

Uso de la Tecnología de Virtualización Xen en el Análisis de Protocolos de Comunicación

José M^a Saldaña, Julián Fernández-Navajas, José Ruiz-Mas

Grupo de Tecnologías de las Comunicaciones – Instituto de Investigación en Ingeniería de Aragón

Dpt. IEC. Centro Politécnico Superior Universidad de Zaragoza

Edif. Ada Byron, 50018, Zaragoza

{jsaldana, navajas, jruiz}@unizar.es

Resumen- Los estudios del Espacio Europeo de Educación Superior (EEES) proponen una educación más centrada en el alumno y en el desarrollo de competencias. La posibilidad de experimentar con sistemas complejos, para conocer los procesos e interacciones entre sus elementos, resulta interesante para el aprendizaje. Concretamente, para los alumnos que terminan un Grado o Máster relacionado con Ingeniería Telemática, puede ser útil disponer de un entorno con un número elevado de nodos conectados en red, en el que analizar o probar protocolos y aplicaciones. En este trabajo se presentan las experiencias adquiridas en el uso de virtualización Xen para la realización de estos trabajos, destacando las competencias que puede adquirir el alumno, y explicando casos de uso en los que las características del entorno resultan especialmente adecuadas para el análisis de protocolos. Además se pone de manifiesto que la virtualización puede también ayudar a los alumnos que están interesados en la iniciación a la investigación.

Palabras Clave- virtualización, EEES, competencias, laboratorio virtual

I. INTRODUCCIÓN

Después de varios años de educación universitaria, los estudiantes de Grado y Máster del Espacio Europeo de Educación Superior (EEES) se enfrentan a su último reto: el Proyecto Fin de Carrera o el Trabajo Fin de Máster, donde deben aplicar los conocimientos adquiridos, y demostrar que son capaces de elaborar un trabajo riguroso con una metodología correcta, para finalmente defenderlo ante un tribunal. Tanto el Proyecto Fin de Carrera como el Trabajo Fin de Máster pueden estar orientados al mundo profesional, o bien enfocarse más hacia la iniciación en la actividad investigadora.

Los nuevos estudios proponen una educación más centrada en el alumno y en el desarrollo de competencias. Por ello, es importante que el alumno pueda tener un control mayor sobre un conjunto de sistemas y los procesos que se desarrollan entre ellos. Así, trabajar con una herramienta que permita a cada estudiante disponer de una red con varios nodos puede resultar una buena solución para poner en práctica los conocimientos adquiridos.

En el caso concreto de la Ingeniería Telemática esos sistemas consisten fundamentalmente en nodos, redes, protocolos de comunicación y aplicaciones. El alumno necesitará en muchas ocasiones utilizar varias máquinas conectadas en red. Esto puede plantear problemas de falta de recursos en el caso de que el número requerido de equipos sea muy elevado. Además, la posibilidad de tener una visión

completa de todas las máquinas implicadas en el protocolo y de las aplicaciones usadas, desde un solo lugar, es una opción a tener en cuenta. Un sistema de este tipo, que sea escalable y fácilmente manejable, puede aportar al alumno un entorno de aprendizaje con grandes potencialidades. En los distintos apartados de este artículo haremos un esfuerzo para remarcar los aspectos específicamente relacionados con la Ingeniería Telemática.

Desde hace unos años, en el Grupo de Tecnologías de las Comunicaciones se viene usando la virtualización Xen para poder proporcionar al alumno una red virtual formada por un conjunto de máquinas virtuales que corren dentro de la misma máquina física. Estas máquinas tienen un sistema operativo completo, con la misma pila de protocolos que una máquina real.

Se ha creado un entorno en el que cada máquina dispone de varios interfaces de red (Fig. 1). Uno de ellos se utiliza exclusivamente para el control, es decir, a través de él se envían a la máquina los comandos para ejecutar las aplicaciones correspondientes. El resto de interfaces son usados para crear la red correspondiente al escenario que se está emulando. De este modo, el tráfico de control no interfiere con el de las pruebas, evitando así perder realismo.

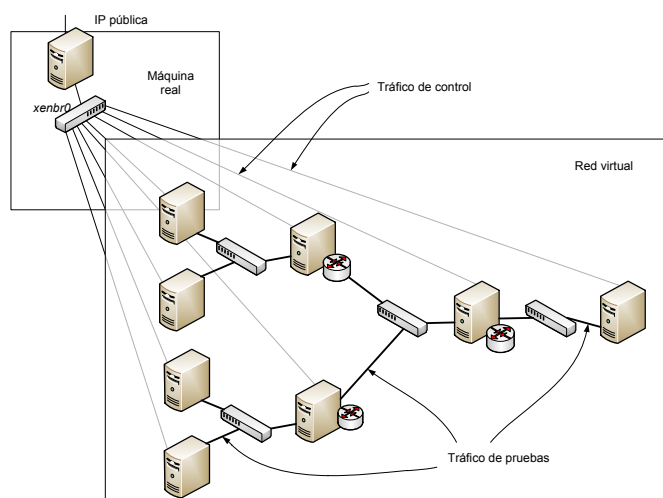


Fig. 1. Máquina real y red de máquinas virtuales

De este modo el alumno puede disponer de un entorno aislado y controlable en el que realizar pruebas de red, requiriendo solamente un PC de gama media, sin especiales características de *hardware*. Se ha utilizado tanto con fines docentes como de investigación y de transferencia de investigación a la empresa.

El resto del artículo se organiza de la siguiente manera: los trabajos relacionados se exponen en la sección II. La sección III expone cuatro casos de uso de la plataforma. La sección IV trata sobre las competencias que se fomentan con este entorno. Después de las conclusiones, se han incluido tres apéndices: el primero habla sobre la virtualización y sus tipos; el segundo expone el modo concreto que utiliza Xen para crear redes, así como las herramientas que hemos usado; el último detalla las características de las máquinas utilizadas.

II. TRABAJOS RELACIONADOS

En la literatura existen muchos *testbed* y emuladores que se utilizan en el ámbito académico. En [1] puede encontrarse un resumen de muchos de los que se han presentado en los últimos años.

En [2] se publicó una comparativa entre soluciones de virtualización. Xen presentó buenos resultados en cuanto a rendimiento, linealidad y aislamiento entre máquinas. Esta es la principal razón por la que se ha elegido esta tecnología de virtualización. En el Apéndice I se puede encontrar una comparativa de los distintos tipos, con las características principales de cada uno.

Con respecto a los aspectos relacionados con la docencia, [3] presentó las experiencias del diseño de un laboratorio de prácticas que permitiese un amplio abanico de configuraciones de red, con objetivos como que los estudiantes pudieran manejar directamente los equipos, y que pudiera haber varios grupos trabajando a la vez. También se buscaba que el laboratorio diera soporte a actividades de investigación.

Y con respecto al uso de virtualización, en [4] se presenta un conjunto de experiencias interesantes del uso de esta tecnología en el ámbito académico, tanto para investigación como para docencia en asignaturas prácticas. Se ha utilizado Xen en algunas de las pruebas, especialmente en las relacionadas con investigación. También se presentan unas encuestas realizadas a profesores y alumnos sobre su experiencia con diferentes plataformas de virtualización. La plataforma más conocida resultó ser VirtualBox, mientras que Xen no lo era tanto, principalmente por su dificultad para usar entornos gráficos. Los principales puntos fuertes de la virtualización que se encontraron fueron su facilidad de instalación y aprendizaje y su disponibilidad pública. Las principales debilidades fueron los requerimientos de *hardware* y el consumo de recursos.

Estos resultados nos parecen interesantes, pues pensamos que Xen puede ser práctico para trabajos más largos, como el Proyecto Fin de Carrera o el Trabajo Fin de Máster, mientras que las tecnologías de virtualización que permiten entornos gráficos pueden ir mejor para sesiones prácticas de muchas asignaturas, ya que requieren menos tiempo de aprendizaje, aunque su potencia pueda ser menor.

Y finalmente, [5] presenta la migración de la infraestructura de un laboratorio, usado para una asignatura sobre seguridad en redes, a un entorno virtual basado en Xen. Se incluyeron más de 40 máquinas virtuales, además de los componentes de red, en una máquina física. Se evitaba que los alumnos se tuvieran que ocupar de la virtualización, haciéndola transparente para ellos. En su trabajo destacan las ventajas de rendimiento de Xen, y sobre todo la reducción de los costes de administración a una sexta parte.

En nuestra plataforma hemos intentado que los estudiantes se familiarizaran con la virtualización Xen, porque pensamos que aprender a manejar esta tecnología puede tener interés en su futuro profesional.

III. CASOS DE USO DE LA PLATAFORMA

En los últimos años hemos empleado esta tecnología para realizar trabajos en diferentes ámbitos, como por ejemplo telefonía y movilidad IP. Pasamos a exponer algunos ejemplos, tanto desde la perspectiva del alumno como docente. En algún caso, el propio alumno que ha realizado el trabajo lo ha publicado posteriormente [6], con lo que se pone de manifiesto la validez de este entorno en la iniciación a la metodología de la investigación, además de su utilización para el aprendizaje en la Universidad.

A. Protocolos de movilidad

Se realizaron pruebas con el protocolo de movilidad MIPv4, realizando algunas medidas de tiempos de traspaso para flujos multimedia. En primer lugar, se usó un simulador para generar los movimientos aleatorios de las máquinas en un escenario, enviando tráfico entre las máquinas mientras se emulaban los movimientos.

Se ha utilizado una versión modificada de *AnSim* [7], aplicación que permite generar los movimientos aleatorios de los nodos, seleccionando parámetros como el tipo de movimiento, el nivel de atenuación o el fichero de escenario. La emulación de los movimientos en las máquinas virtuales se realizó usando *Mackill*, herramienta que filtra a nivel MAC los paquetes que vienen de un nodo no visible en ese momento. Se emuló así el paso de un nodo de una red a otra mientras se desarrolla una sesión multimedia.

MIPv4 utiliza agentes de movilidad, denominados *Home Agent* y *Foreign Agent* (Fig. 2), emulados cada uno en una máquina virtual. Los nodos emulan su movimiento con *Mackill*. Se midieron algunos traspasos, como se ve en la Fig. 3. También se pudo comprobar el funcionamiento del protocolo de movilidad, viendo cómo se realiza el encapsulado en un túnel IP sobre IP cuando el nodo móvil está en una red visitada, y comparando los modos de encapsulado hasta el nodo móvil o hasta el agente externo, y el *reverse* o *triangle tunneling*, pudiendo así ver sus ventajas e inconvenientes. El acceso con *tcpdump* a las trazas completas a nivel *eth* de todas las máquinas permite al alumno ver las direcciones MAC e IP que se utilizan en cada caso. Por ejemplo, se pudo observar cómo el agente local, cuando detecta que el nodo móvil ya no está, envía un mensaje anunciando que la IP del nodo móvil corresponde con su dirección MAC (*proxy ARP*). De este modo, recibe los paquetes destinados al nodo móvil y puede reenviárselos a través de un túnel.

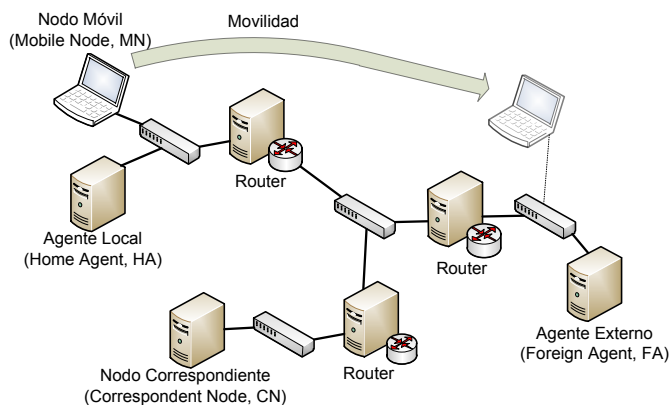


Fig. 2. Escenario de movilidad MIPv4

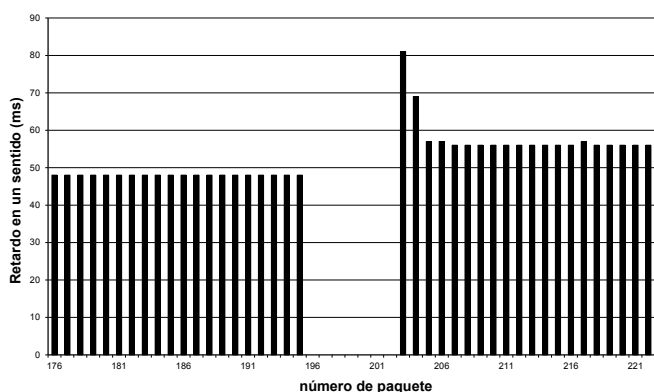


Fig. 3. Medida de un traspaso usando movilidad MIPv4

B. Protocolos de servicios de tiempo real

La plataforma de virtualización también se ha utilizado para la puesta en marcha de una aplicación distribuida de tiempo real, como es el sistema de telefonía IP basado en SIP mostrado en la Fig. 4.

Para emular el centro de datos, se instaló la centralita Asterisk en una máquina virtual y un servidor de bases de datos MySQL en otra. Se emularon cuatro sucursales que accedían a los servicios telefónicos de la centralita a través de un *proxy* SIP instalado en cada oficina. El *proxy* SIP implementa un sistema de control de admisión, y lleva para ello cuenta del número de llamadas que hay establecidas en cada momento. Ese número lo almacena en la base de datos.

En este caso, el escenario con máquinas virtuales fue muy útil para desarrollar el programa que se ejecuta en cada *proxy*. De modo centralizado se podían gestionar los teléfonos, la centralita y los *proxy* para así realizar pruebas de su funcionamiento y conseguir que implementasen un control de admisión con parámetros variables ubicados en la base de datos. Así, por ejemplo, se pudo configurar el *proxy* SIP para que generase un mensaje “302 moved temporarily” cuando no podía aceptar una llamada, y esto provocase que la centralita buscara en su plan de numeración otra sucursal a la que acudir para establecerla (Fig. 5). Una vez más, la facilidad para ver el funcionamiento del protocolo fue de gran ayuda para la realización del trabajo y el correcto funcionamiento del sistema.

Este escenario requirió el uso de 19 máquinas (Fig. 4): cuatro para cada oficina, y otras tres para el centro de datos.

Lógicamente, no se habría podido utilizar un escenario tan extenso con máquinas reales.

C. Seguridad en un sistema de telefonía

En un Proyecto Fin de Carrera se estudiaron soluciones de seguridad y alta disponibilidad para el sistema de telefonía presentado en el apartado anterior. La seguridad se ha estudiado a tres niveles: a nivel de sistema, garantizando la confidencialidad y la autenticidad de los usuarios; a nivel de red, buscando la integridad de los elementos de la red de telefonía; y a nivel de comunicación, garantizando la confidencialidad e integridad de los datos transmitidos. Se llevaron a cabo diversas acciones, como el cifrado de tráfico y el uso de *firewall*. Algunos de los protocolos de seguridad probados fueron *SIPSecure*, TLS, VPNs creadas con IPsec y SRTP. Además se utilizaron herramientas de ataque como *SIPCrack*.

También se pudo probar el funcionamiento de los servidores STUN para solucionar los problemas de NAT, que afectan de modo especial a SIP cuando las direcciones IP van incluidas dentro del cuerpo del mensaje, y el *firewall* no las cambia por hacer NAT.

La alta disponibilidad se estudió desde dos puntos de vista: alta disponibilidad para servicios, que garantiza un funcionamiento continuo del servicio frente a posibles fallos; y alta disponibilidad para datos, garantizando la accesibilidad de éstos frente a fallos en los sistemas de almacenamiento que los proporcionan. Se probaron diversas técnicas, como la monitorización de recursos desde diversos puntos y la redundancia de archivos informáticos y de la información almacenada en bases de datos.

Se utilizaron mensajes *keepalive* para informar de que un servidor sigue en funcionamiento, y también se programaron copias periódicas de la información más sensible de la base de datos de telefonía y del plan de numeración. En la Fig. 6 puede verse la propuesta final del esquema para añadir seguridad y alta disponibilidad al sistema.

D. Calidad de Servicio en servicios de tiempo real

Como último caso de uso explicaremos la aplicación de la plataforma de virtualización para el análisis de los efectos de la multiplexión RTP en la Calidad de Servicio (*Quality of Service*, QoS) para servicios de voz.

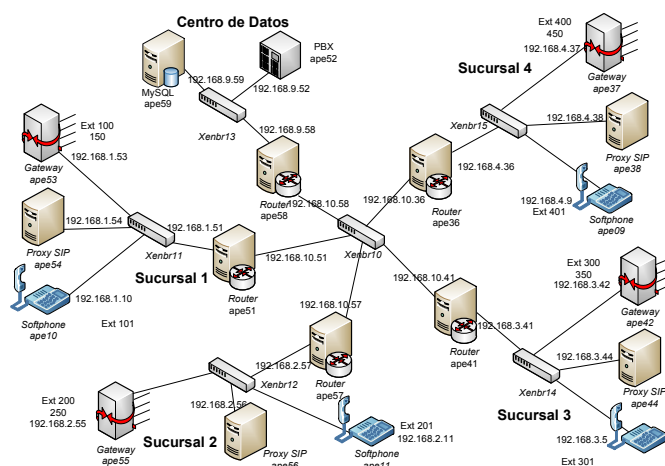


Fig. 4. Escenario de telefonía IP con Asterisk

La multiplexión RTP consiste en unir flujos de datos que tienen el mismo origen y destino, formando paquetes más grandes que comparten una cabecera común, aumentando así la eficiencia de la comunicación, y ahorrando por tanto ancho de banda.

Se seleccionaron dos protocolos de multiplexión [8] y [9], y se compararon entre sí, y también con el uso simple de RTP. Para ello se utilizó un escenario sencillo, en el que una máquina generaba el tráfico, otra implementaba las diferentes políticas del *buffer* del *router*, y otra recogía los paquetes. Posteriormente se realizaba un análisis *offline* de los paquetes recibidos, añadiendo también nuevos retardos y el efecto del *buffer* de *dejitter* (Fig. 7).

En la máquina origen se utilizó D-ITG [10] para generar los flujos de tráfico de voz y un tráfico de fondo. En la máquina *router* se usó *traffic control* (*tc*) para implementar el *buffer*. Se pudieron obtener resultados comparativos entre las tres opciones. Hay que tener en cuenta que al multiplexar se añade un retardo, lo que perjudica a la calidad, pero por otro

lado pueden reducirse las pérdidas de paquetes al disminuir el tráfico total, cosa que mejora la calidad. Así que existe un compromiso, y se deben encontrar las condiciones de tráfico de fondo para las que la multiplexión puede suponer una mejora (Fig. 8). La calidad percibida se midió en términos de Factor R, magnitud definida por la ITU [11].

IV. AYUDA EN LA ADQUISICIÓN DE COMPETENCIAS

La plataforma se empezó a utilizar para Proyectos Fin de Carrera, y posteriormente en el desarrollo de Trabajos Fin de Máster, en el ámbito del Máster Interuniversitario TICRM (Tecnologías de la Información y Comunicaciones en Redes Móviles). Los trabajos realizados han estado orientados tanto a la investigación como a la transferencia a la empresa.

Con el tiempo, se han ido realizando trabajos con escenarios más complejos, que han permitido mejorar el funcionamiento de la plataforma, identificando problemas, y buscando en cada caso la solución más adecuada.

Pasaremos ahora a exponer las competencias que consideramos que se pueden reforzar con el uso de esta plataforma, haciendo referencia a los casos de uso expuestos, que sirven como ejemplos prácticos de lo que han aprendido los alumnos. Las competencias se han extraído de [12].

A. Orden, claridad y precisión

La metodología a seguir en el entorno exige al alumno ser especialmente cuidadoso y metódico para manejar todas las máquinas incluidas en el sistema. El acceso a las máquinas se realiza mediante sesiones remotas, por lo que el alumno debe buscar métodos para identificar claramente con cuáles está trabajando, y entre qué nodos se están enviando los flujos de tráfico.

En el ámbito de la Ingeniería Telemática, un problema frecuente a la hora de medir retardos de red es el de la sincronización entre las máquinas. Este entorno puede ayudar al alumno a ser consciente de este problema y de cómo resolverlo. De hecho, en algún caso, a causa de la desincronización, se pueden obtener resultados ilógicos, como por ejemplo que un paquete se reciba en un momento anterior al que se envió. La precisión alcanzada está en torno al milisegundo, ya que las redes virtuales creadas dentro del ordenador funcionan a velocidad de procesador y resultan mucho más rápidas que los equipos de red habituales.

Al no corresponderse los retardos de red con los que existen en entornos reales, el alumno ha tenido que realizar unas simulaciones previas, seguidas de una fase de envío de tráfico en la plataforma de pruebas, y finalmente un tratamiento *offline* de los resultados obtenidos (Fig. 9). Este

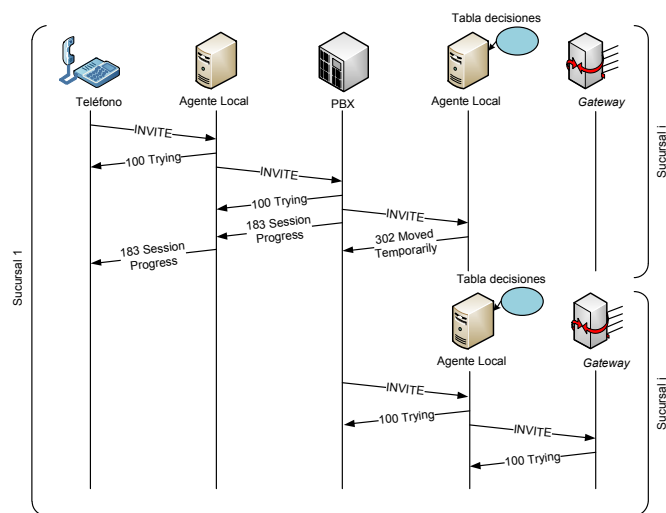


Fig. 5. Esquema del protocolo para el escenario de telefonía IP

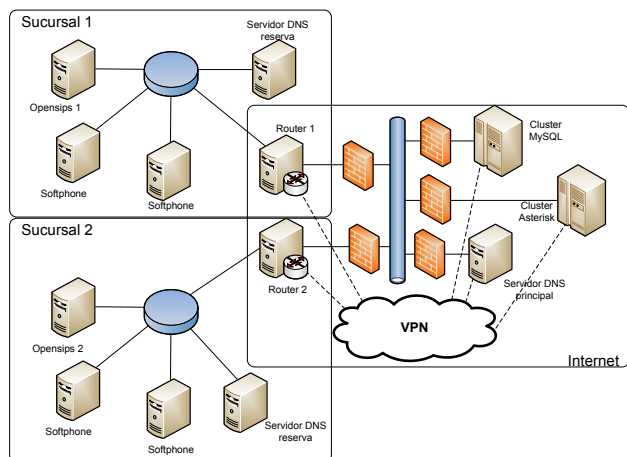


Fig. 6. Esquema del sistema con seguridad y alta disponibilidad incorporadas

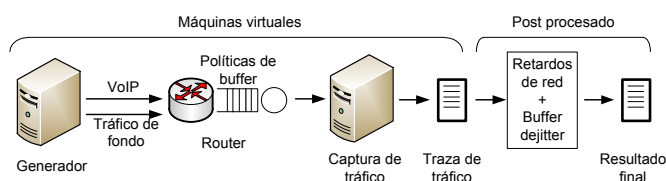


Fig. 7. Esquema de las pruebas de tráfico RTP y multiplexado

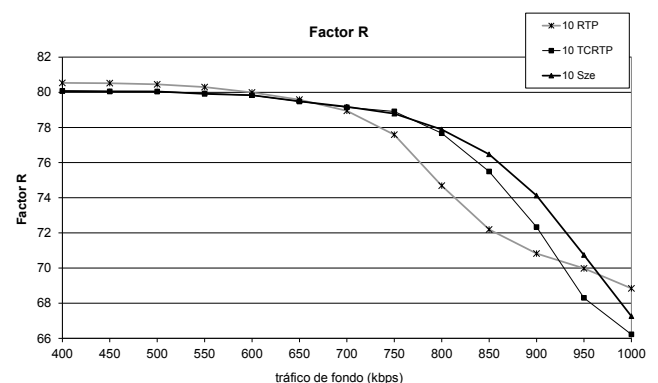


Fig. 8. Comparativa entre métodos de multiplexión

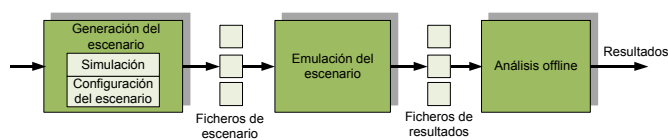


Fig. 9. Esquema de uso de la plataforma

proceso ayuda al alumno a identificar claramente qué retardos se corresponden a la transmisión por la red, o con el procesamiento, etc., cuáles se pueden medir en la plataforma y cuáles deben ser añadidos posteriormente. La integración de distintos entornos de pruebas para obtener un resultado final es una experiencia que requiere orden, claridad y precisión.

B. Pensamiento analítico y pericia

Es muy frecuente que un profesional se enfrente a situaciones en las que una aplicación no funciona correctamente. Para solucionar el problema se deben hacer diferentes pruebas, buscando en qué nivel se está dando el fallo, empezando por lo más básico, para llegar a los niveles más altos. Un error puede deberse a un problema de nivel de enlace, o de *routing*, o estar causado por el NAT, o los puertos usados en la aplicación. En el escenario el alumno puede ir haciendo diferentes pruebas para identificar el problema, ya que tiene acceso a los protocolos de todos los niveles.

Por otro lado, en ocasiones habrá más de un camino para llegar a la misma información. Nuestra versión de Xen crea en la máquina principal un interfaz de red que se corresponde con cada interfaz de cada máquina virtual. Esto puede resultar muy útil para que el alumno comprenda y analice el comportamiento de los protocolos, ya que puede ver y capturar desde un único lugar los paquetes generados por varias máquinas. Como se ha visto en la sección anterior, se pueden obtener esquemas muy completos de los protocolos. Para más detalles se puede consultar el Apéndice II.

C. Pericia en el uso de herramientas

El hecho de usar un número de máquinas virtuales dentro de la misma máquina física imposibilita en la práctica la utilización de entornos gráficos para interactuar con las máquinas. Estos entornos requieren muchos recursos, y resulta imposible que una sola máquina ejecute un gran número de ellos. Por eso el alumno debe gestionar todo el sistema desde interfaz de comandos, que en muchos casos tiene una mayor potencia que un entorno gráfico, y que se utiliza con frecuencia tanto en el mundo de la empresa como en el de la investigación.

Muy relacionado con lo anterior está la capacidad de crear *scripts* que automatizan tareas periódicas. En Linux existen lenguajes de *script* muy potentes, con los que el alumno se puede familiarizar fácilmente.

D. Iniciativa y creatividad

El entorno virtual permite el uso de las mismas aplicaciones que corren en una máquina real. Pero en ocasiones aparecen limitaciones que obligan a emular ciertos comportamientos. Por eso el alumno debe usar su creatividad para encontrar el modo de reproducir o emular en el entorno virtual las situaciones que se dan en un entorno real.

En el caso explicado en el apartado III A, la movilidad IP se emuló mediante la conexión de una máquina por dos

interfaces a dos redes diferentes, haciendo que sólo uno de ellos estuviese activo en cada momento. Al cambiar el interfaz, la máquina asumía que había cambiado de red, y el protocolo correspondiente actuaba. En otros casos (apartados III B y D), al no disponer las máquinas virtuales de tarjeta de sonido, los flujos RTP se tuvieron que emular con un generador de tráfico que corría en paralelo a la aplicación de telefonía.

Otro ejemplo es el modo de emular el ancho de banda del enlace. Al estar conectadas las máquinas con *bridge* virtuales que funcionan a velocidad de procesador, se ha debido recurrir a la herramienta *tc* de Linux, para introducir esas limitaciones y crear un escenario más realista.

E. Búsqueda de información

La necesidad de usar aplicaciones por interfaz de comandos ha llevado a usar programas no comerciales desarrollados en grupos de investigación, que en algunos casos están poco documentados. Para ponerlas en marcha, el alumno ha tenido que recurrir frecuentemente a búsqueda en foros, o incluso a escribir a los propios autores de los programas.

F. Trabajo en equipo y cooperación

Una competencia que consideramos muy importante es la capacidad de trabajar en grupo. En ocasiones varios alumnos han compartido la misma máquina física, creando redes diferentes y separadas con nodos distintos.

En estos casos, los alumnos han podido cooperar de diferentes modos. Por ejemplo, si se necesitaba una máquina virtual que corriese un servidor de bases de datos, un alumno se encargaba de instalarla y de elaborar un pequeño manual con las experiencias y particularidades de la instalación en una máquina virtual. El resto de alumnos se podían beneficiar de este trabajo, centrándose más en lo que a cada uno le correspondía.

La facilidad para copiar máquinas virtuales completas permite que si un alumno ha configurado una aplicación de una determinada manera, pueda proporcionar a otro una copia de esa máquina funcionando correctamente. En muchas ocasiones se invierte mucho tiempo en hacer funcionar una aplicación como se desea, y esta experiencia puede ser compartida.

En una de las fases de trabajo, se dio el caso de que un alumno estaba realizando un trabajo con el sistema de telefonía (caso III B), mientras que otro estaba realizando pruebas con diferentes protocolos de seguridad (caso III C) para ese mismo sistema. Lógicamente, el trabajo de uno puede ser útil para el otro, pues pueden resolver problemas juntos o identificar problemas de otro.

Al trabajar simultáneamente, los alumnos deben coordinarse para realizar pruebas, pues en el momento de las medidas sólo uno de ellos puede estar usando la máquina, para evitar falsearlas, al compartir todas las máquinas los mismos recursos de *hardware*.

G. Otras ventajas para la organización de la docencia

En este apartado trataremos de exponer las ventajas que esta solución puede aportar como ayuda a la docencia de las asignaturas de Ingeniería Telemática.

En primer lugar, queremos destacar un hecho: una máquina virtual se almacena en un fichero del sistema de archivos de la máquina *host*. Esa es una de las grandes

ventajas de la virtualización, que la ha hecho tan popular en el ámbito empresarial: las copias de seguridad se pueden hacer fácilmente, la máquina virtual se puede cambiar de servidor físico sin necesidad de pararla, etc. Esta tecnología aporta una gran flexibilidad y mejora notablemente la disponibilidad en caso de fallo del *hardware*.

Todas estas ventajas también se pueden aplicar en nuestro ámbito. Por ejemplo, un alumno puede replicar una máquina en muy poco tiempo. Basta con duplicar el fichero correspondiente, arrancar la nueva máquina y cambiarle los parámetros necesarios: nombre, dirección IP, etc. Por otro lado, si alguna máquina queda inservible por un mal manejo o configuración, basta con volver a copiarla.

Como caso particular, en la docencia de asignaturas relacionadas con la seguridad, se pueden realizar ataques a una máquina sin temor a cambiar su configuración o dejarla inservible. Basta con volver a copiar el fichero para dejarla en el mismo estado que al principio.

También puede resultar interesante la virtualización en las primeras etapas del desarrollo de una aplicación distribuida (caso III B). Se puede instalar en varias máquinas virtuales conectadas en red de manera conveniente, para así realizar pruebas y mejorar el programa, que luego se puede trasladar fácilmente a máquinas reales.

Y por supuesto, una ventaja está en el ahorro de costes y de tiempo de gestión. Si para una asignatura se requiere un determinado *software*, no hace falta instalarlo en todas las máquinas de un laboratorio. Basta con instalarlo en una máquina virtual y luego replicarla.

V. CONCLUSIONES

El presente trabajo ha mostrado las ventajas que la virtualización Xen puede aportar en el desarrollo de trabajos finales de Grado o Máster en el marco del EEES. También se ha hecho hincapié en las competencias que puede favorecer. Algunas de ellas son el orden y precisión, el pensamiento analítico, la iniciativa, la búsqueda de información y el trabajo en grupo.

Se han expuesto casos de uso con diferentes escenarios, mostrando a través de esos ejemplos la utilidad del sistema de virtualización para este tipo de trabajos. También se han destacado las ventajas específicas para Ingeniería Telemática, poniendo de manifiesto que la virtualización puede aportar otras mejoras a la docencia, como son el ahorro de costes o las facilidades para la gestión en las asignaturas prácticas, así como la capacidad de ofrecer al alumno un número elevado de nodos en los que poder probar protocolos y aplicaciones.

Finalmente, se puede destacar la validez de la plataforma presentada para ayudar a los alumnos interesados en la iniciación a la investigación.

APÉNDICE I. VIRTUALIZACIÓN

En la actualidad, se habla de virtualización con significados muy distintos. Por ejemplo, se puede hablar de “virtualización de la educación” con el significado de la aplicación de las TIC (Tecnologías de la Información y la Comunicación) a la educación. También se puede hablar de “clases virtuales” en oposición a las sesiones presenciales. Para evitar confusiones, queremos dejar claro que en el presente trabajo, el término “virtualización” es estrictamente informático: la creación de una versión virtual de una máquina, que se ejecuta sobre el *hardware* real de otra.

Existe una capa de abstracción que permite a múltiples máquinas virtuales con sistemas operativos distintos ejecutarse en la misma máquina física. Cada máquina virtual tiene su propio *hardware* virtual.

El *software* de virtualización emula un *hardware* suficiente para permitir correr de forma aislada a un sistema operativo cliente. Lógicamente, ese sistema operativo debe estar diseñado para la misma CPU. Muchas máquinas virtuales pueden correr al mismo tiempo en una máquina física.

Por eso, es necesaria una plataforma de virtualización que permita ejecutar un buen número de máquinas virtuales sin penalizar el rendimiento ni necesitar muchos recursos. Aunque no existe un consenso total, algunos autores [13] consideran cuatro tipos fundamentales de virtualización. Pasaremos ahora a verlos.

A. Emulación del Hardware

Se crea una máquina virtual en el sistema operativo de la máquina física, que emula el *hardware* deseado (Fig. 10). Tiende a ser muy lento, porque cada instrucción debe ser simulada en el *hardware* real. Tiene la ventaja de que se puede ejecutar un sistema operativo sin modificar en cualquier procesador. Algunos ejemplos son Bochs y QEMU.

B. Virtualización completa

Se denomina también nativa. Utiliza una máquina virtual que media entre los sistemas operativos clientes y el *hardware* nativo (Fig. 11). No es necesario modificar el sistema operativo cliente, lo que constituye su principal ventaja. Un requisito es que el sistema operativo del cliente debe estar preparado para el *hardware* físico.

Algunas instrucciones protegidas deben ser tratadas de un modo especial por el *hypervisor* o *VMM* (Monitor de máquinas virtuales), porque el *hardware* no es manejado directamente por el sistema operativo, sino por el *hypervisor*, que debe estar constantemente al tanto de estas instrucciones, para tratarlas de una manera especial y permitir su correcta ejecución.

Es un método más rápido que la emulación del *hardware*, pero no consigue mucha eficiencia a causa del *hypervisor*. Ejemplos de este tipo de virtualización son VMware o Virtual Box.

C. Paravirtualización

Es muy semejante al caso anterior, pero integra en el sistema operativo cliente las instrucciones necesarias para evitar que ninguna instrucción deba ser tratada de modo especial (Fig. 12).

Requiere que los sistemas operativos clientes estén compilados para correr en ese entorno de virtualización. Esto supone una desventaja, pero consigue una velocidad similar a la que se daría en un sistema no virtualizado. Por supuesto, varios sistemas operativos diferentes pueden correr a la vez. Ejemplos de este tipo son Xen y UML.

D. Virtualización a nivel de Sistema Operativo

Soporta un solo sistema operativo, y aísla unos grupos de procesos de otros, dándoles acceso solamente a un subconjunto de los recursos. El sistema operativo oculta los grupos de procesos entre sí (Fig. 13). La gran ventaja es que las prestaciones son las mismas que se obtendrían de forma nativa. Dos ejemplos son Linux-VServer y OpenVZ.

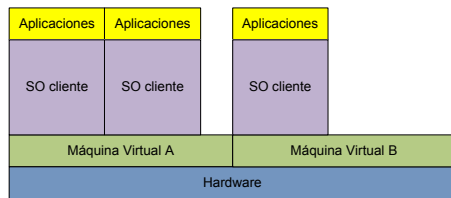


Fig. 10. Emulación del *Hardware*

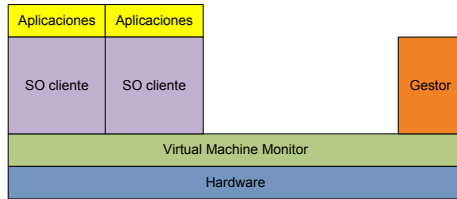


Fig. 11. Virtualización completa

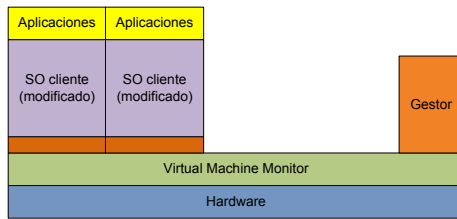


Fig. 12. Paravirtualización

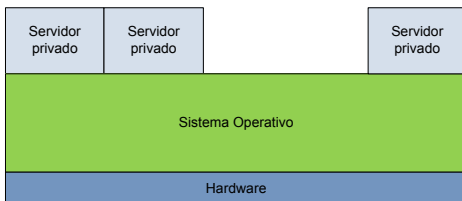


Fig. 13. Virtualización a nivel de Sistema Operativo

E. Solución elegida

La razón principal para optar por la paravirtualización ha sido el interés en que las máquinas virtuales sean rápidas y consuman sólo los recursos necesarios [2]. De esta forma podremos disponer de más máquinas virtuales en una sola máquina física. Nos interesa que, para no falsear las pruebas, funcionen a la misma velocidad que en modo nativo. El problema es la no conveniencia de utilizar entornos gráficos. Esto puede ser un inconveniente serio para las sesiones prácticas de una asignatura, pues el tiempo es limitado y quizá el alumno no pueda hacerse del todo con el entorno, pero tiene mayor potencia para trabajos de larga duración, que son los tratados en el presente trabajo.

Un requisito de esta solución es que las máquinas deben estar preparadas para correr sobre ese *hardware* virtual: algo comparable a compilar el *kernel* para una nueva arquitectura. Eso no supone ningún problema en el caso de máquinas Linux.

APÉNDICE II: MODO DE EMULAR REDES EN XEN

En esta sección explicaremos el método usado para crear redes con máquinas virtuales Xen, así como las herramientas que se han usado para emular las condiciones de red del escenario correspondiente.

Como ya hemos explicado en la introducción, se ha separado la red de control de la de medidas. Esto se ha logrado gracias a que Xen permite crear varias tarjetas de red en cada máquina. La primera máquina que inicia Xen se

llama *dom0*, y es la que arranca todas las máquinas virtuales. Por cada interfaz de cada máquina virtual existe otro en *dom0* (denominado *vifA.B*) que está conectado a él (Fig. 14).

El comando de Linux utilizado para unir interfaces es *brctl*, que permite crear *bridge* que unen las diferentes tarjetas de red. Para crear la red de control, el interfaz *eth0* de cada máquina virtual se ha unido a un *bridge* denominado *xenbr0*, que es accesible desde el dominio principal. Para crear las diferentes topologías de red, basta con crear otros *bridge* y unir a ellos los interfaces correspondientes.

Si se desea crear redes independientes para distintos alumnos, basta con unir sus máquinas virtuales a través de *bridge* que no tengan ninguna máquina en común. Si se quiere crear un nodo que haga de *router*, se pueden conectar sus interfaces a *bridge* diferentes.

El entorno incorpora también emulación del enlace, gracias a la herramienta Linux *traffic control (tc)*, que permite establecer para un interfaz de red varios parámetros, como el ancho de banda máximo, el tiempo máximo en el *buffer*, el tamaño máximo de ráfaga, etc. También se utiliza emulación de red mediante *netem* [14], que permite añadir retardos con diferentes distribuciones estadísticas, así como pérdidas de paquetes, paquetes corruptos, paquetes que llegan fuera de orden, etc.

Los generadores de tráfico que se han usado son JTG [15] y D-ITG [11], que permiten enviar ráfagas con diferentes estadísticas. Ambos disponen de aplicaciones para calcular los resultados en términos de retardo, pérdidas de paquetes, *jitter*, etc.

Finalmente, un asunto a tener en cuenta es que algunos de estos programas requieren tener permisos de *root* en la máquina que se usa, como por ejemplo *xm*, que es el comando utilizado para arrancar y parar máquinas virtuales, o *brctl*, que se usa para crear los *bridge*. Por ello, ha sido necesario definir unos comandos que se pueden utilizar con *sudo* para cada una de las cuentas de usuario creadas. En las máquinas virtuales se ha optado por dar permisos de *root* a los usuarios, puesto que en caso de que dejen de funcionar, se pueden recuperar con facilidad.

APÉNDICE III: MÁQUINAS UTILIZADAS

La máquina utilizada en primer lugar tiene el Sistema Operativo CentOS 5, una distribución Linux derivada de Red Hat Enterprise Linux 5. La versión del núcleo es la 2.6.18-8.1.15. Dispone de un procesador Core 2 Duo a 2.40 Ghz, 2Mb de Cache nivel 2, de 4Gb de RAM, y un disco duro SATA2 de 320 Gb. Las máquinas virtuales tienen instalado también el Sistema Operativo CentOS 5. Los ficheros imagen ocupan 4 Gb por máquina virtual. La versión de Xen instalada es la 3.03-25.0.4.

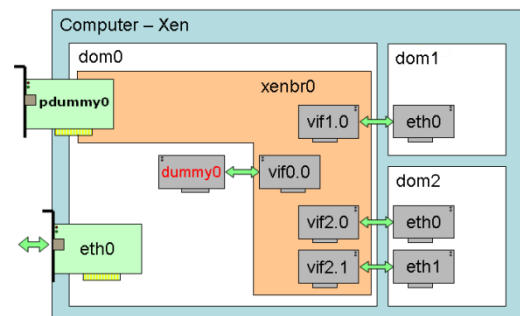


Fig. 14. Esquema de los interfaces de red en Xen

A esta máquina se ha añadido recientemente otra con mejores características: El sistema operativo es CentOS 5.5 x86_64, con un *kernel* 2.6.18-194.32.1.el5xen. La versión de Xen es: *Hypervisor* 3.1 y Dominio 0: 3.0-x86_64. La memoria es de 8 Gb, el procesador es un Intel i5 750 y el disco duro es de 800 Gb.

AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por el Proyecto CPUFLIPI (MICINN TIN2010-17298), por el Proyecto MBACToIP, de la Agencia I+D del Gobierno de Aragón e Ibercaja Obra Social, y por el Proyecto NDCIPI-QQoE de la Cátedra Telefónica, de la Universidad de Zaragoza.

REFERENCIAS

- [1] E. Göktürk, "A stance on emulation and testbeds", in Proc. 21st European Conference on Modelling and Simulation ECMS 2007 developerworks/library/l-linuxvirt/index.html, último acceso 4/4/2011
- [2] B. Quetier, V. Neri, F. Cappello, "Selecting A Virtualization System For Grid/P2P Large Scale Emulation", in Proc. of the Workshop on Experimental Grid testbeds for the assessment of large-scale applications and tools EXPGRID'06, Paris, 2006
- [3] C.E Caicedo, W. Cerroni, "Design of a computer networking laboratory for efficient manageability and effective teaching," *Frontiers in Education Conference, 2009. FIE '09. 39th IEEE*, vol., no., pp.1-6, 18-21 Oct. 2009.
- [4] W. Fuertes, J. E. López de Vergara, F. Meneses, "Educational Platform using Virtualization Technologies: Teaching-Learning Applications and Research Uses Cases," in Proc. II ACE Seminar: Knowledge Construction in Online Collaborative Communities, 2009.
- [5] T. Lindinger, H. Reiser, N. G. Felde, "Virtualizing an IT-Lab for Higher Education Teaching. In für Informatike.V., G., editor, 1. GI/ITG KuVS Fachgespräch "Virtualisierung", pp. 97-104, Paderborn, Germany, 2008
- [6] J. Murillo, J. M. Saldaña Medina, J. Fernández Navajas, J. Ruiz Mas, E. A. Viruete Navarro, J. I. Aznar Baranda. "Análisis de QoS para una Plataforma Distribuida de Telefonía IP" .Actas de las IX Jornadas de Ingeniería Telemática (JITEL 2010). Valladolid. Septiembre. 2010.
- [7] H. Hellbrück, S. Fischer, "Towards analysis and simulation of ad-hoc networks", in ICWN02: Proceedings of the International Conference on Wireless Networks, pp. 69-75, Las Vegas, 2002.
- [8] B. Thompson, T. Koren, and D. Wing, "RFC 4170: Tunneling Multiplexed Compressed RTP (TCRTP)," 2005
- [9] H.P. Sze, S. C. Liew, J.Y.B. Lee, D.C.S.Yip. "A Multiplexing Scheme for H.323 Voice-Over-IP Applications", *IEEE J. Select. Areas Commun*, Vol. 20, pp. 1360-1368, Sept. 2002.
- [10] A. Botta, A. Dainotti, A. Pescapè, "Multi-protocol and multi-platform traffic generation and measurement", INFOCOM 2007 DEMO Session, May 2007, Anchorage (Alaska, USA)
- [11] ITU-T Recommendation G.107, "E-model, a computational model for use in transmission planning," 2003
- [12] M. Riesco González, "El enfoque por competencias en el EEES y sus implicaciones en la enseñanza aprendizaje". *Tendencias Pedagógicas*, núm.13, 2008, pp. 79 a 105.
- [13] M. Tim Jones, "Virtual Linux. An overview of virtualization methods, architectures, and implementations", <http://www.ibm.com/developerworks/linux/library/l-linuxvirt/>
- [14] S. Hemminger, "Network Emulation with NetEm". In *Proceedings of Linux Conference AU*, Canberra, 2005.
- [15] J. Manner, JTG, <http://www.cs.helsinki.fi/u/jmanner/software/jtg/>, último acceso 5/4/2011.