Entornos de verificación de soluciones multi-path BGP

Lisardo Prieto González, José Manuel Camacho Camacho, Francisco Valera Pintor
Ingeniería Telemática,
Universidad Carlos III de Madrid
Avda. de la Universidad, 30, 28911 Leganés (Madrid) España
{lpgonzal, jcamacho, fvalera}@it.uc3m.es

***Resumen*-. Actualmente la utilización simultánea de múltiples caminos en redes de comunicaciones tiene el potencial de generar una serie de importantes beneficios como la mejor utilización de los recursos disponibles o la mayor robustez y protección de las transmisiones. Sin embargo, las soluciones existentes hoy en día para dotar a los routers de un paradigma multi-camino no pasan de ser propuestas aisladas o soluciones concretas intra-dominio. Entre otras cosas, esto se debe a las dificultades de probar y validar las diferentes soluciones como a las dificultades de un posterior despliegue. En este artículo se propone un entorno desarrollado para poder verificar soluciones multi-path inter-dominio (BGP) tanto por la vía de la simulación como por la vía de la implementación real de la solución. Se describirá también cómo se ha validado la propuesta con dos soluciones actualmente en desarrollo, habiéndose detectado así en ellas problemas de convergencia.**

***Palabras Clave*- encaminamiento, multi-path, BGP, simulación, emulación, virtualización, C-BGP, XORP.**

I. INTRODUCCIÓN

Las técnicas de encaminamiento multi-camino (en adelante *multi-path routing*) ponen a disposición de los routers diferentes alternativas para alcanzar un destino concreto que pueden ser usadas de forma concurrente ateniéndose a ciertas restricciones (como que las rutas estén libres de bucles por ejemplo). Esto es posible gracias a la instalación en la tabla de encaminamiento de un router de diferentes „siguientes saltos' hacia el mismo destino para que los utilice todos simultáneamente.

El encaminamiento multi-path tiene una serie de ventajas potenciales muy notables debido a las cuales está recibiendo cada vez más atención. Por ejemplo ([1], [2]):

- Incremento efectivo de la capacidad de la red al permitir que el tráfico se envíe por un mayor número de enlaces.
- Respuesta más rápida a cambios en la red, puesto que ya se han explorado y puesto en funcionamiento diferentes caminos.
- Ingeniería de tráfico escalable, ampliando las posibilidades a la hora de poder configurar los caminos utilizados para optimizar retardos.
- Mejoras en la seguridad, proporcionando protección por ejemplo frente a ataques de denegación de servicio o de inspección de paquetes.

Actualmente existen diferentes soluciones desplegadas en Internet, pero prácticamente todas en el dominio de un único proveedor (intra-dominio) y basadas en protocolos IGP, como el encaminamiento multi-topología de OSPF [3] o el balanceo de carga propuesto por EIGRP [4] o M-PATH [5].

En el caso de entornos inter-dominio, el despliegue de soluciones es mucho menor y aunque es cierto que existen alternativas que proporcionan una mayor variedad de caminos, son en general soluciones muy específicas (habilitar múltiples enlaces entre proveedores, soluciones con encaminamiento basado en fuente, etc. Ver [1]).

No obstante, el creciente interés por las ventajas que ofrecen las alternativas multi-path ha llevado a multitud de propuestas que persiguen dotar de dicha versatilidad al protocolo BGP (ver prospectiva en [1] y [2]).

El proyecto Trilogy (*ICT-2007-216372, Architecting The Future Internet*, [6]) tiene como principal objetivo el desarrollo de nuevas soluciones para la arquitectura de control de Internet (a nivel de routing, control de congestión, etc.). En este proyecto se está prestando una atención especial a diferentes soluciones multi-path, como la propuesta en la capa de transporte, mTCP [7] o diferentes extensiones para multi-path BGP (LP-BGP [8] y MpASS [1]).

Uno de los principales problemas encontrados en el desarrollo de soluciones multi-path BGP es la dificultad para validar dichas propuestas y sobre todo para compararlas con otras alternativas en contextos reales, tanto a nivel de tamaño de la red como a nivel de relaciones reales entre proveedores. En general cada propuesta utiliza un mecanismo diferente de validación (teórico, desarrollo de un software específico para la situación, etc.) y eso evidentemente complica considerablemente la comparación entre ellos o con cualquier otro mecanismo que se pueda proponer.

En este artículo se propone un entorno de verificación de soluciones multi-path BGP, con un doble desarrollo basado en simulación y en emulación real que ha permitido por ejemplo probar las dos soluciones que se han planteado en Trilogy (LP-BGP y MpASS) y detectar problemas de convergencia en dichas soluciones. El artículo describe el trabajo realizado para tratar de objetivar la comparativa entre soluciones multi-path de tal forma que sea sencillo evaluar

diferentes alternativas contrastándolas entre ellas. El entorno es fácilmente extensible y adaptable a las diversas soluciones que se quieran comparar.

El resto del artículo está organizado como sigue. En la sección II se detallan las diferentes alternativas consideradas antes de proceder al desarrollo de extensiones para dos de ellas: C-BGP y XORP. En las secciones III y IV se explican las extensiones desarrolladas y en la sección V se comenta cómo se han utilizado para validar las propuestas de multi-path BGP que se están trabajando en el proyecto Trilogy. Por último, en la sección VI se presentan las principales conclusiones y una serie de líneas de trabajo futuro.

II. ESTADO DEL ARTE

Actualmente se ha visto que las diferentes soluciones que se han propuesto para proporcionar una alternativa multi-path a BGP son validadas o bien únicamente de forma teórica o bien en base a desarrollos específicos para la propuesta en cuestión que impide la comparación entre las diversas alternativas.

Para desarrollar un entorno flexible que permita evaluar diferentes facetas de las soluciones que se estén considerando, se optó por un lado por utilizar técnicas de simulación, que permitiesen valorar de forma sencilla los resultados obtenidos tras la convergencia (ver sección V) sin tener en cuenta los efectos más complejos derivados de la evolución temporal del protocolo. El objetivo de este entorno es poder integrar de forma sencilla las propuestas que se quieran comparar admitiendo incluso configurar diferentes para una red dada cada router con una solución distinta.

De forma complementaria se ha habilitado también una implementación real para permitir lo mismo que la opción de la simulación para poder examinar el detalle de la evolución del protocolo. Por último, se ha recurrido también a técnicas de virtualización con el doble objetivo de simplificar por un lado los experimentos de laboratorio evitando la configuración de equipos físicos y por otro aumentando el tamaño de la red que se puede utilizar en la validación.

En esta sección se describen las diferentes herramientas que se han considerado como alternativas a las opciones finalmente utilizadas.

A. Simulación

Puesto que el objetivo de este enfoque es el de ampliar las funcionalidades de simuladores existentes para darles soporte multi-path, se evaluaron dos de los principales simuladores de código abierto con soporte para BGP: C-BGP [9] y NS-2/BGP++ [10].

C-BGP está especializado en simular el proceso de decisión de BGP basándose en la configuración de cada router, las rutas externas de BGP recibidas y la topología de la red.

El objetivo del simulador es ser usado como una herramienta de investigación para experimentar con procesos de decisión modificados y atributos adicionales de encaminamiento en BGP. También puede ser utilizado por el administrador de un ISP para evaluar el posible impacto de cambios lógicos y topológicos de las tablas de rutas calculadas en sus routers físicos. Los cambios topológicos incluyen caídas de enlaces y de routers. Los fallos lógicos contemplan modificaciones en la configuración de los routers, tales como las políticas de entrada y salida de tráfico o los pesos de los enlaces IGP. Gracias a su eficiencia, C-BGP puede ser utilizado en topologías muy grandes, del mismo tamaño en orden de magnitud que Internet.

Está programado en C y principalmente es utilizado y probado en máquinas con GNU/Linux y MacOS. También se puede utilizar en otras plataformas como FreeBSD, Solaris y Windows.

Por otro lado, BGP++ es una implementación en C++ de BGP para los simuladores de red NS-2 [11] y GTNetS [12]. BGP++ es una modificación del paquete software Zebra BGPd [13] para trabajar con los simuladores citados. BGP++ intenta mantener la mayor parte de la funcionalidad de Zebra BGPd mientras que incorpora dichas características en un potente entorno de simulación. La ventaja de este enfoque es que ahorra esfuerzo de desarrollo ya que los algoritmos no son reescritos y ya han sido validados.

Una característica muy útil de BGP++ es que mantiene la sintaxis de configuración que se utiliza en Zebra BGPd para configurar los routers.

Al igual que en el caso de C-BGP, BGP++ principalmente se utiliza en entornos Linux, pero puede ser compilado para Windows utilizando Cygwin.

Ambas alternativas se han comparado de cara a seleccionar la más adecuada para los objetivos propuestos.

En lo concerniente al tiempo necesario para obtener los resultados de la simulación, en escenarios idénticos las simulaciones tienen una duración menor en C-BGP. Además, el consumo de memoria de C-BGP es significativamente menor. Sin embargo, cabe destacar que el simulador de red C-BGP no permite simular aspectos relacionados con la dinámica de BGP, ya que el modelo de encaminamiento que emplea no simula el envío de mensajes BGP sobre conexiones TCP. Tampoco contempla el establecimiento de sesión ni temporizadores [14], al contrario que BGP++, el cual corre sobre un simulador a nivel de paquetes como es NS-2 y sí los contempla

En una primera revisión del código fuente de ambos se apreció que podría resultar mucho más sencillo modificar C-BGP que BGP++, entre otros motivos porque el último es dependiente del simulador NS-2 (el cual debe ser modificado para su instalación) y además de modificar el protocolo implementado en BGP++, habría que modificar los vínculos con NS-2. Se observó también que C-BGP implementa una serie de métodos para JNI (*Java Native Interface*), esto es, permite que aplicaciones desarrolladas en lenguaje Java puedan utilizar las funcionalidades proporcionadas por el simulador aun estando escritas en diferente lenguaje de programación.

Con respecto a la escalabilidad, C-BGP es capaz de simular topologías del tamaño de Internet con hardware limitado [15], mientras que con NS-2/BGP++ se requiere un entorno con múltiples máquinas trabajando en paralelo de forma distribuida, esto es, requiere un mayor número de recursos para obtener resultados de convergencia.

En conclusión, el entorno de simulación elegido sobre el que realizar las diferentes modificaciones para dar soporte multi-path fue C-BGP, principalmente debido a su eficiencia y bajo consumo de recursos.

B. Emulación real

La utilización de herramientas como C-BGP es realmente útil para obtener resultados preliminares sobre la convergencia y las tablas de rutas generadas por el protocolo bajo análisis con relativa rapidez. Sin embargo tal y como se apuntó en la sección anterior C-BGP prescinde de los aspectos relacionados con la dinámica del protocolo como por ejemplo el orden de ocurrencia de eventos.

Aunque otros simuladores (como NS-2/BGP++) permiten el estudio de la dinámica de los protocolos, para completar el entorno de simulación se ha preferido optar por la implementación real que permita incluso el despliegue de las soluciones en equipos reales.

En esta sección se tratarán dos implementaciones *open source* del protocolo BGP. El objetivo es analizar las posibilidades de modificar software BGP añadiendo soporte multi-path para su posterior utilización sobre un *testbed* real (o virtualizado). En concreto se presentarán en esta sección los paquetes de software para routing Zebra [13] y XORP [16].

1. GNU Zebra – routing software

El paquete de routing Zebra contiene una estructura modular que permite lanzar varios procesos de routing simultáneamente. En otras palabras, Zebra permite ejecutar diferentes protocolos de routing en la misma máquina de manera independiente. Además de las ventajas que esto proporciona a los administradores de la red a la hora de actualizar y configurar el router, puede ser muy interesante de cara a analizar la interacción de las soluciones multi-path BGP con otro protocolos, como por ejemplo los de routing intra-dominio (RIP, OSPF, etc.), sin necesidad de instalar software adicional.

Zebra está disponible para Linux y la mayoría de plataformas BSD. También es posible disponer de Zebra en sistemas Linux embebidos como OpenWRT. Los protocolos soportados por Zebra para IPv4 son los siguientes: RIPv1, RIPv2, RIPng, OSPF, IGMP y BGP4+.

La eficiencia en la implementación de Zebra es una de sus principales ventajas, el consumo de RAM del proceso de BGP de Zebra está entorno a 20Mb. Entre las limitaciones que encontramos para su utilización en nuestros experimentos con multi-path BGP están la escasa documentación para desarrolladores con la que cuenta el proyecto y el uso del plano de *forwarding* del *kernel* del sistema operativo, lo que en caso de querer mantener dos o más entradas por prefijo en la tabla de rutas nos forzaría a parchear el kernel del sistema operativo.

2. XORP

El proyecto de routing XORP es probablemente el paquete software más modular, flexible y completo para crear un router existente a día de hoy. Tanto es así que ha sido adoptado por fabricantes de routers de bajo coste como Vyatta [17] como solución software para sus productos.

Además de la modularidad de su diseño y su disponibilidad para los sistemas operativos más populares (Linux, BSD, Mac OSX y Windows Server 2003), cuenta con una extensa documentación para desarrolladores y una API abierta la cual puede facilitar la implementación de una solución de routing multi-path.

XORP soporta los siguientes protocolos de routing: RIPv1, RIPv2, RIPng, OSPF, IGMP, BGP, MLD, PIM-SM y MIBs para SNMP.

Otra de las ventajas de XORP es su integración con CLICK [18] lo que permite el diseño de un plano de *forwarding* para soporte de multiples entradas por prefijo más sencillo sin necesidad de modificar el *kernel* directamente. Además de cara a contar con más interfaces de red, XORP soporta la creación de routers distribuidos, esto es, un equipo ejecuta los procesos de routing y el resultado de los mismos se instala en varias tablas de rutas, cada una de ellas en un equipo diferente tal y como se muestra en la Figura 1.

Entre las desventajas de XORP se encuentra su falta de eficiencia en términos de consumo de memoria. A diferencia de Zebra que utilizaba entorno a 20Mb, XORP consume unos 100Mb (a lo que hay que añadir el tamaño de la RIB que dependerá de la topología y los prefijos anunciados). El motivo principal es la cantidad de redundancia añadida (ej. cachés de rutas, duplicados de la RIBs, etc.) en el software con el fin de hacerlo más robusto.

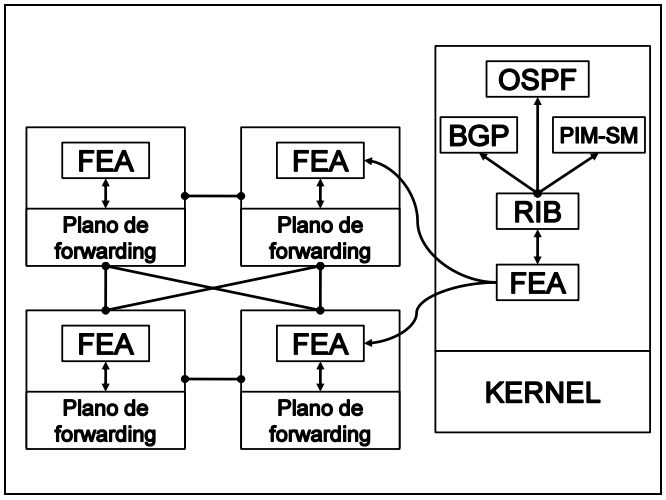

Figura 1. Plano de *forwarding* distribuido en varios equipos.

C. Virtualización

En el apartado anterior presentamos implementaciones de BGP fácilmente extensibles a un escenario multi-path. Si se pretende utilizar estas implementaciones para la verificación de dichas extensiones multi-path va a resultar imprescindible el despliegue de una maqueta de red (*testbed*).

A la hora de verificar los protocolos sobre topologías de cierto tamaño, la necesidad de un número elevado de equipos puede suponer una limitación tanto en términos de coste económico como en términos de la complejidad de su instalación y monitorización. Afortunadamente este problema también se da en los servidores de aplicaciones de Internet lo que ha dado lugar a la proliferación de herramientas de virtualización que permite la configuración y emulación de múltiples máquinas (virtuales) sobre un mismo equipo (físico).

Los principales requisitos que se buscan en una solución de virtualización son: (1) eficiencia y baja sobrecarga de CPU, (2) soporte para virtualizar redes, (3) máximo número de interfaces de red por cada máquina virtual, (4) posibilidad de utilizar imágenes de máquinas virtuales con OpenWRT+Zebra [19] y GNU/Linux+XORP. El uso de dichas imágenes puede facilitar la migración de las modificaciones multi-path de BGP a routers físicos reales como los Linksys WRT54G [20] y los ya citados routers Vyatta.

III. MC-BGP

Para dotar al simulador de red C-BGP de soporte multi-path es necesario realizar una serie de cambios en su arquitectura. Dichos cambios comprenden tanto las estructuras de datos internas que definen los routers BGP como las funciones empleadas para mostrar información y ejecutar el proceso de selección de rutas.

Además de dotar al simulador de soporte multi-path, se han añadido funcionalidades extra, modificando la gramática de comandos empleada por el mismo. Dichas funcionalidades resultan muy útiles a la hora de obtener resultados necesarios en las métricas para la evaluación de las modificaciones sobre BGP. Las funcionalidades añadidas son:

- Mostrar y asignar el tipo de protocolo multi-path a emplear por un determinado router BGP, permitiendo que cada router utilice incluso un protocolo diferente. Esto permite por ejemplo validar el impacto de la introducción de la solución únicamente en una parte de la red (en el núcleo por ejemplo) o la compatibilidad entre soluciones.
- Mostrar la *FIB* (*Forwarding Information Base*) almacenada por un determinado router BGP para poder analizar el detalle de la evolución del protocolo.
- Mostrar el número de bucles detectados al ejecutar el protocolo multi-path en un determinado router BGP para evaluar la convergencia.
- Mostrar y asignar el máximo nivel de agregación de rutas en un determinado router BGP.
- Mostrar el número de mensajes enviado entre los distintos routers hasta que converge. Esto será posteriormente considerado como una métrica más de comparación (ver sección V).

A. Soporte multi-path

El simulador de red C-BGP está compuesto por tres capas principales: planificador, simulación IP y simulación BGP. La capa del planificador es la parte central del simulador. Contiene la secuencia de eventos pendientes que representan los mensajes a ser enviados hacia determinados nodos de la red. El planificador mete en una cola los nuevos eventos cuando un nodo envía un mensaje a otro. Esos mensajes son posteriormente extraídos de la cola y enviados al nodo correspondiente.

El primer componente de la capa de simulación es una representación de la capa IP de la topología de red modelada. Esto es básicamente una estructura de datos que mantiene un grafo de nodos y enlaces. El segundo componente es un modelo estático IGP. Este modelo es responsable de calcular las rutas intra-dominio para cada dominio IGP, basándose en el conocimiento de la topología al completo. Las rutas intra-dominio se almacenan en una tabla de rutas junto con las rutas estáticas (configuradas manualmente). Por último, el tercer componente de la capa de simulación IP es el modelo de router IP el cual es responsable de enviar mensajes a la capa BGP si el mensaje tiene destino local o de reenviar el mensaje a otro nodo si el mensaje debe ser entregado a un destino remoto.

Finalmente, la capa de simulación BGP también contiene una serie de componentes. El primer componente es la configuración de la capa BGP. Esto incluye el grafo de las sesiones BGP al igual que la configuración de los distintos routers BGP en la red modelada. El segundo componente de la capa de simulación BGP es el modelo de routing BGP. Este modelo contiene el proceso de decisión y los filtros de routing. El modelo de routing BGP depende de una serie de tablas de rutas las cuales contienen las rutas BGP conocidas por cada router BGP y las mejores rutas seleccionadas por cada uno de ellos.

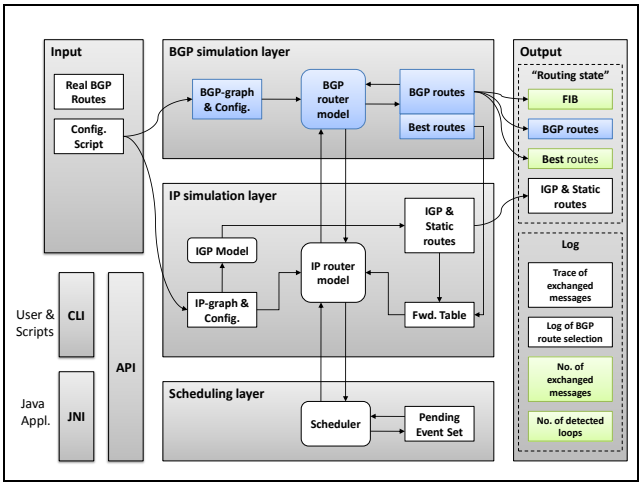

Figura 2. Arquitectura C-BGP

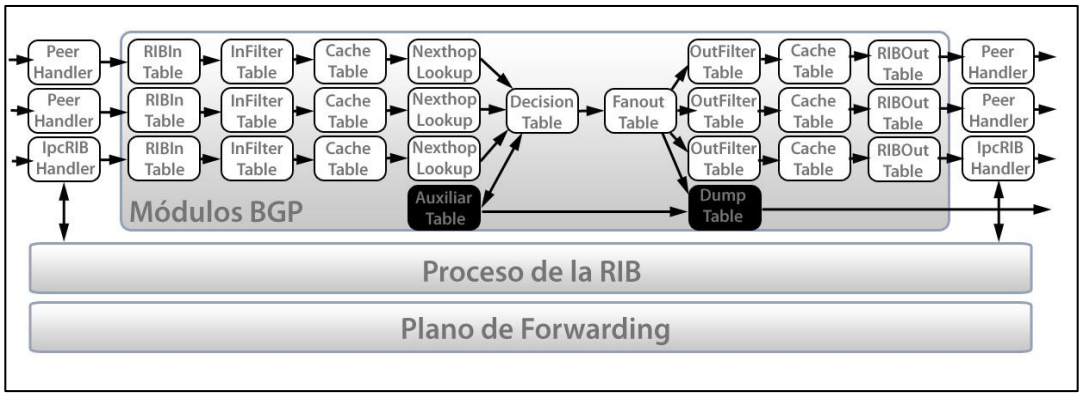

Figura 3. Módulos del proceso BGP de XORP.

En la Figura 2 se pueden apreciar los cambios necesarios sobre la arquitectura de C-BGP para dotarlo de soporte multi-path. Fue necesario ampliar el módulo "*BGP-graph & Config.*" para incluir las opciones relativas a los nuevos parámetros de configuración (tipo de variante de BGP a utilizar, máxima agregación). También se ampliaron los módulos correspondientes al modelo del router BGP ("*BGP Router Model*") para añadir los procesos de decisión alternativos y las estructuras de datos necesarias para almacenar información como los bucles detectados y el módulo correspondiente a las rutas BGP ("*BGP Routes*"), siendo necesario en este caso añadir las estructuras correspondientes a la FIB y a los AS_SETs (estructura utilizada en algunas soluciones y no soportada inicialmente por el simulador) así como las funciones necesarias para manejarlos.

Además, se observó que el simulador carecía de métodos para gestionar los AS_SETs en los AS_PATHs, de forma que se añadieron funciones encargadas de ello, y se modificaron los métodos encargados de mostrar las rutas en la salida, ya que en algunos casos podría ser necesario incluir la información contenida en dichos AS_SETs para el AS_PATH. El soporte de AS_SETs se proporcionó a través de una estructura dinámica de alto rendimiento (array dinámico de enteros con inserción y acceso basado en búsqueda binaria).

Igualmente se implementaron nuevas funciones correspondientes a los módulos "*FIB*", "*Best routes*", "*No. Of Exchanged Messages*" y "*No. of detected loops*" con el fin de mostrar la información relativa a dichos módulos por la salida del simulador.

IV. MXORP

En esta sección se detallan las modificaciones que se pueden realizar en el proceso BGP de XORP para implementar protocolos multi-path. Para comprender las modificaciones realizadas nos referiremos a la estructura del proceso BGP descrita en la Figura 3.

A. El Proceso BGP Estándar

Cada *PeerHandler* representa una sesión BGP sobre TCP. Cada vez que se recibe un UPDATE de BGP, se desglosa en uno o varios mensajes internos ADD_ROUTE, DELETE_ROUTE y REPLACE_ROUTE. Por ejemplo, cuando un *peer* anuncia que se puede alcanzar un nuevo prefijo a través de él a este router, un mensaje ADD_ROUTE atraviesa la cadena de componentes de la rama de entrada a la que está conectado el *PeerHandler*. El mensaje se va modificando a lo largo de la rama, hasta llegar al *DecisionTable* (o ser filtrado) donde la ruta que contiene se somete al proceso de decisión del protocolo BGP. Si la ruta resulta ganadora, se generarán dos mensajes que atravesarán todas las ramas de salida, un DELETE_ROUTE para la antigua ruta ganadora/anunciada y un ADD_ROUTE para la nueva que se traducirá a un UPDATE de BGP al resto de *peers* de este router. Existe una rama especial que no conecta con otro *peer* sino con el proceso XORP que se encarga de popular la decisión en la RIB. En algunos protocolos (ver [1]), la información que atraviesa esta rama puede ser diferente de la que se propaga por el resto (hacia los *peers*).

El módulo *DecisionTable* a la hora de decidir la mejor ruta para un determinado prefijo de red, obtiene todas las rutas anunciadas para ese prefijo, consultando la *RibInTable* de cada una de las ramas de entrada y aplicando el proceso de decisión a ese conjunto de rutas. La ruta seleccionada para ser anunciada al resto de *peers* y ser instalada en la FIB se pasa al módulo *FanoutTable*. Este módulo duplica los mensajes internos resultados de la decisión en cada una de las ramas, así se consigue realizar cambios en el estado del proceso una vez y éstos son propagados independientemente para cada *peer*.

Además del anuncio o retirada de rutas para un determinado prefijo, otro evento bastante común es el establecimiento de una nueva sesión BGP con un *peer* que acaba de arrancar o recuperarse de una caída. Para ese caso particular XORP define un nuevo módulo (*DumpTable*) que se encarga de volcar la información de routing de este router al nuevo *peer* para que pueda construir su RIB de manera consistente. El volcado se realiza en segundo plano y utiliza un complejo proceso de sincronización puesto que nuevos eventos de rutas pueden ocurrir en mitad del volcado y dejar al nuevo *peer* con información inconsistente o desactualizada. Para obtener esta sincronización, el módulo *DumpTable* no obtiene la información a volcar de la RIB directamente, sino que procesa una por una las *RibInTable* de cada rama y envía la nuevo *peer* las rutas que fueron marcadas como propagadas (o ganadoras en terminología XORP) la última vez que el proceso de decisión BGP se ejecutó para un prefijo en concreto.

B. Modificaciones en los bloques para soportar multi-path

Tras introducir el funcionamiento del proceso BGP de XORP, se pueden identificar varios módulos clave para posibles extensiones multi-path. Tanto el filtrado de los mensajes internos como la caché se pueden configurar y deshabilitar mediante los ficheros de configuración y salvo que se necesiten añadir filtros más complejos que los existentes para BGP estándar, estos módulos no deberían de ser modificados para soportar multi-path. El módulo *Next-HopLookup* simplemente se encarga de verificar que exíste un siguiente salto para un prefijo determinado, por lo que tampoco necesita ser modificado.

Todos los módulos excepto el *PeerHandler* implementan una interfaz llamada *RouteTableBase,* la cual permite que los módulos intercambien mensajes internos y soliciten o notifiquen información sobre rutas entre ellos. La principal limitación de la definición de esta interfaz es que sólo permite operaciones sobre una ruta por prefijo en cada llamada entre módulos. Dependiendo de las necesidades de cada implementación se puede elegir entre modificar los mensajes internos para que encapsulen información sobre varias rutas de manera simultánea o sobrecargar la interfaz para que soporte operaciones que utilicen o devuelvan como resultado un set de rutas en lugar de una única ruta. Esta última opción, a pesar de que necesita reescribir todos los módulos que implementan *RouteTableBase* es la más amigable con el procesado que internamente realiza cada módulo, ya que se pueden implementar las funciones multi-path basándose en múltiples llamadas a las funciones originales.

1. Protocolos que preservan mensajes BGP

Una vez modificada la interfaz, para extensiones multi-path en las cuales cada *peer* anuncia única ruta por prefijo, las modificaciones se centran principalmente en (1) el *DecisionTable*, donde el proceso de selección de rutas puede ser modificado para marcar varias rutas como *ganadoras* del proceso. (2) En el *FanoutTable*, ya que en este tipo de modificaciones, sólo una de las rutas resultado del proceso de selección es propagada a través de las ramas que finalizan en un módulo *PeerHandler* para ser anunciada. El resto de rutas se almacenarán en la RIB, por lo que se propagarán por la rama acabada en el *IpcRIBHandler* que comunica con el proceso de control de la RIB.

2. Protocolos con múltiples rutas por anuncio

Si por el contrario, el protocolo que queremos evaluar soporta que cada *peer* pueda anunciar múltiples rutas para un mismo destino (BGP no soporta esta opción) más cambios deben realizarse, especialmente en lo que concierne al *RibInTable* y al *RibOutTable*. En estos módulos toda la información enviada desde/hacia un *peer* queda registrada en una estructura de datos. Si múltiples rutas para un mismo prefijo son anunciadas por un mismo *peer*, los módulos actuales no necesitan ninguna estructura extra de datos (a pesar de que en BGP cada anuncio reemplaza al anterior). El problema viene a la hora de obtener información acerca de un determinado prefijo desde otros módulos en la *RibInTable* o pasar múltiples rutas desde la *RibOutTable* al *PeerHandler*.

Por ejemplo, la función correspondiente de la *RibInTable* obtiene todas las registradas en la estructura de datos cuando se solicita una búsqueda, pero sólo devuelve la marcada como *en uso*. Al sobrecargar la interfaz *RouteTableBase* este problema se soluciona, pasando al siguiente módulo un set de rutas marcadas como *en uso*. En el caso de la *RibOutTable*, si múltiples rutas con el mismo prefijo llegan al módulo mientras el *PeerHandler* está actualmente enviando, el módulo pondrá en la cola de espera para ser anunciada a la ruta más reciente. Esto debe modificarse para que se añadan a la cola múltiples rutas para un prefijo. Como construir los mensajes que intercambian los *peers* es tarea del *PeerHandler* y queda abierto a la definición del protocolo.

3. Protocolos que alteran las rutas recibidas

Si el protocolo crea la(s) ruta(s) a propagar a partir de las que reciba (por ejemplo agregando AS_NUMBERS en un AS_SET, ver [1]), se necesita un módulo adicional que extienda de *RibInTable* para almacenar esta(s) nueva(s) ruta(s) creadas en el proceso de selección (ver *AuxiliarTable* en la Figura 3). Si esto sucede, el procesado del módulo *DumpTable* se simplifica puesto que todas las rutas a volcar al nuevo *peer* se encuentran almacenadas en este módulo adicional.

4. Cambios en la RIB y FIB

El proceso que controla la RIB debe ser modificado para almacenar múltiples rutas por prefijo, la RIB debe contener el set de rutas *válidas* obtenido en el proceso de decisión, y éste puede ser un super-set o un sub-set de las rutas propagadas a los *peers*. Posteriormente, ese conjunto de rutas se instalarán en el plano de *forwarding* para su uso. En la siguiente sección se apuntan algunas alternativas compatibles con XORP para crear una FIB con múltiples entradas.

V. VALIDACIÓN

La validación de los entornos propuestos para verificación de multi-path BGP se realizó utilizando versiones modificadas de C-BGP y XORP. Para la ejecución de los routers XORP se utilizó el sistema de virtualización XenServer [23] por cumplir con todos los requisitos de la sección 2.C y basándonos en la comparativa entre sistemas de virtualización publicada en [24]. Además el proceso que controla la RIB y la FIB en XORP se ha modificado para hacer uso de la librería IPROUTE2 [25] para crear una FIB multi-entrada. La utilización de CLICK para este fin también hubiera sido posible y queda como trabajo futuro.

Para ayudar a evaluar los resultados de las simulaciones / emulaciones se desarrolló una herramienta software llamada "*StatOpology*". Es una aplicación escrita en lenguaje Java, cuyo propósito inicial consistía en convertir diversos formatos de topologías de red al formato admitido por C-BGP. A esta aplicación se le han ido añadiendo funcionalidades, entre ellas se encuentra la de proporcionar información sobre el número de sistemas autónomos de la topología, mostrar los nodos hoja, los Tier-1, el número medio de enlaces y de qué tipo (*peering* o *provider to client*), generar una imagen (tanto vectorial como *raster*) de la topología e incluso cargar la

información correspondiente a las rutas instaladas en los distintos routers BGP tras las simulaciones / emulaciones con el fin de analizar en una tabla datos como son la longitud de los caminos, los nodos involucrados en los caminos, el número de bucles detectados, etc.

Los datos procesados por *StatOpology* pueden ser guardados en un fichero para ser evaluados posteriormente sin tener que analizar de nuevo los resultados de las simulaciones / emulaciones, o las topologías.

Las propuestas multi-path que se han validado son las mencionadas en la introducción (LP-BGP y MpASS) tanto en simulación como en emulación. La idea tras LP-BGP es aplicar una serie de reglas de filtrado sobre el conjunto de rutas que el router ha recibido de sus *peers*. El conjunto resultante de eliminar las rutas durante el filtrado es el set multi-path que el router puede utilizar para hacer el *forwarding* de paquetes. Se puede demostrar que de acuerdo a las *Loop-Free Invariants* introducidas en [5], si de ese conjunto se propaga a los *peers* la de mayor AS_PATH_LENGTH, se garantiza que el resto de rutas están libres de bucles.

Con el fin de comprobar que las modificaciones multi-path realizadas al simulador (mC-BGP) funcionaran adecuadamente se decidió añadir soporte para las dos variantes de BGP multi-path citadas en la introducción: LP-BGP y MpASS. Para realizar la implementación de multi-path utilizando la propagación del camino más largo, fue necesario añadir un proceso de decisión de reglas específico a los cambios anteriormente mencionados, además de una función encargada de determinar si se aplica o no en base al protocolo seleccionado en el router. Estos cambios se aplicaron en las secciones correspondientes a "*BGP-Graph & Config*" y "*BGP Router Model*" de la Figura 2.

En el caso de MpASS, la idea es aplicar la agregación de rutas que se hace entre prefijos más y menos restrictivos a múltiples rutas para un mismo prefijo. La idea es aplicar también un filtrado al set de posibles rutas candidatas para descartar rutas de baja calidad o problemáticas. Sobre las restantes, se aplica el mismo proceso de decisión que en BGP estándar para determinar la mejor de las rutas (en general la de menor AS_PATH_LENGTH). A continuación, la ruta ganadora, se le añade un AS_SET. El contenido de ese AS_SET son todos los AS_NUMBERS del resto de rutas candidatas (ya filtradas) que no formen parte del AS_PATH de la ganadora más el AS_NUMBER local del router. Esta ruta agregada es la que se propaga al resto de routers y el set de rutas candidatas pasa a la RIB del router local.

Para realizar la implementación de multi-path basada en el uso de AS_SETs, además de modificar las funciones del proceso de decisión de BGP también fue necesario hacer uso de las nuevas funciones encargadas de gestionar los AS_SETs. El resto de cambios es semejante a LP-BGP.

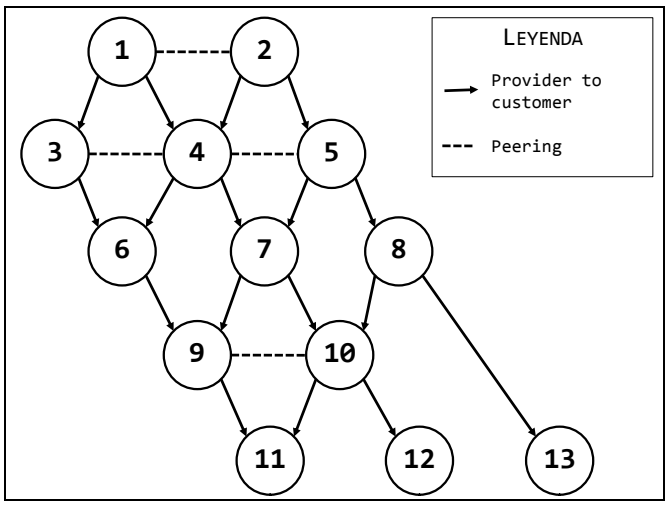

Figura 4. Topología de prueba

VI. CONCLUSIONES

El objetivo principal de la propuesta es poder comparar diferentes soluciones multi-path BGP de tal forma que sea posible no ya solo extraer automáticamente una serie de métricas para cada una objetivando así la comparación, sino que sea posible también combinarlas y evaluar su funcionamiento conjunto.

Para ello se ha combinado por un lado la sencillez de la simulación para valorar rápidamente aspectos como la convergencia así como la implementación real para evaluar la evolución en el tiempo del protocolo.

Todo esto se ha combinado además con herramientas de virtualización para posibilitar las pruebas en topologías con un elevado número de nodos.

Para validar la propuesta se han implementado tanto para el simulador como para el emulador dos soluciones concretas que se están desarrollando en el proyecto Trilogy.

Como parte del trabajo que se sigue desarrollando en esta línea en este proyecto, está previsto una mayor exploración de las posibilidades de esta solución utilizando topologías de un mayor tamaño (las primeras pruebas realizadas con la arquitectura publicada en [22] son muy optimistas pero el proceso de simulación por ejemplo consume muchos recursos y se considera que puede todavía optimizarse más).

VII. AGRADECIMIENTOS

Este artículo ha sido parcialmente financiado por la Comisión Europa a través del proyecto Trilogy (ICT-216372), del VII Programa Marco y por la Cátedra Telefónica-UC3M en Internet del Futuro para la Productividad.

REFERENCIAS

[1] F. Valera, I. van Beijnum, A. García-Martínez, M. Bagnulo. "Next Generation Internet Architectures and Protocols", Ed., B. Ramamurthy, G. Rouskas, and K. Sivalingam, Cambridge University Press, 2010. ISBN: 978052111368

[2] Marcelo Bagnulo, Louise Burness, Philip Eardley, Alberto García-Martínez, Francisco Valera and Rolf Winter. "Joint Multi-path Routing and Accountable Congestion Control".ICT-MobileSummit 2009. June 2009, Santander, Spain

[3] Psenak P., Mirtorabi S., Roy A., Nguyen L., Pillay-Esnault P. “Multi-Topology (MT) Routing in OSPF”.RFC4915.(2007).
[4] Albrightson B., Garcia-Luna-Aceves J., Boyle J. “EIGRP-A fastrouting protocol based on distance vectors”. In Proc.Networld/Interop 94,Las Vegas. (1994). Proceedings. 136-147.
[5] S. Vutukury and J.J. Garcia-Luna-Aceves, “MPATH: A Loop-free Multi-path Routing Algorithm”. ElsevierJournal of Microprocessors and Microsystems, 2000.
[6] Proyecto Trilogy (ICT-2007-216372). “Architecting the Future Internet”. Disponible [Internet]: <http://trilogy-project.org/> [21 de enero de 2011]
[7] A. Ford, C. Raiciu, M. Handley. “TCP Extensions for Multi-path Operation with Multiple Addresses”.IETF draft. Disponible [Internet] <http://tools.ietf.org/html/draft-ford-mptcp-multiaddressed-03> [21 de enero de 2011]
[8] Iljitsch van Beijnum, Jon Crowcroft, Francisco Valera and Marcelo Bagnulo. “Loop-freeness in multi-path BGP through propagating the longest path”.International Workshop on the Network of the Future (Fut-Net 2009).June 2009, Dresden, Germany
[9] Página web oficial del simulador de red C-BGP. Disponible [Internet]: <http://cbgp.info.ucl.ac.be/index.php> [21 de enero de 2011]
[10] Página web oficial del módulo BGP++. Disponible [Internet]: <http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/> [21 de enero de 2011]
[11] Página web oficial del simulador de red NS-2 . Disponible [Internet]: <http://www.isi.edu/nsnam/ns/> [21 de enero de 2011]
[12] Página web oficial del simulador de red GTNetS. Disponible [Internet]: <http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/> [21 de enero de 2011]
[13] Página web oficial del proyecto GNU Zebra. Disponible [Internet]: <http://www.zebra.org/>[21 de enero de 2011]
[14] Arquitectura del simulador C-BGP. Disponible [Internet]: <http://cbgp.info.ucl.ac.be/architecture.php>[21 de enero de 2011]
[15] W. Mühlbauer, A. Feldmann, O. Maennel, M. Roughan, S. Uhlig. “Building an AS-Topology Model that Captures Route Diversity”. ACM SIGCOMM, 2006. Disponible [Internet]: <http://www2.net.in.tum.de/~muehlbaw/sigcomm06.pdf/> [21 de enero de 2011]
[16] Página web oficial del router software XORP. Disponible [Internet]: <http://www.xorp.org/> [21 de enero de 2011]
[17] Página web oficial de Vyatta. Disponible [Internet]: <http://www.vyatta.com/> [21 de enero de 2011]
[18] The Click Modular Router Project. Disponible [Internet]: <http://read.cs.ucla.edu/click/ > [21 de enero de 2011]
[19] Documentación online de OpenWRT, Disponible [Internet]: <http://kamikaze.openwrt.org/docs/openwrt.html> [21 de enero de 2011]
[20] Página web con características hardware de distintos routers existentes en el mercado (wiki de OpenWRT). Disponible [Internet]: <http://oldwiki.openwrt.org/Hardware%282f%29Linksys.html> [21 de enero de 2011]
[21] L. Gao, J. Rexford, “Stable Internet Routing Without Global Coordination”, IEEE/ACM Transactions on networking, Vol.9, No.6, Dec. 2001
[22] Internet Topology Collection. Disponible [Internet]: <http://irl.cs.ucla.edu/topology/> [21 de enero de 2011]
[23] Citrix Xen Server. Disonible [Internet]: <http://www.citrix.com> [21 de enero de 2011]
[24] The Tolly Group. “Test report #209103. Citrix XenServer 5: Optimized Performance for XenApp compared to VMWare ESX 3.5u3”. Abril 2009. Disonible [Internet]: <http://www.tolly.com/> [21 de enero de 2011]
[25] Linux Advanced Routing and Traffic Control. Disponible [Internet]: <http://lartc.org/> [21 de enero de 2011]