Capítulo 1

Manual de Usuario

A largo de este proyecto hemos desarrollado un conjunto de herramientas en *python* que automatizan el proceso de emulación y que agiliza el análisis de los datos recabados. Para facilitar el uso de las mismas hemos desarrollado también un interfaz gráfica ("TTools.py") muy simple que permite invocar cada una de las herramientas.

Figura 1.1: TTools

```
En esta toplogia hay 13 enlaces externos con otros Sistemas Autonomos

Los nodos que mantienen conexiones con AS1 son: MI_1
Los nodos que mantienen conexiones con AS2 son: MI_1
Los nodos que mantienen conexiones con AS3 son: MI_2
Los nodos que mantienen conexiones con AS4 son: RM_2
Los nodos que mantienen conexiones con AS5 son: FI
Los nodos que mantienen conexiones con AS6 son: PD
Los nodos que mantienen conexiones con AS7 son: CT
Los nodos que mantienen conexiones con AS8 son: MI_2
Los nodos que mantienen conexiones con AS9 son: MI_2
Los nodos que mantienen conexiones con AS10 son: RM_2
Los nodos que mantienen conexiones con AS11 son: MI_2
Los nodos que mantienen conexiones con AS12 son: MI_1
Los nodos que mantienen conexiones con AS13 son: TO

Se ha generado el archivo Garr.xml con exito
```

Figura 1.2: Salida del script “GMLtoXMLconverter.py” - Conexiónes eBGP

Para abrir la interfaz simplemente debemos ejecutar `python TTools.py` en la linea de comandos. Como se puede ver en la imagen 1.1 la interfaz consiste en una serie de botones donde cada uno tiene una breve descripción de lo que hace. A continuación veremos las opciones que nos brinda la interfaz.

- Convertir archivo GML a XML: EL lenguaje GML es el más frecuente utilizado para describir información geográfica. La mayoría de las topologías de red que podemos encontrar en la web las encontramos con la extensión “.gml”. Como ya hemos visto para poder correr los algoritmos de localización es necesario otro formato. Esta opción nos permite convertir el archivo a un formato valido para luego poder correr el algoritmo de localización.
  Si queremos convertir un archivo debemos simplemente seleccionar el archivo con extensión “gml” (ver 1.3) e ingresar el número de Sistema Autónomo (ASN) que va a tener la red (ver 1.4). El resultado se guardará en la misma ubicación donde se encuentra el archivo original. También podemos ejecutar esta opción desde la linea de comando. Para esto debemos ejecutar `python GMLtoXMLconverter.py [topo.gml]`. En el caso de que el archivo gml contenga información de los enlaces inter-AS que tiene la red, esta será desplegada en la linea de comandos al finalizar la conversión del formato. En la imagen 1.2 podemos ver el ejemplo de la conversión del archivo “Garr.xml”.

- Preparar emulación de red: Cuando hablamos de preparar la emulación de red nos referimos a crear todos los archivos de configuración de cada uno de los dispositivos más la configuración física de la red. Además, se crea el

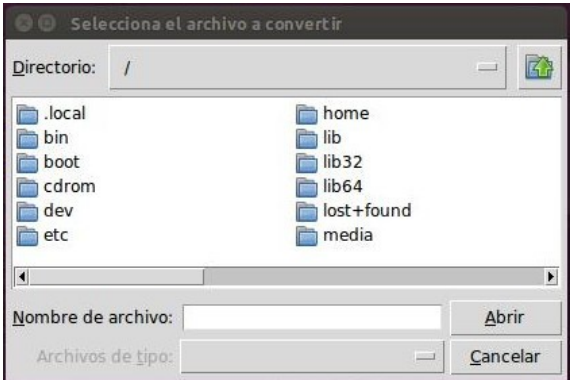

Figura 1.3: Selección de archivo para convertir a formato adecuado

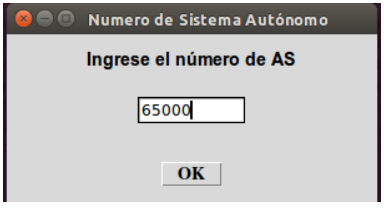

Figura 1.4: Ingresar número de Sistema Autónomo

Figura 1.5: Seleccionar routers de borde

Figura 1.6: Seleccionar ASN vecino eBGP

script "start.py" que contiene la programación de los distintos eventos que participan en la emulación, como por ejemplo la inyección de trazas o el almacenamiento de la tabla "loc_rib" en disco al final de la ejecución.
Para poder preparar la emulación debemos previamente haber utilizado la herramienta RRLoc para correr el algoritmo de localización de RR.
Habiendo realizado lo anterior, para empezar, debemos seleccionar el archivo xml con la información de red. En la siguiente ventana podemos seleccionar la cantidad de sesiones eBGP, los routers de borde de nuestro AS (ver 1.5) y los ASN de los vecinos (ver 1.6). Esto generará una estructura de documentos, los cuales contendrán la configuración de los dispositivos participan en nuestro AS.

- Ejecutar MiniNExT: Una vez que hayamos preparado la emulación de red, podemos correr MiniNExT. Para esto debemos seleccionar la carpeta donde se encuentra los datos de la red que se ha generado previamente. Esto también se puede realizar desde la consola, simplemente ejecutamos el comando `python start.py` estando parado dentro de la carpeta generada y empezará la emulación. Dado que es necesario el ejecutar el comando con privilegios de superusuario, se debe escribir la contraseña del mismo para poder correr MiniNExT.

Figura 1.7: Cargar datos en la base

- Cargar Bases de Datos Mysql: Esta opción nos permite cargar todos los datos en una base mysql. Los datos de las tablas rib_in de cada router son almacenados en binario con el formato MRT [**?**]. Antes de cargar los datos de las tablas rib_in utilizamos la herramienta bgpdump para traducir esta información.
  Los datos son almacenados en cuatro tablas: "loc_rib_ipv4", "loc_rib_ipv6", "rib_in" y "table_dump". Para esto debemos seleccionar la ubicación donde se almacenaron los resultados de la emulación, por defecto se encuentran en la carpeta "/var/log/mininext". Luego de esto se nos abre otra ventana que nos pide el nombre que se la va a dar a la base y el algoritmo utilizado en la emulación (ver 1.7). Una vez completado estos campos comenzamos la carga de la base. Mientras se produce la carga, queda inhabilitado el resto de las opciones de la herramienta TTools.

Figura 1.8: Selecciar trazas para descargar

Nota: Se deben configurar los parámetros de la base antes de utilizar esta herramienta. Para esto se debe editar el archivo "loadDB.py" y modificar los parámetros "HOST_NAME", "USER_NAME" y "PASSWORD".

- Borrar archivos generados por bgpdump: Para cargar las bases de datos es necesario hacer la traducción de los archivos rib_in. Una vez que se realiza la traducción estos archivos se mantienen junto a los archivos originales. Esto produce que se utilice más espacio en el disco de forma innecesaria.
  También se puede ejecutar por fuera del *framewok* a través de la consola escribiendo `python deleteTemporaryFiles` y seleccionado la carpeta donde esta ubicados los resultados de la ejecución.

- Descargar trazas BGP: Esta opción permite descargar la trazas BGP de la pagina "www.ripe.net" y convertirlos a texto plano para después ser utilizado en la emulación. Esta página brinda la posibilidad de descargar trazas BGP de diecisiete colectores [**?**] localizados en distintas partes del globo. Para descargar la traza debemos elegir la ubicación donde queremos que se descargue la traza. Luego debemos seleccionar el colector (estos son nombrados de la forma rrcXY), la fecha y el intervalo de tiempo (ver 1.8). Las trazas descargadas deben llamarse PruebaRRCXY, donde XY indica el número de traza. Por ejemplo, "PruebaRRC03" sera la traza que se inyecte en la tercer sesión eBGP.

1.1. Posible caso de uso

La idea de esta sección es poder mostrar el proceso más común que llevamos a cabo para realizar una emulación.
Como primera instancia buscamos una topología de red que se encuentre en formato gml. En nuestro caso las obtuvimos de la pagina "The topology Zoo" [**?**]. Luego de descargar la topología es necesario convertir el archivo. Como vimos antes, para esto debemos seleccionar la opción "Convertir archivo GML a XML" y seguir los pasos que ya hemos mencionado. Las topologías obtenidas de la pagina "The topology Zoo" pueden, o no, contar con la información de los enlaces que mantiene la red con otros Sistemas Autónomos. Si posee esta información se desplegará en la linea de comandos cuáles son los routers de borde del AS que mantienen sesiones eBGP como se vió en la imagen 1.2. Una vez con la topología en formato xml y con la ayuda de la herramienta RRLoc podemos correr los algoritmos de localización de reflectores de rutas. En esta etapa podremos seleccionar entre un *full-mesh* de sesiones iBGP o entre los algoritmos "Sep", "SepD", "SepS", "Bates", "BatesY", "BatesZ" y "Zhang". El resultado de esta ejecución será un archivo xml que contenga toda la información BGP de la red que queremos emular. Una vez que llegamos a este punto debemos configurar toda la red. Como ya hemos mencionado, esto involucra la configuración de enlaces, interfaces, protocolos entre otros. Para esto simplemente debemos seleccionar la opción "Preparar emulación de red". En este punto es donde será necesario la información correspondiente a las sesiones eBGP del AS.
Lo único que resta para poder llevar a cabo la emulación es conseguir las trazas BGP que queremos inyectar. Estas trazas pueden ser creadas por nosotros mismos, para estudiar el comportamiento ante ciertos mensajes, o podemos emular tráfico BGP real. Si queremos esto último, podemos obtener la trazas reales seleccionando

la opción "Descargar trazas BGP". Ahora si estamos en condiciones de emular la red. Como ya hemos visto lo podemos hacer con el botón "Ejecutar MiniNExT". Una vez que la red se esta emulando, sugerimos esperar a que el protocolo OSPF converja para comenzar a inyectar las trazas. Esto se puede chequear de una manera sencilla escribiendo el comando `sh ip route` dentro de alguno de los routers. Si se aprenden todas la direcciones de *loopback* de los routers quiere decir que OSPF ya convergió. Para poder inyectar la trazas debemos escribir `exit`, de esta forma salimos de la consola del MiniNExT para poder ejecutar lo que se haya programado en el archivo "start.py" y luego vuelve a la consola. Algunos segundos después veremos que se abren nuevas terminales, donde cada una de ellas se corresponde con una traza. En las mismas se puede ver todos los mensajes que son inyectados, tanto *updates* como *withraws*, y en caso de que se produzca algún error también es mostrado en la terminal.
Una vez terminada la inyección de trazas (el tiempo varía según el intervalo de tiempo que hemos seleccionado en la descarga de la traza), debemos escribir nuevamente `exit` en la consola. Esta acción copia los datos que nos interesan al disco y finaliza la emulación. Los datos serán almacenados en la carpeta "/var/log/mininext" por defecto.
Como última paso nos queda cargar los datos obtenidos en la base de datos. Como ya hemos mencionado anteriormente, algunos de estos datos es necesario traducirlos a texto plano antes de cargarlos a la base. La opción "Cargar base de datos Mysql" realiza la traducción de estos datos y luego carga la base.
Recomendemos, luego de cargar la base, borrar los archivos traducidos (los generados por bgpdump) ya que ocupan espacio con información duplicada. Esto se puede hacer de manera sencilla con el botón "Borrar archivos generados por bgpdump" que nos ahorra el tener que entrar a cada uno de los routers para borrar estos archivos. Cuando hablamos de topologías con cientos de routers significa un ahorro importante de trabajo.