

Laboratorios en la nube: un caso práctico

(Recibido: 06/03/2015; Aceptado: 17/05/2015)

Ramon Sanchez Iborra, Maria-Dolores Cano

Grupo de Ingeniería Telemática. Departamento de Tecnologías de la Información y las Comunicaciones
Antiguo Cuartel de Antigones. Plaza del Hospital, N° 1, 30202 Cartagena (Murcia)

Teléfono: 968338872

E-mail: {ramon.sanchez, mdolores.cano}@upct.es

Resumen. La computación en la nube es un nuevo paradigma que ha ganado gran aceptación por parte de los usuarios finales y la comunidad científica. Desde el punto de vista investigador, existen distintas plataformas on-line que permiten acceder a gran cantidad de equipamiento facilitando el diseño e implementación de experimentos. En este trabajo se presenta una descripción de una de las plataformas más empleadas: Emulab. Además, se plantea un experimento real y se detallan las distintas etapas de su diseño, control y recogida de resultados. A partir de los resultados experimentales obtenidos se concluye la incapacidad del protocolo de enrutamiento inalámbrico ad-hoc OLSR para manejar tráfico de vídeo en tiempo real.

Abstract. Cloud computing is a new paradigm that has been widely accepted by both final users and the research community. From the latter's perspective, there are several on-line platforms that allow researchers accessing to a great variety of equipment that ease them designing and implementing research experiments. In this work, it is described of one of the most employed platforms: Emulab. Additionally, a real test-bench is presented, detailing the different experiment stages: design, control, and results collecting. From these results, it is concluded the incapability of the wireless ad-hoc routing protocol OLSR to support real-time video traffic.

Keywords. Computación en la nube; Emulab; QoE

1. Introducción

Desde hace unos años hemos entrado en la era de la computación en la nube [1]. Las altas velocidades de acceso que alcanzan las conexiones de los usuarios finales, así como el desarrollo de servidores con gran capacidad de cómputo, han contribuido a que multitud de servicios deslocalizados hayan cobrado gran auge. Existe a disposición un gran abanico de servicios tales como: almacenamiento, sincronización entre equipos, realización de cálculos complejos, etc. Estos servicios reportan grandes ventajas a sus usuarios como su facilidad de uso, ya que el prestador del servicio se ocupa de las tareas de mantenimiento, actualización de equipos, etc. y el usuario final simplemente accede al servicio "útil" de forma amigable para él. Otro beneficio de cara al usuario final es la capacidad de movilidad que le brindan estos servicios, ya que están disponibles en cualquier parte del mundo a través de Internet, así como su ahorro económico y energético puesto que no es necesario que el usuario final compre equipamiento adicional ni licencias de *Software*.

Además de los servicios más populares, destinados a usuarios cotidianos, también existen distintas plataformas que facilitan el trabajo diario en las tareas de investigación. Si nos centramos en la rama de Ingeniería Telemática, es posible acceder a distintos laboratorios *on-line* que permiten formar redes muy extensas que quedan a completa disposición de los investigadores para realizar experimentos y análisis de muy diversa índole. Ejemplos de estos laboratorios son WAIL (*Wisconsin Advanced Internet Laboratory*) [2], iLab.t [3] o

Emulab [4]. Sin duda, fue este último uno de los precursores a la hora de dejar a disposición pública una importante cantidad de equipos sin coste alguno para los usuarios que deseen hacer uso de ellos. Actualmente, es también uno de los proyectos más activos con más de 10.000 experimentos registrados en sus servidores. Debido a las interesantes ventajas que esta plataforma puede ofrecer tanto en el ámbito académico como en el investigador, en este trabajo se presenta una descripción de la misma, mostrando sus principales características y modos de configuración. Además, se muestra un caso de uso real, en el que es evaluado el funcionamiento del protocolo de enrutamiento inalámbrico *ad-hoc* OLSR (*Optimized Link State Routing*)[5] soportando tráfico multimedia desde una perspectiva de Calidad de Experiencia del usuario (*Quality of user Experience*, QoE). El resto del documento se estructura como sigue. En la Sección 2 se presenta una detallada descripción de la plataforma Emulab. En la Sección 3 se muestran los resultados de un caso real de uso, analizando el funcionamiento del protocolo OLSR soportando tráfico de vídeo. Finalmente, se presentan las conclusiones extraídas de este trabajo.

2. Emulab

2.1. Descripción

Esta plataforma fue puesta en marcha por la Universidad de Utah (Estados Unidos) en el año 2000 [6]. Dispone de una gran cantidad de equipamiento de red accesible para el diseño de experimentos. A modo de ejemplo, reseñar que dispone de 588

ordenadores con tarjeta de red Ethernet, de los cuales 118 también cuentan con tarjeta de red inalámbrica,

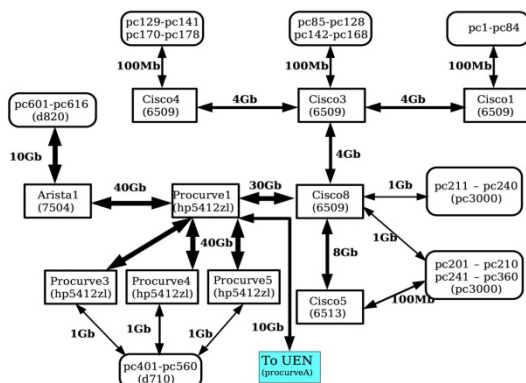


Fig. 1. Red Emulab (extraída de [4]).

13 encaminadores y *switches* y 9 servidores de diversa índole. En la Fig. 1 se muestra la topología lógica de la red que conecta todo este equipamiento. Esta variedad de equipos permite diseñar experimentos a gran escala tanto en número de nodos formando la red como en la cantidad de servicios que ésta puede albergar. Una de las características más interesantes que proporciona Emulab es la posibilidad de cargar en los nodos un sistema operativo editado por los usuarios. Existe una serie de imágenes de sistemas operativos Linux (Ubuntu, Fedora, etc.) que se toman como base. Una vez arrancado el equipo con uno de estos sistemas operativos ya pre-configurados para funcionar correctamente en los equipos de Emulab, se pueden instalar nuevos paquetes y configurar el sistema operativo “a medida”. Antes de finalizar la sesión es posible realizar una imagen del sistema en su último estado, de modo que en los siguientes usos de la plataforma, no es necesario volver a configurar todos los equipos ya que éstos tendrán instalado todo el *Software* necesario para comenzar con la experimentación. Para poder realizar un experimento, es necesario darse de alta en la plataforma y, una vez aceptado el registro por parte de los administradores de Emulab, acceder al sistema a través de su interfaz web [4] o de una aplicación cliente con interfaz de usuario programada en Java. En las siguientes subsecciones se detallan los pasos para diseñar y controlar un experimento.

2.2. Diseño, seguimiento y finalización de un experimento

Después de haber accedido al sistema (a partir de ahora se realizará una descripción del proceso desde la interfaz web de Emulab) es necesario crear la topología de la red a implementar. Para ello se utiliza el lenguaje de descripción de red *NS* [7], indicando los nodos que formarán la red y las conexiones entre ellos, además de otras características de la red como la introducción de retardos emulados en los enlaces. Es posible indicar qué tipo de nodos concretos se requieren atendiendo a sus características (con mayor o menor capacidad de cómputo, con o sin tarjeta inalámbrica, etc.) o permitir a la plataforma que

seleccione de forma automática los nodos físicos que formarán parte de la red descrita. Una vez que se ha realizado la reserva de equipos, Emulab implementa

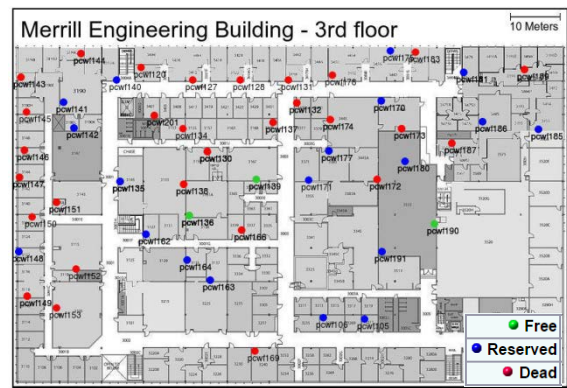


Fig. 2. Mapa de nodos inalámbricos en Emulab (extraída de [4]).

la topología lógica de la red de forma que se emula la topología física diseñada por el usuario. Tras la finalización de este proceso, denominando *swap in*, los nodos se pondrán en marcha a la espera de interacción por parte del usuario. Con el fin de configurar los nodos de la red y dar funcionalidad a la misma, es posible acceder a los equipos mediante conexiones seguras SSH. De este modo, desde una única pantalla se puede tener el control de multitud de equipos de forma simultánea. Para completar la interacción entre el usuario y los equipos a su disposición, es posible copiar y descargar ficheros de ellos mediante transacciones SFTP a un espacio habilitado por la plataforma para tal fin. Este espacio es compartido por todos los nodos en la red por lo que no es necesario subir el mismo fichero repetidas veces a cada uno de los nodos. Así, copiar *scripts* de configuración en los equipos o descargar ficheros de resultados resulta una tarea sencilla y eficiente. Los experimentos tienen una duración limitada para evitar el bloqueo de los equipos de forma que no estén disponibles para otros investigadores. Así, la terminación de un experimento (*swap out*) se puede realizar de forma manual o se producirá de forma automática cuando pase un determinado tiempo.

3. Caso de uso

Tal y como se señaló en secciones anteriores, a modo de ejemplo se ha diseñado un experimento con nodos inalámbricos con el fin de evaluar la capacidad del protocolo de enrutamiento *ad-hoc* OLSR para soportar tráfico multimedia (*video-streaming*).

3.1. Escenario

Se ha establecido una red formada por 15 nodos inalámbricos situados en forma de malla sin seguir ningún patrón definido. Emulab provee un mapa con la situación de cada uno de los nodos para tener un mejor entendimiento de la red formada (ver Fig. 2). Para este experimento se emplearon nodos del tipo pc2400w. Se utilizó una imagen pre-configurada de la distribución de Linux Ubuntu 12.04 (64 bits) con distintos paquetes necesarios cargados en el Kernel de Linux, como el *olsrd daemon* v0.6.6.2. Para las

conexiones inalámbricas se empleó el estándar IEEE 802.11g a 54 Mbps en su configuración por defecto. Las tarjetas inalámbricas fueron configuradas para que emitieran a una potencia de 1mW. El tráfico de vídeo fue generado mediante el *Distributed-Internet Traffic Generator* v2.8.1 [8] a distintas tasas de transmisión. Con el fin de evitar desviaciones puntuales en los resultados obtenidos, se crearon 5 grupos aleatorios de pares transmisor-receptor y se establecieron 3 comunicaciones independientes de 60 segundos entre cada par transmisor-receptor.

3.2. Resultados

La Fig. 3 muestra la Calidad de Experiencia en términos de MOS (*Mean Opinion Score*), calculada mediante el modelo matemático presentado en [9], para un número variable de flujos de vídeo en el sistema enviados a distintas tasas de transmisión. Observar el pobre desempeño de OLSR soportando este tipo de tráfico: tan sólo en el caso de una única transmisión en el sistema con las tasas más bajas (70 Kbps y 100 Kbps) el MOS queda por encima del valor mínimo de calidad recomendado (3.6). Para el resto de escenarios se obtiene el valor mínimo en la escala MOS (1). Para entender estos resultados, la Fig. 4 muestra las pérdidas de paquetes en los mismos escenarios evaluados anteriormente. Observar como los paquetes perdidos crecen a medida que el número de flujos en el sistema y la tasa de transmisión de vídeo también lo hacen. Considerando que el límite máximo establecido por el ITU-T para pérdida de paquetes en transmisiones de vídeo es del 1.5%, los valores obtenidos en la red gestionada por OLSR resultan inaceptables. Así, podemos concluir que OLSR en su configuración por defecto no se muestra capaz de gestionar altas cargas de tráfico por lo que la calidad del servicio operando sobre él cae de forma dramática.

4. Conclusiones

En este trabajo se han mostrado las grandes ventajas que las plataformas de computación en la nube proporcionan a los investigadores. Concretamente, se ha presentado una descripción de la plataforma Emulab, una de las más empleadas en la actualidad. A través de ella es posible acceder de forma remota y con total libertad de configuración a un elevado número de equipos, lo cual abre un infinito abanico de posibilidades tanto en el ámbito académico como en el investigador. A modo de prueba de concepto, se ha diseñado un experimento con nodos inalámbricos con el fin de evaluar la capacidad del protocolo de enrutamiento *ad-hoc* OLSR para manejar alta carga de tráfico como la que supone la transmisión de vídeo en tiempo real. De los resultados obtenidos, se concluye que la calidad que percibe un usuario (QoE) al establecer una comunicación de vídeo en una red *ad-hoc* gestionada por OLSR resulta inaceptable a partir de los 100 Kbps de tasa de transmisión. Además, cuando más de un flujo de vídeo concurre en el sistema, también se produce una degradación en

las prestaciones de la red que provocan una dramática caída de la calidad de las transmisiones.

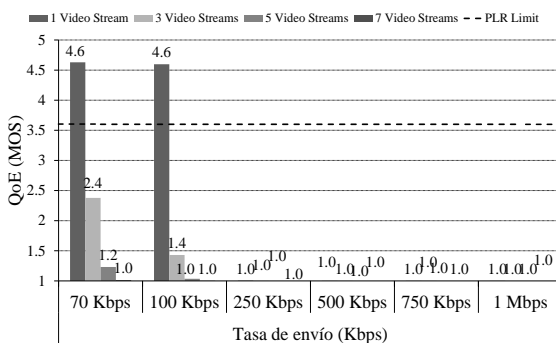


Fig. 3. QoE obtenida para distinto número de flujos de vídeo simultáneos enviados a diferentes tasas de transmisión

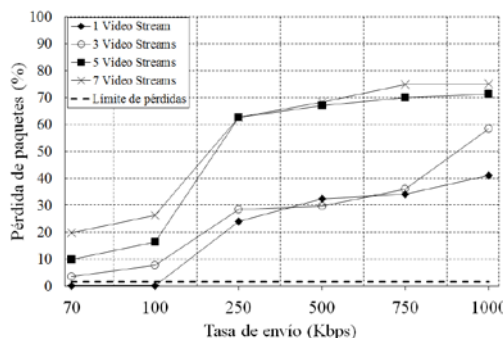


Fig. 4. Pérdida de paquetes obtenida para distinto número de flujos de vídeo simultáneos enviados a diferentes tasas de transmisión

Agradecimientos

Este trabajo ha sido financiado por el MINECO/FEDER con el proyecto TEC2013-47016-C2-2-R (COINS).

Referencias

- [1] M. R. Media, "Global Cloud Computing Market Forecast 2015-2020," 2014. [Online]. Available: <http://www.marketresearchmedia.com/?p=839>.
- [2] "Wisconsin Advanced Internet Laboratory (WAIL)." [Online]. Available: <http://wail.cs.wisc.edu/>.
- [3] "iLab.t." [Online]. Available: <http://ilabt.iminds.be/>.
- [4] "University of Utah Flux Research Group, Emulab: The Utah Network Testbed." [Online]. Available: <http://www.emulab.net>.
- [5] T. Clausen and P. Jacquet, "Optimized Link State Routing (OLSR) Protocol," *RFC 3626*, 2003.
- [6] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An integrated experimental environment for distributed systems and networks," *ACM SIGOPS Oper. Syst. Rev. - OSDI '02 Proc. 5th Symp. Oper. Syst. Des. Implement.*, vol. 36, no. SI, pp. 255 – 270, 2002.
- [7] "The Network Simulator (NS-2)." [Online]. Available: <http://www.isi.edu/nsnam/ns/>.
- [8] A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," *Comput. Networks*, vol. 56, no. 15, pp. 3531–3547, Oct. 2012.
- [9] J. Joskowicz and J. C. L. Ardao, "Enhancements to the opinion model for video-telephony applications," in *5th International Latin American Networking Conference- LANC '09*, 2009, p. 87.