de control, retardo, etc.) pueden ser construidas a nivel lógico haciendo uso de la misma arquitectura física, sin necesidad de hacer cambios en la infraestructura (reconexiones de cables, etc.).

Esta reconfigurabilidad se consigue haciendo uso de la tecnología 802.1q [11] para construir redes virtuales sobre Ethernet, de forma que, con las configuraciones adecuadas en los conmutadores y en el sistema operativo de los OCCs, puede conseguirse cualquier topología deseada. Lo único que permanece fijo e invariante en las distintas topologías lógicas es la topologia troncal de transporte basada en el anillo de fibra óptica que conforma el plano de transporte; en los OCCs satélite existe plena flexibilidad. La conexión entre OCCs puede ser directa, o realizarse mediante un túnel GRE (*Generic Routing Encapsulation*) [12], que pasa por el *linkserver*, si se desea emular condiciones de transmisión específicas entre un par de nodos concretos.

Cabe resaltar también que el uso de la tecnología 802.1q permite configurar una red de operación para el testbed, simplificando de esta forma enormemente las tareas de administración. Concretamente, esta red utiliza el identificador 200 (elegido arbitrariamente).

A pesar de la reconfigurabilidad del testbed, las operaciones de configuración que han de ser realizadas en conmutadores, *linkserver* y OCCs resultarían en un enorme consumo de tiempo, si estas debieran ser efectuadas manualmente por el administrador o usuario. Para solucionar este problema, se ha desarrollado la herramienta ADNETCONF (*ADrenaline NETwork CONFigurator*) [13], que automatiza estos procesos de reconfiguración y evita así problemas asociados a

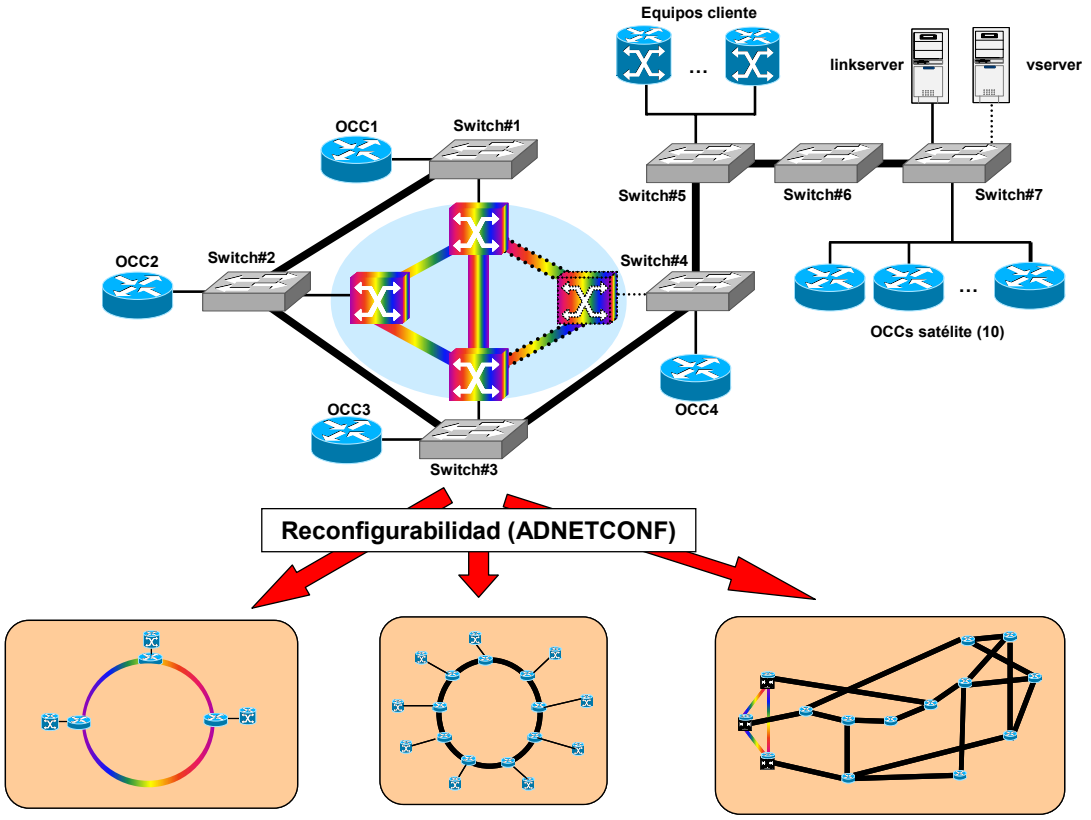

Figura 3. Arquitectura física de ADRENALINE y reconfigurabilidad con ADNETCONF

operaciones manuales. Realmente, la herramienta va más allá y permite configurar mediante una sencilla interfaz gráfica orientada a usuario (*user friendly*) no sólo la topología lógica, sino también la configuración de los procesos a ejecutar en los OCCs, además de integrar funciones de monitorización con el objetivo de comprobar en tiempo real el estado de la red y los procesos del plano de control.

2.3. Capacidades del plano de control

Actualmente, ADRENALINE está formado por 3 OCCs con hardware óptico asociado, si bien (como se ha señalado en la sección 2.1) está previsto en un futuro cercano un cuarto nodo, junto con 10 OCCs satélite. Por tanto, topologías lógicas de hasta 14 OCCs podrán ser implementadas y configuradas haciendo uso de la herramienta ADNETCONF.

Sin embargo, disponer de mayor capacidad resulta de interés por dos motivos. En primer lugar, se necesitan redes complejas con el fin de validar y estresar los mecanismos de encaminamiento y los algoritmos de aprovisionamiento y restauración de conexiones ópticas que se desarrollan en el testbed, y poder de esta forma justificar y discutir la eficiencia y optimización de los mismos. En segundo lugar, para modelar redes basadas en topologías reales, que puedan ser utilizadas para extraer conclusiones en el testbed extrapolables a dichas redes o, simplemente, utilizarlas como modelos de referencia. Por ejemplo, la red académica española RedIris [3] consta de 19 nodos y la red estadounidense NSFNet [4] de 24. Ninguna de ellas podría ser implementada completamente por ADRENALINE dado que sólo dispone de 14 OCCs.

La opción en principio más directa para expandir las capacidades del testbed es ampliar la infraestructura, es decir, añadir nuevos OCCs físicos. Sin embargo, esta solución plantea diversos problemas. En primer lugar, el coste de equipamiento (tanto en PCs como, posiblemente, en infraestructura de red para conectarlos, como por ejemplo nuevos conmutadores). En segundo lugar, el coste en esfuerzo de administración (por ejemplo, instalación de los nuevos sistemas operativos, configuración de infraestructura de red, mantenimiento activo de equipos, etc.), más importante que el anterior. Finalmente, es una solución poco escalable que tan sólo amplía el límite, pero no lo elimina. Si, por ejemplo, se decide ampliar el testbed con 5 OCCs más, únicamente se está solucionando parcialmente el problema ya que se podría construir una topología como la de RedIris, pero no una como la de NSFNet.

Otra opción de ampliación, más versátil y que no adolece de los problemas anteriormente mencionados es el uso de virtualización. Esta solución se describirá en la siguiente sección.

3. Extensión virtual del plano de control

Desde un punto genérico, puede definirse la *virtualización* como una técnica que permite encapsular una unidad de proceso para su ejecución dentro de un entorno controlado en un *equipo anfitrión* que emula el entorno real de la forma más transparente posible. Dependiendo a que nivel se sitúe la virtualización podemos distinguir entre virtualización de procesos (por ejemplo, *jails* de FreeBSD [14]), sistemas operativos (Xen [15], QEMU [16]) o incluso un equipo completo (como VMware [17]).

En el caso de ADRENALINE, el objetivo es proporcionar, de una forma sencilla y con el mínimo coste de equipamiento y gestión, OCCs virtuales que se comporten exactamente igual (o lo más parecido posible desde un punto de vista funcional y de rendimiento) a OCCs satélites implementados con equipos físicos reales y que, adicionalmente, se integren de forma transparente en el sistema de configuración del testbed implementado por la herramienta ADNETCONF.

Las siguientes subsecciones describen la tecnología UML (*User Mode Linux*) y su utilización mediante el recubrimiento VNUML (*Virtual Network User Mode Linux*).

3.1. User Mode Linux

UML [18] es una técnica de virtualización a nivel de sistema operativo consistente en una modificación de las fuentes del núcleo de Linux que permiten su ejecución como proceso de usuario encima del núcleo convencional de Linux (el que ejecuta el equipo anfitrión). Cada uno de los procesos UML tiene asociados sus propios recursos (espacio de memoria, procesos, sistemas de ficheros, dispositivos de red, etc.) y, en definitiva, constituye una máquina virtual dentro de la cual otros procesos se ejecutan.

La funcionalidad del núcleo UML es exactamente la misma que la de un núcleo convencional de Linux (de hecho, el proceso de compilación es análogo), por lo que la transparencia funcional es total. Es decir, la interfaz que ofrece el sistema operativo a las aplicaciones y procesos que se ejecutan en las máquinas virtuales es la misma que en una máquina convencional, por lo que las aplicaciones no “ven” ninguna diferencia entre ambos entornos y se comportan exactamente igual desde el punto de vista funcional (Fig. 4).

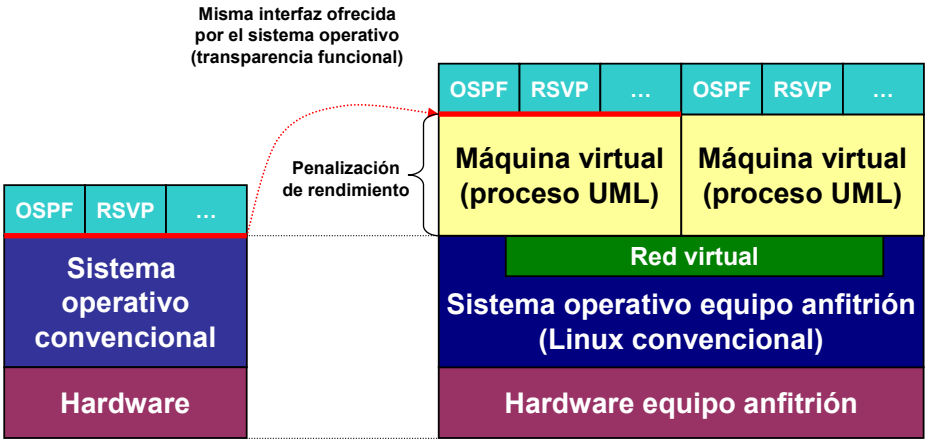

Figura 4. Comparación entre máquinas reales y máquinas virtuales UML

En lo que respecta a la transparencia en rendimiento, es decir, que los procesos ejecutándose en las máquinas virtuales lo hagan con la misma eficiencia que en la máquina convencional equivalente, el uso de virtualización implica siempre una penalización. Si bien, por tanto, la transparencia en rendimiento completa nunca puede alcanzarse, las técnicas de virtualización actuales, y en concreto UML, están muy optimizadas y son muy eficientes, por lo que la penalización podría resultar despreciable. Existen de hecho, resultados experimentales que lo prueban en el contexto de QoS de aplicaciones web [19].

UML no sólo proporciona la manera de crear máquinas virtuales, sino que provee mecanismos para la creación de redes virtual. Estas redes pueden ser utilizadas para interconectar máquinas virtuales entre sí (formando topologías arbitrariamente complejas si se desea) o con equipos externos, haciendo uso de las interfaces físicas del equipo anfitrión. Estas redes virtuales permiten una gran flexibilidad, siendo posible interconexión a nivel 2 o a nivel 3 (en este caso, el equipo anfitrión actuando como *router*) y la integración con redes virtuales 802.1q externas (como las que se usan en ADRENALINE para creación de topologías lógicas).

Una de las principales restricciones de UML es que sólo permite crear máquinas virtuales basadas en Linux. No es posible crear, por ejemplo, máquinas virtuales basadas en MS Windows. Sin embargo, dado que en ADRENALINE los OCCs se implementan en GNU/Linux (como se explica en la sección 2.1) esto no supone ningún problema.

3.2. Virtual Network User Mode Linux

De la descripción de la subsección anterior puede deducirse que UML es una herramienta potente con múltiples posibilidades. Sin embargo, se trata de un software complejo de utilizar si se pretende construir escenarios que incluyan muchas máquinas virtuales y topologías complicadas de red. Además, es necesario tener un buen conocimiento de ciertos detalles relacionados con sistemas operativos GNU/Linux (dispositivos tap, sockets UNIX, *bridges* virtuales, etc.) para construir escenarios "a mano". La motivación con la que VNUML [5] ha sido desarrollado es evitar esta complejidad, de forma que el usuario tenga que concentrarse únicamente en la definición del sistema emulado, es decir, en un conjunto de OCCs virtuales en el caso de ADRENALINE.

VNUML tiene dos componentes. Por una parte, un lenguaje de especificación de escenarios de red, basado en XML (*eXtensible Markup Language*) [20], con su propia sintaxis (especificada dentro del propio modelo del lenguaje a través de validación por DTD –*Document Type Definition*) y semántica (especificada en la documentación que acompaña a la herramienta). Se trata de un lenguaje de alto nivel, centrado en conceptos de redes como son nodo, enlace, interfaz de red, etc., y cuyo objetivo es aislar al usuario de detalles complejos sobre UML o virtualización.

El segundo componente es la herramienta VNUML propiamente dicha, que consiste en un *parser* (interprete) del lenguaje anteriormente descrito. El programa interpreta el escenario descrito en XML e interactúa con el sistema anfitrión y las máquinas virtuales para crear escenarios, eliminarlos o ejecutar secuencias programas de comandos en los mismos.

Así pues, el ciclo de trabajo típico tiene dos etapas como se representa en la Fig. 5. En la primera, el usuario escribe el escenario deseado en un fichero XML en el lenguaje VNUML. En la segunda, el usuario utiliza la herramienta usando el fichero como entrada para crear y gestionar el escenario virtual. Por otra parte, el uso de XML, tecnología ubicua hoy en día para el intercambio de datos entre aplicaciones, permite el desarrollo e integración de terceras aplicaciones de un modo sencillo (por ejemplo, la interfaz gráfica vnumlGUI [21], que permite "dibujar" el escenario en vez de tenerlo que describirlo en un fichero XML).

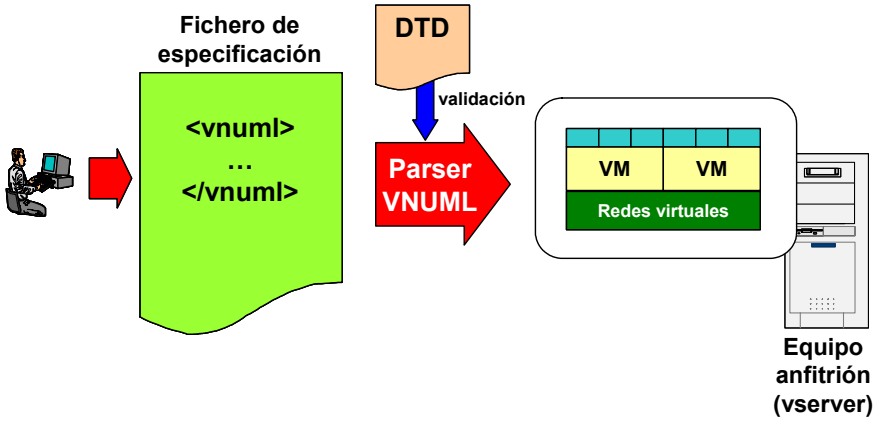

Figura 5. Ciclo de trabajo con VNUML

Es de destacar que VNUML no es una técnica de virtualización propiamente dicha, sino un *front-end* que permite utilizar una de ellas (concretamente, UML, que es el *back-end*) para construir escenarios de red genéricos[1]. En los últimos años ha sido utilizada en multitud de proyectos (relacionados con infraestructuras de *routing* avanzado en IPv6 [22], plataformas de provisión de servicios [23], laboratorios docentes [24][25] o seguridad [26]), lo que prueba su robustez y flexibilidad para su uso también en ADRENALINE. Finalmente, mencionar que la herramienta se desarrolla siguiendo un modelo open-source bajo licencia GPL y puede ser descargada, modificada y utilizada libremente[2].

En la siguiente sección describiremos como se utiliza VNUML para implementar la extensión virtual de ADRENALINE.

4. Implementación

El equipo anfitrión que alberga la extensión virtual del plano de control de ADRENALINE se implementa en un PC, denominando *vserver*, conectado a la infraestructura de conmutadores del testbed y utilizando el mismo hardware que el resto de los OCCs físicos (Intel Xeon 3GHz, 1 GB RAM). El *vserver* ejecuta Debian GNU/Linux y tiene instalada la herramienta VNUML.

Como se adelantaba en la sección 3, el objetivo es proporcionar OCCs satélites virtuales que se integren en ADRENALINE y que puedan intervenir de forma transparente en las topologías diseñadas y construidas con la herramienta ADNETCONF. Estos OCCs virtuales se ejecutarán como máquinas virtuales UML dentro del *vserver* y para gestionarlos se utiliza el escenario VNUML mostrado en el Apéndice I. A continuación analizaremos sus características más destacadas (para una explicación completa y detallada de cada una de las etiquetas ver [27]).

En la sección <global> las etiquetas más significativas son <vm_mgmt> y <filesystem>. La primera, sin entrar en detalles, habilita una red de gestión (1.0.0.0/24) que es usada por VNUML para realizar ciertas acciones de administración (por ejemplo, la ejecución automática de secuencias de comandos programadas con <exec> que se describe un poco más adelante). Esta red sólo existe en el contexto del *vserver* y no interfiere con ninguna de las otras utilizadas en ADRENALINE. Por otra parte, la etiqueta <filesystem> especifica el sistema de ficheros que los OCCs virtuales utilizan (se utilizan técnicas automáticas de *copy-on-write* para que no haya problemas al usar un mismo fichero). Este sistema de ficheros incluye todos los procesos necesarios para el funcionamiento del OCC (los que aparecen en Fig. 2), y es de destacar que son exactamente las mismas (a un nivel de fichero binario ejecutable) que las de los OCCs convencionales.

La etiqueta <net> configura redes virtuales. En un caso genérico, habría tantas etiquetas como redes virtuales fueran necesarias. En este caso, sólo necesitamos una para crear la red "corevl". El atributo "external" indica que esta red ha de ser interconectada a nivel 2 con el exterior, utilizando la tarjeta eth1 del *vserver*. Por consiguiente, la interconexión entre las máquinas virtuales y el conmutador 7 (al que se está conectado el *vserver* físicamente, Fig. 3) se produce a nivel 2, siendo el *vserver* transparente. En definitiva, a todos los efectos, los OCCs virtuales quedan interconectados igual que los OCCs convencionales, por lo que la transparencia a nivel de red es total (por ejemplo, en el momento que ADNETCONF cree las redes virtuales 802.1q interconectando OCCs entre sí en una cierta topología lógica para ADRENALINE).

Finalmente, las etiquetas <vm> definen las distintas máquinas virtuales. En su interior, la etiqueta <if> se utiliza para especificar que se creará una interfaz de

[1] Los términos *front-end* y *back-end*, utilizados comúnmente en ingeniería del software, se refieren a la parte de la aplicación que interactúa con el usuario (*front-end*) y al núcleo que implementa realmente la funcionalidad (*back-end*).

[2] En http://www.dit.upm.es/vnuml, puede descargarse la herramienta, así como encontrar documentación, referencias, ejemplos de uso y soporte a través de listas de correo de usuarios y desarrolladores.

red eth1 (eth0 queda reservara para la red de gestión VNUML antes mencionada) dentro de la máquina virtual, conectando con la red "corevl" (especificada en el atributo "net"). La otra etiqueta relevante es <exec> que sirve para construir dos secuencias de comandos ("up" y "down") cuya utilización se explicará más adelante.

Suponiendo que esta especificación se encuentre escrita en un fichero *adrenaline-virtual.xml*, basta con ejecutar '*vnumlparser.pl –t adrenaline-virtual.xml'* en el *vserver* para que, después de unos instantes, queden arrancados los OCCs virtuales. Es necesario, sin embargo, un paso adicional de configuración para configurar la red de operación del testbed (que necesita ADNETCONF para acceder a todo OCC). VNUML permite la ejecución de secuencias de comandos automáticamente en las máquinas virtuales, agrupándolos con grupos de etiquetas <exec>. En concreto es necesario ejecutar la secuencia "up", que crea la red virtual 802.1q con identificador 200 (ver sección 2.2) y la configura con la IP apropiada (el rango 10.1.1.130 en adelante está reservado para OCCs virtuales), mediante el comando: '*vnumlparser.pl –x up@adrenaline-virtual.xml'*.

El apagado de las máquinas virtuales se realiza de forma análogamente sencilla: *'vnumlparser.pl –d adrenaline-virtual.xml'*.

Notar que todos los OCCs virtuales tienen una configuración homogénea, variando únicamente su nombre y la dirección IP en la red de operación (eth1.200). Corresponde a ADNETCONF (y no a VNUML) gestionar las topologías y las configuraciones de los procesos que se ejecutarán en estos OCCs virtuales en el momento de su uso.

Observar la alta escalabilidad que se obtiene mediante el uso de VNUML. El ejemplo del Apéndice I tan sólo muestra dos OCCs, pero si en un momento dado se requiere utilizar más, basta con repetir la etiqueta <vm> tantas veces como se desee (modificando únicamente la dirección IP y el nombre en las copias). Es decir, añadir nuevas máquinas virtuales es casi tan sencillo como cortar-y-pegar en un fichero de texto. Compárese con el elevado coste que implica añadir nuevos OCCs satélites físicos.

5. Conclusiones y líneas de trabajo futuro

ADRENALINE es un avanzado testbed de experimentación de redes ópticas inteligentes basado en DWDM y GMPLS que puede extraer un gran beneficio de la utilización de técnicas de virtualización. Este artículo ha descrito como mediante el uso de VNUML es posible proporcionar un número ilimitado (en teoría) de OCCs virtuales de una forma muy sencilla, sin entrar en las complejidades que muchas veces implican las técnicas de virtualización y que suelen suponer una "barrera de entrada" para los administradores y técnicos no familiarizados con ellas. Estos OCCs virtuales se integran de forma completamente transparente con los mecanismos de configuración de topologías lógicas del testbed, implementados mediante la herramienta ADNETCONF.

Es de destacar también el paralelismo entre ADNETCONF y VNUML, ya que ambas herramientas siguen una misma filosofía de funcionamiento. En ambos casos, el usuario elabora un modelo del escenario de red deseado y luego ejecuta un procedimiento de despliegue automático, con la diferencia de que con ADNETCONF este despliegue se realiza en equipos físicos de ADRENALINE haciendo uso de las capacidades de las redes IP para construir redes superpuestas, mientras que con VNUML se crea una infraestructura virtual *ad hoc*. No obstante, la tecnología y el diseño de ambas herramientas (modelado de redes en XML, *parser* de lenguaje, etc.) son muy similares. De hecho, parte de esta tecnología de despliegue de redes está actualmente pendiente de patente [28]. Se contempla como línea de trabajo futuro la generalización de modelos de red, posiblemente mediante la utilización de técnicas más potentes de modelado (ontologías, *Unified Modeling Language*, etc.) y la definición de una arquitectura genérica para desarrollar este tipo de modelos y las herramientas que los utilizan.

El principal inconveniente que existe con la utilización de OCCs virtuales es la penalización en su rendimiento. El problema no es tanto la penalización marginal de cada uno de ellos, ya que UML es bastante eficiente en ese sentido (como se explica en la sección 3.1), sino porque el uso de muchos OCCs virtuales pueda sobrecargar el *vserver*. Si bien existen experiencias de uso de VNUML con redes virtuales de más de 50 nodos [29], lo que implicaría un número de OCCs virtuales muy razonable para implementar la mayor parte de las topologías de red de interés deseables, se contempla como línea de trabajo futuro el estudio del impacto de la virtualización. En primer lugar, estableciendo un conjunto de métricas que permitan evaluar la "calidad" del OCC virtual y, posteriormente, calculando dichas métricas en función del número de OCCs virtuales ejecutándose en el *vserver*. En todo caso, localizado el punto crítico, basta con aumentar el número de *vservers*.

Además de la aplicación para ADRENALINE descrita en este artículo, también se está trabajando en el uso de virtualización para mejorar los equipos clientes de experimentación y pruebas, ya que los broadband testers comerciales que se usan actualmente tienen APIs (*Application Programming Interface*) poco flexibles e inadecuadas en algunos casos. La posibilidad de controlar y obtener estadísticas de un conjunto de nodos clientes de la

red óptica de forma centralizada y en tiempo real resulta crítica para implementar un sistema de pruebas de estas características. Una solución es usar el equipo anfitrión (*vserver* o uno adicional) como controlador centralizado y utilizar los mecanismos de comunicación directa entre este y las máquinas virtuales que, a diferencia de otros (como la red), funcionan en tiempo real.

Apéndice I

A continuación (Tabla 1) se muestra la especificación VNUML utilizada para efectuar la extensión virtual del plano de control ADRENALINE, cuya implementación se describe en la sección 4.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia (MEC) mediante el proyecto RESPLANDOR bajo el contrato TEC2006-12910/TCM.

Referencias

[1] *Decálogo tecnológico*. El Universal (edición digital del 4 de Septiembre de 2006). Documento online: http://www.eluniversal.com.mx/articulos/34348.html (última comprobación 31 de Octubre de 2006).

[2] R. Muñoz, C. Pinart, R. Martínez, J. Sorribes, G. Junyent, M. Maier y A. Amrari., "ADRENALINE testbed: Integrating GMPLS, XML and SNMP in transparent DWDM network", IEEE Communications Magazine, vol. 43, no. 6, pp. s40-

Tabla 1. Especificación VNUML para la provisión de OCCs virtuales en ADRENALINE

```
<vnuml>
 <global>
  <version>1.7</version>
  <simulation_name>occs_virtual</simulation_name>
  <ssh_version>2</ssh_version>
  <automac/>
  <netconfig stp="on" />
  <vm_mgmt type="net" network="1.0.0.0" mask="24" offset="0">
    <mgmt_net sock="/var/run/vnuml/Mgmt_net.ctl" hostip="1.0.0.1" />
  </vm_mgmt>
  <vm_defaults>
     <filesystem type="cow">/usr/share/vnuml/filesystems/root_fs_tutorial</filesystem>
     <kernel>/usr/share/vnuml/kernels/linux</kernel>
  </vm_defaults>
 </global>

 <net name="corevl" mode="virtual_bridge" external="eth1" />

 <vm name="occ0-v">
   <mem>128M</mem>
   <if id="1" net="corevl" />
   <exec type="verbatim" seq="up">modprobe 8021q</exec>
   <exec type="verbatim" seq="up">vconfig add eth1 200</exec>
   <exec type="verbatim" seq="up">ifconfig eth1.200 10.1.1.130 netmask 255.255.255.0</exec>
   <exec type="verbatim" seq="up">route add default gw 10.1.1.1</exec>
   <exec type="verbatim" seq="down">vconfig rem eth1.200</exec>
 </vm>

 <vm name="occ1-v">
   <mem>128M</mem>
   <if id="1" net="corevl" />
   <exec type="verbatim" seq="up">modprobe 8021q</exec>
   <exec type="verbatim" seq="up">vconfig add eth1 200</exec>
   <exec type="verbatim" seq="up">ifconfig eth1.200 10.1.1.131 netmask 255.255.255.0</exec>
   <exec type="verbatim" seq="up">route add default gw 10.1.1.1</exec>
   <exec type="verbatim" seq="down">vconfig rem eth1.200</exec>
 </vm>

 <!-- este ejemplo solo muestra dos máquinas virtuales, occ0-v y occ1-v, pero pueden añadirse tantas
      como se desee, usando etiquetas vm análogas a los dos aquí mostradas, cambiado tan solo nombre
      de la máquina virtual y dirección IP en la red eth1.200 -->

</vnuml>
```

s48, Agosto 2005.

[3] Infraestructura de red de RedIris. Disponible en: http://www.rediris.es/red/ (última comprobación 31 de Octubre de 2006).

[4] R. Hülsermann, S. Bodamer, M. Barry, A. Betker. C. Gauger, M. Jäger, M. Höhn y J. Späth, "A set of typical network scenarios for network modelling", in Proc. 5th ITG-Workshop on Photonic Networks, Mayo 2004.

[5] *VNUML*. Disponible online en: http://www.dit.upm.es/vnuml (última comprobación 31 de Octubre de 2006).

[6] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", IETF RFC3945, Octubre 2004.

[7] D. Katz, K. Kompella y D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", IETF RFC 3630, Septiembre 2003.

[8] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, y G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", IETF RFC 3209, Diciembre 2001.

[9] R. Muñoz, P. Vázquez, I. Martínez, M. Requena, C. Pinart, G. Junyent, "Optical Transport Network of the ADRENALINE Testbed: GMPLS Metropolitan All-Optical Tuneable AWG-based R-OADM ring", Proc IEEE/Create-net International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2006), Barcelona, Marzo 2006.

[10] C. Pinart, G. Junyent, "On managing optical services in future control plane enabled IP/WDM networks", IEEE/OSA Journal of Lightwave Technology, Vol. 23, Issue 10, pp. 2868-2876, Octubre 2005.

[11] "802.1q: Virtual LANs", IEEE 802.1Q Working Group, IEEE, 2001.

[12] D. Farinacci, S. Hanks, D. Meyer, P. Traina, "Generic Routing Encapsulation (GRE)", IETF RFC 2784, Marzo 200.

[13] F. Galán, R. Muñoz y R. Martínez, "Demo of ADNETCONF: ADRENALINE's tool for dynamic configuration of GMPLS-based all optical transport networks", TridentCom 2006, Barcelona, Marzo 2006.

[14] P. Hope, "Using Jails in FreeBSD for Fun and Profit", Login: The Magazine of Usenix & Sage, n. 3, vol. 27, Junio 2002.

[15] P. Barham et al, "Xen and the art of virtualization", 19th ACM Symposium on Operating Systems Principles, pp. 164-177, Bolton Landing, NY (EEUU), 2003.

[16] F. Bellard, "QEMU, a Fast and Portable Dynamic Translator", Usenix Annual Technical Conference FREENIX Track, pp. 41-46, Anaheim, CA (EEUU), 2005.

[17] J. Nieh, O. C. Leonard, "Examining VMware", Dr. Dobb's Journal, Agosto 2002.

[18] J. Dike, "User Mode Linux", Proc.5th Annual Linux Showcase & Conf., Oakland, 2001.

[19] Fidel Liberal, "Propuesta de un modelo y una metodología para la gestión de la calidad en los servicios de telecomunicación", Tesis Doctoral, Universidad del País Vasco, Septiembre 2005.

[20] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation, Octubre 2000.

[21] *vnumlGUI, a graphical user interface for VNUML*. Disponible en: http://pagesperso.erasme.org/michel/vnumlgui/ (última comprobación 31 de Octubre de 2006).

[22] D. Fernández, F. Galán y T. de Miguel. "Study and Emulation of IPv6 Internet Exchange (IX) based Addressing Models", IEEE Communications Magazine, vol. 42(1), pp. 105-112, Enero 2004.

[23] F. Galán, E. García, C. Chávarri, M. Gómez y D. Fernández, "Design and Implementation of an IP Multimedia Subsystem (IMS) Emulator Using Virtualization Techniques", 13th HP OpenView University Association (HP-OVUA) Workshop, Niza (Francia), Mayo 2006.

[24] F. Galán, D. Fernández, F. J. Rúiz, O. Walid, y T. de Miguel, "A Virtualization Tool in Computer Network Laboratories", 5th International Conference on Information Technology Based Higher Education and Training (ITHET'04), Estambul (Turquía), Mayo 2004.

[25] D. Fernández, F. J. Ruiz, F. Galán, V. Burillo, y T. de Miguel, "Uso de técnicas de virtualización para mejorar la docencia en laboratorios de redes de comunicaciones", V Jornadas de Ingeniería Telemática (JITEL 2005), Vigo, Septiembre 2005.

[26] F. Galán, D. Fernández, "Use of VNUML in Virtual Honeynets Deployment", IX Reunión Española sobre Criptología y Seguridad de la Información (RECSI), Barcelona, Septiembre 2006.

[27] VNUML Language Reference. Disponible en: http://www.dit.upm.es/vnuml/doc/current/reference/ (última comprobación 31 de Octubre de 2006).

[28] F. Galán, R. Muñoz, "Method for Logical Deployment, Undeployment and Monitoring of a Target IP Network", PCT/EP2006/009960, Octubre 2006 (solicitud de patente).

[29] VNUML OSPF Networking Laboratory. Disponible en: http://www.dit.upm.es/vnuml/doc/examples/ospf_lab/ospf_lab.html (última comprobación 31 de Octubre de 2006).