Revista Tecnológica ESPOL, vol. 20, N. 1, 1-8,(julio, 2007)

Análisis y Estudio del plano de control de un Open Router basado en el Sistema Operativo Linux y en el Open Source Software Xorp

Olga Jaramillo[1], Rebeca Estrada[1], Raffaele Bolla[2]
[1]Escuela Superior Politécnica del Litoral (ESPOL), Facultad de Ingeniería en Electricidad y Computación (FIEC), Campus Gustavo Galindo, Km 30.5 via Perimetral, Apartado 09-01-5863. Guayaquil, Ecuador
[2]Universidad de Génova, Departamento de Informática, Sistemática y Telemática (DIST), Vía Opera Pia 13 Apartado 16145, Génova, Italia.
ojaramil@fiec.espol.edu.ec, restrada@espol.edu.ec, Raffaele.Bolla@unige.it

Resumen

Los aparatos comerciales son caracterizados de arquitecturas cerradas que impiden el análisis de nuevos mecanismos, en la actualidad se han desarrollado proyectos de tipo open-source (código abierto) que permiten a programadores desarrollar aparatos, como el llamado Open Router, que compitan con los aparatos comerciales. El objetivo de este articulo ha sido el de realizar un análisis de prestaciones open-source que concierne al plano de control utilizando los protocolos de enrutamiento BGP y OSPF soportados por el software de código abierto Xorp. Realizamos pruebas que permitan examinar el desempeño del Open Router como por ejemplo el tiempo de convergencia de dicho aparato al producirse un cambio en la topología de la red. Los resultados obtenidos fueron confrontados con las características de aparatos comerciales, demostrando que el Open Router es una prueba ideal en el campo de la investigación y una buena alternativa a bajo costo.

Palabras Claves: *Open Router, protocolos de enrutamiento, Linux, Xorp, BGP, OSPF.*

Abstract

The commercial devices are characterized of closed architectures that prevent the analysis of new mechanisms, actually new projects had been developed, like open-source, that allow programmers to develop new devices, as the Open-Router, to compete with the commercial devices. The objective of this article has been the one to make an analysis of the features open-source that concerns the control view using the routing protocols BGP y OSPF supported by the open code software Xorp.

We made tests that allow examining the performance of the Open Router, as for example what occurs with the convergence time when a change in the network's topology had happened. The obtained results were faced with the features of the commercial devices, proving that the Open Router is an ideal device on the investigation field and a good choice at low cost.

***Key words**: Open Router, routing protocols, Linux, Xorp, BGP, OSPF.*

1. Introducción

La tecnología de red que ha tenido la mayor difusión es la suite protocolar TCP/IP, nacida para el transporte de datos y hoy candidata a convertirse en la tecnología de referencia para futuras redes globales.

El triunfo obtenido por Internet proviene en gran parte del hecho que nació como una suite protocolar abierta: todos los protocolos, las arquitecturas y las estructuras han sido creadas y descritas públicamente.

Lo contrario a que casi todos los aparatos comerciales son caracterizados de arquitecturas "cerradas", es decir los detalles arquitecturales no son de dominio público.

En los años 80 los programadores comenzaron a limitar los derechos de usos de sus software, haciendo de esta manera imposible realizar, analizar y validar nuevos mecanismos y protocolos sobre aparatos comerciales. Fue de esta manera que gran parte de estos grupos focalizaron el propio interés sobre proyectos de código abierto (open-source) basados en el sistema operativo Linux.

El conjunto de estos instrumentos open-source constituye una arquitectura de red que está en grado de competir con la mayor parte de los protocolos comerciales. Esta arquitectura se llama Open Router.

A pesar de los muchos elementos arquitecturales del Open Router que ya han sido descritos, todavía faltan, estudios que analicen el perfomance obtenido en este tipo de aparatos, lo que se convierte en nuestro objetivo. Realizaremos pruebas que muestren el desempeño del Open Router, analizaremos índices de prestación como el tiempo de convergencia y confrontaremos dichos resultados con las características de un aparato comercial. El Open Router dispone de una plataforma Linux equipada con un kernel 2.6, y de una plataforma de enrutamiento open source llamada Xorp.

Este trabajo fue desarrollado en el laboratorio de Telemática del Departamento de Informática, Sistemática y Telemática (DIST) de la Universidad de Génova, Italia.

2. Arquitectura del Open Router

El Open Router está basado en el chipset ServerWorks GC-LE, cuya arquitectura es ilustrada en figura 1. El chipset está en grado de soportar un sistema basado sobre doble procesador Xeon con dual memory channel y bus PCI-C a 133 Mhz por 64 bits de paralelismo.

El procesador Intel Xeon se basa sobre el core del Pentium 4 con 603 pines que permite configuraciones multiprocesador con soporte Hyper-Threading. Deriva de la arquitectura NetBurst de Intel y es uno de los procesadores que mejor se prestan a arquitecturas server high-end gracias a la notable dimensión de las cache L2/L3. En particular en nuestro testbed hemos usado dos procesadores Xeon con reloj de 2.4 GHz y 512 KB de cache.

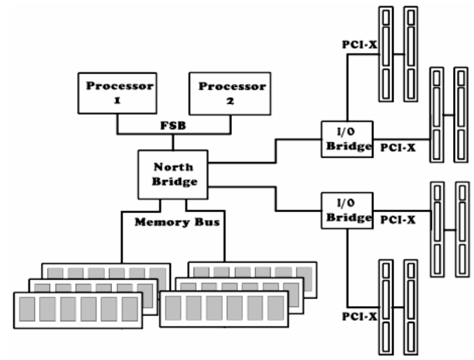

Figura 1. Esquema de la mainboard Supermicro X5DL8-GC.

2.1 Tarjetas de red

Como descrito en [1], las tarjetas presentes en el mercado tienen diferentes niveles de prestaciones máximas y un diverso nivel configurable. Después de haber realizado las debidas consideraciones nuestra selección recae sobre las tarjetas de red Gigabit Ethernet Intel Pro/1000 XT Server, basadas sobre el chipset Intel 8254x y dotadas de controladores PCI-X a 133 Mhz. Estas proporcionan, además, un amplio rango de configuración para varios parámetros como por ejemplo, la longitud de los buffer de recepción y transmisión, la unión de las interrupciones y otros aspectos importantes [2].

2.2 Arquitectura Software

El kernel, en general, constituye el núcleo fundamental de un SO, en cuanto es el elemento que, trabaja directamente con el hardware, administra los recursos del sistema proporcionando a los procesos y a los servicios software una interfaz de alto perfil, que no debe por lo tanto, tener en consideración las peculiaridades de la arquitectura hardware.

Linux [3-4] en particular pertenece a la familia de SO Unix-like como BSD, ha sido desarrollado por Linus Torvalds en 1991 como SO para PC basado en el microprocesador 80386 de Intel. Una de las mayores ventajas de Linux es el hecho que no es un SO comercial: su código fuente es "abierto" y disponible a cualquiera para posibles desarrollos.

2.3 Router Linux

El elemento crucial en la operación de enrutamiento de paquetes IP es el kernel, donde son realizadas todas las operaciones de los niveles de línea, red y transporte (figura. 2).

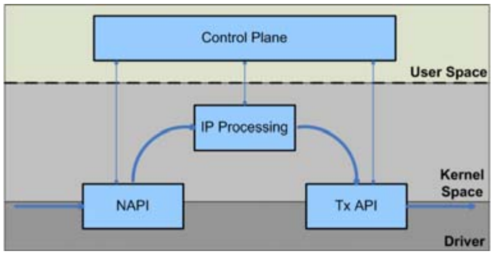

Figura 2. Diagrama de bloques de la estructura software de un PC Linux router

El mecanismo de forwarding de todos los kernel Linux está constituido de una cadena de tres diferentes módulos: una "API de recepción" que administra la recepción de los paquetes (NAPI), un módulo que extrae la elaboración del nivel IP y, una "API de transmisión" que se ocupa de las operaciones de envío hacia las interfaces de salida.

3. Protocolos de enrutamiento

Para transferir los paquetes entre una pareja de hosts, la capa de red tiene el deber de determinar el camino fuente-destino. El núcleo de cada uno de los protocolos de enrutamiento es el algoritmo de enrutamiento cuyo objetivo es: dado un grupo de routers, conectados mediante enlaces, un algoritmo de enrutamiento encuentra un "buen" camino (a "costo mínimo") desde la fuente hacia el destino.

3.1 Open Shortest Path First (OSPF)

El protocolo de enrutamiento OSPF regula las modalidades con las que logra hacer llegar a todos, la información necesaria para la compilación de las tablas de enrutamiento y forwarding, que consienten el enrutamiento de los paquetes a lo largo del camino de costo mínimo. El protocolo de enrutamiento OSPF, actúa en modo automático y dinámico. Para hacer esto OSPF se basa sobre el algoritmo estado del enlace (link-state) en colaboración con algoritmos de tipo SPF (Shortest Path First).

Todas la comunicaciones entre routers se realizan mediante el intercambio de paquetes OSPF, que son transmitidos en el Internet junto al IP user traffic. OSPF provee por si mismo el control de la transmisión mediante un mecanismo de acknowledgment, y no requiere por lo tanto el soporte de un protocolo de transporte. Un paquete OSPF es intercambiado solamente entre routers adyacentes.

3.2 Border Gateway Protocol (BGP)

El protocolo Border Gateway versión 4, definido en [5], es el protocolo estándar para el enrutamiento inter-dominio en el Internet actual, conocido como BGP4 o simplemente como BGP. Es un protocolo de red usado para conectar entre ellos varios routers que pertenecen a sistemas autónomos distintos, y que son llamados gateway. BGP es caracterizado como un protocolo vector de caminos (path vector), porque los routers BGP contiguos, llamados pares BGP (BGP peer), se intercambian información detallada de los caminos que información sobre los costos.

BGP es un protocolo distribuido, los routers BGP calculan independientemente sus tablas, y se comunican solo con los routers BGP directamente conectados.

La información global sobre los caminos hacia destinos remotos se propaga de AS a AS mediante el intercambio de información de enrutamiento BGP entre parejas de routers BGP directamente conectadas.

El funcionamiento de BGP gira alrededor a tres actividades, todas legadas a los anuncios sobre caminos:
- Recepción y filtraje de anuncios en los caminos por parte de vecinos directamente conectados.
- Selección del camino.
- Envío de anuncios a los vecinos

4. Plataforma Extensible de Código Abierto (XORP)

Xorp es un software de enrutamiento que garantiza a los aspectos de estabilidad y latencia la atención necesaria que otros sistemas software no pueden garantizar debido a que los desarrolladores del software han adoptado como vínculo primario la escalabilidad.

El diseño de Xorp consiste de un framework compuesto de procesos de enrutamiento, cada uno compuesto de bloques modulares mediante el cual las rutas fluyen.

Xorp, por lo tanto, deriva de estrategias utilizadas para subdividir el plano de control y los protocolos de enrutamiento individuales, en componentes que facilitan sea la extensión que el mejoramiento del perfomance.

4.1 Arquitectura modular de Xorp

4.1.1 Procesos de gestión y de enrutamiento

La figura 3 muestra la estructura software general del plano de control de un router en lo que se refiere al enrutamiento, poniendo en evidencia los principales procesos involucrados y las relaciones más importantes entre ellos.

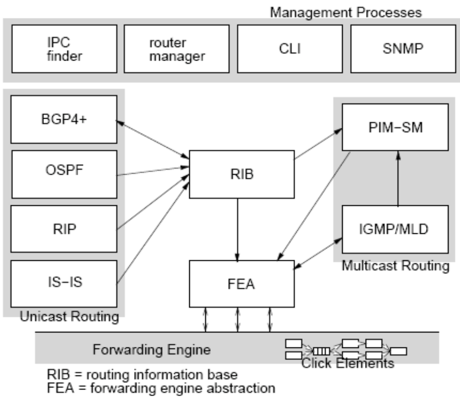

Figura 3. Management Process

RIB: decide que protocolo usar entre las diferentes alternativas.

BGP: las rutas BGP entrantes normalmente individúan más bien un next-hop router hacia el destino que un vecino.

FEA: garantiza una API estable para la comunicación con uno o más forwarding engine.

PIM-SM e IGMP: proveen las funcionalidades de multicast enrutamiento, donde PIM realiza el forwarding e IGMP informa al PIM de la existencia de receptores locales.

IS-IS: este modulo, todavia no está implementado.

IPC: garantiza la comunicación entre los procesos de Xorp y las aplicaciones de enrutamiento construidas.

Pasando a los procesos de gestión citamos:

Router Manager: El Router Manager basándose en los file de configuración del router da inicio, configura y bloquea el protocolo de enrutamiento y otras funcionalidades del router.

SNMP: El Router Manager está configurado para dirigir el agente SNMP con el fin de supervisar la administración de los aparatos conectados a la red.

4.1.2 Comunicación entre procesos

Los procesos se comunican entre ellos utilizando un mecanismo IPC extenso llamado XRLs. Este mecanismo es soportado por cada proceso; cuando un proceso desea comunicarse con otro escribe un XRL, diseccionado al proceso genérico (BGP, OSPF, RIP, etc) y lo envía.

El llamado Finder interpreta "resuelve" este XRL genérico, en una forma que especifica definitivamente cual forma de comunicación debe producirse; el resultado consiste en la individuación del protocolo de transporte que será utilizado para la comunicación, como por ejemplo TCP, y cada parámetro de la misma, así como hostname y puerta.

El código de Xorp ha sido escrito en C++, gracias a las características de ser un lenguaje orientado a objetos y garante de buen perfomance.

5. Resultados

Ilustraremos los testbed experimentales utilizados y describiremos con detalles los resultados obtenidos.

5.1 Configuración de BGP con Xorp

Hemos iniciado con algunos tests que conciernen el proceso de instauración de la adyacencia entre 2 routers BGP que pertenecen a sistemas autónomos diferentes.

```
protocols {
      bgp {
         bgp-id: 10.0.2.1
         local-as: 65001
         peer 10.0.2.2 {
            local-ip: 10.0.2.1
            as: 65002
            next-hop: 10.0.2.1
            local-port:179
            peer-port: 179
            holdtime: 120
            }
      }
}
```

Figura 4. file configbgp.boot

La figura 4 representa el file de configuración Xorp que permite establecer una comunicación entre los routers de la figura 5.

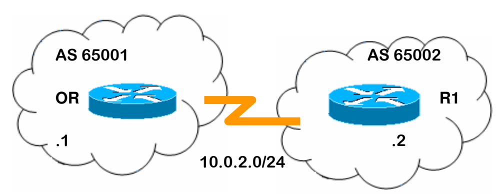

Figura 5. Conexión BGP entre routers de AS diferentes

La figura 5 muestra que el OR se encuentra en el AS 65001 y un router virtual R1 en el AS 65002. Conectamos una de las interfaces del OR a una de las puertas Gigabit del Router Tester.

Con la ayuda de la shell de Xorp podemos monitorear dicha conexión y así verificar el estado de los BGP peerings y de la tabla de enrutamiento.

Para visualizar los detalles de los pares conectados, se utiliza:

```
root@openrouter> show bgp peers detail
    Peer 1: local 10.0.2.1/179 remote 10.0.2.2/179
    Peer ID: 10.0.2.2
    Peer State: ESTABLISHED
    Admin State: START
    Negotiated BGP Version: 4
    Peer AS Number: 65002
    Updates Received: 2,  Updates Sent: 0
    Keep Alive Time: 30 s
    Configured Hold Time: 120 seconds,
```

Podemos ver la tabla de enrutamiento BGP mediante el comando:

```
root@openrouter> show bgp routes
    Status Codes: * valid route, > best route
    Origin Codes: i IGP, e EGP, ? incomplete
    Prefix        Nexthop      Peer        AS Path
 ------          -------         ----          -------
*> 10.0.2.0/24  10.0.2.2     10.0.2.2    65002 e
```

Ahora pasamos a analizar las modalidades de comunicación entre una pareja de pares conectados, y el proceso que siguen para establecer una sesión BGP. Para esto consideramos la situación ilustrada en figura 6, que muestra una interconexión BGP entre 3 AS diferentes.

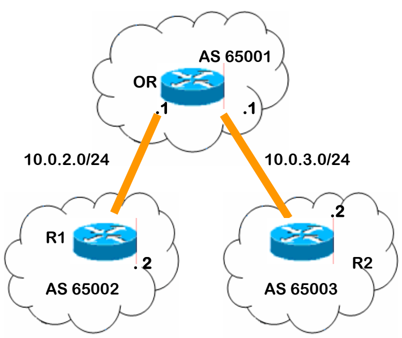

Figura 6. Conexión BGP entre routers de 3 AS diferentes

El OR se encuentra en el AS 65001, y desea establecer una comunicación BGP con R1 y R2 que se encuentran en los AS 65002 y 65003 respectivamente. El nuevo file de configuración es referido a continuación:

```
protocols {
   bgp {
      bgp-id: 10.0.2.1
      local-as: 65001
      peer 10.0.2.2 {
         local-ip: 10.0.2.1
         as: 65002
         next-hop: 10.0.2.1
                      }
   peer 10.0.3.2 {
         local-ip: 10.0.3.1
         as: 65003
         next-hop: 10.0.3.1
         }
      }
   }
```

El primer mensaje que el OR enviará a R1 y a R2 será el mensaje Open (figura 7), usado para intercambiar información de configuración y negociar los primeros parámetros necesarios para la sesión de peering.

```
Marker = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Message Length = 37
Message Type = Open (1)
Version = 4
Autonomous System (AS) Number Of The Sender = 65001
Hold Time (In Seconds) = 120
BGP Identifier = 10.0.2.1
```

Figura 7. Mensaje Open

Si R1 y R2 aceptan la conexión BGP con el OR, ambos enviarán un mensaje Keepalive.

En este punto el OR envía el primer mensaje de update a R1 (figura 8) y a R2 y recibe los update de R1 y R2.

```
Marker = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Message Length = 54
Message Type = Update (2)
Unfeasible Routes Length = 0
Total Path Attribute Length = 27
Attribute: Type = NEXT HOP (Length = 4)
    Flags = 0x40
        IP Address Of Border Router = 10.0.2.1
Attribute: Type = AS_PATH (Length = 6)
        Flags = 0x40
            Path Segment Type = AS_SEQUENCE
            Number Of ASs In Path Segment Value Field = 2
            Autonomous System Number 0 = 65001
            Autonomous System Number 1 = 65003
IP Address Prefix/Length = 10:0:3/24
```

Figura 8. Mensaje Update que recibe R1

El mensaje que recibe R2 es similar al de figura 8, con la diferencia que tendrá la información de enrutamiento entre el OR y R2.

5.1.1 Tiempo de convergencia

Cuando un router decide eliminar una o varias rutas, éste envía un mensaje update con campo Unfeasible Routes Length con valor diferente de cero, indicando la presencia del Withdrawn Routes, que a su vez, contienen las rutas que han sido eliminadas. El router que recibe éste mensaje debe recalcular su tabla de enrutamiento BGP, eligiendo el nuevo camino hacia el destino.

Ahora que hemos aclarado el mecanismo de eliminación de una o más rutas, procederemos a efectuar un test para entender como reacciona nuestro sistema cuando es anunciada la eliminación de una ruta. El parámetro clave para éste tipo de análisis es el tiempo de convergencia (switching time), es decir el intervalo temporal que trascurre entre la recepción de un mensaje update, indicando un cambio topológico tal de modificar el next-hop para uno o más destinos, y el primer paquete

IP recibido sobre el enlace directamente conectado al SUT y perteneciente al nuevo camino.

Con el fin de testear y validar el mecanismo de eliminación de una ruta en Xorp hemos realizado una prueba en la cual el router 10.0.1.2 anuncia la red virtual 192.0.0.1 que puede ser alcanzada sea por R1 que por R2 (figura 9).

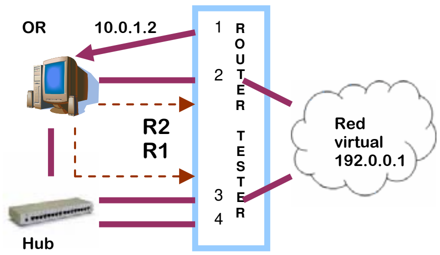

Figura 9. Topología de testing

Cuando el enlace R1-red virtual es deshabilitado, R1 envía un mensaje update al OR indicando a R2, mediante un mensaje update, que ha ocurrido un cambio topológico, y ambos recalculan sus tablas de enrutamiento.

Cuando el OR actualiza su tabla de enrutamiento, elegirá R2 como el nuevo camino para enviar paquetes hasta 192.0.0. En la figura 10 mostramos los resultados obtenidos en la simulación realizada según el esquema de figura 9.

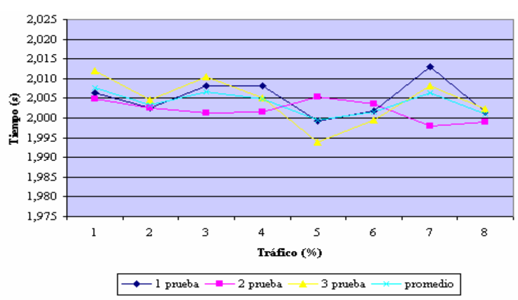

Figura 10. Tiempo de convergencia

El tiempo de convergencia corresponde en promedio, cercano a 2 segundos para todas las pruebas realizadas, tiempo razonable para recalcular un nuevo camino, considerando el hecho que BGP por eficiencia solo envía las diferencias de la actualización precedente.

5.2 Testbed experimentales sobre OSPF

5.2.1 Tiempo de desconexión en una comunicación OSPF

Hemos operado nuestro test usando la siguiente configuración (figura 11):

-Un solo flujo monodireccional que atraviesa el OR desde una puerta Gigabit a otra.

Figura 11. Configuración de bechmarking

Un parámetro importante incluido en el paquete hello es el Router Dead Interval que especifica el número de segundos que pasa antes de declarar un router inactivo, 40 segundos, es decir si el router no recibe al menos 1 paquete hello en éste tiempo el estado de la comunicación OSPF pasará al estado Down porque considerará que el router vecino no es alcanzable., y cuando el router recibe un paquete hello donde no se ha incluido en el campo neighbour Router el ID del receptor lleva automáticamente la NDS al estado Init, y por lo tanto el otro router no está en grado de mantener la comunicación bidireccional activa.

Para conocer el tiempo que permanece inactiva una comunicación OSPF debemos conocer la probabilidad de pérdida de un paquete hello. Con el número de paquetes enviados (offered) y el número de paquetes recibidos (througput) podemos obtener la probabilidad deseada (PrDrop).

$$\Pr Drop(n) = \Pr Drop + \Pr Drop^2 + ... + \Pr Drop^n$$

Donde n es la cantidad de paquetes hello perdidos.

Si perdemos 4 paquetes hello consecutivos, los routers se desconectarán. Para tener una idea del tiempo que permanece inactiva la comunicación, hagamos uso de la siguiente expresión matemática:

$$TiempoDeDesconexión = \Pr Drop(n) * TiempodeConexión$$

En nuestro test hemos establecido una comunicación OSPF por 15 minutos (900 segundos). En este lapso de tiempo la comunicación OSPF se perdió por decenas de segundos. Esta prueba le realizamos a diferentes cantidades de tráfico. La tabla 1 muestra los valores teóricos y prácticos del tiempo de desconexión de nuestras pruebas.

Tráfico (%)	90	80	70	60
Teórico (s)	85,7339	56,25	30,36235	11,11111
Práctico (s)	90	70	15	10

Tabla 1. Tiempo de desconexión

El kernel Linux 2.6 usado en nuestras pruebas hace que el throughput sea solo del 40% del tráfico enviado. El OR genera una gran cantidad de interrupts, que son considerados prioritarios para su correcto funcionamiento y por esto descarta paquetes que considera no importantes, entre ellos los paquetes hello.

Según el tráfico aumenta, la probabilidad de perder los paquetes hello es mayor, como muestra la figura 12 donde se confrontan los resultados obtenidos teóricamente con los resultados obtenidos en nuestro test.

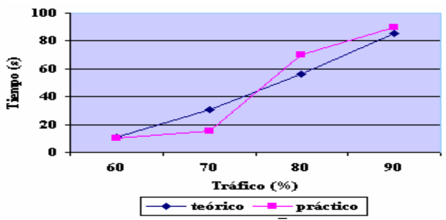

Figura 12. Tiempo de desconexión

5.2.2 Tiempo de convergencia

Para efectuar esta prueba hemos usado 3 interfaces del OR conectadas a 3 puertas Gigabit del Router Tester, como muestra figura 13.

Para alcanzar la red simulada por el Router Tester, el OR utilizará caminos óptimos que incluyen los enlaces A y B.

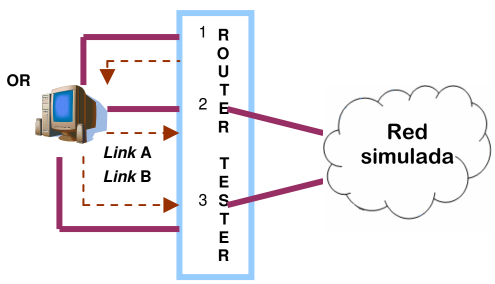

Figura 13. Topología de testing

El OR es atravesado por un flujo de tráfico de datos proveniente de la red simulada detrás de la puerta 1 del Router Tester, que es destinado a la red simulada detrás de las puertas 2 y 3. Consideremos la siguiente condición inicial: el peso del enlace A es menor al del enlace B, por lo tanto, el camino elegido será el enlace A para llegar a la red simulada por el Router Tester.

Si la métrica del enlace A sufre un cambio, se genera un LSA que será enviado al OR. Si este valor de métrica es mayor al del enlace B, la tabla de enrutamiento será modificada para enviar los paquetes hacia la red de destino sobre el nuevo camino.

Los tests mostrados en este párrafo han considerado el estudio de las variaciones del tiempo de convergencia en función del aumento de los nodos pertenecientes a la red simulada.

Decidimos realizar estos experimentos manteniendo constante el trafico de datos generado por el Router Tester y a carga baja por motivos de capacidad del Router Tester Agilent, para lograr capturar los tiempos en que se producen los cambios topológicos y el tiempo en que el nuevo camino recibe el primer paquete IP.

En la figura 14 mostramos el resultado obtenido de las simulaciones efectuadas según el esquema de figura 13.

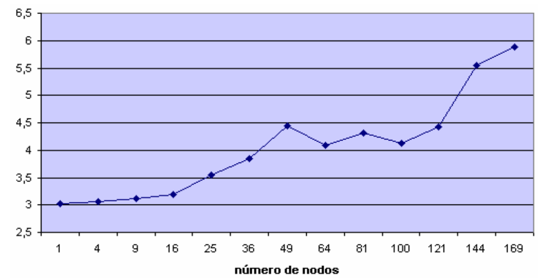

Figura 14. Tiempo de convergencia con trafico al 10%

Podemos notar que los valores del tiempo de convergencia (en el orden de los segundos) son crecientes con el numero de nodos de la red (en nuestra prueba simulamos una red de máximo 169 nodos), seguramente atribuible al hecho que, la estructura de memorización ordenada de los nodos de Xorp fue originalmente implementada mediante un árbol binario, y para conmutar el tráfico sobre el nuevo camino de costo mínimo, se debe realizar el cálculo de las rutas hacia cada destino.

5.3 Confrontación de resultados

Hasta el momento pruebas con un router comercial usando el protocolo BGP en el laboratorio en que se desarrollaron estas pruebas, no se han efectuado como se realizaron con el protocolo OSPF y el router Juniper M10.

Grupos interesados en networking han analizado el impacto del tiempo de convergencia en distintas redes, han creado modelos para dicho estudio. En [6] se establece que el tiempo de convergencia de BGP oscila entre 30 hasta 400 segundos, esto depende de la cantidad de AS que forman la red y de los modelos a seguir según los resultados que se desean obtener.

Las condiciones de nuestras pruebas son diferentes a las expresadas en [6], nosotros usamos 3 AS y ellos 60 AS, por lo tanto al confrontar los resultados nuestro Open Router quedaría en desventaja. Si consideramos el tiempo de convergencia obtenido, 2 segundos, para nuestras condiciones podemos ver que el OR se desempeña favorablemente y que es un óptimo objetivo de futuras investigaciones para mejorar dicho tiempo con redes más grandes y complejas. El avance en el estudio de plataformas de software abierto, como Linux y Xorp, ayudarán a dicho objetivo.

En el caso de OSPF, podemos confrontar nuestros datos con [7]. El test "tiempo de convergencia" fue previamente realizado por un estudiante de tesis de la Universidad de Génova para determinar dicho tiempo en el router comercial Juniper M10. En dichas pruebas se estableció que el tiempo de convergencia, para una red de máximo 1800 nodos, es de 5.4 segundos, en nuestro test el número máximo de nodos es de 169 con un tiempo de convergencia de casi 6 segundos, como podemos ver la diferencia es grande, dicha diferencia es atribuible a las limitaciones en la simulación de la topología ya que al hacer las pruebas con una mayor cantidad de nodos el Open Router se paraba.

Si consideramos el hecho de que son 169 nodos, con un tráfico del 10% a un tiempo de 6 segundos podemos concluir que el desempeño del Open Router es bueno, que es un tiempo considerable y aceptable para que pueda actualizar la tabla de enrutamiento al producirse un cambio en la topología.

6. Conclusiones

Con los resultados presentados en este trabajo, hemos contribuido al estudio de los aparatos Internet y de arquitectura abierta.

Hemos demostrado como esta categoría de aparatos abiertos puede constituir un banco de prueba ideal en el campo de la investigación, y una alternativa a bajo costo a los aparatos comerciales.

Los test realizados en este trabajo han sido ejecutados con conformidad a la RFC 2544 [8]. El estudio de la estructura modular de Xorp nos ha permitido analizar el código fuente, individuar donde está implementada la actualización de las rutas, y de observar los tiempos empleados para el recalculo de las tablas de enrutamiento.

Los tests efectuados usando los protocolos de enrutamiento BGP y OSPF para el tiempo de convergencia muestran que los valores obtenidos son índices de un buen desempeño de nuestro Open Router.

El continuo estudio y actualización de estas plataformas permitirán el progreso del código para un mejor desempeño de los aparatos de arquitectura abierta.

Los resultados de este trabajo representan una buena base de partida para futuros desarrollos que podrían considerar, por ejemplo, el impacto del plano de control sobre las prestaciones globales del sistema, y de sucesivas actividades de investigación de otras plataformas.

7. Referencias

[1] P. Gray, A. Betz, "*Performance Evaluation of Copper-Based Gigabit Ethernet Interfaces*", 27th Annual IEEE Conference on Local Computer Networks (LCN'02), Tampa, Florida, Novembre 2002, pp.679-690.

[2] Intel Corporation. "The Intel PRO/1000 XT Server Adapter". URL: http://www.intel.com/network/connectivity/products/pro1000xt.htm

[3] B. Hubert et al. "*Linux advanced routing & traffic control HOWTO*". URL: http://lartc.org

[4] S. Radhakrishnan. "Linux – Advanced *networking* overview".

[5] Y. Rekhter, T. Li. "*Request for Comments 1771: A Border Gateway Protocol 4 (BGP-4)*". Marzo 1995 URL: http://www.ietf.org/rfc/rfc1771.txt

[6] Dan Pei, Xiaoliang Zhao, Lan Wang, Daniel Massey, Allison Mankin, S. Felix Wu, Lixia Zhang *"Improving BGP Convergence Through Consistency Assertions"*

[7] Andrea Rolleri *"Analisi di prestazioni del piano di controllo e di inoltro in Core Router ad architettura aperta"*. Febrero 2006

[8] S. Bradner, J. McQuaid. "Request for Comments 2544: Benchmarking Methodology for Network Interconnect Devices". Marzo 1999 URL: http://www.ietf.org/rfc/rfc2544.txt