



UNIVERSIDAD CARLOS III DE MADRID
Departamento de Ingeniería Telemática



PROYECTO FIN DE CARRERA

Desarrollo de un entorno de evaluación para protocolos de encaminamiento multi-path interdominio

Autor: D. Lisardo Prieto González

Tutor: Prof. Dr. D. Francisco Valera Pintor

Titulación: Ingeniería Informática

Leganés, Mayo de 2010

Título: DESARROLLO DE UN ENTORNO DE EVALUACIÓN PARA
PROTOCOLOS DE ENCAMINAMIENTO MULTI-PATH INTERDOMINIO

Autor: D. Lisardo Prieto González

Tutor: Prof. Dr. D. Francisco Valera Pintor

TRIBUNAL

Presidente: Prof. Dr. D. Alberto García Martínez

Vocal: Prof. Dr. D. David Expósito Singh

Secretario: Prof. Dr. D. Norberto Fernández García

Agradecimientos

A ese descomunal conjunto de circunstancias que ha originado nuestra existencia [y que nos ha encauzado a la vida que llevamos].

Resumen

El objetivo de este proyecto es el desarrollo de un entorno de simulación versátil y potente en el que validar protocolos de encaminamiento *multi-path* interdominio sobre topologías de red. Estas simulaciones son muy importantes, ya que nos permiten determinar cómo se va a comportar un determinado protocolo de encaminamiento, o una variante del mismo sin necesidad de realizar un elevado desembolso económico destinado a la adquisición de hardware en el que realizar pruebas reales, o a invertir una gran cantidad de tiempo tratando de predecir matemáticamente el comportamiento de un modelo generalmente simplificado.

En primer lugar se describen las fases iniciales, empezando por la aproximación preliminar a una solución basada en el uso de equipos hardware reales, y seguidamente por una solución basada en máquinas virtuales. También se justifica por qué ambas alternativas fueron descartadas para dar lugar a una tercera solución basada en utilizar simuladores de red. Se detallan los estudios comparativos realizados entre distintas aplicaciones existentes y se justifica por qué ampliar el simulador C-BGP en vez de las otras alternativas. A partir de aquí se describen los cambios necesarios en la estructura del simulador para dar soporte *multi-path*, y la validación del mismo mediante la implementación de dos variantes de BGP *multi-path* propuestas en el proyecto Trilogy: LP-BGP y MpASS.

Debido a diversas dificultades encontradas a lo largo del desarrollo del simulador, se plantea la creación de una herramienta, *StatOpology*, esta comienza siendo una aplicación de consola para generar scripts, convertir entre formatos de ficheros de topologías y proporcionar información sobre las mismas, pero que acaba siendo una potente y versátil aplicación multiplataforma con interfaz gráfica de usuario que cumple los estándares de accesibilidad y que sigue una serie de patrones de diseño que hacen que su código fuente sea reutilizable, muy escalable y fácilmente mantenible.

Seguidamente se muestran los resultados de aplicar el simulador y las dos variantes *multi-path* (LP-BGP y MpASS) a dos pequeñas topologías de prueba y a una tercera topología

con un tamaño de una magnitud similar a la de Internet. Aquí se puede comprobar cómo desde un solo computador pueden obtenerse en un tiempo muy razonable unos resultados para los que en otro caso necesitaríamos una gran cantidad de hardware.

Tras la validación de resultados se detallan los costes asociados a los distintos medios empleados para la realización del proyecto, y se adjunta un diagrama de Gantt en el que se muestra la distribución de tiempo en las distintas fases del proyecto.

Finalmente se completa la memoria con el glosario de términos, la bibliografía y los anexos. En el primero se explican los comandos añadidos al simulador C-BGP junto con ejemplos de uso y en el segundo se incluye el manual de usuario de la herramienta **StatOpology** anteriormente mencionada.

Palabras clave: encaminamiento, *multi-path*, BGP, simulación, virtualización, versátil, económico, C-BGP, StatOpology.

Abstract

The current project details the steps taken to get a powerful and versatile environment, to simulate multi-path routing protocols on network topologies. These simulations are very important because they allow us to determine how it will behave a particular routing protocol or a variation thereof without making a major financial outlay for the purchase of hardware on which make actual testing, nor investing a lot of time trying to make a mathematical approach that predicts such behavior.

First, initial stages are described, beginning with the first approach to a solution based on the use of real hardware devices, followed by a solution based on virtual machines. Also why these solutions were dropped to make way for a solution based on using network simulators. A detailed comparative study between different existing applications is provided next, and is justified by what extend the C-BGP simulator instead of the other existing alternatives. From here we describe the necessary changes in the structure of the simulator to allow multi-path support, and validation of the same by implementing two variants of BGP multi-path proposed in Trilogy project: LP-BGP and MpASS.

Due to various difficulties encountered during the development of the simulator, we propose the creation of a tool, *StatOpology*, which starts as a console application to generate scripts, convert between topology file formats and provide information about them, but it becomes a powerful and versatile application with graphical user interface that meets accessibility standards and follows a series of design patterns that make its source code reusable, highly scalable and easily maintainable.

In the next part, results of applying the simulator with the two variants of BGP multi-path (LP-BGP and MpASS) are shown. It was tested in a couple of small test topologies and in a third topology with a size of a similar magnitude to the Internet. Reading these results you can realize that with a single computer and in a reasonable time is possible to get some results for those who otherwise would have been necessary a large amount of hardware.

After validating the results, next you can find the detailed cost for the items used in the budget for the project. A Gantt diagram is also included. It shows the distribution of time in the different phases of the project.

Finally, the document gets completed with a glossary of terms, the bibliography and the annexes. In the first one, the new commands added to C-BGP are explained with examples of use. The second one contains *StatOpology* user manual.

Keywords: routing, multi-path, BGP, simulation, virtualization, versatile, economic, C-BGP, StatOpology.

Índice general

1. INTRODUCCIÓN Y OBJETIVOS	13
1.1 Introducción	13
1.2 Objetivos	14
1.3 Estructura de la memoria	15
1.4 Medios empleados.....	16
1.4.1 Hardware	16
1.4.2 Software	17
2. ESTUDIO TECNOLÓGICO	19
2.1 Introducción	19
2.2 Equipos reales	20
2.3 Virtualización de routers	21
2.3.1 Análisis sobre diferentes sistemas de virtualización	22
2.3.2 Características de la virtualización.....	27
2.4 Simuladores de red.....	28
2.4.1 Características de los simuladores	28
2.4.2 Análisis sobre diferentes simuladores existentes	29
2.5 Conclusiones	32
3. DESARROLLO DEL ENTORNO DE EVALUACIÓN <i>MULTI-PATH</i>.....	34
3.1 Introducción	34
3.2 De C-BGP a mC-BGP.....	35
3.2.1 El router BGP	38
3.2.2 Las rutas	39
3.2.3 El proceso de decisión.....	40
3.2.4 La interfaz con el usuario	41
3.2.5 Implementación de nuevos protocolos.....	41
3.3 StatOpology.....	43
3.4 Conclusiones	48
4. VALIDACIÓN DEL ENTORNO	49
4.1 Introducción	49
4.2 StatOpology.....	50

4.3 mC-BGP	56
4.3.1 LP-BGP	56
4.3.2 MpASS	56
4.4 Métricas de evaluación para las soluciones.....	57
4.5 <i>Multi-path</i> BGP: pruebas y resultados	59
4.5.1 Simulando “Internet”. Primer intento	61
4.5.2 Simulando “Internet”. Segundo intento	62
4.5.3 Simulando “Internet”. Tercer intento (éxito)	63
4.6 Conclusiones	64
5. PRESUPUESTO	65
5.1 Diagrama de tareas	65
5.2 Desglose de costes.....	68
5.2.1 Personal	68
5.2.2 Material	70
5.2.3 Transporte	72
5.2.4 Costes indirectos	73
5.2.5 Resumen de costes	74
5.2.6 Totales	74
6. CONCLUSIONES Y TRABAJO FUTURO	76
7. GLOSARIO	78
8. REFERENCIAS.....	81
9. ENLACES DE INTERÉS.....	84
10. C-BGP - COMANDOS AÑADIDOS.....	87
11. STATOPOLOGY - MANUAL DE USUARIO	90

Índice de figuras

Fig. 1. OpenWrt en ejecución bajo VirtualBox.....	27
Fig. 2. Ejecución de <i>mC-BGP</i>	35
Fig. 3. Arquitectura de <i>mC-BGP</i>	36
Fig. 4. Diagrama de componentes de StatOpology	43
Fig. 5. Ventana principal de StatOpology	45
Fig. 6. Logo de StatOpology	46
Fig. 7. Gráfico generado para una topología de prueba	47
Fig. 8. Topología extraída de [GR01]	50
Fig. 9. Grafo generado para la topología.....	52
Fig. 10. Análisis de la topología de Internet	54
Fig. 11. Información estadística sobre la topología de Internet	55
Fig. 12. Topología empleada inicialmente para la validación de soluciones <i>multi-path</i> ..	59
Fig. 13. Primer intento de simular la topología de Internet	61
Fig. 14. Segundo intento de simular la topología de Internet. Inyección progresiva de prefijos.....	62
Fig. 15. Simulación de la topología de Internet. Sub-simulaciones de prefijos en la topología completa	63
Fig. 16. Desglose de tareas llevadas a cabo y su duración en días.....	66
Fig. 17. Diagrama de Gantt con las distintas fases del proyecto.....	67
Fig. 18. Manual de StatOpology	91

Índice de tablas

Tabla 1. Características técnicas del router WRT54G	21
Tabla 2. Características técnicas de la máquina virtual con OpenWRT	22
Tabla 3. Prestaciones del equipo sobremesa	23
Tabla 4. Prestaciones del equipo portátil	23
Tabla 5. Virtualización con VMware Server 1.0.6	23
Tabla 6. Virtualización con VirtualBox 2.0	24
Tabla 7. Virtualización con QEMU 0.9.1	25
Tabla 8. Virtualización con VNUML 1.8.3	26
Tabla 9. Comparativa entre soluciones de virtualización	32
Tabla 10. Comparativa entre simuladores de red	33
Tabla 11. Distribución del código fuente de C-BGP	42
Tabla 12. Leyenda para los grafos generados con StatOpology	46
Tabla 13. Topología de red en formato interpretable por <i>mC-BGP</i>	51
Tabla 14. Información sobre la topología	51
Tabla 15. Código fuente empleado por <i>GraphViz</i> para generar un grafo	52
Tabla 16. Script para simular la topología en <i>mC-BGP</i>	53
Tabla 17. Métricas de evaluación para las soluciones	58
Tabla 18. Estimación de coste por cada tipo de trabajo / hora	68
Tabla 19. Coste asociado a las actividades y recursos empleados	69
Tabla 20. Coste asociado al equipamiento hardware empleado	70
Tabla 21. Coste asociado al software empleado	71
Tabla 22. Coste asociado al material fungible	72
Tabla 23. Coste asociado al transporte durante el proyecto	72
Tabla 24. Costes indirectos	73
Tabla 25. Resumen de costes asociados al proyecto	74
Tabla 26. Presupuesto del PFC	74

Capítulo 1

Introducción y objetivos

*"Que la fuerza nos acompañe."
Almirante Ackbar. (Star Wars: Ep. VI)*

1.1 Introducción

Los protocolos de **encaminamiento** son de vital importancia en las redes de comunicaciones. Especialmente en Internet. Sin embargo, realizar modificaciones en dichos protocolos o desarrollar otros nuevos y determinar cómo se van a comportar en un entorno real resulta extremadamente complicado a nivel **matemático/teórico**.

Por otra parte, los protocolos de encaminamiento *multi-path* presentan una serie de ventajas muy importantes respecto a los protocolos convencionales, las cuales hacen que estén recibiendo cada vez más atención. Algunas de ellas son (ver [VBG10+] y [BBE09+]):

- Aumento de la seguridad de la red derivados del reparto del tráfico entre diferentes enlaces, proporcionando protección frente a ataques de denegación de servicio o de inspección de paquetes.
- Incremento efectivo de la capacidad de la red ya que el tráfico puede enviarse por un mayor número de enlaces.
- Respuesta más rápida ante cambios en la red, puesto que se han explorado y probado diferentes caminos alternativos.
- Ingeniería de tráfico escalable. Se pueden optimizar los retardos de la red mediante una configuración adecuada de los caminos utilizados.

El creciente interés por las ventajas ofrecidas por las alternativas *multi-path* ha derivado en una multitud de propuestas que pretenden dotar de dicha versatilidad al protocolo BGP (ver [VBG10+] y [BBE09+]).

El proyecto Trilogy (*ICT-2007-216372, Architecting The Future Internet*, [TRLGY]) tiene como principal objetivo el desarrollo de nuevas soluciones para la arquitectura de control de internet (mecanismos de encaminamiento, control de recursos y control social y comercial). En dicho proyecto se está prestando una especial atención a diferentes soluciones *multi-path*, como la propuesta en la capa de transporte, mTCP [FRH10] o diferentes extensiones para *multi-path* BGP (como LP-BGP [BCV09+] y MpASS [VBG10+]).

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 compararas con otras alternativas en contextos reales, tanto a nivel de tamaño de la red como a nivel de relaciones reales entre proveedores. Por lo general, cada propuesta emplea un mecanismo diferente de validación (teórico, mediante desarrollo de software específico para la solución, etc.) lo cual complica considerablemente la comparación entre ellos o con cualquier otro mecanismo que se pueda proponer.

En este proyecto, desarrollado bajo el marco de la *Cátedra Telefónica-UC3M en Internet del Futuro para la Productividad*, se documentan los pasos y decisiones tomadas para realizar un entorno de verificación y evaluación de soluciones *multi-path* BGP, mediante una herramienta de análisis (StatOpology) y un simulador de red modificado (C-BGP), los cuales han permitido, por ejemplo, probar las dos soluciones planteadas en Trilogy (LP-BGP y MpASS), y detectar problemas de convergencia de dichas soluciones. El entorno es fácilmente extensible y adaptable a las diversas soluciones que se quieran comparar.

1.2 Objetivos

El objetivo de este proyecto es el de obtener un entorno que nos permita, de una forma sencilla y amigable, realizar pruebas y simulaciones sobre soluciones *multi-path* BGP, con el fin de verificar su funcionamiento y evaluar cómo se comportarán en entornos reales. Todo ello sin necesidad de realizar una inversión económica en el hardware necesario que haría falta para realizar dichas pruebas (en algunas ocasiones, inviable). En base a ese objetivo principal, se proponen los siguientes requisitos:

- El entorno de simulación y pruebas debe ser lo más económico posible. Al menos debe suponer una inversión menor a la equivalente que supondría la adquisición de los medios hardware y software necesarios para realizar las pruebas y simulaciones en un entorno real.
- El entorno debe ser sencillo de administrar, amigable al usuario.
- El entorno debe ser flexible, permitiendo evaluar múltiples soluciones de encaminamiento *multi-path* sobre BGP.

- El entorno debe ser escalable, permitiendo con un número pequeño de cambios incluir implementaciones de las nuevas soluciones *multi-path* a validar.
- El entorno debe devolver resultados en un tiempo razonable en relación al tamaño de la topología que se quiera comprobar.

1.3 Estructura de la memoria

Para facilitar la lectura del proyecto, se incluye a continuación un breve resumen de cada capítulo.

En el capítulo 2 se describe cada una de las fases por las que ha pasado el proyecto, desde el inicio hasta el final, aportando información sobre las dificultades encontradas, las soluciones propuestas, y las distintas características de los elementos involucrados en cada fase. Dicho capítulo comienza describiendo el análisis del problema, en el cual se explica lo que se pretendía conseguir en un principio, y cuál fue la solución inicial. Siguiendo con la línea temporal, se describe dicha solución en el siguiente apartado. El uso de hardware real (routers específicos y cables de red) para realizar pruebas sobre modificaciones *multi-path* de BGP. A continuación se propone la siguiente solución, la virtualización de un router y el estudio de prestaciones de distintos sistemas de virtualización para determinar el problema máximo que se podría simular teniendo en cuenta las restricciones del hardware que se está empleando. Finalmente, en un intento de abarcar problemas de mayor tamaño y sacrificando características irrelevantes para la solución que se pretende conseguir, se decide optar por las simulaciones en vez de la virtualización. En este punto se describen las diferentes alternativas existentes en el mercado para simular topologías con protocolos de encaminamiento y se explica por qué la decisión de utilizar C-BGP como base sobre la que desarrollar el entorno de validación.

En el capítulo 3 se describen los cambios realizados al simulador de red C-BGP para dar soporte a variantes *multi-path* del protocolo. Además de esto, se describe la aplicación [Statology](#), desarrollada inicialmente para convertir diferentes formatos de topología a otros admisibles por el simulador, pero que ha ido ganando funcionalidades hasta convertirse en una potente herramienta con la que, entre otras características, se pueden analizar diversos aspectos de las topologías.

En el capítulo 4 se explican brevemente los dos protocolos de encaminamiento *multi-path* BGP (LP-BGP y MpASS) con los que se ha verificado el correcto funcionamiento y aplicabilidad del entorno. Además se exponen las métricas seleccionadas para comparar soluciones, y se describen las pruebas realizadas sobre el mismo, esto es, sobre la herramienta [Statology](#) y sobre el simulador *mC-BGP*.

En el capítulo 5 se realiza un desglose de las tareas y su duración correspondiente a lo largo del proyecto. Además de esto, se proporciona un presupuesto detallado seguido del coste total supuesto para la realización del proyecto.

En el capítulo 6 se exponen las conclusiones que se han obtenido tras la realización del proyecto y algunas de las posibles vías de trabajo futuro que podrían ser desarrolladas en base al mismo.

Al final del documento se puede consultar un glosario de términos y acrónimos, una bibliografía que contiene las fuentes empleadas para la realización del proyecto y un par de anexos. El primero incluye los comandos que se han añadido al simulador C-BGP para obtener información adicional y configurar características avanzadas en las simulaciones. El segundo anexo incluye el manual de usuario de la herramienta [StatOpology](#).

1.4 Medios empleados

A lo largo del proyecto ha sido necesario emplear una serie de recursos "*hardware*" (dispositivos físicos) y "*software*" (aplicaciones informáticas) con el fin de recopilar información, realizar tareas de análisis, diseño e implementación de código, llevar a cabo simulaciones y pruebas, documentar los distintos procesos, etc. A continuación se muestran los distintos medios utilizados de manera detallada y clasificados en función de su tipo.

1.4.1 Hardware

Para llevar a cabo este proyecto se ha contado con los siguientes medios hardware:

- Ordenador personal en el que se realizó el trabajo en el laboratorio. Dicho PC cuenta con las siguientes características:
 - Procesador Intel Core i7 (2.67 GHz)
 - 12 Gb de memoria RAM DDR3 conectada en TriChannel
 - Disco duro Seagate de 750 Gb conectado por SATA-II
 - Tarjeta gráfica GeForce 9600 GTS
- Servidor de la universidad con alta capacidad de procesamiento en el que se lanzaron las simulaciones finales con la topología de gran tamaño (en magnitud, similar a Internet). El acceso a esta máquina se realizó vía SSH. Dicho servidor cuenta con:
 - 2 procesadores Intel(R) Xeon(R) CPU X5550 (2.67 GHz)
 - 48 Gb de memoria RAM DDR3
 - 2 discos duros de 1 Tb en SAS
- Ordenador personal utilizado en casa para realizar tareas de implementación, simulación, documentación y pruebas comparativas entre sistemas de virtualización y simuladores de red. Dicho PC cuenta con las siguientes características:
 - Procesador Intel Core 2 Quad Q9450 (2.67 GHz)
 - 8 Gb de memoria RAM DDR3 conectada en DualChannel
 - 2 discos duros. El primero, de 500 Gb. El segundo, de 1 Tb. Ambos conectados por SATA-II

- Tarjeta gráfica GeForce 9600 GTS
- Netbook Asus EeePC 1201N, utilizado para realizar tareas de implementación y documentación. Dicho PC cuenta con las siguientes características:
 - Procesador Intel Atom 330 (1.60 GHz)
 - 4 Gb de memoria RAM DDR2 conectada en DualChannel
 - Disco duro Hitachi de 250 Gb conectado por SATA-II
 - Tarjeta gráfica GeForce 9400 (nVidia ION)
- Ordenador portátil Acer Aspire 5672WLMi utilizado para realizar tareas, documentación y pruebas comparativas entre sistemas de virtualización y simuladores de red. Dicho PC cuenta con las siguientes características:
 - Procesador Intel Core Duo T2300 (2 núcleos a 1.67 GHz)
 - 3 Gb de memoria RAM DDR2 conectados en DualChannel
 - Disco duro Seagate de 100 Gb conectado por SATA
 - Tarjeta gráfica ATI Mobility Radeon X1400
- Disco duro externo USB + SATA de 320 Gb para transporte de datos.
- Memoria flash de 16 Gb marca Patriot para transporte de datos.
- 2 monitores TFT/LCD de 17 pulgadas (trabajo en el laboratorio).
- Monitor TFT/LCD de 24 pulgadas (trabajo en casa).
- Conexión a Internet de alta velocidad, 100 Mbps <TVEL> (trabajo en el laboratorio).
- Conexión a Internet de alta velocidad, 3 Mbps (trabajo en casa).
- Red *gigabit ethernet* instalada entre equipos del laboratorio con salida a la red del mismo a través de un *switch fast ethernet*.
- Router Linksys WRT54G utilizado para pruebas.

1.4.2 Software

Análogamente al apartado anterior, para la realización del proyecto se ha contado con el siguiente software:

- Sistemas operativos:
 - Ubuntu GNU/Linux 9.10
 - Instalado en el PC del laboratorio
 - Instalado en el PC de casa
 - Instalado en el netbook
 - Debian GNU/Linux 5.0.4
 - Instalado en el servidor de la universidad
 - OpenWrt
 - Instalado en el router Linksys WRT54G
 - Instalado en la máquina virtual (router virtual)
 - Microsoft Windows 7 Professional (64-bit)
 - Instalado en el PC de casa
 - Instalado en el netbook
 - Microsoft Windows Vista Business (64-bit)
 - Instalado en el PC de casa antes de Windows 7
- Software de virtualización
 - VirtualBox

- VMWare Server
- QEMU
- VNUML
- Software ofimático
 - Suite OpenOffice
 - Suite Microsoft Office 2007
 - Microsoft Word 2007
 - Microsoft Excel 2007
 - Microsoft PowerPoint 2007
 - Microsoft Project Professional 2007
 - Altova UModel 2008 y 2010 (trial)
 - Notepad ++
 - GNU vim
- Entornos de desarrollo software
 - NetBeans IDE
 - Eclipse IDE
- Editores gráficos
 - Inkscape
 - TheGimp
 - Paint.NET
 - IrfanView
- Compresores
 - WinRAR
 - unrar, TAR, GZ, BZ2
- Compiladores
 - GCC 3.X y 4.X
 - Javac (correspondiente a Java JDK 1.6)
 - Python
 - Perl
- Gestor de procesos, monitorización de procesador y memoria
 - Gnome System Monitor
 - HTOP
 - Microsoft Process Explorer
- Acceso remoto y transferencia de ficheros
 - SSH v2
 - SCP
 - FileZilla
 - WinSCP
 - Putty
- Otro software (simuladores de red e implementaciones del protocolo BGP)
 - C-BGP
 - NS+
 - Quagga
 - Zebra

Capítulo 2

Estudio tecnológico

*"Tus ojos pueden engañarte, no confíes en ellos."
Obi-Wan Kenobi. (Star Wars: Ep. IV)*

2.1 Introducción

En este capítulo se describen las distintas alternativas estudiadas, comenzando por el análisis del problema consistente en examinar el protocolo BGP y determinar los cambios necesarios para proporcionar soporte *multi-path*. Así pues, es necesario recopilar información y documentación sobre el protocolo. Como punto de partida, pueden resultar de gran ayuda las siguientes referencias: [BEI10], [RLH06] y [BRC00+].

Una vez estudiado el núcleo del problema y determinados los posibles cambios necesarios para modificar el protocolo se propone la primera alternativa sobre la que basar el entorno de evaluación. Dicha alternativa consiste en hacer uso de equipos reales para realizar las pruebas. A continuación se justifica el por qué no utilizar esta alternativa, sugiriendo otra más económica, basada en el uso de máquinas virtuales. En dicho punto se realiza un análisis entre diversas herramientas de virtualización gratuitas existentes en el mercado y se justifica la elección de una de ellas para realizar las modificaciones pertinentes sobre un router virtualizado. Finalmente se propone la solución que ha sido tomada en cuenta para el desarrollo del entorno: modificar el simulador de red C-BGP dotándolo de estructuras y funciones necesarias a la hora de implementar en el mismo variantes *multi-path* del protocolo BGP.

2.2 Equipos reales

Una posible alternativa consiste en realizar la modificación del protocolo BGP sobre una implementación existente en un router real de bajo coste. Concretamente en un Linksys WRT54G funcionando con una distribución basada en GNU/Linux.

Se analizaron las características de las dos distribuciones GNU/Linux para este tipo de routers, *DD-Wrt* <DDWRT> y *OpenWRT* <OWRT>, siendo elegida esta última, por ofrecer en su repositorio de paquetes instalables una implementación de BGP bajo el paquete *Quagga* (con licencia de código abierto y ampliamente extendida).

Una vez seleccionados los componentes necesarios, se requiere la realización de pruebas sobre un router real modificado con el objetivo de aprender sobre los comandos del sistema y el modo de configurarlo para finalmente probar a montar una pequeña red funcionando con BGP convencional.

El primer problema encontrado al aplicar esta solución es relativo a la disponibilidad de los routers Linksys WRT54G, puesto que dichos dispositivos son empleados por los alumnos del laboratorio en sus prácticas de encaminamiento. Por otra parte, realizar la instalación de cada variante *multi-path* del protocolo y configurar cada dispositivo hardware de forma independiente es una tarea muy tediosa y poco práctica.

Debido al elevado número de routers que serían necesarios para realizar pruebas con arquitecturas de red significativamente grandes se descartó esta opción. Sin embargo, la idea de utilizar al menos el sistema operativo del router con soporte BGP, es buena, ya que permite entre otras cosas obtener resultados más complejos, derivados de la evolución temporal del protocolo (de ahora en adelante, *dinámicas del protocolo*). Sin embargo, el uso de hardware real supone un considerable coste económico asociado a la adquisición de los equipos. Coste que puede verse mitigado si en lugar de utilizar routers reales se emula su comportamiento por software de forma que se comporte como si fuese real (*virtualizar*). Esto nos lleva al planteamiento de una segunda alternativa, expuesta a continuación.

2.3 Virtualización de routers

El primer paso para realizar la virtualización del router Linksys WRT54G debe comenzar en recopilar información sobre el dispositivo y sobre intentos de virtualización llevados a cabo por otras personas. Para ello fue de gran ayuda la documentación de la página oficial de OpenWRT [DOWRT]. Además de cómo preparar una distribución OpenWRT para ser virtualizada, es necesario conocer las especificaciones técnicas del router a fin de obtener resultados lo más fieles posible a la realidad. Dichas especificaciones fueron obtenidas a través de la página de hardware soportado por DD-WRT [WRTTS].

Modelo	WRT54G
Revisión del hardware	<i>1.1</i>
Procesador	<i>Broadcom</i>
Modelo de la CPU	<i>BCM5352</i>
Velocidad de la CPU	<i>200 MHz</i>
Overclock medio estable	<i>250 MHz</i>
RAM	<i>16 MB</i>
Memoria flash	<i>4 MB</i>
Tipo de conector de antena	<i>RP-TNC</i>
Puertos Ethernet	<i>4 LAN, 1 WAN</i>
Sistema Operativo	<i>OpenWRT v7.06</i>

Tabla 1. Características técnicas del router WRT54G

Una vez conocidas las especificaciones hardware sobre el modelo de router a virtualizar, se preparó una imagen virtual del mismo. El resultado de compilar los distintos módulos y configurarlos adecuadamente fue empaquetado en una máquina virtual con las siguientes características:

Hardware a emular	WRT54G
Nº de procesadores	1
RAM	16 MB
Imagen de disco	20 MB
Puertos Ethernet virtuales	5
Puertos serie	1
Sistema Operativo	OpenWRT v7.06

Tabla 2. Características técnicas de la máquina virtual con OpenWRT

Antes de comenzar con las modificaciones necesarias sobre la implementación de BGP contenida en *Quagga* para dotarla de soporte *multi-path* se realizaron un par de tareas necesarias para conseguir resultados óptimos: determinar qué sistema de virtualización iba a ser el más adecuado y realizar una prueba con una topología formada por máquinas virtuales conectadas en red para observar el comportamiento de las mismas.

2.3.1 Análisis sobre diferentes sistemas de virtualización

Para determinar cuál de los diferentes sistemas de virtualización del mercado podría ser el más apropiado se realizó un estudio entre los cuatro principales sistemas de virtualización del mercado con licencia de uso gratuita. Para cada uno de ellos se evaluaron características técnicas, de rendimiento, limitaciones y aspectos positivos que aportan.

2.3.1.1 Software de virtualización

Los sistemas software de virtualización estudiados fueron:

- VMware Server v1.0.6
- VirtualBox v2.0
- QEMU v0.9.1
- VNUML v1.8.3 (estable)

2.3.1.2 Hardware empleado para el análisis

Las pruebas fueron realizadas sobre dos equipos distintos: un ordenador de sobremesa con características de servidor y un ordenador portátil de gama media.

Equipo 1	Sobremesa
Procesador	Intel® Core™ 2 Quad Q9450 (4 núcleos a 2.66 Ghz)
RAM	4 GB DDR3 a 1333Mhz en Dual-Channel
Discos duros	2 discos Seagate (500GB + 1TB) interfaz SATA-II
Sistema operativo 1	Microsoft® Windows Vista™ Business (64-bit)
Sistema operativo 2	Linux Ubuntu 8.04 (64-bit)

Tabla 3. Prestaciones del equipo sobremesa

Equipo 2	Portátil (Acer Aspire 5672WLMi)
Procesador	Intel® Core™ Duo T2300 (2 núcleos a 1.66 Ghz)
RAM	3 GB DDR2 a 667Mhz en Dual-Channel
Discos duros	1 disco Seagate (100GB) interfaz SATA
Sistema operativo 1	Linux Ubuntu 8.04 (32-bit)

Tabla 4. Prestaciones del equipo portátil

2.3.1.3 Virtualización con VMWare Server v1.0.6

Las pruebas de virtualización se realizaron tanto en las versiones de Windows como de Linux. Los consumos de memoria se muestran en la siguiente tabla:

	Windows	Linux
Tamaño de instalación	126 MB	187 MB
Tamaño de imagen virtual	20.4 MB	20.4 MB
RAM utilizada (VMware)	33 MB	35.5 MB
RAM utilizada (imagen virtual)	17.2 MB	17.2 MB

Tabla 5. Virtualización con VMware Server 1.0.6

Características adicionales de VMware Server:

- Relativa facilidad para configurar las redes virtuales. En otros productos de VMware se facilita una aplicación de ventana para editar las interfaces virtuales de forma más rápida y sencilla (Virtual Network Editor).
- La versión Server (al igual que la versión Workstation) de VMware sólo admiten un máximo de 4 interfaces de red virtuales por máquina, lo cual no permite una emulación 100% fiable (los routers disponen de 5 interfaces).
- Lanzar las máquinas virtuales es extremadamente sencillo, aunque si se utiliza la interfaz gráfica, llegado a un determinado número de máquinas se hace complicado (a partir de 15 ya es complicado debido al gran número de pestañas).
- El acceso a las máquinas virtuales se puede realizar mediante SSH (previa configuración de la red virtual) y a través de la interfaz gráfica de VMware Server.
- La licencia de uso de VMware Server aunque implica registro en la web es gratuita.

2.3.1.4 Virtualización con VirtualBox 2.0

Análogamente a VMWare Server, las pruebas de virtualización se realizaron tanto en las versiones de Windows como de Linux. Los consumos de memoria se muestran en la siguiente tabla:

	Windows	Linux
Tamaño de instalación	74.3 MB	21.3 MB
Tamaño de imagen virtual	20.4 MB	20.4 MB
RAM utilizada (VirtualBox)	51 MB	6.7 MB
RAM utilizada (imagen virtual)	17.2 MB	17.2 MB

Tabla 6. Virtualización con VirtualBox 2.0

Características adicionales de VirtualBox:

- Al igual que en VMware, sólo se admiten un máximo de 4 interfaces de red virtuales por máquina, lo cual es un problema.
- No presenta características de servidor de máquinas virtuales. Esto es, las máquinas deben ser lanzadas en local, y si se cierra la aplicación se apagan todas las que estuvieran activas.
- Gestionar las máquinas virtuales desde la aplicación se hace complicado cuando se sobrepasa cierto número de máquinas lanzadas, al igual que en VMware al utilizar la interfaz gráfica.
- El acceso a las máquinas virtuales se puede realizar mediante SSH (previa configuración de la red virtual) y a través de la interfaz gráfica de VirtualBox.

2.3.1.5 Virtualización con QEMU 0.9.1

En este caso, también se realizaron las pruebas de virtualización sobre los sistemas operativos Windows y Linux. La siguiente tabla muestra los resultados comparativos de consumo de memoria:

	Windows	Linux
Tamaño de instalación	21.6 MB	13.4 MB
Tamaño de imagen virtual	20.4 MB	20.4 MB
RAM utilizada (QEMU)	19.9 MB	2.9 MB
RAM utilizada (imagen virtual)	17.2 MB	17.2 MB

Tabla 7. Virtualización con QEMU 0.9.1

Características adicionales de QEMU:

- QEMU consiste en una aplicación de consola (línea de comandos). Resulta bastante tedioso crear / lanzar máquinas virtuales. Más aún configurar la red virtual. A pesar de todo es el sistema de virtualización que menos recursos consume de entre los que presentan una interfaz gráfica para trabajar con la máquina virtual.
- Para facilitar el trabajo con QEMU, se han empleado dos aplicaciones que sirven como interfaz gráfica (en realidad lo que hacen es lanzar en *background* el proceso QEMU con los parámetros seleccionados). Dichas aplicaciones son:
 - Qemu Manager 5.0 (Windows)
 - QtEmu 1.0.5 (Linux)
- Se pueden definir más de 16 interfaces virtuales de red, con lo que no hay problema en emular el router con todas sus características.
- Las máquinas virtuales deben ser lanzadas de forma independiente, cada una en su propia ventana.
- El acceso a las máquinas virtuales se puede realizar mediante SSH (previa configuración de la red virtual) y a través de la interfaz de QEMU.
- En Linux puede aprovechar la tecnología UML (*User Mode Linux*).

2.3.1.6 Virtualización con VNUML v1.8.3

La virtualización de tipo *User Mode Linux* sólo pudo ser probada en dicho sistema operativo, ya que Windows no presenta esa característica. Los consumos de memoria utilizando este sistema son los siguientes:

	Windows	Linux
Tamaño de instalación	-	7.4 MB
Tamaño de imagen virtual	-	20.4 MB
RAM utilizada (VNUML)	-	13.5 MB
RAM utilizada (imagen virtual)	-	17.2 MB

Tabla 8. Virtualización con VNUML 1.8.3

Características adicionales de VNUML (Virtual Network User-Mode-Linux):

- VNUML presenta una interfaz de consola (línea de comandos). No dispone de ningún tipo de interfaz gráfica en la que ver las máquinas virtuales lanzadas ni su estado.
- Permite definir las redes e interfaces de cada una de las máquinas a lanzar mediante scripts escritos en .xml.
- Lanzar y configurar las máquinas virtuales es extremadamente sencillo una vez conocido el lenguaje de scripting.
- El acceso a las máquinas virtuales sólo puede realizar mediante SSH ya que VNUML no dispone de interfaz gráfica en la que ver el estado de las máquinas virtuales.
- Existe un “LiveDVD” que permite utilizar el entorno VNUML en una distribución Linux pre-configurada, con un coste de recursos mínimo.
- De momento no existe un parche para los kernels de 64 bits (lo cual implica que al tener que ser lanzada en sistemas de 32 bits no se pueden emplear más de 4 GB de memoria total en el sistema).

2.3.1.7 Conclusiones sobre el análisis

Teniendo en cuenta el consumo de recursos y las características ofrecidas por cada sistema de virtualización, quedan **descartados VMware Server y VirtualBox** ya que el máximo número de interfaces de red virtuales admitido es 4.

Por otra parte, entre QEMU y VNUML es preferible utilizar **VNUML** para virtualizar routers, ya que aunque no exista ninguna interfaz gráfica para configurarlo, **consume menos recursos** y está **orientado al tema de virtualización de redes**. Además **resulta más sencillo gestionar una red** más o menos grande de máquinas virtuales interconectadas que con QEMU.

Finalmente, teniendo en cuenta las características técnicas de los equipos sobre los que se han realizado las pruebas, y el consumo medio por sistema de virtualización + máquinas virtuales se estima que el número máximo de máquinas virtuales ejecutándose en paralelo bajo VNUML para cada equipo es de aproximadamente:

Equipo 1: 43 máquinas virtuales

Equipo 2: 17 máquinas virtuales

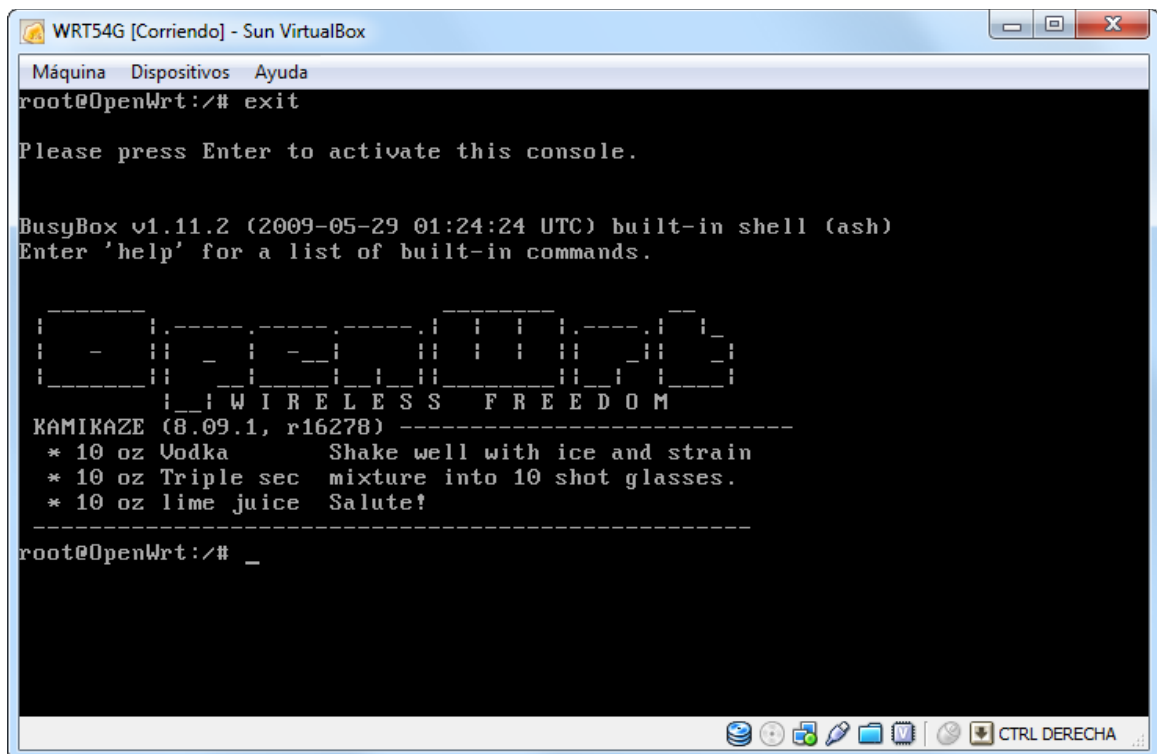


Fig. 1. OpenWrt en ejecución bajo VirtualBox

2.3.2 Características de la virtualización

Como se puede comprobar por los resultados del análisis, la virtualización puede resultar una forma de evaluar y verificar realmente útil si se cumplen los siguientes requisitos:

1. La red a simular no es excesivamente grande (< 70 nodos).
2. Se necesitan valores muy ajustados a la realidad, como retardos en la red o evaluar qué ocurre en los distintos nodos en caso de caída de uno o varios de ellos.

Tras barajar la posibilidad de simular topologías muy grandes, en orden de magnitud igual al de Internet, se concluyó que la alternativa de usar máquinas virtuales no iba ser suficientemente buena, ya que a la hora de verificar los protocolos, la necesidad de un número elevado de equipos puede suponer una limitación importante tanto en términos de coste económico como en términos de la complejidad de su instalación y monitorización. Se requiere una alternativa que suponga un menor consumo de recursos y que resulte más sencilla de administrar y configurar. Esto nos lleva a la siguiente solución, la cual será la elegida sobre la que realizar el desarrollo del entorno de evaluación.

2.4 Simuladores de red

Ya que la idea consiste en probar cómo se comportan las modificaciones *multi-path* BGP en redes grandes con un número elevado de sistemas autónomos interconectados, se decidió emplear simuladores de red. Fue preciso realizar un estudio para determinar el que mejor se ajusta a las necesidades requeridas. Las alternativas evaluadas fueron C-BGP <CBGP> y NS-2 <NS2> con BGP++ <BGP++>. Tras realizar una serie de pruebas de rendimiento, se decidió optar por la alternativa C-BGP.

A continuación se muestran las principales características observadas en los simuladores de red y el estudio comparativo realizado sobre prestaciones entre C-BGP y NS-2.

2.4.1 Características de los simuladores

Aparentemente, hacer uso de simuladores de red sólo tiene asociado una serie de ventajas:

- Suelen ser escalables, permitiendo ampliar su funcionalidad.
- Por lo general es posible evaluar un gran número de parámetros.
- El tamaño de las redes sobre las que realizar las pruebas puede llegar a ser muy grande.
- Existen varios simuladores de código abierto, gratuitos, muy potentes, extendidos y probados.
- La complejidad de las simulaciones es variable. Es decir, si únicamente se desea comprobar si un protocolo de encaminamiento converge, pueden no ser tenidos en cuenta factores como los retardos de red, caídas, etc. Sin embargo, si queremos realizar una simulación más precisa, en los que sea necesario evaluar factores como los anteriormente citados, podemos configurar la misma para conseguir dicho propósito, o bien, ampliar las características del simulador, implementando en su código dichas propiedades.

Sin embargo, no todas sus características son ventajosas. Algunos de sus principales inconvenientes son los siguientes:

- El consumo de memoria que hacen no es lineal en base a los datos de entrada, pudiendo llegar a ser excesivo en algunas simulaciones.
- Por lo general, no hacen uso de computación en paralelo. Esto podría ahorrar bastante tiempo de simulación.
- El código fuente suele estar poco comentado y resulta muy difícil trazarlo y modificarlo.

2.4.2 Análisis sobre diferentes simuladores existentes

En este apartado se pretende comentar y comparar las características propias entre los dos principales simuladores de red de código abierto existentes en el mercado, válidos para obtener nuestro objetivo: un entorno de simulación de protocolos versátil, económico y escalable.

Los simuladores de red analizados, como se ha comentado previamente, han sido C-BGP y NS-2 con el módulo BGP++.

2.4.2.1 C-BGP

C-BGP está orientado a simular el proceso de decisión de BGP en redes compuestas por muchos routers. Para realizar dicho propósito, utiliza la configuración de cada router, las rutas externas de BGP recibidas y la topología de la red.

Soporta el proceso de decisión de BGP (entero), importar y exportar filtros, *route-reflection* y algunos atributos experimentales, como *redistribution communities*.

C-BGP puede 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 fallos de enlaces y de routers. Los fallos lógicos incluyen cambios 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 grades, del mismo tamaño en orden de magnitud que Internet.

C-BGP está escrito en el lenguaje de programación C. Principalmente es utilizado y probado en máquinas con GNU/Linux y MacOS. También se puede utilizar en otras plataformas como FreeBSD, Solaris e incluso bajo Windows, utilizando la API Cygwin. Debería compilar en cualquier entorno con soporte POSIX. C-BGP se distribuye bajo la licencia de código libre LGPL.

Las pruebas fueron realizadas compilando el código fuente de la versión 1.3.0 bajo GNU/Linux.

2.4.2.2 NS-2 con el módulo BGP++

BGP++ es una implementación en C++ de BGP para los simuladores de red NS-2 y GTNetS <GTNS>. BGP++ es una modificación del paquete software Zebra BGPd <ZBRA> para trabajar con los simuladores citados. Zebra es una implementación de código abierto de BGP y de otros protocolos de encaminamiento para plataformas UNIX. 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 protocolos 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. Además, BGP++ soporta las siguientes características: BGP *confederations*, *route reflection*, *flap dampening* y *route-refresh*.

Principalmente se utiliza en entornos Linux, pero puede ser compilado para Windows utilizando Cygwin o MinGW.

Análogamente al caso de C-BGP, las pruebas fueron realizadas tras realizar la compilación del código fuente de la versión 1.05 de BGP++ y de la versión 2.33 de NS-2 bajo GNU/Linux.

2.4.2.3 Comparación entre los dos simuladores de red

En primer lugar, para instalar C-BGP fue necesario obtener e instalar una serie de bibliotecas de código abierto que no vienen por defecto instaladas en las distintas versiones de GNU/Linux como Debian o Ubuntu. Dichas bibliotecas son **libpcrc**, **libreadline** y **libgds**. Mientras que existen paquetes pre compilados para **libpcrc** y **libreadline** en la mayor parte de plataformas, hizo falta descargar **libgds** y compilarlo manualmente. Finalmente, la compilación y ejecución se realizó como viene siendo habitual con los programas de código abierto, es decir, mediante la secuencia de comandos:

```
./configure
make
sudo make install
```

En el caso de BGP++, la tarea resultó más complicada, ya que fue necesario descargar y compilar por un lado el simulador de red NS-2 y por otro el módulo BGP++. Además, la compilación de BGP++ requiere tener instalado Tcl/Tk (con sus cabeceras) junto con dos paquetes adicionales: **tccl** y **otcl**. Una vez instalados todos los paquetes requeridos, fue necesario parchear la versión de NS-2 mediante un archivo incluido con los archivos fuente de BGP++. Adicionalmente se probó a instalar **pdns v2** (*Parallel Distributed NS* <PDNS>), sin embargo no se pudo probar debido a la necesidad de tenerlo en varias máquinas a la vez. Una vez configurados los requisitos de instalación de **pdns** y parcheados los ficheros fuente de NS-2, la compilación e instalación resultó análoga a la de C-BGP.

En la prueba realizada se apreció cómo C-BGP resultaba más rápido obteniendo la información necesaria para cumplir los objetivos del entorno. Inicialmente comprobar la convergencia de la red y obtener las tablas de rutas de cada sistema autónomo. Además de esto, el consumo de memoria de C-BGP resultó significativamente menor. Posiblemente porque NS-2 trabaja con más parámetros a la hora de realizar la simulación. Al fin y al cabo, BGP++ es la implementación completa del protocolo BGP (como ya se comentó anteriormente) y adaptada para NS-2. 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 [CBGPA], al contrario que BGP++, el cual corre sobre un simulador a nivel de paquetes como es NS-2, el cual sí los contempla. Como se ha comentado, para los resultados que se esperan obtener del entorno, algunos de estos parámetros resultan innecesarios (como por ejemplo el valor de temporizadores), ocupando memoria y generando un mayor tiempo a la hora de realizar las simulaciones, resultando esto un punto a favor del simulador C-BGP.

Por otro lado, 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.

Finalmente, con respecto a la escalabilidad, C-BGP es capaz de simular topologías del tamaño de Internet con hardware limitado [MFM06+], 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. Este último detalle resultó bastante atractivo a la hora de tomar la decisión final sobre cuál iba a ser el entorno de partida a modificar.

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

2.5 Conclusiones

En este capítulo se han descrito las distintas alternativas estudiadas con el objetivo de desarrollar un entorno de evaluación versátil, escalable, mantenible y económico para protocolos de encaminamiento *multi-path* interdominio. En concreto, para variantes *multi-path* de BGP.

En primer lugar se apostó por la opción de emplear routers reales, pero tras observar el coste asociado a cada equipo, tanto económico como en tiempo de instalación y configuración de las variantes del protocolo en cada uno de ellos, se decidió descartar dicha alternativa.

En segundo lugar se decidió pasar a virtualizar el sistema operativo empleado por los routers anteriormente citados. Dicho sistema operativo es una pequeña distribución GNU/Linux orientada a sistemas embebidos. Para determinar el sistema de virtualización a emplear se realizó una comparativa de características y consumo de recursos. A continuación se muestra una tabla-resumen sobre dicha comparativa. Los valores asociados a los tamaños en memoria y disco son los correspondientes a las versiones de GNU/Linux para cada sistema de virtualización.

Característica	VMware Server v1.0.6	VirtualBox v2.0	QEMU v0.9.1	VNUML v1.8.3
Tamaño de instalación	187 MB	21.3 MB	13.4 MB	7.4 MB
Tamaño de imagen virtual	20.4 MB	20.4 MB	20.4 MB	20.4 MB
RAM utilizada (entorno)	35.5 MB	6.7 MB	2.9 MB	13.5 MB
RAM utilizada (imagen virtual)	17.2 MB	17.2 MB	17.2 MB	17.2 MB
Administración	Gráfica	Gráfica	Gráfica + Consola	Consola
Configuración	Gráfica	Gráfica	Gráfica + Consola	Lenguaje de scripting específico
Acceso a la máquina virtual	Gráfico + SSH	Gráfico + SSH	Gráfico + SSH	SSH
Máximo número de interfaces de red por máquina virtual	4	4	16	255 (*)

Tabla 9. Comparativa entre soluciones de virtualización

(*) En realidad el valor depende del valor asignado bajo la etiqueta `<vm_mgmt>` del script de configuración. Si se encuentra presente la etiqueta `<automatic>` entonces se aplica la restricción máxima a 255 interfaces (valor por defecto).

Como se puede apreciar en la Tabla 9, la solución de virtualización que más ventajas presenta es VNUML, puesto que además de consumir pocos recursos hardware, permite realizar configuraciones masivas gracias a un lenguaje de scripting específico (en otras soluciones hay que realizar la configuración máquina a máquina desde una interfaz gráfica). Otro aspecto destacable es que el número de interfaces de red por máquina virtual satisface con creces el requerido para virtualizar el router Linksys WRT54G.

Sin embargo, se observó que utilizando este tipo de solución, el máximo número de máquinas virtuales por equipo era de aproximadamente 50, valor muy inferior al requerido para simular una topología de red del tamaño de Internet.

Finalmente se optó por modificar un simulador de red a fin de dotarlo de soporte *multi-path*. Análogamente al caso anterior se realizó un estudio comparativo entre los candidatos más apropiados a ser modificados. A continuación se muestra una tabla con el resumen de características de ambos.

Característica	NS-2 con BGP++	C-BGP
Entorno de ejecución	GNU/Linux, Windows	GNU/Linux, Windows
Computación distribuida	Sí (con módulo <i>pdns v2</i>)	No
Tiempo de simulación	Medio - Alto	Muy bajo
Simula dinámicas del protocolo	Si	No
Consumo de memoria	Medio - Alto	Medio - Bajo
Escalabilidad	Media - Alta	Alta - Muy alta
Integración con otro software	No de forma nativa	Si, gracias a JNI

Tabla 10. Comparativa entre simuladores de red

Aunque el simulador de red NS-2 con el módulo BGP++ presenta importantes ventajas respecto a C-BGP, como son la simulación de dinámicas del protocolo y la posibilidad de ejecutarse a la vez entre diferentes máquinas (*clustering*) para obtener resultados, el segundo simulador se ajusta más a las necesidades requeridas para desarrollar el entorno, como son el tiempo empleado y la cantidad de memoria requerida para obtener los resultados. Principalmente por estos motivos se tomó la decisión de emplear este último como base para el desarrollo del entorno de evaluación.

Capítulo 3

Desarrollo del entorno de evaluación *multi-path*

*"No. No lo intentes. Hazlo, o no lo hagas, pero no lo intentes."
Maestro Yoda. (Star Wars: Ep. V)*

3.1 Introducción

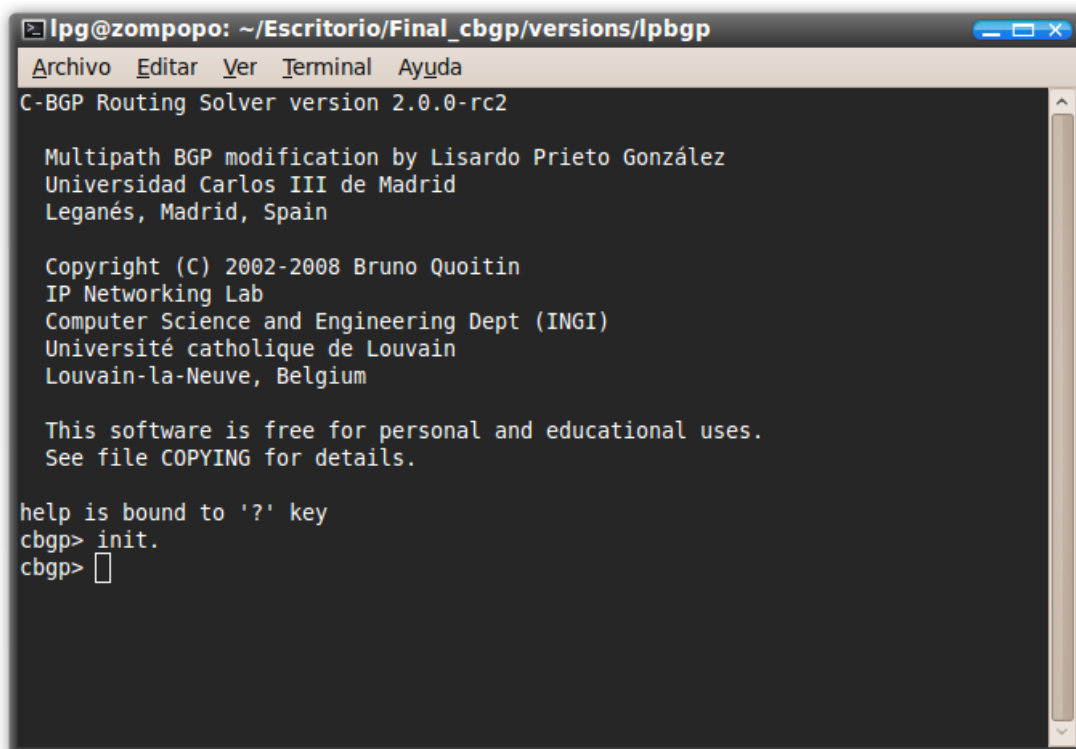
Para ampliar la funcionalidad existente del simulador C-BGP y dotarlo de soporte *multi-path* se requieren una serie de cambios repartidos entre los distintos módulos que lo componen. En este capítulo se explica la arquitectura del simulador y se detallan dichos cambios sobre la misma.

Para completar el entorno de evaluación, además de modificar el simulador se requiere una herramienta que permita gestionar las topologías, convirtiéndolas a un formato interpretable por el simulador y que obtenga datos estadísticos de las mismas. También es deseable que dicha herramienta tenga características adicionales, como la posibilidad de generar scripts para efectuar las simulaciones de forma automática o la posibilidad de interpretar los datos obtenidos como resultado de la simulación. El desarrollo y características de dicha herramienta se muestran tras el siguiente apartado.

3.2 De C-BGP a mC-BGP

Para dotar al simulador de red C-BGP de soporte *multi-path* (de ahora en adelante *mC-BGP*) 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.

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.



```
lpq@zompopo: ~/Escritorio/Final_cbgp/versions/lpbgp
C-BGP Routing Solver version 2.0.0-rc2

Multipath BGP modification by Lisardo Prieto González
Universidad Carlos III de Madrid
Leganés, Madrid, Spain

Copyright (C) 2002-2008 Bruno Quoitin
IP Networking Lab
Computer Science and Engineering Dept (INGI)
Université catholique de Louvain
Louvain-la-Neuve, Belgium

This software is free for personal and educational uses.
See file COPYING for details.

help is bound to '?' key
cbgp> init.
cbgp> 
```

Fig. 2. Ejecución de *mC-BGP*

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 encaminamiento BGP. Este modelo contiene el proceso de decisión y los filtros de encaminamiento. El modelo de encaminamiento 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.

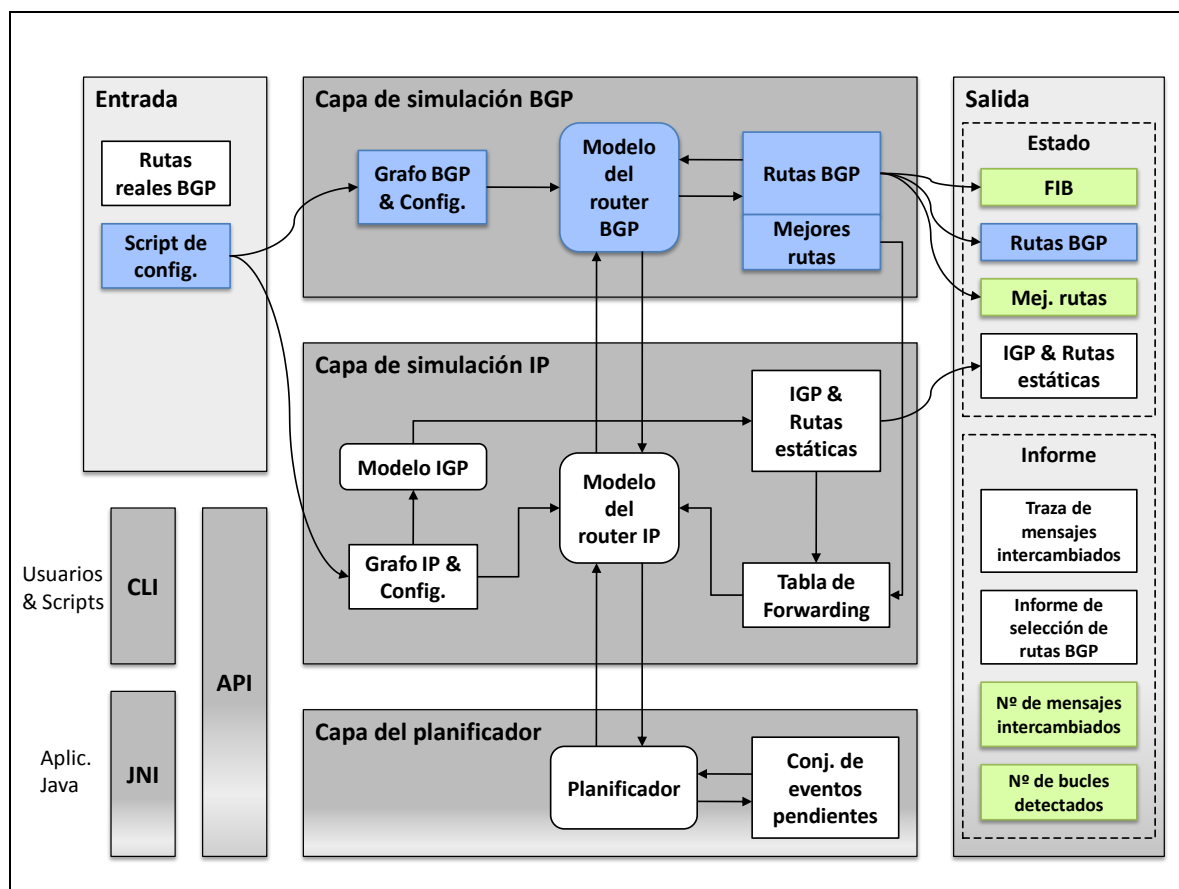


Fig. 3. Arquitectura de *mC-BGP*

En la Fig. 3 se pueden apreciar los cambios necesarios sobre la arquitectura de C-BGP para dotarlo de soporte *multi-path*. Fue necesario ampliar el módulo “Grafo BGP & Configuración” 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” con el fin de 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”,

siendo necesario en este caso añadir las estructuras correspondientes a la **FIB** y a los **AS_SETs** (estructura utilizada en algunas soluciones *multi-path* y no soportada inicialmente por el simulador).

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).

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 (ver apartado 4.4). 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.

Para abordar los cambios anteriormente citados fue necesario implementaron nuevas funciones, correspondientes a los módulos "*FIB*", "*Mejores rutas*", "*Nº de mensajes intercambiados*" y "*Nº de bucles detectados*" con el fin de mostrar la información correspondiente a dichos módulos por la salida del simulador. Para el primer caso, la salida corresponde a la *Forwarding Information Base* almacenada por los routers BGP seleccionados mediante el comando encargado de obtener dicha información. La información mostrada por pantalla es coherente con el formato de salida estándar proporcionado por el simulador para, por ejemplo, mostrar la RIB. En el segundo caso, mostrará todas las "mejores rutas" seleccionadas por el simulador hasta la convergencia de la red. El formato de salida será análogo al caso anterior. Para el número de mensajes intercambiados, la salida corresponderá a un número entero, siendo este la suma de todos los mensajes intercambiados por todos los routers BGP de la topología hasta el momento en que ha convergido. Finalmente, el número de bucles detectados podrá obtenerse en casos concretos de protocolos *multi-path* en los cuales se implemente la detección de

bucles. La salida corresponderá a los bucles detectados por un determinado router de la topología a lo largo de la simulación.

Se comenzó modificando la versión 1.3.0 del simulador de red C-BGP, pero debido a las apariciones de versiones posteriores a lo largo del proyecto, se realizaron tareas de análisis y modificación del código fuente de las mismas para aprovechar las nuevas funcionalidades proporcionadas por ellas. La última versión sobre la que se han aplicado las modificaciones ha sido la 2.0.0-rc2, publicada el 4 de agosto del 2009.

En los siguientes apartados se va a describir de una forma más específica los principales cambios necesarios en los componentes que forman C-BGP para dotar al simulador de soporte *multi-path*.

3.2.1 El router BGP

El simulador C-BGP está dividido en una serie de módulos independientes que interaccionan entre sí para lograr el funcionamiento del conjunto. Uno de los módulos principales es el que define los tipos de elementos que puede gestionar el simulador mediante una serie de estructuras de datos y registros. En dicho módulo se define el router BGP.

En la versión no modificada del simulador, el router BGP contiene una serie de campos necesarios para procesar y almacenar la información con la que trabaja. En concreto, para almacenar las tablas de rutas utiliza una estructura de datos eficiente (*árbol trie*) pero que no permite repetición entre sus elementos. Es decir, no se pueden almacenar varias rutas para un mismo prefijo. Esta estructura es utilizada para gestionar la **Loc-RIB** (es decir *Local Routing Information Base*), el **Adj_RIB_IN** (es decir, la **RIB** de las rutas propagadas por los vecinos) y el **Adj_RIB_OUT** (la **RIB** con las rutas que queremos propagar a los vecinos). En el caso que nos concierne, necesitamos una estructura de datos eficiente y de crecimiento dinámico (es decir, a lo largo del tiempo de ejecución del simulador) pero que permita almacenar múltiples entradas para un mismo prefijo. Dicha estructura es necesaria para almacenar la **FIB** (o tabla de rutas de *forwarding*) del protocolo *multi-path*, puesto que en este caso, se pueden presentar varias entradas para el mismo prefijo (o lo que es lo mismo, varias rutas).

En la versión modificada, *mC-BGP* se decidió emplear una estructura de datos incluida en la biblioteca utilizada por el simulador: **libgds**, de forma que la mantenibilidad del código permanezca inalterada. La estructura en cuestión se trata de un array dinámico con inserción y acceso por búsqueda binaria, para el cual hubo que implementar las funciones de comparación entre elementos, e inicializarlo con soporte para elementos repetidos, o lo que es lo mismo, soporte de múltiples rutas para el mismo prefijo de red. La adición de esta nueva estructura supuso modificar las funciones de creación (inicialización) y destrucción del router, puesto que al tratarse de una estructura dinámica se basa en punteros y de no liberarse la memoria utilizada por la misma podrían ocasionarse problemas en el sistema (memoria libre pero marcada como ocupada, con la consiguiente degradación de rendimiento en la máquina que ejecuta el simulador).

Debido a que uno de los objetivos del presente proyecto consiste en la posibilidad de simular múltiples protocolos *multi-path* en la misma topología y al mismo tiempo, también es necesario dotar al router BGP de una variable que indique el tipo de protocolo que va a ser utilizado por el mismo. Junto con dicha variable también será necesario implementar un par de funciones, correspondientes a la consulta y modificación de dicho valor.

Finalmente, se añadieron un par de parámetros adicionales a la estructura que define el router con el objetivo de obtener más información de la simulación y poder objetivar mejor la comparación de resultados mediante el uso de las métricas sobre estos nuevos valores. Dichos parámetros son la cantidad máxima de agregación permitida, o lo que es lo mismo en términos del simulador, la cantidad máxima de rutas que se pueden instalar para cada prefijo y el número de bucles detectados. Ambos parámetros requieren, como en el caso anterior, de una serie de funciones para consultar y modificar sus valores desde otros módulos que conforman la estructura del simulador.

3.2.2 Las rutas

En el mismo módulo en el que se definen las estructuras y variables que conforman el router BGP, son definidos los elementos que conforman las rutas BGP. La estructura que implementa una ruta BGP es compleja. Consta de diversos elementos, como el prefijo de la ruta, el vecino del cual ha aprendido dicha ruta, los *flags* (mejor, posible, elegible, etc.), el rango y los atributos para dicha ruta.

A su vez, los atributos de una ruta están definidos mediante una estructura que contiene el NEXT_HOP (siguiente salto), el ORIGIN (origen), el LOCAL_PREF (preferencia local), los valores necesarios para la gestión de comunidades, etc. Uno de los atributos más importantes contenidos por dicha estructura es el AS_PATH. El AS_PATH contiene una lista de todos los Sistemas Autónomos que conforman un camino para ir desde un origen hacia un destino concreto en su orden de recorrido.

Si se analiza la estructura que define el AS_PATH, se puede apreciar que proporciona soporte para los AS_SETs (estructura de datos que almacena un conjunto de Sistemas Autónomos sin un orden pre-establecido). El problema es que en el código del simulador no se encuentran definidas las funciones necesarias para manejar los AS_SETs contenidos en el AS_PATH.

Para facilitar la gestión de los AS_SETs, se ha definido bajo la estructura que conforma los atributos de una ruta un nuevo campo correspondiente a dicho AS_SET. La estructura utilizada para implementar el AS-Set, análogo al caso de los routers BGP, se encuentra definida en la biblioteca **libgds**. En este caso, por términos de eficiencia se ha elegido un array de enteros dinámico puesto que los AS_NUM deben ser números enteros. La inserción y acceso a los elementos en el mismo se realiza mediante búsqueda binaria, resultando muy eficiente en operaciones de consulta y adición de nuevos ASes, y a diferencia del caso planteado para los routers BGP, al tratarse de un conjunto, se ha inicializado la estructura sin posibilidad de repetición de elementos. Esto es, en caso de

intentar almacenar un elemento que ya existe en la estructura, no se realiza dicha acción, existiendo únicamente una instancia de cada elemento.

Al definir una nueva estructura para almacenar los AS_SETs, la gestión de los mismos se simplifica enormemente respecto al caso de tener que manejarlos dentro de la subestructura del AS_PATH, puesto que no hay que determinar el tipo del elemento almacenado por dicha estructura antes de decidir qué operación es posible aplicar (el elemento podría ser un AS de la ruta o bien una parte de un AS_SET). La adición de esta nueva estructura requiere la implementación de las funciones encargadas de gestionarla (inserción, consulta y eliminación).

3.2.3 El proceso de decisión

El proceso de decisión de BGP consiste en una serie de reglas que se aplican sobre las rutas recibidas de los vecinos con el fin de determinar si se han encontrado caminos nuevos y si dichos caminos son mejores que los que ya se conocen para un prefijo determinado.

En BGP convencional, el proceso de decisión se ejecuta hasta que queda una o ninguna ruta para el mismo prefijo. En caso de empate se pueden utilizar estrategias de *tie-break* o desempate, quedando como resultado una única ruta resultado. En el caso de BGP *multi-path*, interesa que en caso de haber múltiples rutas que cumplan las restricciones impuestas por las reglas del proceso de decisión, sean almacenadas todas ellas y no se realice ningún "desempate". Así pues, es necesario modificar cada una de las funciones encargadas de aplicar cada regla del conjunto de reglas de decisión sobre las rutas recibidas a fin de eliminar el proceso de desempate. De esta forma, como resultado del filtrado se pueden mantener varias rutas para un mismo prefijo. Rutas que deberán ser instaladas en la nueva estructura implementada en el router, la **FIB**.

Una vez finalizado el proceso de decisión sobre las rutas entrantes, se ha implementado la opción de realizar otro proceso de decisión distinto sobre la **FIB** con el fin de obtener una única ruta por prefijo. Estas rutas pueden ser instaladas en la estructura **Adj-RIB-OUT** para ser posteriormente propagadas a los vecinos del router BGP.

En caso de necesitar aplicar un proceso de decisión distinto, o aplicar las reglas que lo componen en distinto orden, se puede implementar o modificar la estructura que lo almacena. Dicha estructura contiene las funciones que conforman dicho proceso de decisión en el orden secuencial de aplicación. De esta forma, si se eliminan o varían entradas en ella, el proceso de decisión resultará alterado. Esto puede ser deseable dependiendo del protocolo *multi-path* a implementar en el simulador.

En caso de necesitar diferentes procesos de decisión, es posible clonar la estructura anterior con un nombre distinto, e implementar las funciones necesarias a aplicar. Estas acciones deben ser realizadas en el mismo módulo que contiene la estructura inicial del proceso de decisión genérico.

Este último caso podría ser necesario si se desean simular diferentes protocolos *multi-path* BGP en una misma topología a la vez. Entonces sería necesario modificar la función encargada de aplicar el proceso de decisión para utilizar en cada caso el proceso correspondiente a cada router BGP en base a su valor almacenado por el campo "protocolo a utilizar".

3.2.4 La interfaz con el usuario

Además de los módulos encargados del cálculo de la simulación, se requieren otros módulos que conforman la parte en la que el simulador interactúa con el usuario. En una parte de estos módulos, correspondiente a las funcionalidades de línea de comandos, se ha definido la gramática del simulador, esto es, el conjunto de instrucciones y operaciones permitidas sobre los distintos elementos modelados en C-BGP.

La inclusión de mejoras y nuevas características requiere una modificación de dicha gramática a fin de incluir nuevas directivas que permitan al usuario aprovechar estas nuevas funcionalidades (ver Anexo I). La modificación de la gramática implica modificar las funciones utilizadas por el gestor de la interfaz de forma que se permita reconocer nuevos comandos, además de implementar las funciones asociadas a dichos nuevos comandos.

Las funciones asociadas a los nuevos comandos van a hacer uso de las nuevas funciones implementadas para la gestión de los atributos añadidos en los módulos citados anteriormente (router BGP, rutas, etc.), de forma que se conserve la estructura modular del simulador, y resulte sencillo mantener y ampliar su código fuente.

Por otra parte, la inclusión de la nueva estructura AS_SET requiere una modificación en la forma en la que son mostradas las rutas en la salida por defecto. Como se comentó anteriormente, el código original del simulador no brinda soporte a la gestión de esta estructura, por lo que es necesario añadir el soporte en las funciones encargadas de mostrar las rutas para que en caso de detectarse un AS_SET sea mostrado con el formato adecuado.

3.2.5 Implementación de nuevos protocolos

Gracias a los cambios aplicados a los distintos módulos en la estructura de C-BGP (ahora *mC-BGP*), implementar nuevos protocolos *multi-path* no debería resultar especialmente complicado. A continuación se adjunta una tabla-guía con la estructura de directorios asociada al código fuente, y los módulos correspondientes a cada uno de ellos (para más información visitar <CBGPSRC>).

DIRECTORIO	CONTENIDO
src	Directorio base del código fuente
src/bgp	Protocolo BGP
src/cli	Interfaz de línea de comandos
src/external	Código externo (<i>libbgpdump</i>)
src/hlp	Protocolo HLP
src/jni	Java Native Interface
src/net	Topología de red, encaminamiento y <i>forwarding</i>
src/sim	Planificador del simulador
src/ui	Interfaz de usuario

Tabla 11. Distribución del código fuente de C-BGP

Para implementar una variante *multi-path* BGP en el simulador hay que tener en cuenta las características de dicha modificación del protocolo. En caso de presentar un nuevo proceso de decisión, será necesario modificar el módulo correspondiente (ver apartado 3.2.3) con el fin de añadir ese nuevo proceso, y de modificar la función decisora encargada de asignar a cada router BGP el proceso de decisión correspondiente al valor configurado para cada uno de ellos durante la simulación (por defecto, el proceso de decisión estándar de BGP).

En caso del que el protocolo envíe mensajes BGP en momentos específicos, se deberán añadir en dichos puntos las operaciones necesarias para incrementar la variable correspondiente al número de mensajes enviados. Esto siempre y cuando se desee obtener dicho valor como parte del resultado de la simulación.

Igualmente, si se desea especificar un máximo nivel de agregación de rutas, en el módulo correspondiente al proceso de decisión será necesario añadir las operaciones adecuadas para filtrar las rutas a partir del valor seleccionado como máximo nivel de agregación en cada router BGP. Análogamente, en el mismo módulo será necesario añadir las operaciones pertinentes en caso de que el protocolo posea un mecanismo de detección de bucles, a fin de incrementar el contador individual de cada router BGP correspondiente al número de bucles detectados. Esto siempre y cuando se desee obtener dicho valor como parte del resultado de la simulación.

3.3 StatOpology

Para ayudar a evaluar los resultados de las simulaciones se desarrolló una herramienta software llamada “StatOpology”. Es una aplicación escrita en lenguaje Java, ideada de forma modular, con el objetivo de poder añadir características nuevas sin modificar el código correspondiente al resto de módulos. Las funcionalidades inicialmente implementadas fueron las de convertir diversos formatos de topologías de red al formato admitido por C-BGP y obtener información estadística de las mismas. Esta necesidad surgió tras el primer intento de validación del simulador *mC-BGP* utilizando dos topologías de gran tamaño <SUBRT><INETT>.

A lo largo del proyecto se le han ido añadiendo funcionalidades, y se ha modificado su arquitectura. En la Fig. 4 se puede apreciar el diagrama de componentes utilizado para el diseño de la aplicación en su estado actual.

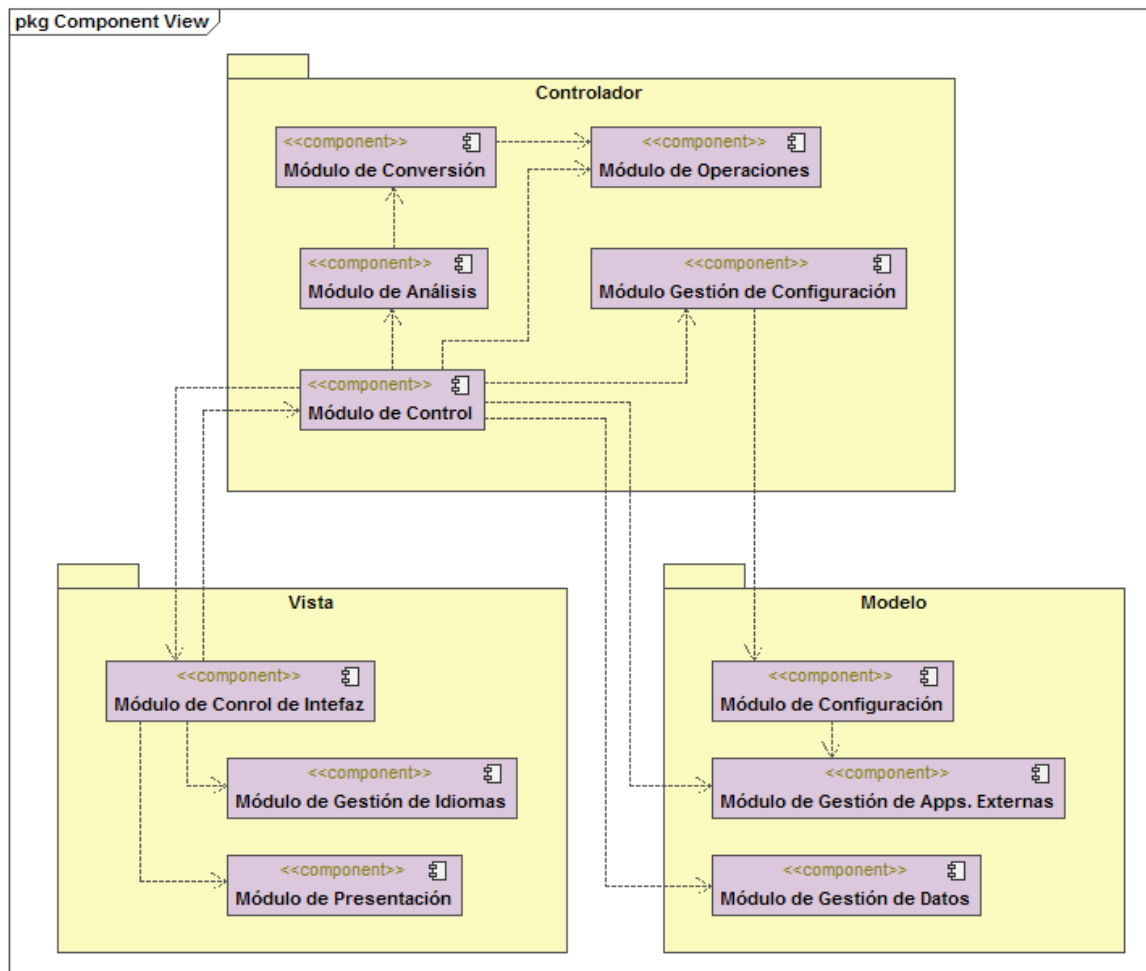


Fig. 4. Diagrama de componentes de StatOpology

En una fase inicial, la aplicación se ejecutaba en línea de comandos, recibiendo como entrada un fichero específico con datos de una topología de red. Los datos de dicho fichero son analizados, almacenados en memoria en una serie de estructuras de datos intermedias, optimizadas para realizar operaciones de inserción y búsqueda, y posteriormente, tras el acceso a dichas estructuras, los datos son convertidos al formato deseado. También se desarrolló un módulo para mostrar estadísticas sobre la topología analizada. Los datos mostrados son los siguientes:

- **Número de nodos** (Sistemas Autónomos o routers BGP para el simulador)
 - **Nodos de tipo "Tier-1"**: aquellos que no tienen ningún proveedor.
 - **Nodos de tipo "Gran ISP"**: aquellos que tienen un número mayor ó igual a 50 clientes.
 - **Nodos de tipo "Pequeño ISP"**: aquellos que tienen entre 50 y 5 clientes.
 - **Nodos de tipo "Reducido"**: aquellos que tienen menos de 5 clientes.
 - **Nodos de tipo "Hoja"**: aquellos que no tienen clientes, sólo proveedores.
- **Número total de enlaces** (entre nodos)
 - **Número medio de enlaces de tipo P2P**
 - **Número medio de enlaces hacia proveedores**
 - **Número medio de enlaces hacia clientes**

Estos datos pueden resultar de suma utilidad a la hora de evaluar una topología, pues proporcionan información sobre la conectividad entre los nodos de la red, el tipo de nodos que la componen y los valores totales de los mismos. Además se pueden obtener dos tipos de listado asociados a las distintas clases de nodo:

- IP correspondiente a cada Sistema Autónomo para el tipo de nodo elegido.
- AS_NUM (número de Sistema Autónomo) para el tipo de nodo elegido.

Con los listados obtenidos es posible obtener resultados específicos sobre las soluciones calculadas por el simulador, ya que gracias a ellos es posible determinar la clase a la que corresponde cualquier nodo de la topología. De esta forma, por ejemplo, es posible evaluar las rutas entre nodos hoja únicamente (ver métricas de la Tabla 17 en el apartado 4.4).

En paralelo al desarrollo de dicha ampliación se diseñó e implementó una funcionalidad aprovechando el conocimiento de la topología previamente analizada. Esta funcionalidad se ocupa de generar ficheros de comandos interpretables por el simulador C-BGP con el objetivo de lanzar simulaciones pre-configuradas mediante la herramienta, sin necesidad de hacerlo de forma manual. Como se verá en el apartado 4.2, esta funcionalidad ha sido ampliada varias veces para poder ejecutar simulaciones sobre topologías del mismo tamaño en magnitud que Internet.



Fig. 5. Ventana principal de StatOpology

La siguiente ampliación consistió en un "limpiador" de tablas de rutas obtenidas como resultado del simulador. El objetivo de dicho módulo es el de eliminar la información innecesaria de los resultados para permitir la importación directa de las rutas obtenidas en el software *Matlab* a fin de obtener datos y gráficas acorde a una serie de métricas (ver apartado 4.4) de evaluación de protocolos *multi-path* interdominio.

Para facilitar la usabilidad de *StatOpology* se decidió reestructurar, dotando a la aplicación de interfaz gráfica de usuario (ver Fig. 5). Inherentemente asociado a dicho cambio hubo que realizar un análisis y diseño siguiendo los paradigmas de la programación orientada a objetos, con el fin de conseguir una aplicación modular, extensible, reutilizable y fácilmente mantenible. Se aprovecharon las funcionalidades previas y se añadieron otras nuevas. Igualmente se desarrolló un logo en formato vectorial (svg) mediante la aplicación *InkScape* <INKS> y siguiendo algunas normas de diseño de logotipos sugeridas por un experto en la materia (ver [SIM05]). La parte principal del logo (la "o" central) es utilizada como icono de los ficheros asociados a la aplicación.



Fig. 6. Logo de StatOpology

La aplicación sigue una serie de normas de accesibilidad, entre las que se encuentran el acceso a los principales elementos de los menús mediante atajos de teclado y la posibilidad de utilizar la aplicación sin necesidad de ratón, es decir, únicamente con el teclado. También se ha pensado en la distribución de los elementos que componen la aplicación para conseguir una usabilidad lo más intuitiva posible.

Como funcionalidades nuevas se presenta la posibilidad de generar una imagen (tanto vectorial como *raster*) de la topología (ver Fig. 7) gracias a la herramienta *GraphViz* <GVIZ>. En caso de no disponer de dicha herramienta en el sistema, se generará un script interpretable por la misma para generar las imágenes asociadas a la topología.

Los gráficos generados siguen una serie de códigos de color para diferenciar de forma rápida y visual los distintos tipos de elementos que lo componen. A continuación se muestra una leyenda:








COLOR	TIPO DE ELEMENTO	SIGNIFICADO
	Círculo con borde	Nodo tipo "Tier-1"
	Círculo	Nodo tipo "Gran ISP"
	Círculo	Nodo tipo "Pequeño ISP"
	Círculo	Nodo de tipo "Reducido"
	Círculo	Nodo de tipo "Hoja"
	Línea con flecha	Enlace de cliente a proveedor
	Línea sin flecha	Enlace tipo <i>peering</i>

Tabla 12. Leyenda para los grafos generados con StatOpology

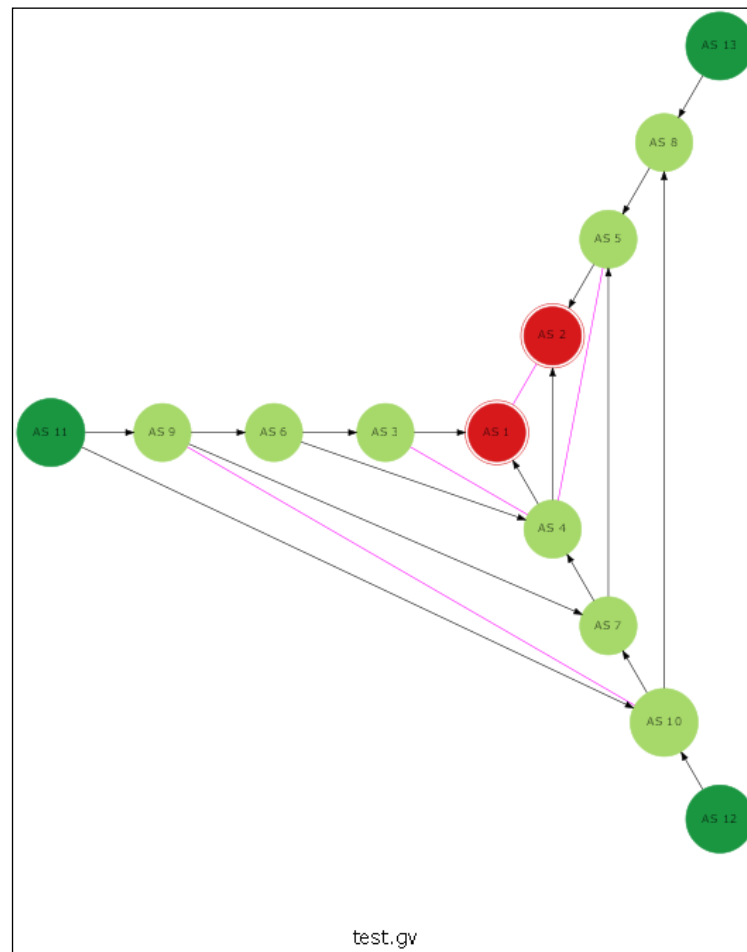


Fig. 7. Gráfico generado para una topología de prueba

También se encuentra en desarrollo un módulo para realizar el análisis de los resultados, con el fin de evitar la necesidad de utilizar Matlab a la hora de obtener datos para las métricas, así como refinar el apartado de configuración, en el cual se puede seleccionar el idioma de la aplicación, las rutas a programas externos (*GraphViz*, simulador C-BGP, etc.) o incluso el directorio de trabajo por defecto.

Es importante destacar que la aplicación cuenta con un menú de ayuda con dos funcionalidades: consultar el manual de usuario y acceder a un cuadro de diálogo con información sobre la aplicación y desde el que contactar con el autor del programa vía correo electrónico.

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, o las topologías de red.

3.4 Conclusiones

Como se ha podido observar, el proceso de adaptación del simulador requiere una serie de cambios importantes en algunos de los módulos que componen su arquitectura. Dichos cambios se basan principalmente en la modificación de estructuras de datos internas, añadiendo campos nuevos (como la FIB, o el contador de bucles detectados) y de modificaciones/adiciones de funciones para proporcionar características nuevas (como el selector de procesos de decisión en base al protocolo seleccionado, o las funciones de gestión de los AS_SETs). Estos cambios son una extensión de la versión inicial del simulador, esto es, se puede evaluar por defecto el protocolo convencional BGP. Por otro lado, los desarrolladores que deseen implementar variantes *multi-path* del protocolo pueden hacer uso de estas nuevas funciones y estructuras para conseguir su objetivo en muy poco tiempo, ampliando aún más las funcionalidades ofrecidas por el simulador.

Para poder validar el simulador se ha desarrollado una completa herramienta, que entre otras funcionalidades, permite convertir entre formatos las topologías de entrada al simulador y proporcionar datos estadísticos sobre las mismas. Se han seguido una serie de “buenas prácticas” en su diseño, de forma que sus módulos presenten una alta cohesión y bajo acoplamiento. Para ello se ha seguido un modelo de desarrollo software basado en el Proceso Unificado (ver [JBR00]), empleado patrones de diseño software (ver [GHJ95+]) en su arquitectura.

Capítulo 4

Validación del entorno

*"Su carencia de fe resulta molesta."
Darth Vader. (Star Wars: Ep. IV)*

4.1 Introducción

El proceso de validación del entorno puede dividirse en tantas partes como componentes lo forman, así pues, para realizar la verificación de que todo funciona como debe es necesario probar por un lado la herramienta *StatOpology* y por otro el simulador *mC-BGP*. En el primer caso, las pruebas han consistido en topologías de diversos tamaños, obtener información sobre las mismas y finalmente generar una serie de scripts útiles para llevar a cabo las simulaciones en *mC-BGP*. Para el simulador, el proceso de validación ha consistido en la implementación de las dos variantes *multi-path* propuestas por el proyecto Trilogy (LP-BGP y MpASS) y comprobar su funcionamiento empleando las topologías y scripts obtenidos de *StatOpology*. Es por ello que se presentará en primer lugar la validación de *StatOpology*, y a continuación la del simulador *mC-BGP*.

4.2 StatOpology

Como se comentó en el apartado 3.3, la herramienta *StatOpology* ha ido creciendo en funcionalidad poco a poco a lo largo del proyecto. Tras implementar cada una de las partes que la componen se realizaron pruebas para evaluar su correcto funcionamiento. Para cada módulo específico de la aplicación se realizaron pruebas de caja negra y de caja blanca. Dichas pruebas principalmente consistieron en evaluar los resultados obtenidos por cada función específica en cada caso específico y compararlos con los que se esperaba obtener, junto con una prueba de funcionalidades generales análoga a las anteriores, siendo éste un proceso de verificación de correcto funcionamiento.

Tras comprobar su correcto funcionamiento, *StatOpology* se empleó para generar los scripts y análisis de las topologías empleados en las pruebas con *mC-BGP*.

A continuación se muestra como ejemplo el análisis y prueba de funcionalidades de la herramienta para una topología extraída de [GR01], (ver Fig. 8). Esta topología ha sido seleccionada debido a que, como puede observarse en el grafo, presenta un enlace de *peering* entre los nodos 1 y 3 que puede originar ciclos entre ellos en el caso de que se configuren preferencias en la selección de rutas. En esta situación no está garantizada la convergencia del protocolo y podría resultar interesante observar qué ocurre al aplicar sobre la misma un enfoque *multi-path*.

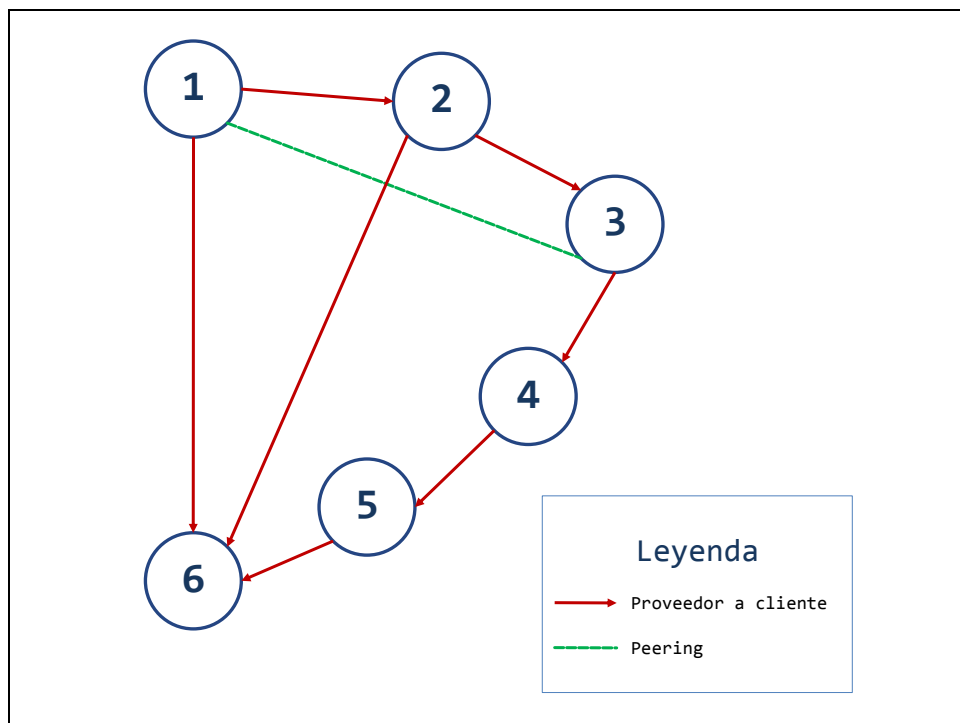


Fig. 8. Topología extraída de [GR01]

Las operaciones realizadas por **StatOpology** a lo largo de la validación han sido:

1. Obtención de la representación de la topología en un formato interpretable por *mC-BGP*

1	2	1
1	6	1
1	3	0
2	6	1
2	3	1
3	4	1
4	5	1
5	6	1

Tabla 13. Topología de red en formato interpretable por *mC-BGP*

2. Obtención de información estadística sobre la topología

```
==== StatOpology results ====

Topology file: 'rexford.rel'

Number of nodes: 6
- 'Tier-1' nodes: 1
- 'Large ISP' nodes: 0
- 'Small ISP' nodes: 0
- 'Tiny ISP' nodes: 4
- 'Leaf' nodes: 1

Number of links: 8
- Average lins per node (P2P):0.16666666666666666
- Average lins per node (to providers):1.1666666666666667
- Average lins per node (to customers):1.1666666666666667

=====
```

Tabla 14. Información sobre la topología

3. Obtención del grafo correspondiente a la topología y el código fuente empleado por la herramienta *GraphViz* para su generación

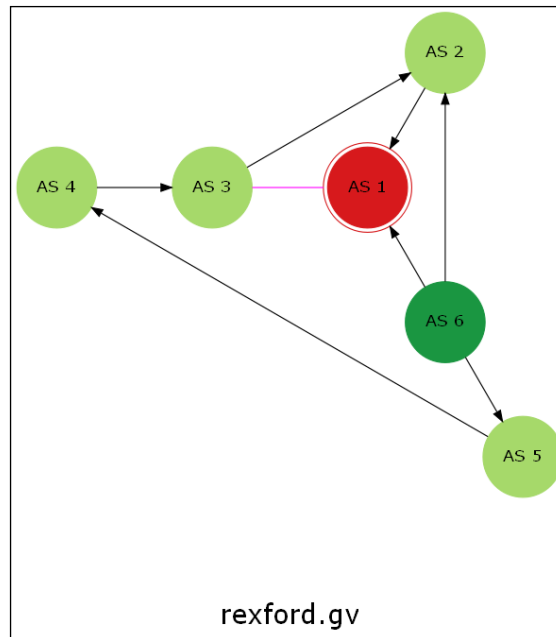


Fig. 9. Grafo generado para la topología

```
/* Generated by StatOpology. To be used with Graphviz or equivalent app. */  
  
digraph "rexford.gv" {  
    ranksep=2;  
    ratio=auto;  
    graph [ fontname = verdana, fontsize = 25, label = "\n\nrexford.gv", size = "7,7" ];  
    node [shape = doublecircle, style = filled, colorscheme = rdylgn5, color = 1, fontname = verdana, fontsize = 15]; "AS 1";  
    node [shape = circle, style = filled, colorscheme = rdylgn5, color = 2, fontname = verdana, fontsize = 15];  
    node [shape = circle, style = filled, colorscheme = rdylgn5, color = 3, fontname = verdana, fontsize = 15];  
    node [shape = circle, style = filled, colorscheme = rdylgn5, color = 4, fontname = verdana, fontsize = 15]; "AS 2" "AS 3"  
    "AS 4" "AS 5";  
    node [shape = circle, style = filled, colorscheme = rdylgn5, color = 5, fontname = verdana, fontsize = 15]; "AS 6";  
    "AS 2" -> "AS 1";  
    "AS 6" -> "AS 1";  
    "AS 3" -> "AS 1" [color = "magenta", arrowhead = "none"];  
    "AS 3" -> "AS 2";  
    "AS 6" -> "AS 2";  
    "AS 4" -> "AS 3";  
    "AS 5" -> "AS 4";  
    "AS 6" -> "AS 5";  
}
```

Tabla 15. Código fuente empleado por *GraphViz* para generar un grafo

4. Obtención del script a emplear en el simulador de red para probar la topología

```
bgp topology load "rexford.rel"
bgp topology install
bgp topology policiecs
bgp topology run
bgp router 0.1.0.0 add network 0.1/16
bgp router 0.2.0.0 add network 0.2/16
bgp router 0.3.0.0 add network 0.3/16
bgp router 0.4.0.0 add network 0.4/16
bgp router 0.5.0.0 add network 0.5/16
bgp router 0.6.0.0 add network 0.6/16
bgp router 0.1.0.0 set algorithm 2
bgp router 0.2.0.0 set algorithm 2
bgp router 0.3.0.0 set algorithm 2
bgp router 0.4.0.0 set algorithm 2
bgp router 0.5.0.0 set algorithm 2
bgp router 0.6.0.0 set algorithm 2
print "### Pre-sim. Queue info ###\n"
sim queue info
sim run
print "### Post-sim. Queue info ###\n"
sim queue info
print "### FIBs ###\n"
print "/// 1 (router 0.1.0.0 )\n"
bgp router 0.1.0.0 show fib
print "/// 2 (router 0.2.0.0 )\n"
bgp router 0.2.0.0 show fib
print "/// 3 (router 0.3.0.0 )\n"
bgp router 0.3.0.0 show fib
print "/// 4 (router 0.4.0.0 )\n"
bgp router 0.4.0.0 show fib
print "/// 5 (router 0.5.0.0 )\n"
bgp router 0.5.0.0 show fib
print "/// 6 (router 0.6.0.0 )\n"
bgp router 0.6.0.0 show fib
print "### Msgs ###\n"
show msgs
print "### Loops detected ###\n"
print "/// 1 (router 0.1.0.0 )\n"
bgp router 0.1.0.0 show loops
print "/// 2 (router 0.2.0.0 )\n"
bgp router 0.2.0.0 show loops
print "/// 3 (router 0.3.0.0 )\n"
bgp router 0.3.0.0 show loops
print "/// 4 (router 0.4.0.0 )\n"
bgp router 0.4.0.0 show loops
print "/// 5 (router 0.5.0.0 )\n"
bgp router 0.5.0.0 show loops
print "/// 6 (router 0.6.0.0 )\n"
bgp router 0.6.0.0 show loops
```

Tabla 16. Script para simular la topología en *mC-BGP*

Además de realizar una comprobación de las distintas funcionalidades presentadas por *Statology*, se validó la herramienta mediante la conversión y el análisis de una topología con más de 30.000 nodos (tamaño similar al de Internet). Dicha topología se obtuvo de <INETT>, y tras realizar un primer análisis de la misma con *Statology* se observaron una serie de detalles interesantes, mostrados a continuación.

En primer lugar, se encontraron una serie de relaciones repetidas. Dichas relaciones se mostraron en el log del programa y fueron automáticamente descartadas. Además de las relaciones repetidas, se encontraron diversos AS cuyo valor no es un número entero. Tras estudiar la RFC5396 en la que se describe el proceso deseable de nombrado para los AS (ver [HM08]) se decidió descartar dichos AS y sus relaciones asociadas. Como trabajo futuro se plantea la alternativa de modificar el analizador de *Statology* de modo que en lugar de descartar dichos AS y sus relaciones se utilice una re-numeración de los mismos a fin de que puedan seguir presentes en la topología convertida.

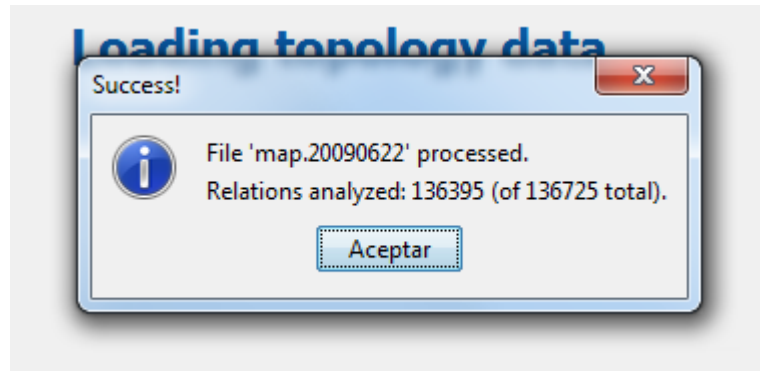


Fig. 10. Análisis de la topología de Internet

En la Fig. 10 se puede observar el resultado de importar (análisis) la topología de Internet en *Statology*. De todas las relaciones presentes en el fichero se descartaron 330 por poseer valores de AS con distinto formato al especificado, o bien por anunciar relaciones previamente definidas.

Una vez cargados los datos de la topología en *Statology* se procedió a la obtención de los scripts necesarios para realizar las simulaciones en *mC-BGP*. Se crearon dos tipos de script, necesarios para probar los enfoques implementados: LP-BGP y MpASS. En el apartado 4.5 se exponen las pruebas realizadas con dicha topología y los problemas encontrados a la hora de realizar la simulación en *mC-BGP*.

Igualmente se intentó generar el grafo asociado a la topología, mediante el uso combinado de *Statology* con *GraphViz*, pero debido a que las pruebas se realizaron sobre una versión de 32-bits de este último, no fue posible obtener las imágenes (vectorial y *raster*) al ser abortada la ejecución tras sobrepasar los 4 GB de memoria RAM. A pesar de ello, *Statology* produjo el script necesario para generar el grafo en versiones de 64-bit (o superiores).

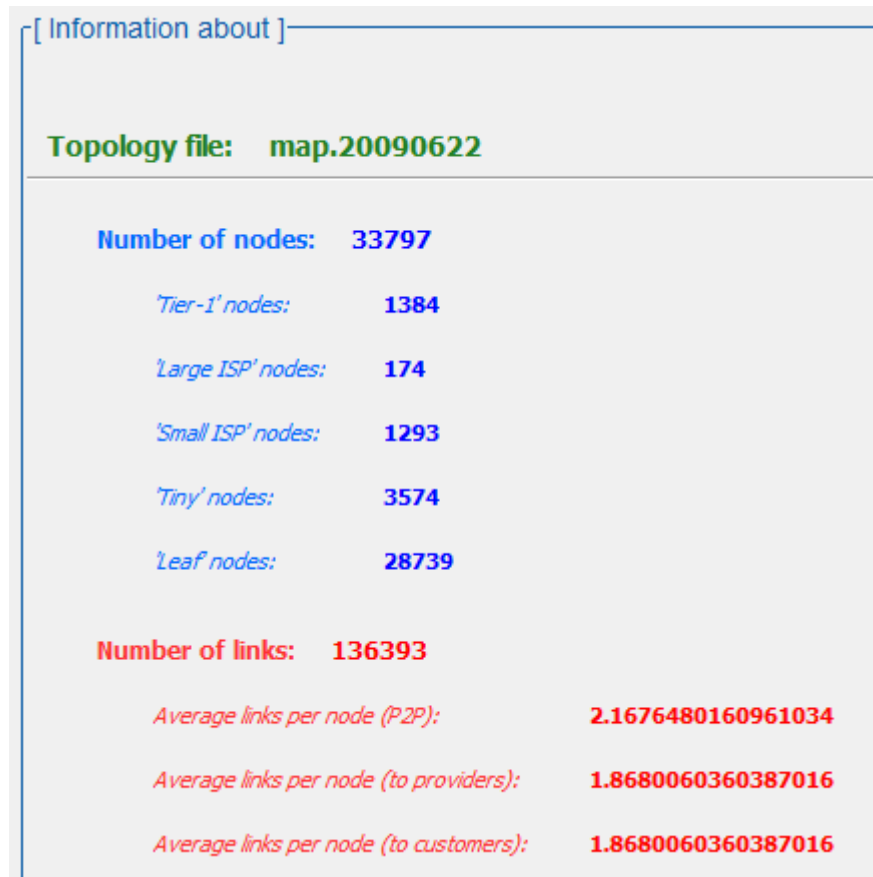


Fig. 11. Información estadística sobre la topología de Internet

En la Fig. 11 se muestran los valores estadísticos asociados a la topología de Internet. Se puede apreciar que la topología está compuesta por 33.797 Sistemas Autónomos, de los cuales 28.739 son nodos hoja. El número medio de enlaces es aproximadamente de 2 por nodo en cada tipo (proveedor a cliente y *peering*). Dicho valor es el correspondiente al grado medio del grafo, interesante para conocer la vecindad media de los nodos que componen la red.

4.3 mC-BGP

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 dichas implementaciones fue necesario añadir un proceso de decisión de reglas específico para cada variante, además de una función encargada de determinar si cuándo aplicar dicho proceso en base al protocolo seleccionado en el router. Estos cambios se realizaron en las secciones correspondientes a "*Grafo BGP & Config*" y "*Modelo del router BGP*" de la Fig. 3.

4.3.1 LP-BGP

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 [VGL00], 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.

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, dicho proceso de decisión se aplica en el orden siguiente:

- Preferir LOCAL-PREF más alto.
- Preferir AS-PATH más largo.
- Preferir ORIGIN más alto.
- Preferir MED más bajo.
- Preferir eBGP sobre iBGP.
- Preferir NEXT-HOP más cercano.

4.3.2 MpASS

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.

A continuación se muestra el proceso de decisión aplicado, en orden, para el filtrado inicial de las posibles rutas candidatas:

- Preferir LOCAL-PREF más alto.
- Preferir ORIGIN más alto.
- Preferir MED más bajo.
- Preferir eBGP sobre iBGP.
- Preferir NEXT-HOP más cercano.
- Preferir CLUSTER-ID-LIST más corta.
- Preferir dirección del vecino más baja.

4.4 Métricas de evaluación para las soluciones

Tras realizar la implementación de los dos enfoques *multi-path* propuestos en el proyecto Trilogy (LP-BGP y MpASS) comentados anteriormente se decidieron probar una serie de topologías para comprobar el resultado de dichos enfoques.

Con el objetivo de poder comparar los resultados de las simulaciones se aplicaron una serie de métricas desarrolladas para tal efecto. Dichas métricas se exponen en la siguiente tabla:

Métricas relacionadas con las rutas obtenidas
Desde cualquier AS hacia otro AS
Número medio de rutas diferentes que podría seguir un paquete
Número de rutas que comparten segmentos comunes
Número de rutas que tienen m nodos en común
Longitud media de las rutas obtenidas
Desde cualquier AS “hoja” hacia otro AS “hoja”
Número medio de rutas diferentes que podría seguir un paquete
Número de rutas que comparten segmentos comunes
Número de rutas que tienen m nodos en común
Longitud media de las rutas obtenidas
Métricas relacionadas con el intercambio de mensajes
Número de mensajes BGP procesados hasta la convergencia
Métricas relacionadas con el estado almacenado en los nodos
Número de entradas en la FIB de cada router para cada prefijo destino
Promedio del anterior por cada prefijo
Promedio de ASes involucrados en una ruta iniciada en un AS “hoja” y acabada en otro AS “hoja”
Número de bucles detectados

Tabla 17. Métricas de evaluación para las soluciones

En el proceso de generación de resultados acorde a dichas métricas se empleó el simulador de red *mC-BGP*, la herramienta *StatOpology* y se desarrollaron una serie de funciones en Matlab para el tratamiento de los datos generados por las dos aplicaciones anteriores a fin de obtener valores y gráficas concretas como solución a las métricas de los puntos 1, 3 y 4. El resto de valores se obtuvieron directamente del simulador *mC-BGP* o de *StatOpology*.

4.5 *Multi-path* BGP: pruebas y resultados

Para los experimentos de validación sobre las soluciones *multi-path* se diseñó una topología específica, de pequeño tamaño para hacer pruebas iniciales. Dicha topología puede verse en la Fig. 12. A la hora de realizar el diseño se tuvo en cuenta que existieran varios nodos de tipo “Tier-1”, varios nodos tipo “hoja” con distinto nivel de conectividad y posibilidad de ser alcanzados. Igualmente se establecieron una serie de enlaces de *peering* entre varios nodos a distintos niveles.

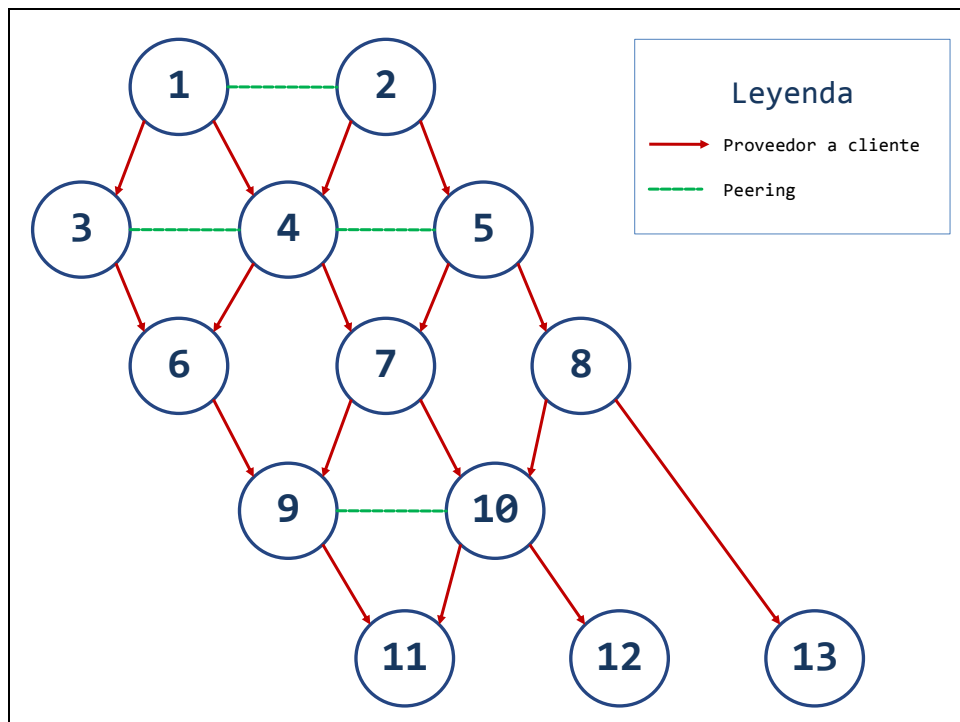


Fig. 12. Topología empleada inicialmente para la validación de soluciones *multi-path*

También se utilizó la topología de la Fig. 8. Esta topología puede producir situaciones inseguras en cuanto a convergencia (ver punto 4.2). De esta forma puede evaluarse el comportamiento de las variantes *multi-path* en circunstancias extremas para el caso de BGP convencional. Tanto para el enfoque propuesto por LP-BGP como por MpASS, se alcanzó la convergencia. Cabe destacar que tras cargar la topología en el simulador se emplearon los valores por defecto para enlaces y relaciones, es decir, no se realizó ningún tipo de configuración adicional.

Alternativamente se compararon los resultados obtenidos para la topología de la Fig. 12 con los obtenidos mediante la implementación de soporte *multi-path* en un router XORP <XORP> realizada por José Manuel Camacho Camacho, becario de doctorado del departamento de Ingeniería Telemática de la Universidad Carlos III de Madrid. En las pruebas realizadas para la validación se utilizaron variantes con diferentes valores de

configuración y filtrado (ej. diferentes LOCAL_PREFs). Los resultados obtenidos para esta topología coincidieron para ambos entornos, siendo las rutas instaladas en la FIB de los routers idénticas tanto en simulación como en emulación.

Los resultados de estabilidad y convergencia también se repitieron en ambos escenarios. Para configuraciones que provocan la pérdida de las propiedades "valley-free" de la topología se identificaron comportamientos no deseados en ambos protocolos. En concreto, para determinadas secuencias de eventos los protocolos oscilaban y no llegaban a converger. En otros órdenes de eventos no se cumplía que la solución a la que convergían los protocolos fuera determinista, dándose diferentes estados finales en las FIBs de los routers.

La no convergencia cuando se utilizaban estas configuraciones en particular se detectaba con *mC-BGP*, sin embargo la falta de trazas de mensajes entre routers durante el proceso de simulación imposibilitaba la identificación de las causas de la no convergencia. Gracias a la utilización del entorno de emulación, fue posible identificar las secuencias de eventos que daban lugar a las oscilaciones y a la falta de determinismo en los protocolos. Estos resultados han servido de base para futuras mejoras de los protocolos y para demostrar el potencial de estos entornos para la verificación de soluciones *multi-path BGP*.

4.5.1 Simulando “Internet”. Primer intento

Una vez realizadas las pruebas de validación con topologías de pequeño tamaño se procedió a realizar una verificación utilizando la topología de Internet <INETT>. El proceso de simulación para dicha topología se realizó en varias etapas, debido a la detección de una serie de problemas surgidos en el proceso de simulación.

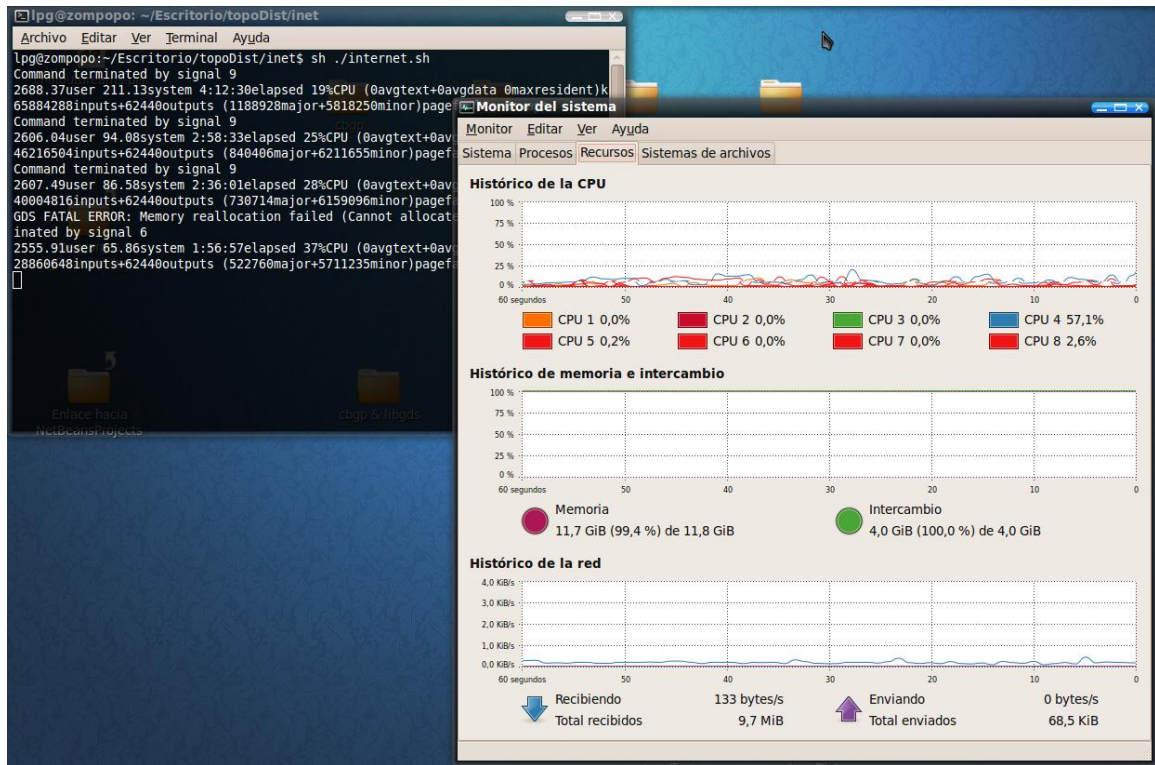


Fig. 13. Primer intento de simular la topología de Internet

El primer intento de simulación se lanzó la topología asignando cada AS a un nodo de la red, cada uno de ellos con un prefijo diferente. Se observó que al cabo de poco tiempo se llenaba completamente la memoria del sistema (ver Fig. 13), forzando dicha situación el cierre de la aplicación (*sigkill*). El problema es ocasionado porque se deben computar todas las entradas para todos los routers BGP de la topología, con el consiguiente crecimiento exponencial de las tablas de rutas y de la nueva tabla de *forwarding* implementada para almacenar las distintas posibles rutas a utilizar por cada nodo.

4.5.2 Simulando “Internet”. Segundo intento

Tras buscar información sobre cómo simular topologías de gran tamaño (ver [MFM06+]) se realizó un segundo intento previa modificación de la herramienta *StatOpology*. En dicha modificación se implementó la característica de generar scripts de simulación con inyección progresiva de prefijos, esto es, computar una serie de nodos, almacenar los resultados, añadir los prefijos correspondientes a otro conjunto de nodos, re-simular, y así sucesivamente hasta completar el proceso con todos los AS que conforman la topología. Tras simular se observó el mismo problema que en el caso anterior, es decir, la memoria se llenaba completamente provocando la salida forzada del simulador (ver Fig. 14), aunque en este caso ocurría al cabo de más tiempo.

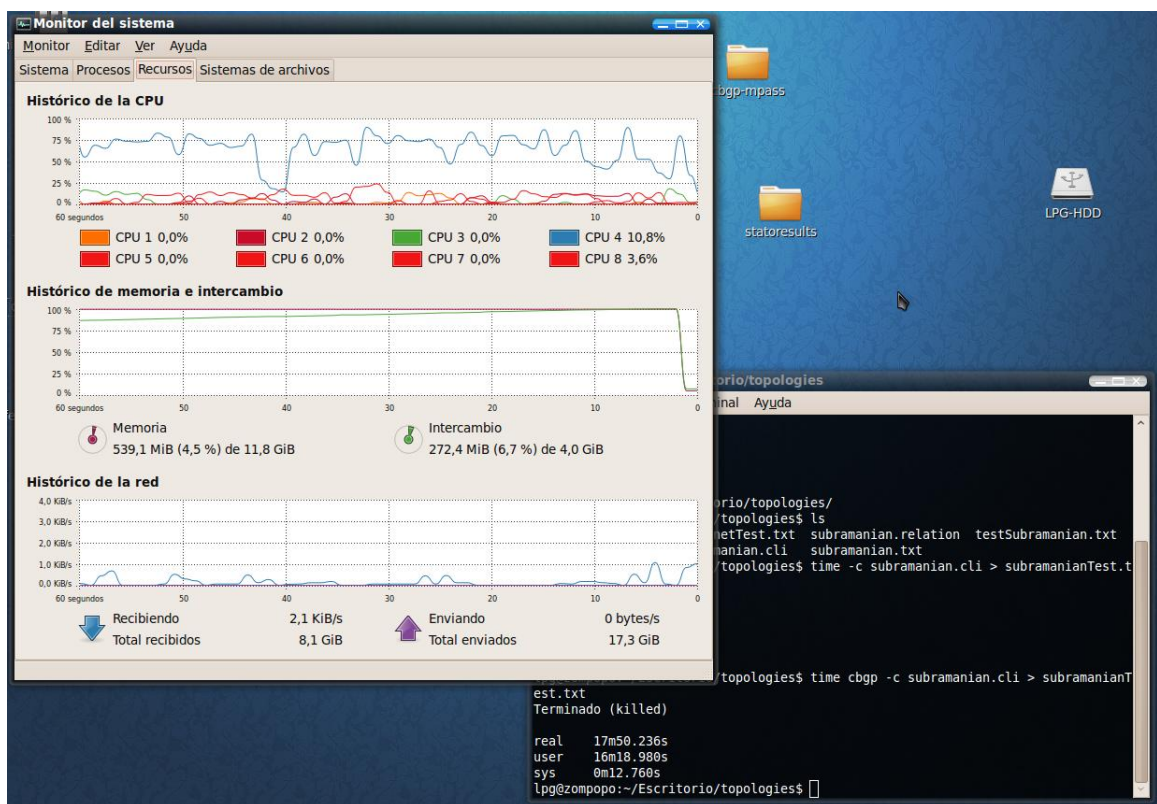


Fig. 14. Segundo intento de simular la topología de Internet. Inyección progresiva de prefijos

4.5.3 Simulando “Internet”. Tercer intento (éxito)

Tras comprobar que la inyección de prefijos tampoco era efectiva se implementó una segunda variante de generación de scripts en la herramienta *StatOpology*, para realizar simulaciones parciales sobre toda la topología pero con una serie de prefijos cada vez, almacenando los resultados parciales de las mismas para generar al final de todas las sub-simulaciones un resultado completo con todas las rutas por sistema autónomo (ver Fig. 15). En este caso la simulación resultó un éxito. Se realizaron 676 simulaciones parciales, siendo la duración media de cada una de ellas 9 minutos aproximadamente, y resultando en una duración total de 4 días, 1 hora, 14 minutos y 11 segundos.

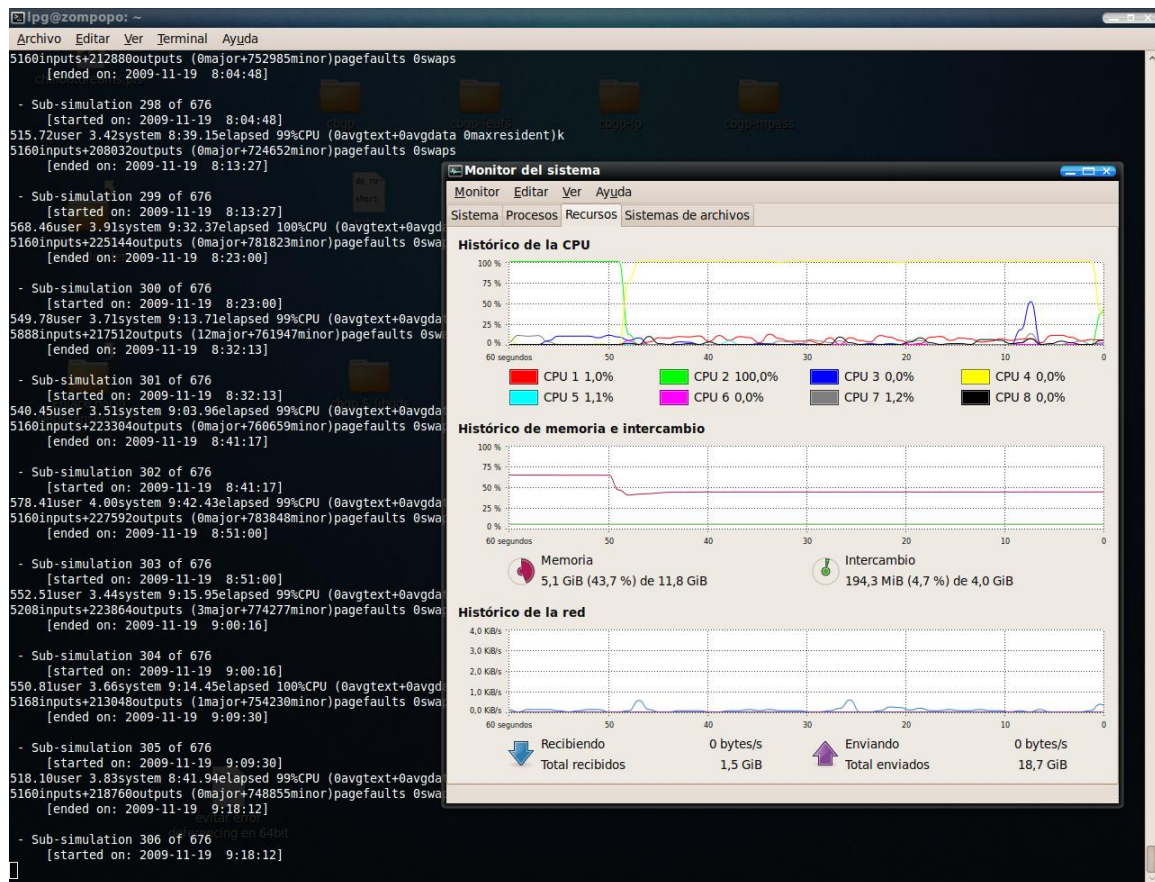


Fig. 15. Simulación de la topología de Internet. Sub-simulaciones de prefijos en la topología completa

En este caso, la solución combinada de todos los resultados de las simulaciones generó un fichero de texto de aproximadamente 72 GB con todas las tablas de rutas calculadas para los diferentes AS de la topología.

4.6 Conclusiones

Tras realizar los procesos de validación oportunos se comprobó que tanto el simulador *mC-BGP* como la herramienta *Statopology* funcionan como se esperaba.

mC-BGP ha proporcionado resultados sobre la convergencia de los protocolos *multi-path* implementados, y las tablas de encaminamiento calculadas por los routers BGP a lo largo de la simulación. *Statopology* por su parte ha permitido convertir y analizar una topología del tamaño de Internet, así como generar los scripts necesarios para ejecutar su simulación de manera satisfactoria. Se ha podido observar como en este caso, resultó necesario ampliar la funcionalidad de *Statopology* varias veces, hasta obtener un tipo de script adecuado para realizar dicha simulación, puesto que los enfoques iniciales saturaban la memoria del computador de pruebas.

Por otra parte, se han propuesto una serie de métricas que permiten comparar de una forma objetiva las soluciones generadas para los posibles protocolos *multi-path* implementados en el simulador. La obtención de los datos de dichas métricas se realiza mediante la herramienta *Statopology* y gracias a una serie de sencillos scripts realizados en Matlab. Como trabajo futuro queda pendiente la adaptación e inclusión de dichos scripts en *Statopology*, de forma que se puedan obtener los mismos resultados sin necesidad del software Matlab.

Capítulo 5

Presupuesto

*"¡Mira el tamaño de esa cosa!"
Wedge Antilles. (Star Wars: Ep. IV)*

5.1 Diagrama de tareas

A lo largo del presente proyecto se han descrito las distintas alternativas existentes para realizar simulaciones sobre redes de comunicaciones y se ha justificado mediante diversos análisis comparativos por qué se ha elegido C-BGP como base de un sistema de simulación para determinar cómo se comportarán diferentes protocolos de encaminamiento *multi-path*. Se han descrito las modificaciones necesarias realizadas al simulador, los distintos problemas encontrados, las soluciones propuestas y se ha verificado su correcto funcionamiento una vez implementado. Se podría considerar que a lo largo del proyecto se han dado fases alternativas de búsqueda de información, análisis de problemas, diseño de soluciones e implementación de las mismas, seguidas de numerosas verificaciones. En paralelo a dichas fases, ha estado siempre presente la documentación de las tareas realizadas.

Para elaborar la planificación del proyecto se ha utilizado la herramienta Microsoft Office Project Professional 2007. A continuación se muestra el diagrama de Gantt generado mediante dicha herramienta, con las distintas tareas desarrolladas a lo largo del proyecto:

	Nombre de tarea	Duración
1	[-] Análisis del problema	54 días
2	Recopilación de información sobre BGP	28 días
3	Estudio de alternativas multipath	6 días
4	Diseño de una posible solución	4 días
5	[-] Virtualización	26 días
6	Análisis del hardware a virtualizar	1 día
7	Virtualización del hardware	5 días
8	Estudio comparativo sobre sistemas de virtualización existentes	15 días
9	Pruebas de maqueta virtual	5 días
10	[-] Simulación	365 días
11	Búsqueda de simuladores aplicables	5 días
12	Estudio comparativo sobre simuladores existentes	4 días
13	[-] Modificación de C-BGP	180 días
14	Análisis del código fuente de C-BGP	110 días
15	Diseño de nuevas estructuras de datos y funciones necesarias para dar soporte multipath	15 días
16	Implementación del código	30 días
17	Validación y pruebas	25 días
18	[-] Desarrollo de StatOpology	175 días
19	Captura y análisis de requisitos de usuario	7 días
20	Diseño de la aplicación	4 días
21	Implementación	6 días
22	Validación y pruebas	1 día
23	Captura y análisis de nuevos requisitos	2 días
24	Ampliación del diseño de la aplicación	2 días
25	Implementación de las mejoras	3 días
26	Validación y pruebas	1 día
27	Captura y análisis de nuevos requisitos	5 días
28	Ampliación del diseño de la aplicación	8 días
29	Implementación de las mejoras	15 días
30	Validación y pruebas	1 día
31	Desarrollo del manual de usuario	3 días
32	[-] Adaptación de LP-BGP en C-BGP	43 días
33	Análisis del código fuente de C-BGP	23 días
34	Diseño de cambios necesarios	5 días
35	Implementación del código	10 días
36	Validación y pruebas	5 días
37	[-] Adaptación de MpASS en C-BGP	70 días
38	Análisis del código fuente de C-BGP	10 días
39	Diseño de cambios necesarios	20 días
40	Implementación del código	25 días
41	Validación y pruebas	15 días
42	Diseño e implementación de topologías de prueba	1 día
43	Obtención de resultados (acorde a métrica definida)	13 días
44	Validación / verificación de resultados	5 días
45	Documentación	459 días

Fig. 16. Desglose de tareas llevadas a cabo y su duración en días



Fig. 17. Diagrama de Gantt con las distintas fases del proyecto

5.2 Desglose de costes

Como se puede apreciar en el diagrama anterior, la duración del proyecto ha sido de aproximadamente 20 meses, con una media de trabajo de cuatro horas diarias. A lo largo de este tiempo han surgido una serie de gastos que ha sido necesario cubrir. Las estimaciones sobre los mismos se exponen a continuación.

5.2.1 Personal

Para la realización del proyecto se ha necesitado cubrir las tareas específicas realizadas por un analista de sistemas, un arquitecto de software, un diseñador, un programador y un experto en calidad y pruebas. Además, ha sido necesario un tutor que ha realizado el trabajo de jefe de proyecto, proporcionando asistencia en los momentos necesarios.

A continuación se muestra una estimación de los costes que generan cada uno de esos roles por hora de trabajo:

Personal	
Cargo	Coste (por hora)
Analista	35,00 €
Arquitecto	40,00 €
Diseñador	40,00 €
Gestión de calidad y pruebas	35,00 €
Jefe de proyecto	45,00 €
Programador	25,00 €

Tabla 18. Estimación de coste por cada tipo de trabajo / hora

Aplicando estos costes al desglose de horas mostrado en la siguiente tabla:

Actividades y recursos							
Actividad	Personal						Totales
	Jefe de proyecto	Analista	Arquitecto	Diseñador	Pruebas	Programador	
Análisis del problema	20	138	12	0	0	0	170
Recopilación de información sobre BGP		110	0	0	0	0	110
Estudio de alternativas <i>multi-path</i>		20	4	0	0	0	24
Diseño de una posible solución		8	8	0	0	0	16
Uso de máquinas virtuales	10	42	0	0	60	0	112
Análisis del hardware a virtualizar		2	0	0	0	0	2
Virtualización del hardware		20	0	0	0	0	20
Análisis de sistemas de virtualización		20	0	0	40	0	60
Pruebas de maqueta virtual		0	0	0	20	0	20
Uso de simuladores de red	120	458	100	32	172	516	1398
Búsqueda de simuladores aplicables		20	0	0	0	0	20
Análisis de simuladores existentes		15	0	0	0	0	15
Modificación de C-BGP		240	80	12	80	300	712
Desarrollo de StatOpology		40	20	20	12	80	172
Adaptación de LP-BGP en C-BGP		40	0	0	30	40	110
Adaptación de MpASS en C-BGP		80	0	0	50	80	210
Diseño e implementación de topologías de prueba		1	0	0	0	0	1
Generación de resultados (métrica)		10	0	0	0	16	26
Validación de resultados		12	0	0	0	0	12
Horas totales	150	638	112	32	232	516	1680
Coste	6.750,00 €	22.330,00 €	4.480,00 €	1.280,00 €	8.120,00 €	12.900,00 €	55.860,00 €

Tabla 19. Coste asociado a las actividades y recursos empleados

El coste derivado del personal asciende a la cantidad de 55.860 euros.

5.2.2 Material

El coste derivado del uso del material puede ser dividido en tres apartados: por un lado el coste de los dispositivos hardware empleados a lo largo del proyecto, por otro, las licencias de software necesarias para llevarlo a cabo y finalmente los materiales fungibles, entre los que podemos encontrar los discos vírgenes empleados para grabar información, memorias flash y material de oficina (impresiones, papel, bolígrafos...).

5.2.2.1 Equipos

Para estimar el coste de los equipos se ha realizado el cálculo de la amortización de su precio respecto al tiempo que han sido usados en base al coste supuesto en el momento de adquisición.

En la siguiente tabla se muestra el coste derivado del uso de los equipos:

Equipos					
Nombre	Precio unitario	Unidades	Coste(ud.)/mes	Nº meses	Total
Servidor 2x Intel Xeon, 48 GB DDR3	4.500,00 €	1	125,00 €	2	250,00 €
PC Intel core i7, 12 GB DDR3	1.300,00 €	1	36,11 €	18	650,00 €
PC Intel Core 2 Quad, 8 GB DDR3	1.100,00 €	1	30,56 €	20	611,11 €
NetBook Asus EeePC 1201N	435,00 €	1	12,08 €	5	60,42 €
Portátil Acer Aspire 5672WLMi	736,90 €	1	20,47 €	3	61,41 €
Router Linksys WRT54g	56,90 €	1	1,58 €	2	3,16 €
Total					1.636,10 €

Tabla 20. Coste asociado al equipamiento hardware empleado

El coste derivado del uso de equipos asciende a la cantidad de 1.636,10 euros.

5.2.2.2 Software

Se ha intentado abaratar al máximo en cuanto a coste por software se refiere. Para ello se han empleado un gran número de programas de código abierto bajo el sistema operativo GNU/Linux, así como el máximo posible de aplicaciones *freeware* bajo Windows. Los costes por licencia del software más significativo se desglosan en la siguiente tabla:

Software			
Concepto	Licencias	Coste/licencia	Coste
Microsoft Windows 7 Professional (*)	3	0,00 €	0,00 €
Microsoft Windows Vista Pro. (*)	2	0,00 €	0,00 €
Debian GNU/Linux	1	0,00 €	0,00 €
Ubuntu GNU/Linux	3	0,00 €	0,00 €
Microsoft Office Project Pro. 2007 (*)	1	0,00 €	0,00 €
Microsoft Office 2007	2	155,00 €	310,00 €
NetBeans IDE	1	0,00 €	0,00 €
Eclipse IDE	1	0,00 €	0,00 €
Total			310,00 €

Tabla 21. Coste asociado al software empleado

(*) Las licencias de estos productos *Microsoft* se obtuvieron gracias al acuerdo *MSDNAA* entre *Microsoft Corp.* y la *Universidad Carlos III de Madrid*.

En el apartado 1.4.2 se puede apreciar un desglose detallado de las aplicaciones empleadas para llevar a cabo el proyecto. Las aplicaciones que no se muestran en la Tabla 21 se proporcionan bajo licencia gratuita de uso o han sido utilizadas en sus periodos de prueba.

El coste derivado del uso de software asciende a la cantidad de 310 euros.

5.2.2.3 Fungible

En este apartado se incluyen los costes por memorias flash, discos duros externos, y CD / DVD vírgenes utilizados para transportar datos, así como los costes originados por el uso de material de oficina. Esto es, impresiones, fotocopias, encuadernaciones, bolígrafos, etc.

En la siguiente tabla se muestra una estimación de este tipo de costes a lo largo del proyecto:

Material fungible			
Concepto	Cantidad	Coste unitario	Total
Memorias USB	1	37,00 €	37,00 €
Discos vírgenes (CD / DVD)	15	0,44 €	6,60 €
Disco duro portátil 320 GB	1	97,90 €	97,90 €
Material de oficina	1	50,00 €	50,00 €
Total			191,50 €

Tabla 22. Coste asociado al material fungible

El coste derivado del uso de material fungible asciende a la cantidad de 191,50 euros.

5.2.3 Transporte

Para acudir al lugar de trabajo se empleó el transporte público y un vehículo propio. Para los desplazamientos mediante transporte público se adquirieron abonos transporte mensuales tipo B2. En el caso del transporte mediante vehículo propio (Seat Ibiza 100 c.v.), se estima que el coste de carburante al mes es de unos 40 euros.

Transporte	
Tipo	Coste
Público (abono transporte)	210,00 €
Privado (vehículo propio)	760,00 €
Total	970,00 €

Tabla 23. Coste asociado al transporte durante el proyecto

El coste derivado del transporte a lo largo de los 20 meses de duración del proyecto asciende a la cantidad de 970 euros.

5.2.4 Costes indirectos

En esta categoría se incluyen los costes por alquiler del local completamente acondicionado para el trabajo, durante los 20 meses de duración de proyecto, el consumo de luz y agua, el servicio de limpieza, la red de telefonía y comunicaciones necesaria, así como un colchón económico para cubrir posibles bajas del personal y un seguro a todo riesgo.

Costes indirectos	
Concepto	Coste
Alquiler de local	4.800,00 €
Luz, agua	1.200,00 €
Servicio de limpieza	1.200,00 €
Teléfono fijo y red de comunicaciones	1.400,00 €
Cobertura por bajas (* ¹)	5.800,00 €
Seguro a todo riesgo (* ²)	160,00 €
Total	14.560,00 €

Tabla 24. Costes indirectos

(*¹) En la cobertura por bajas se aplica en caso de tener que contratar personal para cubrir bajas por enfermedad o incapacidad.

(*²) El seguro a todo riesgo incluye cobertura de personal y bienes materiales.

La cantidad económica asociada a los costes indirectos a lo largo del proyecto asciende a la cifra de 14.560 euros.

5.2.5 Resumen de costes

Calculando la suma económica necesaria para cubrir los costes derivados de los apartados anteriores, se necesitan 73.527,60 euros para realizar el proyecto. Sin embargo, esta cantidad no incluye el impuesto sobre valor añadido ni los márgenes de beneficio y riesgo asociados a este tipo de proyectos.

Resumen de costes	
Concepto	Cantidad
Mano de obra	55.860,00 €
Equipos	1.636,10 €
Software	310,00 €
Fungible	191,50 €
Transporte	970,00 €
Costes indirectos	14.560,00 €
Total	73.527,60 €

Tabla 25. Resumen de costes asociados al proyecto

5.2.6 Totales

Llegados a este punto se puede obtener el precio final necesario para llevar a cabo este proyecto. Se muestra en la siguiente tabla:

Presupuesto del PFC	
Concepto	Cantidad
Coste total	73.527,60 €
Riesgo (20%)	14.705,52 €
Beneficio (20%)	14.705,52 €
Total sin I.V.A.	102.938,64 €
I.V.A. (16%)	16.470,18 €
Total	119.408,82 €

Tabla 26. Presupuesto del PFC

Teniendo en cuenta los costes desglosados en los apartados anteriores, el presupuesto total de este proyecto asciende a la cantidad de CIENTO DIECINUEVE MIL CUATROCIENTOS Y OCHO CON OCHENTA Y DOS euros.

Leganés a 31 de Marzo de 2010

El ingeniero proyectista



Fdo. Lisardo Prieto González

Capítulo 6

Conclusiones y trabajo futuro

*"La fuerza estará contigo... siempre."
Obi-Wan Kenobi. (Star Wars: Ep. IV)*

El objetivo principal de la propuesta es obtener un entorno con el que poder comparar diferentes soluciones *multi-path* BGP. De esta forma, no sólo es posible extraer automáticamente una serie de métricas para cada una, objetivando así la comparación, sino que también resulta posible combinar dichas soluciones y evaluar su funcionamiento conjunto.

Al comienzo del proyecto se buscaron alternativas factibles para el desarrollo de un entorno de evaluación para protocolos de encaminamiento *multi-path* interdominio. Ante la ausencia de soluciones existentes, se barajaron distintas posibilidades sobre cómo afrontar el problema. Se descartaron por motivos económicos y de escalabilidad algunas de ellas, y se llegó a la conclusión de que la simulación por software es una herramienta potente, que cumple los requisitos deseados, y además, mucho más económica que el resto de las alternativas.

El desarrollo de este proyecto ha satisfecho los objetivos propuestos, cumpliendo los requisitos establecidos inicialmente. El resultado ha sido un simulador del protocolo BGP con soporte *multi-path* y una completa herramienta de análisis, *Statology*. Con esta última es posible convertir distintos formatos de topologías al aceptado por el simulador, obtener información estadística sobre las mismas, generar scripts que permitan realizar las simulaciones de forma automática, e incluso interpretar los resultados generados por el simulador de forma que sea posible la aplicación directa de algunas de las métricas de evaluación, o bien proporcionar parte de dichos resultados interpretados a una serie de

scripts en lenguaje Matlab que permiten obtener los resultados para las métricas restantes. Además, dicha herramienta ha sido diseñada siguiendo una serie de metodologías y paradigmas de desarrollo software que la hacen fácilmente mantenible, modificable y escalable. Cabe destacar que **StatOpology** dispone de una interfaz gráfica accesible e intuitiva, la cual permite realizar al usuario tareas complejas de forma sencilla.

Para validar la propuesta se analizaron una serie de topologías con **StatOpology**, y a continuación, se simularon en *mC-BGP*. En el caso de éste último, el proceso de validación consistió en la implementación de dos soluciones *multi-path* concretas que se están desarrollando en el proyecto Trilogy: LP-BGP y MpASS. Los resultados para una de las topologías simuladas pudieron ser contrastados con los obtenidos en un proyecto paralelo, consistente en la implementación de dichas variantes *multi-path* en routers software (XORP). Ambos resultados han resultado evaluados y son equivalentes.

Con el objetivo de probar la escalabilidad de la solución, se probó a simular una topología del tamaño de Internet publicada en <INETT>. Tras probar con diferentes enfoques, se consiguió realizar la simulación de forma exitosa, aunque el tiempo empleado fue de aproximadamente 4 días. En cualquier caso, los resultados fueron obtenidos en una única máquina. Si se hubiera empleado alguna de las alternativas propuestas al principio del proyecto, hubiera sido necesario emplear una cantidad considerablemente superior de recursos hardware para obtener estos resultados.

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 topología de Internet anteriormente mencionada son muy optimistas, pero el proceso de simulación, por ejemplo, consume muchos recursos y se considera que puede todavía optimizarse más). También cabe destacar que debido a las modificaciones realizadas en el simulador, es posible probar nuevas propuestas *multi-path* para BGP, de una forma sencilla y relativamente rápida

De una forma paralela, se están añadiendo funcionalidades ya implementadas a la interfaz gráfica de la herramienta **StatOpology**, con el fin de hacerla más completa y versátil. Entre otras características se pretende añadir una serie de módulos encargados de obtener los resultados necesarios para las métricas restantes sin necesidad de recurrir a Matlab. También es deseable la opción de lanzar las simulaciones de forma automática desde la herramienta, tras haberlas configurado previamente en la interfaz.

Por último, se está trabajando en la documentación exhaustiva de **StatOpology**, para poder proporcionar una completa API de desarrollo mediante el uso de *Javadoc* a cualquier persona que quiera trabajar sobre la herramienta.

Glosario

*"Domino más de seis millones de formas de comunicación y ésta no es una clave usada por la Alianza... Podría tratarse de un código imperial."
C-3PO. (Star Wars: Ep. V)*

Accesibilidad	Grado en el que todas las personas pueden utilizar un objeto independientemente de sus capacidades técnicas, cognitivas o físicas.
Agregación	Acción de insertar una ruta en la tabla de rutas.
Array	Estructura de datos formada por una zona de almacenamiento contiguo que contiene una serie de elementos del mismo tipo.
AS	Del inglés " <i>Autonomous System</i> ". Ver Sistema Autónomo .
BGP	Protocolo de red mediante el cual se intercambia información de encaminamiento entre sistemas autónomos.
Cola	Estructura de datos en la que el orden de inserción de elementos determina el orden de extracción (primero en entrar primero en salir).
Convergencia	En una red, estado en el que se ha conseguido el equilibrio entre los distintos nodos que la componen.
Dual-Channel	Tecnología para memorias aplicada a computadores que permite el incremento de rendimiento gracias al acceso simultáneo a dos módulos distintos de memoria.
Encaminamiento	Función de buscar un camino entre todos los posibles en una red.
Escalabilidad	Propiedad de un sistema que indica la habilidad para extender el margen de operaciones sin perder calidad.

IGP	Del inglés " <i>Interior Gateway Protocol</i> " hace referencia a los protocolos de encaminamiento utilizados por los Sistemas Autónomos. Dichos protocolos pueden ser de "estado de enlace" o de "vector de distancias".
Ingeniería de tráfico	Funciones necesarias para planificar, diseñar, proyectar, dimensionar, desarrollar y supervisar redes de telecomunicaciones en condiciones óptimas de acuerdo a la demanda de servicios, márgenes de beneficios de la explotación, calidad de la prestación y entorno regulatorio y comercial.
ISP	Del inglés " <i>Internet Service Provider</i> ". Empresa que brinda conexión a Internet a sus clientes.
LP-BGP	Modificación <i>multi-path</i> del protocolo BGP que evita los bucles mediante la propagación de la ruta más larga. (ver [BCB08+])
Mantenibilidad	Propiedad de un sistema que representa la cantidad de esfuerzo requerido para conservar su funcionamiento normal o para restituirlo en caso de fallo.
Métrica	Propiedad asociada a una ruta que puede ser utilizada para su evaluación y comparación con otras.
MpASS	Modificación <i>multi-path</i> del protocolo BGP que evita los bucles mediante la propagación de AS_SETs. (ver [VBG08+])
OpenWRT	Distribución GNU/Linux orientada a dispositivos embebidos. (ver <OWRT>)
Prefijo	Parte inicial de una dirección de red. Utilizado en asignación de direcciones y encaminamiento.
Protocolo	Conjunto de estándares que controlan la secuencia de mensajes que ocurren durante una comunicación entre las entidades que forman una red.
Raster	Aplicado a imágenes digitales, es una estructura o fichero de datos que representa una rejilla rectangular de píxeles (puntos de color).
SATA	Del inglés " <i>Serial Advanced Technology Attachment</i> ". Interfaz de transferencia de datos entre la placa base de un computador y algunos dispositivos de almacenamiento, como el disco duro.
Sistema Autónomo	Conjunto de redes y dispositivos router IP que se encuentran administrados por una sola entidad (típicamente un ISP), o en algunas ocasiones varias, que poseen una política de rutas propia e independiente. Cada Sistema Autónomo es como una Internet en pequeño.

SSH	Del inglés “ <i>Secure Shell</i> ”. Es el nombre de un protocolo y del programa que lo implementa. Sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora remota mediante un intérprete de comandos a través de un canal de datos seguro.
SVG	Del inglés “ <i>Scalable Vector Graphics</i> ”. Especificación para describir gráficos vectoriales bidimensionales en formato XML.
Topología	Disposición física en la que se conecta una red de dispositivos.
Tri-Channel	Tecnología para memorias aplicada a computadores que permite el incremento de rendimiento gracias al acceso simultáneo a tres módulos distintos de memoria.
UML	Del inglés “ <i>Unified Modeling Language</i> ”. Es el lenguaje de modelado de sistemas software más empleado en la actualidad.
Usabilidad	Facilidad con que las personas pueden utilizar una herramienta específica con el fin de alcanzar un objetivo concreto.
Vectorial	Aplicado a imágenes digitales, son aquellas formadas por objetos geométricos independientes (segmentos, polígonos, arcos, etc.), cada uno de ellos definidos por distintos atributos matemáticos de forma, posición, color, etc.
VNUML	Del inglés “ <i>Virtual Network User-Mode Linux</i> ”. Es una herramienta de virtualización de propósito general orientada al ámbito de redes. Está disponible únicamente en el sistema operativo GNU/Linux.
XML	Del inglés “ <i>Extensible Markup Language</i> ”. Metalenguaje extensible de etiquetas desarrollado por el W3C.
Zebra / Quagga	Paquete de software libre que gestiona protocolos de encaminamiento basados en TCP/IP. Soporta entre otros BGP-4. (ver <ZBRA>)

Referencias

"¿Quién es más loco: el loco o el loco que sigue al loco?"
Obi-Wan Kenobi. (Star Wars: Ep. IV)

- [BBE09+] Marcelo Bagnulo, Louise Burness, Philip Eardley, Alberto García-Martínez, Francisco Valera and Rolf Winter. "Joint Multi-path Routing and Accountable Congestion Control". ICT-Mobile Summit 2009. June 2009, Santander, Spain
- [BCB08+] *Multi-Pathology by Propagating BGP's Longest Path*. Iljitsch van Beijnum, Jon Crowcroft, Marcelo Bagnulo, Francisco Valera. 2008
- [BCV09+] 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
- [BEI10] Iljitsch van Beijnum. "BGP. Building Reliable Networks with the Border Gateway Protocol". Caps. 1, 2, 4, 5 y 10. Ed. O'Reilly Media. Marzo 2010. Formato e-book. ISBN: 9781449383954. Disponible [Internet]: <<http://my.safaribooksonline.com/9780596002541>> [07 de mayo de 2010]
- [BGB94] Albrightson B., Garcia-Luna-Aceves J., Boyle J. "EIGRP-A fastrouting protocol based on distance vectors". In Proc. Network/Interop 94, Las Vegas. (1994). Proceedings. 136-147.
- [BRC00+] T. Bates, Y. Rekhter, R. Chandra, D. Katz. *Multiprotocol Extensions for BGP-4*. Internet Engineering Task Force. Request for Comments: 2858. Junio 2000. Disponible [Internet]: <<http://tools.ietf.org/html/rfc2858>> [07 de mayo de 2010]
- [CBGPA] *Arquitectura del simulador C-BGP*. Disponible [Internet]: <<http://cbgp.info.ucl.ac.be/architecture.php>> [07 de mayo de 2010]

- [DOWRT] *Documentación online de OpenWRT*. 2. Development Issues. Disponible [Internet]: <<http://kamikaze.openwrt.org/docs/openwrt.html>> [07 de mayo de 2010]
- [DSA] *Diccionario de Sinónimos y Antónimos*. Espasa Calpe 2005. Disponible [Internet]: <<http://www.wordreference.com/sinonimos/>> [07 de mayo de 2010]
- [FRH10] 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>> [07 de mayo de 2010]
- [GHJ95+] E. Gamma, R. Helm, R. Johnson, J. M. Vlissides. “*Design Patterns: Elements of Reusable Object-Oriented Software*”. Ed. Addison-Wesley Professional. 1995. ISBN: 9780201633610
- [GR01] L. Gao, J. Rexford, “Stable Internet Routing Without Global Coordination”, IEEE/ACM Transactions on networking, Vol.9, No.6, Dec. 2001
- [HM08] G. Huston, G. Michaelson. *Textual Representation of Autonomous System (AS) Numbers*. Internet Engineering Task Force. Request for Comments: 5396. Enero 2008. Disponible [Internet]: <<http://www.ietf.org/rfc/rfc5396.txt>> [07 de mayo de 2010]
- [JBR00] I. Jacobson, G. Booch, J. Rumbaugh. *El Proceso Unificado de Desarrollo de Software*. Pearson Addison-Wesley. Año 2000. ISBN: 9788478290369
- [MFM06+] 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/~muehlbau/sigcomm06.pdf>> [07 de mayo de 2010]
- [PSA07+] Psenak P., Mirtorabi S., Roy A., Nguyen L., Pillay-Esnault P. “*Multi-Topology (MT) Routing in OSPF*”. RFC4915.(2007).
- [RAE07] Real Academia de la Lengua Española. *Diccionario de la Lengua Española*. Edición 23^a. 2007. ISBN: 84-670-0039-2. Disponible [Internet]: <<http://www.rae.es>> [07 de mayo de 2010]
- [RLH06] Y. Rekhter, T. Li, S. Hares. *A Border Gateway Protocol 4 (BGP-4)*. Internet Engineering Task Force. Request for Comments: 4271. Enero 2006. Disponible [Internet]: <<http://www.ietf.org/rfc/rfc4271.txt>> [07 de mayo de 2010]
- [SIM05] G. Simon. “*How to make an awesome logo*”. Disponible [Internet]: <<http://www.garysimon.net/logotutorial/>> [07 de mayo de 2010]
- [TRLGY] Proyecto Trilogy (ICT-2007-216372). “*Architecting the Future Internet*”. Disponible [Internet]: <<http://trilogy-project.org/>> [07 de mayo de 2010]

- [VBG08+] *Multi-path BGP: motivations and solutions*. F. Valera, I. van Beijnum, A. García-Martínez, M. Bagnulo. Universidad Carlos III de Madrid, IMDEA Networks. 2008
- [VBG10+] 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
- [VGL00] S. Vutukury and J.J. Garcia-Luna-Aceves, “*MPATH: A Loop-free Multi-path Routing Algorithm*”. Elsevier Journal of Microprocessors and Microsystems, 2000.
- [WRTTS] *Especificaciones técnicas del router WRT54G*. Disponible [Internet]: <http://www.dd-wrt.com/wiki/index.php/Supported_Devices> [07 de mayo de 2010]

Enlaces de interés

"Ayudarte podría, sí."

Maestro Yoda. (Star Wars: Ep. V)

- <BGP++> *Página web oficial del módulo BGP++.* Disponible [Internet]:
<<http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/>> [07 de mayo de 2010]
- <CBGP> *Página web oficial del simulador C-BGP.* Disponible [Internet]:
<<http://cbgp.info.ucl.ac.be/index.php>> [07 de mayo de 2010]
- <CBGPSRC> *Página web con enlaces a la organización del código fuente del simulador C-BGP.* Disponible [Internet]:
<<http://cbgp.info.ucl.ac.be/wiki/index.php?n=Main.Internals>> [07 de mayo de 2010]
- <DDWRT> *Página web oficial del proyecto DD-Wrt.* Disponible [Internet]:
<<http://www.dd-wrt.com/site/index/>> [07 de mayo de 2010]
- <GTNS> *Página web oficial del simulador de red GTNetS.* Disponible [Internet]:
<<http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/>> [07 de mayo de 2010]
- <GVIZ> *Página web oficial de la herramienta GraphViz.* Disponible [Internet]:
<<http://www.graphviz.org/>> [07 de mayo de 2010]
- <INETT> *Página web oficial de "Internet Topology Collection".* Disponible [Internet]: <<http://irl.cs.ucla.edu/topology/>> [07 de mayo de 2010]
- <INKS> *Página web oficial del editor InkScape.* Disponible [Internet]:
<<http://www.inkscape.org/>> [07 de mayo de 2010]
- <NS2> *Página web oficial del simulador de red NS-2.* Disponible [Internet]:
<<http://www.isi.edu/nsnam/ns/>> [07 de mayo de 2010]

- <OWRT> *Página web oficial del proyecto OpenWRT.* Disponible [Internet]:
<<http://openwrt.org/>> [07 de mayo de 2010]
- <PDNS> *Página web oficial de Parallel/Distributed NS.* Disponible [Internet]:
<<http://www.cc.gatech.edu/computing/compass/pdns/index.html>> [07 de mayo de 2010]
- <POTAROO> *Informe sobre análisis de las tablas de rutas BGP.* Disponible [Internet]:
<<http://bgp.potaroo.net/>> [07 de mayo de 2010]
- <QEMU> *Página web oficial de Qemu.* Disponible [Internet]:
<<http://www.qemu.org/about.html>> [07 de mayo de 2010]
- <SUBRT> *Subramanian et al network topology.* Disponible [Internet]:
<<http://cbgp.info.ucl.ac.be/downloads/as.relation>> [07 de mayo de 2010]
- <TVEL> *Test de velocidad alojado en adsl4ever.* Localización: España. Disponible [Internet]: <<http://www.velocidad.info/>> [07 de mayo de 2010]
- <VBOX> *Página web oficial de VirtualBox.* Disponible [Internet]:
<<http://www.virtualbox.org/>> [07 de mayo de 2010]
- <VMWAR> *Página web oficial de VMWare.* Disponible [Internet]:
<<http://www.vmware.com/>> [07 de mayo de 2010]
- <VNUML> *Página web oficial de VNUML.* Disponible [Internet]:
<http://www.dit.upm.es/vnumlwiki/index.php/Main_Page> [07 de mayo de 2010]
- <XORP> *Página web oficial del router software XORP.* Disponible [Internet]:
<<http://www.xorp.org/>> [07 de mayo de 2010]
- <ZBRA> *Página web oficial del proyecto GNU Zebra.* Disponible [Internet]:
<<http://www.zebra.org/>> [07 de mayo de 2010]

ANEXOS

"¡WHARRRRGGH!"
Chewbacca. (Star Wars: Ep. IV)

Anexo I

C-BGP - Comandos añadidos

En esta sección se describen los comandos añadidos al simulador de redes C-BGP para permitir el trabajo con protocolos de encaminamiento *multi-path* y para obtener información adicional de la simulación.

bgp router *dirección* set algorithm

Este comando permite definir el protocolo de encaminamiento que se aplicará en el router correspondiente a *<dirección>*.

Los valores aplicados pueden ser:

- **0** (por defecto): BGP estándar
 - **1** LP-BGP
 - **2** MpASS
- Ejemplo de uso:

```
cbgp> bgp router 0.1.0.0 set algorithm 0
```

bgp router dirección show algorithm

Este comando permite obtener el protocolo de encaminamiento que se aplicará en el router correspondiente a <dirección>.

- Ejemplo de uso:

```
cbgp> bgp router 0.1.0.0 show algorithm  
Algorithm for AS 1 is 2 (MpASS)
```

bgp router dirección show all-possible

Este comando permite obtener la FIB del router correspondiente a <dirección> junto con todas las rutas para cada prefijo que contenían el mismo siguiente salto para alcanzar igual destino.

- Ejemplo de uso:

```
cbgp> bgp router 0.1.0.0 show all-possible  
0.2.0.0/16 0.2.0.0 80 4294967295 2 i  
0.3.0.0/16 0.3.0.0 100 4294967295 3 i  
0.4.0.0/16 0.4.0.0 100 4294967295 4 i  
0.5.0.0/16 0.2.0.0 80 4294967295 2 5 i  
0.6.0.0/16 0.3.0.0 100 4294967295 3 6 i  
(...)
```

bgp router dirección show fib

Este comando permite obtener la FIB del router correspondiente a <dirección>.

- Ejemplo de uso:

```
cbgp> bgp router 0.1.0.0 show fib  
0.2.0.0/16 0.2.0.0 80 4294967295 2 i  
0.3.0.0/16 0.3.0.0 100 4294967295 3 i  
0.4.0.0/16 0.4.0.0 100 4294967295 4 i  
0.5.0.0/16 0.2.0.0 80 4294967295 2 5 i  
0.6.0.0/16 0.3.0.0 100 4294967295 3 6 i
```

bgp router dirección show loops

Este comando permite obtener el número de bucles detectados por el router correspondiente a <dirección>.

- Ejemplo de uso:

```
cbgp> bgp router 0.1.0.0 show loops  
Number of loops detected: 15
```

bgp router dirección set max-aggregation

Este comando permite definir el número máximo de entradas por prefijo que se permitirán en la FIB para el router correspondiente a <dirección>.

- Ejemplo de uso:

```
cbgp> bgp router 0.1.0.0 set max-aggregation 25
```

bgp router dirección show max-aggregation

Este comando permite mostrar el número máximo de entradas por prefijo que se permitirán en la FIB para el router correspondiente a <dirección>.

- Ejemplo de uso:

```
cbgp> bgp router 0.1.0.0 show max-aggregation  
Max aggregation for AS 1 is 0 (no limit)
```

show msgs

Este comando permite mostrar el número de mensajes enviados entre los distintos routers a lo largo de la simulación.

- Ejemplo de uso:

```
cbgp> show msgs  
Number of BGP messages sent: 786
```

Anexo II

StatOpology - Manual de usuario

A continuación se adjuntan las páginas que componen el manual de usuario de la herramienta *StatOpology*.



Fig. 18. Manual de StatOpology


ÍNDICE DE CONTENIDOS

Introducción	3
Requisitos mínimos	3
Ventana principal.....	4
Menú "File"	5
Subopción "Load statOpology data (.sto)"	5
Subopción "Save statOpology data (.sto)"	6
Subopción "Exit"	7
Menú "Topology"	9
Subopción "Import topology (map format)"	9
Subopción "Import topology (.rel)"	10
Subopción "Export topology (map format)"	10
Subopción "Export topology (.rel)"	11
Subopción "Show topology info"	11
Subopción "Generate topology graph"	12
Subopción "Reset data"	14
Menú "Simulation"	16
Subopción "Generate script for C-BGP"	16
Subopción "Launch simulation in C-BGP"	16
Subopción "Analyze results"	16
Menú "Preferences"	17
Subopción "Configuration"	17
Menú "Help"	17
Subopción "Help"	18
Subopción "About StatOpology"	18

ÍNDICE DE FIGURAS

Figura 1. Ventana principal de <i>StatOpology</i>	4
Figura 2. Menú “File” desplegado	5
Figura 3. Ventana de selección de fichero (cargar datos)	5
Figura 4. Carga de información sobre la topología de red	6
Figura 5. Ventana de selección de fichero (guardar datos).....	7
Figura 6. Información de la topología guardada con éxito.....	7
Figura 7. Cuadro de diálogo mostrado al salir de la aplicación.....	8
Figura 8. Menú “Topology” desplegado	9
Figura 9. Resultado de importar topología.....	10
Figura 10. Datos exportados con éxito.....	11
Figura 11. Información sobre la topología	12
Figura 12. Guardar archivo interpretable por Graphviz	13
Figura 13. Éxito al guardar el fichero interpretable por Graphviz.....	13
Figura 14. Grafos de la topología generados con éxito (por Graphviz).....	13
Figura 15. Ejemplo de archivos generados en Microsoft® Windows 7™	14
Figura 16. Reset en <i>StatOpology</i>	14
Figura 17. Ventana de información sobre la topología sin datos que mostrar	15
Figura 18. Menú “Simulation” desplegado.....	16
Figura 19. Menú “Preferences” desplegado	17
Figura 20. Menú “Help” desplegado	17
Figura 21. Acerca de StatOpology	18
Figura 22. Ponerse en contacto con el autor vía correo electrónico	19
Figura 23. Ejemplo de envío de correo tras hacer click en el nombre del autor.....	19


INTRODUCCIÓN

Stat**pology** es una aplicación multiplataforma que permite el análisis de topologías de red, la conversión entre formatos de varios archivos de topologías, su representación gráfica en forma de grafo (*) y configurar simulaciones para la topología procesada, permitiendo elegir las características que se evaluarán en la simulación así como configurar de forma independiente diversos aspectos de cada uno de los nodos involucrados. Además de esto, permite realizar de forma automática la simulación (**) y posteriormente mostrar los resultados.

(*) Para poder realizar la representación gráfica de la topología se requiere tener en el sistema la aplicación “**Graphviz**” y configurar la ruta a sus binarios en el apartado de configuración.

(**) Para poder realizar la simulación de manera automática, se requiere tener en el sistema el simulador de redes “**C-BGP**” y configurar la ruta a sus binarios en el apartado de configuración.

REQUISITOS MÍNIMOS

Cuando **Stat****pology** fue desarrollado, se hizo pensando en los distintos sistemas operativos existentes. De esta forma se decidió crear una aplicación accesible desde el máximo número de sistemas posibles. Así pues los requisitos mínimos son los siguientes:

- Sistema operativo con soporte a Java Runtime Environment 1.6 o superior.
- Resolución de pantalla igual o superior a 800x600 píxeles y 16 bit de color.
- 128 Mb de memoria RAM.

VENTANA PRINCIPAL

Al ejecutar **StatOpology** aparecerá en pantalla una ventana como la siguiente:



Figura 1. Ventana principal de **StatOpology**

Es la ventana principal. Desde aquí tenemos acceso a un gran número de funcionalidades a través de la barra superior de menú. En los siguientes puntos explicaremos cada una de ellas.

MENÚ “FILE”

Al desplegar el menú “File” nos encontramos con las siguientes opciones:

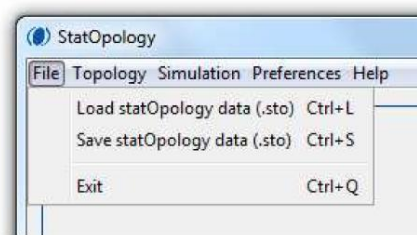


Figura 2. Menú “File” desplegado

SUBOPCIÓN “LOAD STATOPOLOGY DATA (.STO)”

Esta opción permite cargar un fichero con información de una topología previamente analizada por StatOpology. Para acceder a esta función, también se puede utilizar el atajo de teclado **Ctrl+L**.

Una vez accedida a esta función, se mostrará una ventana de selección de fichero como la siguiente:

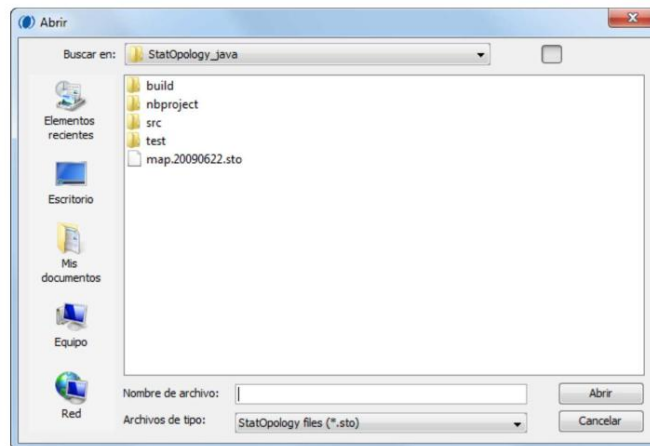


Figura 3. Ventana de selección de fichero (cargar datos)

Tras seleccionar un fichero con extensión **.sto** se deberá realizar *click* en el botón “Abrir”. Se mostrará una ventana como la que sigue:

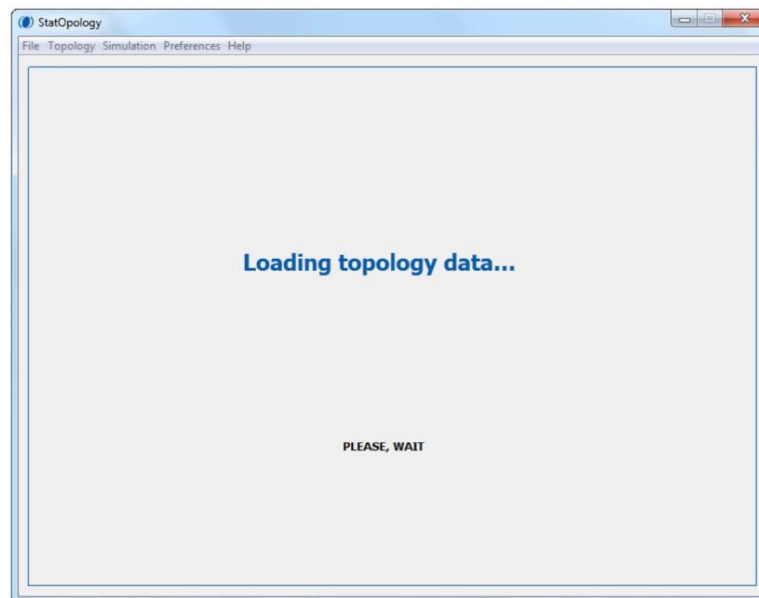


Figura 4. Carga de información sobre la topología de red

Y concluida la carga, se mostrará una ventana con la información sobre la topología (ver subopción “*Show topology info*”, figura 11).

SUBOPCIÓN “SAVE STATOPOLOGY DATA (.STO)”

Esta opción permite salvar en un fichero con extensión **.sto** la información de sobre la topología en uso por **StatOpology**. Para acceder a esta función, también se puede utilizar el atajo de teclado **Ctrl+S**.

Una vez accedida a esta función, se mostrará una ventana de selección de fichero destino como la siguiente:

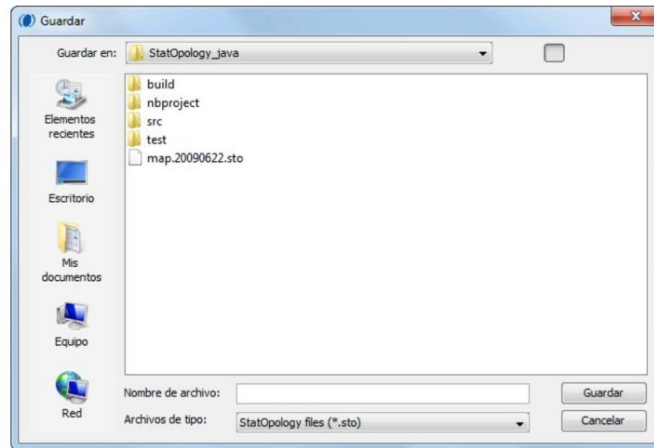


Figura 5. Ventana de selección de fichero (guardar datos)

Una vez guardados los datos, se mostrará el siguiente cuadro de diálogo:



Figura 6. Información de la topología guardada con éxito

SUBOPCIÓN "EXIT"

Esta opción permite finalizar la ejecución de *StatOpology*. Realiza la misma función que hacer *click* en el *aspa de la ventana* (botón de *cerrar ventana*). Para acceder a esta función, también se puede utilizar el atajo de teclado **Ctrl+Q**.

Por seguridad, al utilizar esta función se muestra un cuadro de diálogo como el que sigue:

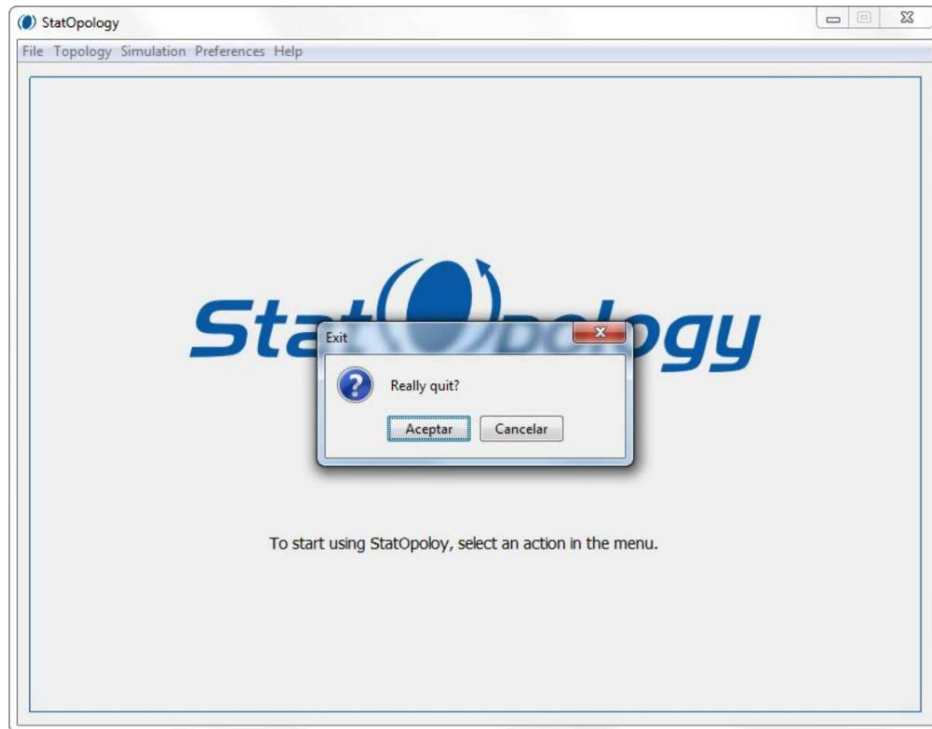


Figura 7. Cuadro de diálogo mostrado al salir de la aplicación

Para salir de la aplicación se deberá pulsar el botón **“Aceptar”**. Por el contrario si se desea seguir utilizando *StatOpology*, por ejemplo, para salvar los datos en curso, bastará con hacer *click* en el botón **“Cancelar”**.

MENÚ “TOPOLOGY”

Al desplegar el menú “Topology” nos encontramos con las siguientes opciones:

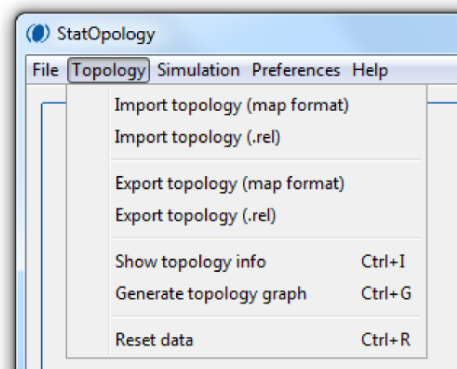


Figura 8. Menú “Topology” desplegado

SUBOPCIÓN “IMPORT TOPOLOGY (MAP FORMAT)”

Esta opción permite importar un fichero con información de una topología en formato “map”. El formato “map” se caracteriza porque sus datos representan relaciones entre sistemas autónomos. Es ampliamente utilizado entre software y expertos encargados de recuperar información topológica sobre redes del tamaño de Internet.

Cada fila del archivo está compuesta por los siguientes valores:

<AS1> <AS2> <Rel>

Donde <AS1> y <AS2> corresponden a números enteros con valor positivo inferior a 65536 relativos al identificador de cada AS involucrado en la relación, y <Rel> indica el tipo de relación, pudiendo ser de *peering* (p2p) o bien de *proveedor a cliente* (p2c) / *cliente a proveedor* (c2p).

Al acceder a esta funcionalidad, se mostrará un cuadro de selección de archivos análogo al de la figura 3, seguido de una ventana de carga como la de la figura 4 una vez seleccionado el archivo de tipo “map”. Una vez importados los datos se mostrará un cuadro de diálogo con información sobre los datos importados y una ventana con información sobre la topología (ver figura 11).

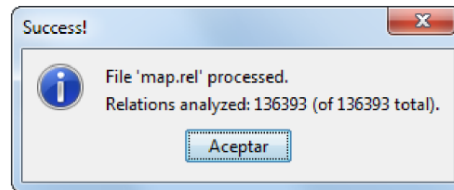


Figura 9. Resultado de importar topología

SUBOPCIÓN “IMPORT TOPOLOGY (.REL)”

Esta opción permite importar un fichero con información de una topología en formato “rel”. El formato “rel” se caracteriza porque sus datos representan relaciones entre sistemas autónomos. Es utilizado por el simulador de protocolos de encaminamiento *C-BGP*.

Cada fila del archivo está compuesta por los siguientes valores:

<AS1> <AS2> <Rel>

Donde <AS1> y <AS2> corresponden a *números enteros con valor positivo inferior a 65536* relativos al identificador de cada AS involucrado en la relación, y <Rel> indica el tipo de relación mediante un número positivo entero igual a **0** (*peering*) ó a **1** (*cliente a proveedor*).

Al acceder a esta funcionalidad, se mostrará un cuadro de selección de archivos análogo al de la figura 3, seguido de una ventana de carga como la de la figura 4 una vez seleccionado el archivo de tipo “rel”. Una vez importados los datos se mostrará un cuadro de diálogo con información sobre los datos importados (figura 9) y una ventana con información sobre la topología (ver figura 11).

SUBOPCIÓN “EXPORT TOPOLOGY (MAP FORMAT)”

Esta opción permite exportar los datos de la topología en uso al formato “map”. Al utilizar esta función, se mostrará un cuadro de selección de archivo destino similar al de la figura 5. Una vez confirmado el archivo de destino se mostrará un cuadro de información de éxito como el siguiente:

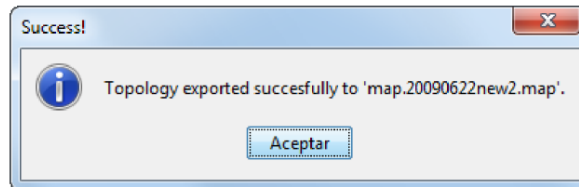


Figura 10. Datos exportados con éxito

SUBOPCIÓN “EXPORT TOPOLOGY (.REL)”

Esta opción permite exportar los datos de la topología en uso al formato “**rel**”. Al utilizar esta función, se mostrará un cuadro de selección de archivo destino similar al de la figura 5. Una vez confirmado el archivo de destino se mostrará un cuadro de información de éxito análogo al de la figura 10.

SUBOPCIÓN “SHOW TOPOLOGY INFO”

Esta opción permite mostrar información relativa a la topología cargada en StatOpology. Algunos de los datos mostrados son: número de nodos tipo *Tier-1*, nodos *hoja*, número medio de enlaces entre nodos...

En la ventana correspondiente a esta opción se muestra también un botón para salvar los datos a un fichero de texto (.txt). Al pulsar el botón se muestra un diálogo de selección de archivo destino análogo al de la figura 5.

Para acceder a esta funcionalidad también se puede utilizar el atajo de teclado **Ctrl+I**.

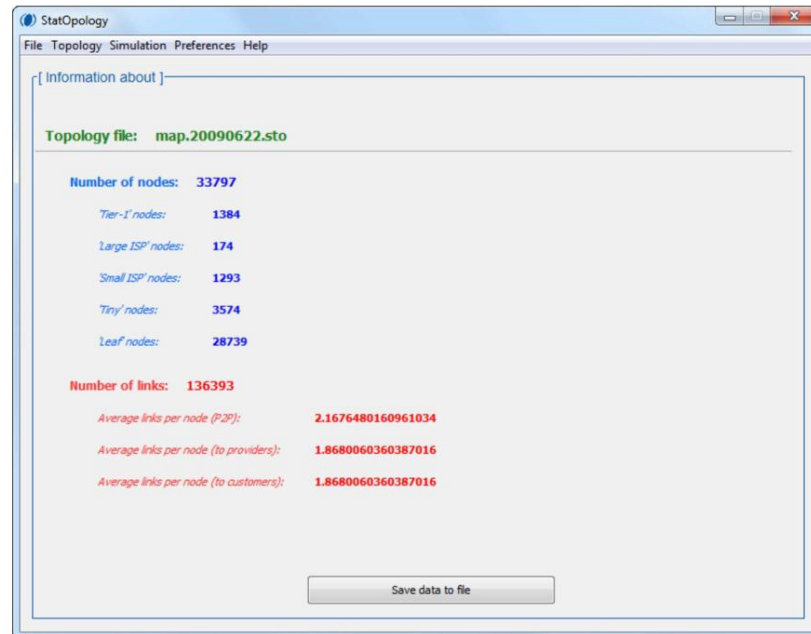


Figura 11. Información sobre la topología

SUBOPCIÓN “GENERATE TOPOLOGY GRAPH”

Esta opción genera un fichero interpretable por la herramienta “**Graphviz**”. Dicha herramienta permite representar los datos de dicho fichero como un grafo relacional en el que ver la topología de una forma mucho más intuitiva.

En caso de tener “**Graphviz**” instalado en el sistema, si se configura la ruta a sus binarios en **StatOpology**, esta opción generará también los grafos de la topología en formato **.png** (*imagen*) y **.svg** (*gráficos vectoriales*).

Para acceder a esta funcionalidad también se puede utilizar el atajo de teclado **Ctrl+G**.

Una vez ejecutada, se mostrará un diálogo para seleccionar el archivo en que guardar el fichero interpretable por “**Graphviz**”. Si se ha configurado en las opciones de **StatOpology** el acceso al mismo, se generarán también los grafos correspondientes con igual nombre al fichero interpretable pero diferente extensión (**.png** y **.svg**).

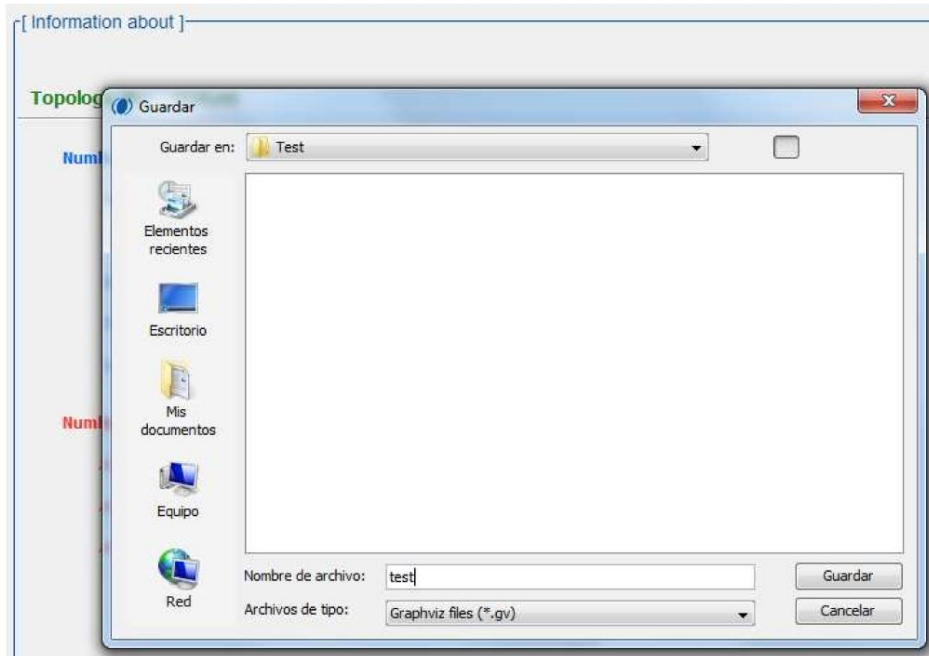


Figura 12. Guardar archivo interpretable por Graphviz

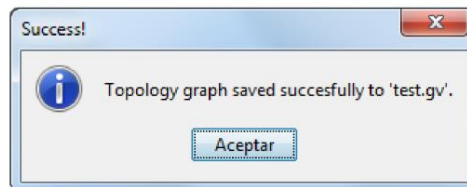


Figura 13. Éxito al guardar el fichero interpretable por Graphviz

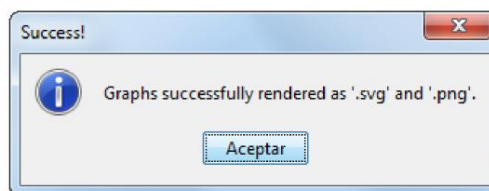


Figura 14. Grafos de la topología generados con éxito (por Graphviz)

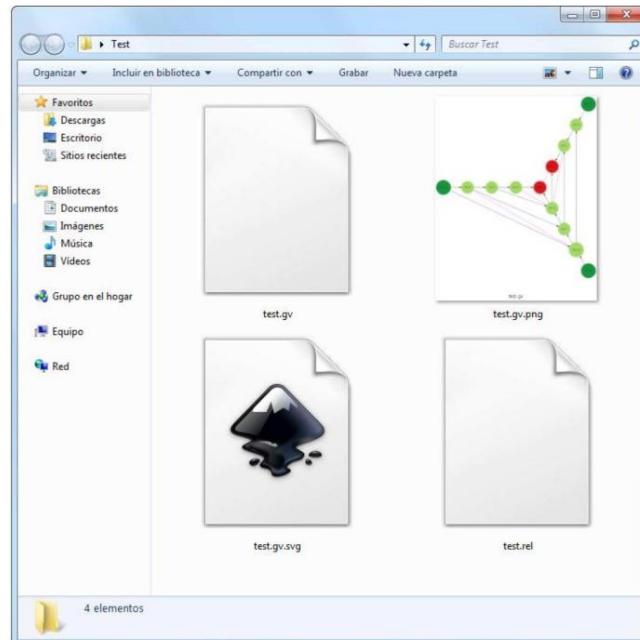
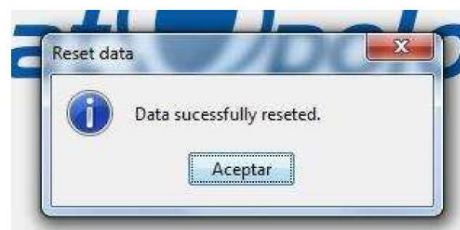


Figura 15. Ejemplo de archivos generados en Microsoft® Windows 7™

SUBOPCIÓN “RESET DATA”

Esta opción limpia la memoria de *StatOpology*, dejándolo en el mismo estado que cuando se inicia. Toda la información de topologías cargadas en el sistema se descarta al usar esta opción.

Existe un atajo de teclado para ejecutar dicha funcionalidad. Sólo hay que presionar **Ctrl+R**.

Figura 16. Reset en *StatOpology*

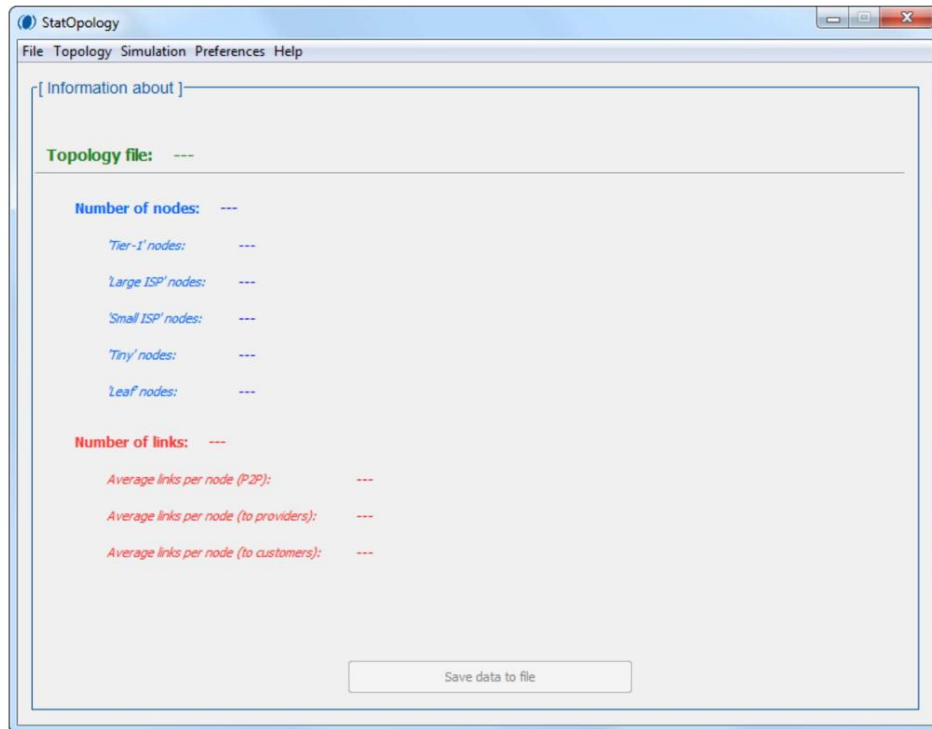


Figura 17. Ventana de información sobre la topología sin datos que mostrar

MENÚ “SIMULATION”

Al desplegar el menú “Simulation” nos encontramos con las siguientes opciones:

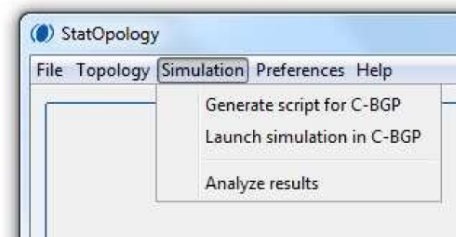


Figura 18. Menú “Simulation” desplegado

SUBOPCIÓN “GENERATE SCRIPT FOR C-BGP”

Esta opción se encuentra disponible en la interfaz de consola. Actualmente se está implementando en la interfaz gráfica.

SUBOPCIÓN “LAUNCH SIMULATION IN C-BGP”

Esta opción se encuentra disponible en la interfaz de consola. Actualmente se está implementando en la interfaz gráfica.

SUBOPCIÓN “ANALYZE RESULTS”

Esta opción se encuentra disponible en la interfaz de consola. Actualmente se está implementando en la interfaz gráfica.

MENÚ “PREFERENCES”

Al desplegar el menú “**P**references” nos encontramos con las siguientes opciones:

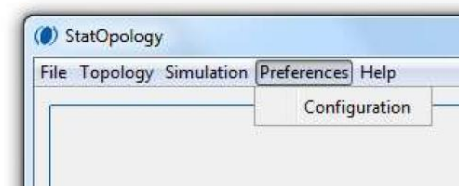


Figura 19. Menú “Preferences” desplegado

SUBOPCIÓN “CONFIGURATION”

Esta opción se está implementando en la interfaz gráfica. Se adjuntará ayuda sobre la misma en cuanto esté disponible.

MENÚ “HELP”

Al desplegar el menú “**H**elp” nos encontramos con las siguientes opciones:

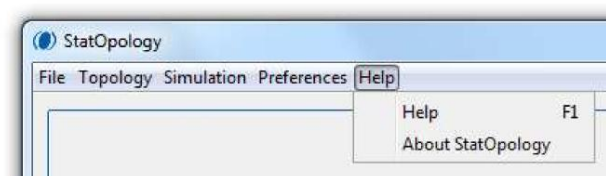


Figura 20. Menú “Help” desplegado

SUBOPCIÓN "HELP"

Esta opción muestra la ayuda en pantalla de **StatOpology**, o lo que es lo mismo, el fichero actual. El método alternativo para acceder a dicha opción es pulsar la tecla **F1** del teclado.

SUBOPCIÓN "ABOUT STATOPOLOGY"

Esta opción muestra información relativa a **StatOpology** y a su autor. En caso de realizar click en el nombre del autor se abrirá la aplicación de correo por defecto instalada en el sistema ya configurada para enviar un mail al creador de **StatOpology**.

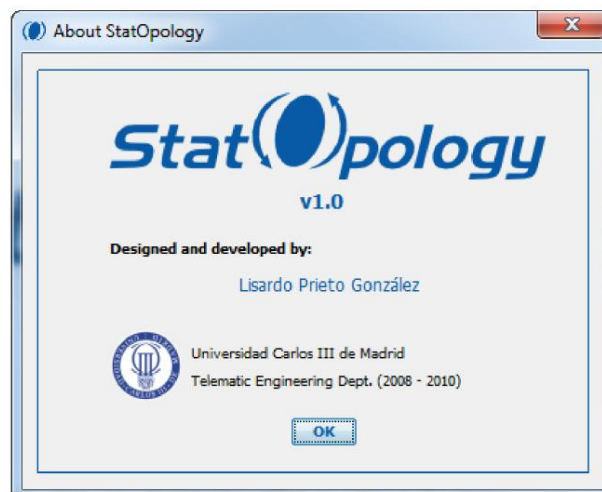


Figura 21. Acerca de StatOpology



Figura 22. Ponerse en contacto con el autor vía correo electrónico

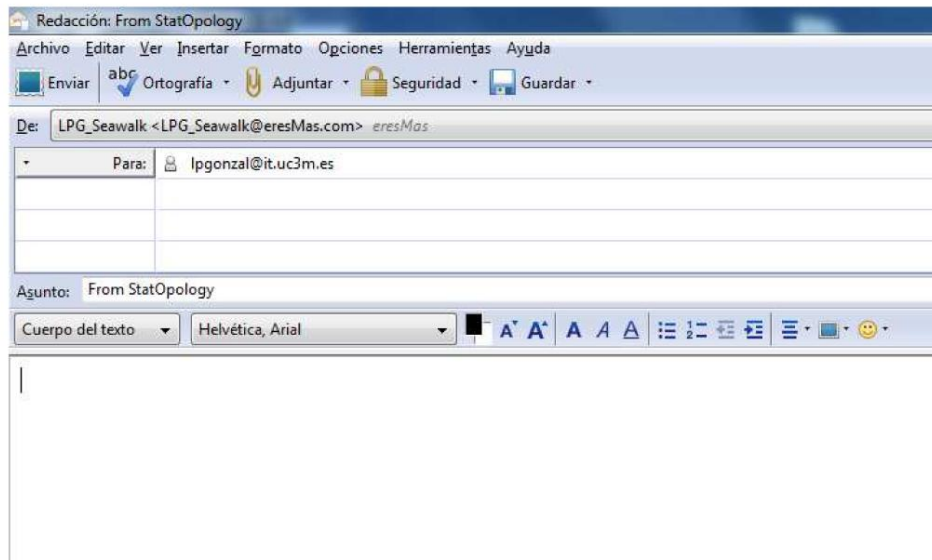


Figura 23. Ejemplo de envío de correo tras hacer click en el nombre del autor