Sistema de recomendación de políticas de tráfico BGP
BGP traffic policies recommendation system

Trabajo de Fin de Máster
Curso 2021–2022

Autor
Alberto Caballero Gámez

Director
Juan Carlos Fabero Jiménez

Máster en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Documento maquetado con TEXiS v.1.0.

Este documento está preparado para ser imprimido a doble cara.

Sistema de recomendación de políticas de tráfico BGP
BGP traffic policies recommendation system

Trabajo de Fin de Máster en Ingeniería Informática
Departamento de Arquitectura de Computadores y Automática

Autor
Alberto Caballero Gámez

Director
Juan Carlos Fabero Jiménez

Convocatoria: Junio 2022
Calificación: 10

Máster en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

13 de julio de 2022

Agradecimientos

Mi agradecimiento está dirigido principalmente a Juan Carlos Fabero Jiménez por darme la posibilidad de desarrollar este TFM, por su gran implicación y la ayuda incondicional que siempre me ha prestado, sin todo ello no hubiese sido posible la consecución de este trabajo.

También agradecer a los docentes y la Facultad de Informática de la UCM, todos los conocimientos adquiridos durante estos años. Aunque ha sido un camino difícil, la recompensa ha sido altamente satisfactoria.

Por último, pero no menos importante, agradecer a todos mis familiares, amigos y compañeros, el apoyo y la fuerza que me han transmitido para solventar los retos y cumplir mis metas. Muchas gracias por haber estado y estar siempre conmigo.

Abstract

BGP traffic policies recommendation system

Information and communication technologies are areas of research that are constantly growing. The numerous advances in the sector provide tools to access a wide variety of information and services from anywhere in the world. These tools could be summarized in a couple of words, the Internet.

The Internet is a global system whose operation is possible thanks to complex mechanisms and protocols developed throughout history. Each of these mechanisms is responsible for managing a specific feature, being BGP (Border Gateway Protocol) one of the most relevant protocols on which the Internet is based. However, this protocol, which is responsible for the exchange of global routing information, is managed and configured locally by different ISPs (Internet Service Provider), technology companies, universities, government agencies and scientific institutions. This causes the interests of some entities to intervene in the routing of network traffic, sometimes causing certain problems.

This academic project presents a study about the different problems that this protocol harbors, providing a means to observe the events that occur and recommending possible configurations to avoid unexpected service interruptions or unwanted prefix hijacking.

Keywords

BGP (Border Gateway Protocol), AS (Autonomous System), Outages, Hijacks, BGPMon, BGPStream

Resumen

Sistema de recomendación de políticas de tráfico BGP

Las tecnologías de la información y comunicación son áreas de investigación en constante crecimiento. Los numerosos avances del sector proporcionan herramientas para acceder a una gran variedad de información y servicios desde cualquier parte del mundo. Estas herramientas se podrían resumir en una única palabra, Internet.

Internet es un sistema de carácter global cuyo funcionamiento es posible gracias a complejos mecanismos y protocolos desarrollados a lo largo de la historia. Cada uno de estos mecanismos se encarga de gestionar una característica concreta, siendo BGP (*Border Gateway Protocol*) uno de los protocolos más relevantes sobre los que se sostiene Internet. Sin embargo, este protocolo que se encarga del intercambio de información de encaminamiento global, es gestionado y configurado de manera local por los diferentes ISP (*Internet Service Provider*), empresas tecnológicas, universidades, agencias gubernamentales e instituciones científicas. Esto hace que los intereses particulares de algunas entidades intervengan en el encaminamiento del tráfico de red, causando en ocasiones ciertos problemas.

En este trabajo se presenta un estudio acerca de los diferentes problemas que alberga este protocolo, proporcionando un medio para observar los eventos que se producen y recomendando posibles configuraciones con el fin de evitar interrupciones de servicio inesperadas o el secuestro indeseado de prefijos.

Palabras clave

BGP (*Border Gateway Protocol*), AS (*Autonomous System*), *Outages*, *Hijacks*, BGPMon, BGPStream

Índice

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Internet es un sistema que proporciona multitud de servicios a la sociedad a través de diversos protocolos de comunicación. Cada uno de estos protocolos, con funciones determinadas, hacen posible el funcionamiento de la fuente de información más grande del mundo. Para partir de una base, es importante conocer que estos protocolos son clasificables mediante el modelo OSI (*Open System Interconnection*) o el modelo TCP/IP, los cuales contienen la misma información pero utilizan una distribución de capas distinta (Figura 1.1).

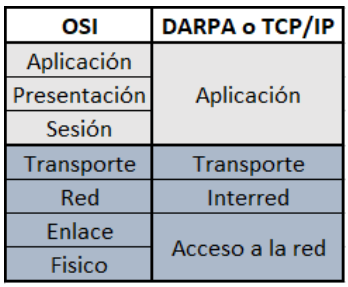


Figura 1.1: Capas del Modelo OSI y Modelo TCP/IP

Es complejo conseguir la coordinación de cada uno de los elementos de Internet, ya que en ocasiones los protocolos encargados de dicha tarea incluyen características de configuración que pueden afectar a su correcto funcionamiento. Los modelos anteriores definen las diferentes etapas que deberán atravesar los datos para su envío. La capa de red es la encargada del direccionamiento lógico de datos, que mediante protocolos como IP (*Internet Protocol*) u OSPF (*Open Shortest Path First*) determina la mejor ruta para el envío óptimo de los mismos.

El objeto principal de este TFM es el estudio del protocolo BGP. Este protocolo, cuya función no es clasificable en una única capa del modelo OSI, se encarga de la construcción de las rutas globales de Internet que definen cómo alcanzar a cada uno de los diferentes destinos presentes en la red. Sin embargo, como se verá más adelante, este protocolo también contiene algunas peculiaridades que son dignas de estudio.

1.1. Motivación

Internet se puede ver como un gran grafo, donde los vértices serían los encargados de gestionar y realizar el intercambio de la información. El protocolo BGP proporciona mecanismos para el intercambio de información de encaminamiento entre los diferentes vértices que componen esta red. Dentro de la terminología BGP, estos vértices o nodos se denominan sistemas autónomos (AS, *Autonomous System*) y se identifican mediante un número único o ASN (*Autonomous System Number*).

De esta forma, el protocolo se encarga de establecer las rutas óptimas por las que se dirigirá el tráfico. Para formalizar estas rutas, el protocolo incorpora diferentes opciones de configuración que se gestionan de manera local por diferentes AS. Sin embargo, queda por definir qué es una ruta óptima, puesto que cada sistema autónomo puede tener unos intereses particulares distintos.

La gran versatilidad a la hora de configurar y determinar las diferentes rutas que seguirá el tráfico de red supone un gran beneficio a favor del protocolo BGP, permitiendo que las organizaciones responsables del encaminamiento lo hagan en pro de sus propios intereses. Sin embargo, una alteración en la configuración de las políticas de encaminamiento de BGP condiciona el tránsito de red, pudiendo derivar en la sustracción de información sensible para las diferentes entidades geopolíticas mundiales o incluso su aislamiento en Internet.

A lo largo de la historia de BGP se han experimentado multitud de problemas como consecuencia de una mala configuración. Sin ir mas lejos, una de estas consecuencias se dio el 4 de Octubre de 2021 [26], cuando Facebook, debido a una mala configuración de las tablas de rutas de BGP, quedó inaccesible a través de Internet, produciendo la caída en cadena de las redes sociales (Whatsapp e Instagram) más utilizadas por los usuarios y la interrupción del servicio de otras empresas dependientes de Facebook como proveedor.

Por todo ello, existen multitud de estudios que pretenden identificar cuándo uno de estos sistemas autónomos no trabaja como debería, es decir, cuándo un AS no intercambia la información adecuada debido a una mala configuración local, ya sea de forma deliberada o no.

De la misma forma, también se encuentran herramientas a disposición del público que se encargan de monitorizar las actualizaciones BGP, tratando de determinar un posible anuncio[1] fraudulento (*Hijack*), una posible caída de servicio (*Outage*) o una filtración indebida de prefijos (*BGP Leak*) durante el intercambio de información entre los diferentes AS. Este es el ámbito en el que se encuentran las herramientas BGPStream [19] y BGP-Mon [28], que son accesibles a través de la plataforma de Cisco Crosswork Cloud [5]. Estas herramientas son fundamentales para BgpRS, puesto que mediante la plataforma social Twitter se encargan de publicar información relevante acerca de los eventos que pueden significar algún riesgo significativo en BGP.

1.2. Objetivos

En este TFM se presenta un sistema de recomendación BGP en forma de aplicación. Esta aplicación, que ha sido denominada como BgpRS (*BGP Recommendation System*), mediante los datos publicados por Cisco Systems proporciona recomendaciones para la

[1] Intercambio de información de encaminamiento entre sistemas autónomos (Sección 3.1.1)

configuración de *routers* BGP con el fin de intentar evitar aquellos sistemas autónomos que impliquen algún riesgo. Además, como característica adicional, BgpRS también proporciona herramientas para la visualización de datos, concediendo al usuario la capacidad de realizar un estudio histórico sobre los eventos generados por los diferentes países en BGP.

La presente guerra entre la Federación Rusa y Ucrania es un posible escenario para probar que BGP no es un simple protocolo de red. Como los AS de estos países pueden producir eventos *Outage* y *Hijack* diariamente, si mediante BgpRS se seleccionan diferentes momentos en el tiempo antes y después de la guerra, y se comparan los sucesos BGP de cada país, es posible visualizar el impacto de la guerra sobre Internet.

Algunos errores de configuración de los *routers* BGP también desencadenan eventos como los mencionados anteriormente. Por esta razón, surge la idea de clasificar los sistemas autónomos según su histórico de incidentes. Las tendencias obtenidas mediante los datos de estos eventos permiten establecer la reputación de los AS y que la aplicación BgpRS informe a los administradores BGP sobre la forma más adecuada para tratar a un AS cuyo comportamiento es errático, utilizando para ello instrucciones `vtysh` contempladas en Quagga o FRRouting y que son fácilmente aplicables a Cisco CLI.

1.3. Plan de trabajo

Durante el funcionamiento del protocolo BGP se producen numerosos eventos: aparición o desaparición de rutas, incorporación de nuevos AS, etc. En este trabajo no es necesario el uso de cada uno de estos eventos, ya que solo se necesitan aquellos que impliquen una caída de servicio (*Outage*) o un cambio en el origen de prefijos (*Hijack*).

La extracción de estos datos supone un requisito necesario para la implementación de BgpRS. Sin embargo, cabe destacar que su filtrado y clasificación requiere de un gran número de recursos no disponibles, que son accesibles mediante espejos o reflectores de rutas BGP (LG, BGP *Looking Glasses*). Por dicha razón, el tratamiento de los datos será realizado por las herramientas BGPStream y BGPMon, que internamente utilizan los LG para consolidar su información. De esta forma, mediante el uso de esta información será posible mantener un histórico consistente que servirá a la aplicación BgpRS.

Estas herramientas de Cisco se encargan de nutrir su propia base de datos histórica. Sin embargo, el acceso a sus datos solo es posible a través de su API de pago. Por otro lado, Cisco también publica en Twitter de forma gratuita la información de los eventos que identifica mediante estas herramientas. Uno de los objetivos de BgpRS es proporcionar una opción gratuita para el estudio de los eventos que se producen en BGP. Esto se realiza a través de la extracción de la información disponible en Twitter, por lo que la API de esta red social es una herramienta esencial para obtener gran parte de la información necesaria.

Una vez recopilada esta información, se realizará una clasificación por países, asociando el evento con el AS que lo produjo y la organización al mando. De esta forma, será posible visualizar si el impacto de ciertas situaciones internacionales incide sobre BGP.

Por último, a través de los diferentes datos obtenidos, se procederá a construir la mencionada funcionalidad de recomendación de BgpRS. Este sistema será capaz de determinar cómo es de necesario tomar medidas para un AS concreto, proporcionando instrucciones de configuración para evitar redirigir el tráfico por aquellos AS que hayan sido considerados como no fiables.

Capítulo 2

Estado del arte

Internet permite el intercambio de información de manera casi inmediata entre distintos puntos geográficamente distribuidos. En los años 60 tuvo lugar la guerra fría. Este hecho condujo a los Estados Unidos a crear una red militar exclusiva que permitiese el intercambio de información militar desde cualquier parte del país incluso ante un posible ataque enemigo, provocando que el departamento de defensa estadounidense ARPA (*Advanced Research Project Agency*) diese lugar a lo que hoy se conoce como Internet.

Desde entonces las necesidades de las redes de comunicación han aumentado notablemente, necesitando del soporte de un mayor número de elementos, medios y otras características hasta construir el sistema global que hoy conocemos.

2.1. Ámbito de estudio

Como ya se ha mencionado, la funcionalidad de BGP no es clasificable en una única capa del modelo OSI. Este protocolo, surge ante la necesidad de un encaminamiento dinámico capaz de sostener el gran volumen de rutas y redes que existen hoy en día. Desde los inicios de Internet ha existido la necesidad de encontrar un protocolo capaz de mantener dichas características. Por esta razón, existe una gran evolución y actualizaciones sobre protocolos de encaminamiento contemplados en distintos RFC[1].

Existen multitud de aproximaciones y descripciones de protocolos de encaminamiento realizados a lo largo del tiempo. En este aspecto, BGP surge en 1989 y es en el RFC 1105 [15] donde se encuentra una de sus primeras especificaciones. La sucesión de los RFC 1163 [16], 1265 [21], 1267 [17] y 1654 [23] plasman la evolución y la precedencia de la versión 4 de BGP utilizada actualmente que se contempla por primera vez en el RFC 1771 [20] publicado en 1995.

La versión 4 del protocolo BGP contiene todos y cada uno de los aspectos necesarios para que, bien configurado, funcione correctamente. Esta configuración es realizada por los diferentes administradores de sistemas o nodos BGP. No obstante, cabe destacar que incluso un sistema BGP bien configurado, puede presentar comportamientos no deterministas. Este es el caso de las denominadas *Wedgies* descritas en el RFC 4264 [9].

[1] *Request For Comments:* `https://www.ietf.org/rfc/`

2.2. Configuración y herramientas BGP

Como es de suponer, la correcta configuración de los nodos BGP no es una cuestión trivial. Además, como BGP es un protocolo global que se gestiona de manera local, puede inducir a una configuración personalizada con el fin de cubrir intereses particulares de los diferentes AS. En este aspecto y a lo largo del tiempo de uso de este protocolo de encaminamiento, se han podido observar multitud de problemas derivados de sus características. Por este motivo, existe una gran variedad de herramientas disponibles para observar sus comportamientos incoherentes y el alcance de cada uno de los mismos.

Cisco Systems, que es una de las entidades más influyentes en el campo de telecomunicaciones, mediante su API Cisco Crosswork posibilita la obtención de información sobre los sistemas BGP. Para comprender toda la información publicada por esta compañía es necesario conocer los diferentes elementos que componen una red BGP.

Para comenzar por lo más básico y para dar un punto de referencia con un ejemplo gráfico, Internet o la red de redes, es representable como un gran grafo con una gran cantidad de nodos donde cada nodo es un sistema autónomo que a su vez contiene usuarios, páginas *web* u otros servicios. Si nos fijamos en la figura 2.1 y tratamos de obtener la ruta con menor número de saltos entre dos nodos cualesquiera, se estaría tratando de realizar mentalmente aquello de lo que se encarga BGP por defecto y sin ser configurado. Establecer un camino óptimo entre dos sistemas autónomos, no es sencillo. En ocasiones, la ruta con menor número de saltos no es siempre la mejor solución, ya que en Internet se encuentran factores como el ancho de banda o el coste económico de utilizar una ruta que influyen a la hora de determinar cuál es la ruta óptima. Por esta razón, BGP es configurable y permite establecer prioridades entre diferentes rutas.

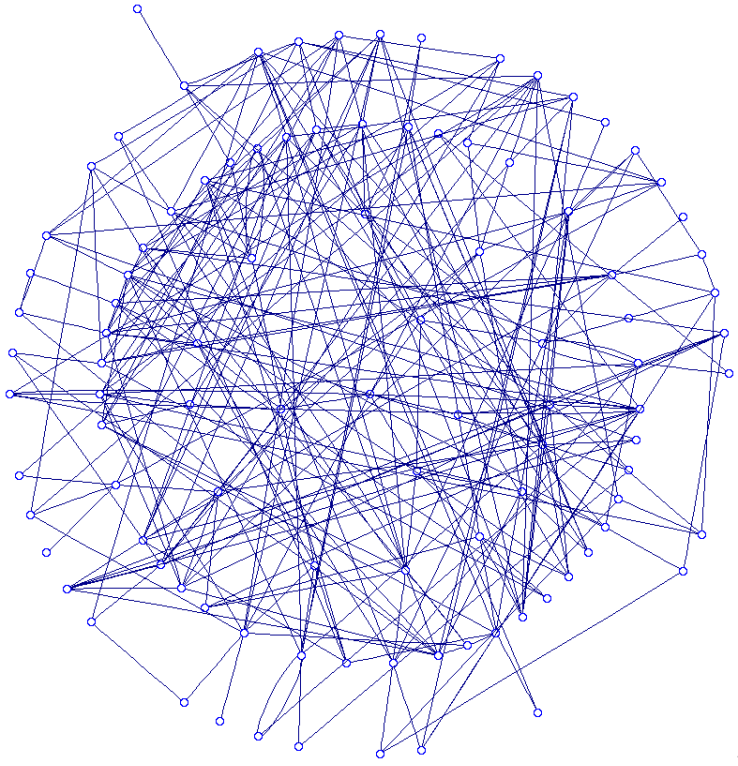


Figura 2.1: Representación de Internet mediante grafo

Existen multitud de factores complejos a los cuales BGP ofrece solución. Otro de los problemas a los que se enfrenta este protocolo, es la necesidad del dinamismo en la selección de rutas. Esto se debe a que si un nodo de la ruta deja de dar servicio por alguna razón, la comunicación desde el nodo inicial hasta el nodo final se vería interrumpida. Con el objetivo de recuperar esta comunicación, la opción más sencilla sería recalcular la ruta evitando el nodo que generó el problema. Sin embargo, Internet es realmente grande y es posible que estas situaciones se den en múltiples nodos, dificultando enormemente la ejecución de dicha solución.

No obstante, aunque BGP brinda solución a los problemas anteriormente descritos, se sabe que presenta comportamientos no deterministas o *Wedgies* que provocan dificultades ante la recuperación de las rutas que se eligieron como óptimas tras un fallo. Cuando se establecen dichas rutas por medio de atributos BGP y se produce un fallo, es posible que aparezcan problemas como que el tráfico siga rutas no intencionadas o que aparezcan estados inesperados.

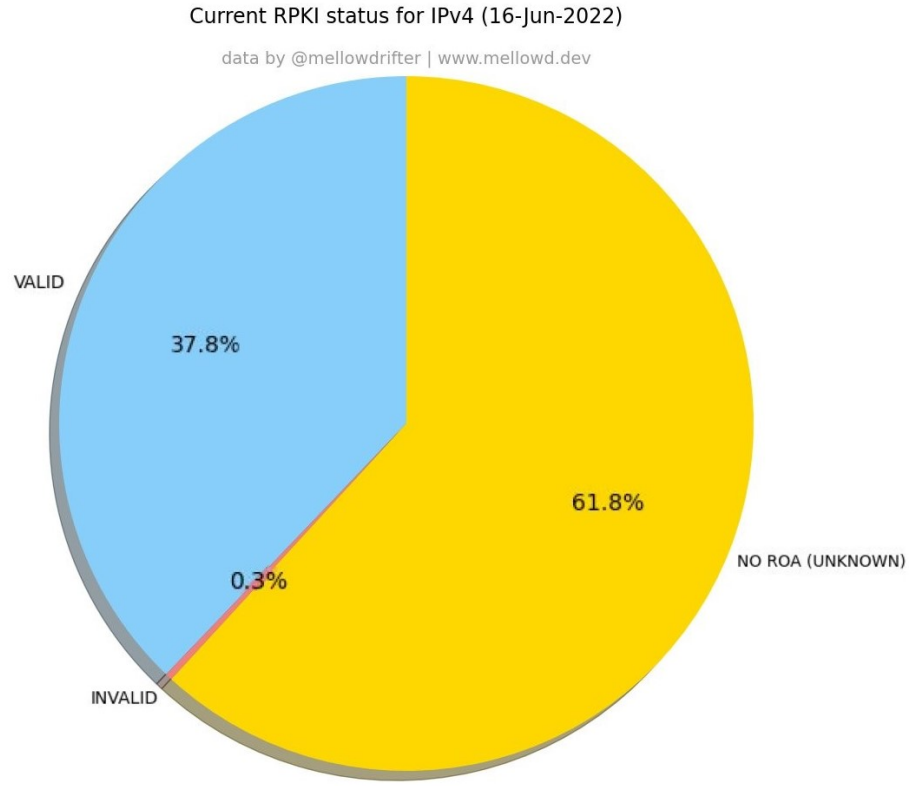
Además de esto, es posible que una mala configuración accidental o intencionada, concluya en la alteración del tráfico de determinados sistemas autónomos. La configuración de los atributos BGP en un AS, no solo afecta al sistema en cuestión, sino que también puede tener repercusiones globales afectando a terceros.

Estos problemas son los ya denominados *Outages*, *Hijacks* o *Leaks*. Los *Outages* son eventos inevitables puesto que el *hardware* de los dispositivos encargados de realizar las labores de encaminamiento se deteriora irremediablemente. Por ello, la única solución posible para estos eventos es mantener implementaciones robustas de recuperación ante fallos.

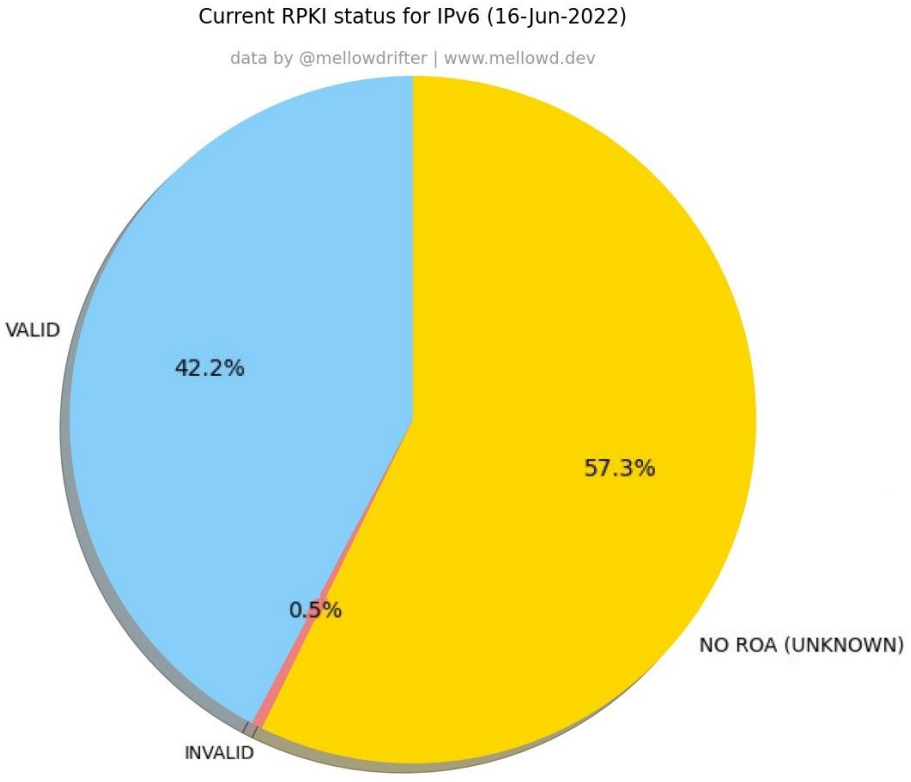
Por el contrario, los *Hijacks* y *Leaks*, sí son eventos evitables, que además ocurren con más frecuencia de la que deberían y que suponen un riesgo de seguridad en el protocolo BGP y en Internet. Esto promueve la búsqueda de soluciones por parte de grandes entidades del sector de las redes, dando lugar a propuestas como BGPsec [14] y RPKI (*Resource Public Key Infrastructure*) [13].

Estas herramientas trabajan conjuntamente para proporcionar un medio de validación de los prefijos anunciados durante el encaminamiento de BGP. Para realizar esta comprobación, RPKI se encarga de rechazar los anuncios originados desde un falso AS, mientras que BGPsec se encarga de comprobar la legitimidad de la ruta atravesada por dicho anuncio. Sin embargo, como se observa en la figura 2.2, la completa adopción de estas herramientas todavía es incipiente.

Por todo ello, en realidad Internet es menos robusta de lo que parece, no debido al protocolo BGP en sí mismo, sino debido a la constante lucha de intereses promovida por sus características de configuración, siendo este el motivo que impulsa el estudio y la elección de este Trabajo de Fin de Máster.



(a) Estado de adopción de RPKI en IPv4



(b) Estado de adopción de RPKI en IPv6

Figura 2.2: Estado de adopción de RPKI en IPv4 e IPv6, *fuentes:* `https://twitter.com/bgp4_table` *y* `https://twitter.com/bgp6_table`

Capítulo 3

Protocolo BGP

En este capítulo se describirán las características más relevantes del protocolo BGP con el fin de comprender de forma más extensa el motivo y el alcance de este TFM. La mayor parte de este contenido ha sido obtenido durante el periodo de estudios del Máster de Informática, en concreto de la asignatura de Redes de Nueva Generación y la documentación elaborada por Juan Carlos Fabero Jiménez [7].

3.1. Sistemas Autónomos

Un sistema autónomo es un conjunto de redes IP y encaminadores que se encuentran bajo el control de una o varias organizaciones que emplean una política de encaminamiento común. Los encaminadores de una red BGP pueden intercambiar mensajes de manera interna (iBGP, *Internal BGP*) o externa (eBGP, *External BGP*). De esta forma, dos *routers* que intercambian información de encaminamiento mantienen una sesión iBGP cuando ambos pertenecen al mismo AS; y mantienen una sesión eBGP cuando pertenecen a dos AS distintos. Un ejemplo de esto se ve reflejado en la figura 3.1.

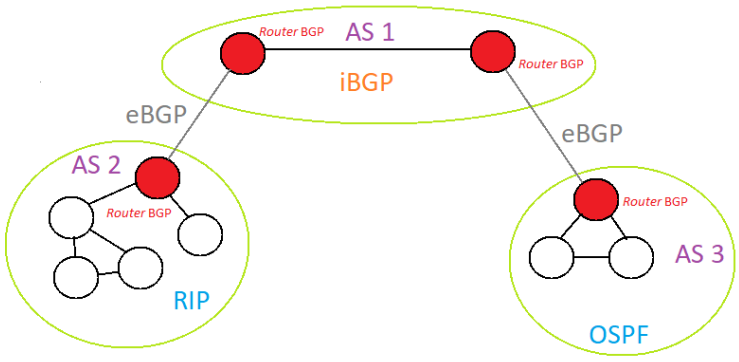


Figura 3.1: Encaminamiento interno y externo BGP

Cada sistema autónomo está identificado por un número único que se denomina ASN (*Autonomous System Number*). Estos identificadores son delegados por IANA (*Internet Assigned Numbers Authority*), la autoridad responsable de la coordinación global de los diferentes recursos que utiliza Internet.

Los ASN utilizan números de 16 o 32 bits para identificar a cada una de las organizaciones que existen en Internet. Estos números se han simplificado en la imagen 3.1, enumerando los sistemas autónomos del 1 al 3 para mayor claridad.

IANA distribuye en forma de bloques los diferentes ASN a cada uno de los RIR (Regional Internet Registry) del mundo. Estas regiones están compuestas por organizaciones como ISP (*Internet Service Providers*), empresas tecnológicas, universidades, agencias gubernamentales e instituciones científicas. Actualmente existen los cinco Registros Regionales de Internet que se ven reflejados en la tabla 3.1.

Región	Nombre del registro	Sitio web
América del Norte	ARIN	`www.arin.net`
Europa	RIPE NCC	`www.ripe.net`
Pacífico Asiático	APNIC	`www.apnic.net`
América Latina	LACNIC	`www.lacnic.net`
África	AfriNIC	`www.afrinic.net`

Tabla 3.1: Tabla de registros regionales de Internet

3.1.1. Definiciones

Para comprender mejor el protocolo BGP es necesario primeramente enumerar los distintos componentes y tecnicismos que se utilizan. Por una parte, BGP utiliza la relación de vecindad de tal forma que dos AS denominados **vecinos** son aquellos encaminadores que, situados en diferentes AS o en un mismo AS, intercambian políticas e información de encaminamiento. Este intercambio de información se realiza a través de anuncios.

Un **anuncio** es el envío de información de encaminamiento a los sistemas vecinos del AS que originó el mismo. Estos anuncios son de diversos tipos y se describirán en mayor detalle en la sección 3.2.1.

En cuanto a la forma en la que se generan estos anuncios, existe un AS **origen** encargado de construir el mensaje, insertando en éste la información de encaminamiento necesaria. De la misma manera, también existe un AS receptor que deberá **aceptar** el mensaje para utilizar la información recibida y, en caso necesario, propagarla a través de la red. Este comportamiento se ve reflejado en la figura 3.2.

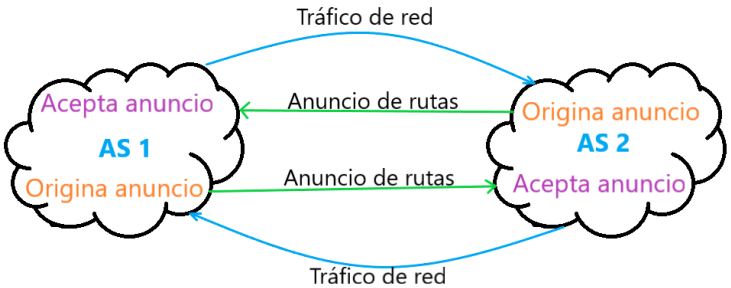


Figura 3.2: Representación del comportamiento de los sistemas autónomos en una red BGP

De esta forma, si existen dos AS nombrados en la imagen anterior como AS1 y AS2, para que estos intercambien información de encaminamiento, es necesario que AS2 anuncie primeramente su existencia a AS1, y que posteriormente AS1 acepte dicho anuncio para comenzar el flujo de datos.

Este flujo de datos transitará en el sentido opuesto a los anuncios de rutas, y para que el mismo sea bidireccional deberá realizarse el proceso anterior en sentido contrario, es decir, que AS2 anuncie su redes y que AS1 las acepte.

3.1.2. Tipos de Sistemas Autónomos

Los sistemas autónomos se clasifican según la conexión y los tipos de operación que realicen como Sistemas *Multihomed*, Sistemas de Tránsito o Sistemas *Stub*, los cuales se definen a continuación.

- **Sistemas *Multihomed*:** son aquellos que están conectados a más de un AS, pero que no permiten el tráfico de tránsito a través de ellos. La característica principal de este tipo de AS es su capacidad de seguir manteniendo su conexión a internet aun en caso de fallo de alguno de los AS a los que esté conectado, haciendo uso del resto de conexiones redundantes que dispone y que se encuentran operativas. Un ejemplo gráfico de este sistema autónomo se ve reflejado en la figura 3.3.

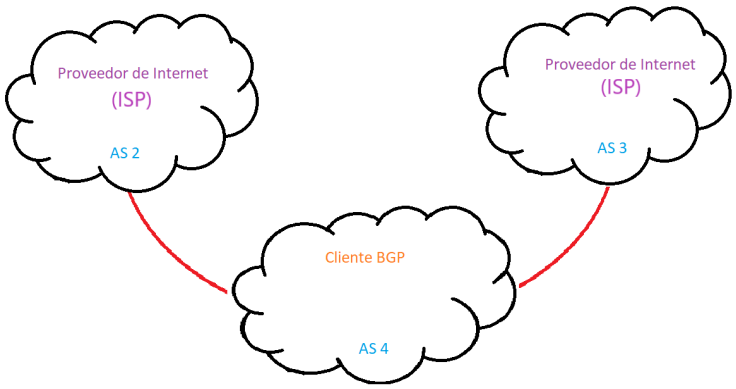


Figura 3.3: Conexión y tipo de operación de los Sistemas *Multihomed*

- **Sistemas de Tránsito:** son aquellos sistemas que proporcionan conexión entre distintos AS. Un ejemplo de sistema de tránsito es el perteneciente a los proveedores de servicio en Internet. La figura 3.4 refleja el comportamiento de este tipo de sistemas. En ella se observa que el AS1, situado como nodo central, permite el tránsito del tráfico a través de sí mismo comportándose como un ISP.

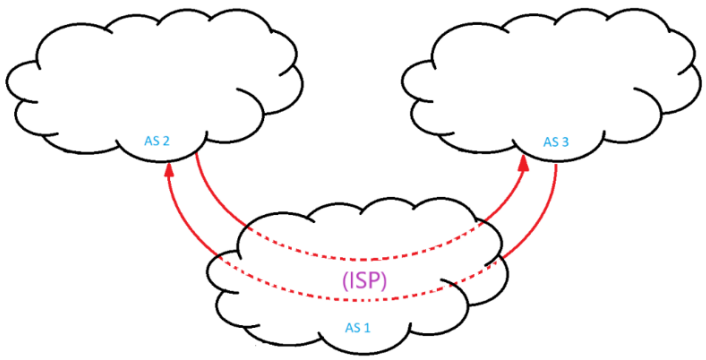


Figura 3.4: Conexión y tipo de operación de los Sistemas de Tránsito

- **Sistemas *Stub*:** son aquellos sistemas finales o clientes que solo están conectados a otro AS que actúa de proveedor. Estos sistemas pueden haber establecido vecindad con otro AS, aunque su relación no se refleje en los servidores públicos (*peering*). Este es el caso de sectores de transporte o entornos financieros. Un ejemplo de este tipo de sistemas se ve reflejado en la figura 3.5.

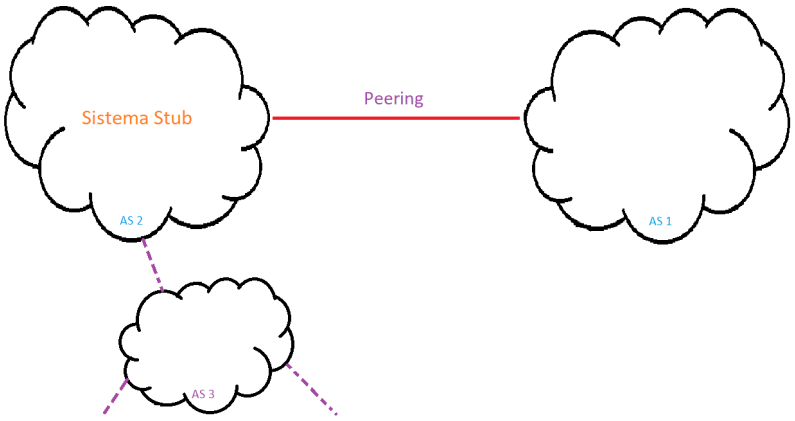


Figura 3.5: Conexión y tipo de operación de los Sistemas *Stub*

3.2. Encaminamiento BGP

Para comenzar es necesario conocer que BGP necesita del protocolo IP, en su versión 4 (IPv4) o versión 6 (IPv6), para establecer sus tablas de encaminamiento. En el caso de utilizar la versión 4 de este protocolo, el encaminamiento se realiza en el nivel correspondiente al `NetID`. Esto genera una entrada en la tabla por cada destino de red, produciendo tablas demasiado grandes e insostenibles. Para solventar dicho problema, se utiliza la notación CIDR (*Classless Inter-Domain Routing*). Este tipo de notación permite la agrupación de varias redes por cada entrada de la tabla, dando lugar a los prefijos de red.

La gestión de estos prefijos de red es realizada por los diferentes RIR, que distribuyen los mismos y el ASN que los engloba a las diferentes organizaciones. Por su parte, estas organizaciones son las encargadas de mantener y gestionar los diferentes AS que le hayan sido asignados.

Llagados a este nivel, se puede comenzar a hablar sobre aspectos más concretos del encaminamiento BGP. Este protocolo utiliza multitud de mensajes y atributos, configurables desde cada AS, para realizar su función. En esta sección se hablará de cada uno de ellos, comprobando la necesidad de que cada AS establezca acuerdos particulares.

3.2.1. Mensajes BGP

Como se ha mencionado en la sección 3.1.1, el AS encargado de originar los anuncios también es el responsable de construir el mensaje incorporando la información necesaria. Estos mensajes son de distintos tipos y se describen a continuación.

- **Mensaje *OPEN*:** se utiliza para establecer la sesión BGP. En este tipo de mensajes se incluye el identificador del sistema autónomo que originó el mensaje, así como un valor inicial para el `hold-timer`.

- **Mensaje *KEEPALIVE*:** se utiliza para confirmar el inicio de sesión BGP y tiene la función adicional de controlar el tiempo de `hold-timer`. De esta forma, si no se recibe el mensaje `KEEPALIVE`, en un tiempo inferior al indicado por el `hold-timer`, se producirá el fin de la sesión BGP.
- **Mensaje *NOTIFICATION*:** se utiliza para la notificación de errores producidos por fallos en las cabeceras del mensaje, problemas en los mensajes `UPDATE`, errores en las máquinas de estados, cierres administrativos o cuando se supera el tiempo de `hold-timer`. Por ejemplo, cuando se vence el tiempo de `hold-timer` antes de la recepción del mensaje `KEEPALIVE` se genera un mensaje de notificación para indicar que la sesión BGP ha terminado, produciendo la invalidación de las rutas del AS originario del mensaje.
- **Mensaje *UPDATE*:** se utiliza para el intercambio de información de encaminamiento entre los sistemas autónomos vecinos. Este es uno de los mensajes más relevantes del protocolo y será de gran importancia para el presente trabajo. El envío de estos mensajes se produce después de establecer la sesión BGP, enviando la tabla completa de encaminamiento al AS vecino con las nuevas rutas y las rutas eliminadas. El formato de datos de este tipo de mensajes se observa en la tabla 3.2.

Cabecera
Tamaño del campo de rutas que son inalcanzables
Rutas que son inalcanzables
Tamaño del campo de atributos de ruta
Atributos de ruta
Información de alcanzabilidad de red

Tabla 3.2: Formato de datos de los mensajes de tipo `UPDATE`

Por último, cabe destacar que este tipo de mensajes anuncian información de encaminamiento sobre una ruta o conjunto de rutas, de forma que con el uso de un prefijo y sus atributos se determine la ruta que seguirá el tráfico de red.

3.2.2. Atributos de los anuncios

Los atributos de los anuncios que viajan dentro de los mensajes `UPDATE` se utilizan para dirigir el proceso de selección de ruta. Los administradores de los AS, mediante la manipulación de estos atributos, pueden definir determinadas políticas de tráfico siguiendo ciertos factores económicos, acuerdos comerciales o políticas internacionales. Es altamente probable que algunos de estos factores influyan negativamente sobre la selección de la ruta óptima, dando lugar a pensar que Internet podría funcionar mucho mejor a cómo lo hace actualmente.

Estos atributos son de diversos tipos, y en este apartado se detallarán algunos de los más comunes. Por un lado, se encuentran los **atributos bien conocidos y obligatorios**, que deben ser reconocidos en todas las implementaciones de BGP y han de estar presentes en todos los mensajes `UPDATE`.

De manera opuesta, existen atributos que pueden no estar contenidos en los mensajes `UPDATE`, aunque sean reconocidos en las implementaciones BGP. Este es el caso de los atributos **bien conocidos y no obligatorios**.

Por último, según su nivel de propagación los atributos pueden ser **transitivos** o **no transitivos**. La diferencia entre estos dos atributos reside en que los transitivos se deben propagar, al contrario que los no transitivos. Además, este tipo de atributos, al contrario que los bien conocidos, pueden no ser especificados en todas las implementaciones BGP.

- ***ORIGIN:*** este atributo bien conocido y obligatorio se utiliza para indicar el origen de la ruta. Puede tomar tres valores: IGP (*Interior Gateway Protocol*), EGP (*Exterior Gateway Protocol*) e *Incomplete*. El valor IGP significa que el anuncio es interno al propio AS (anuncio explícito). El valor EGP se considera obsoleto actualmente. El valor *Incomplete* indica que la ruta se ha aprendido mediante algún otro medio distinto de BGP (inyección desde otro protocolo, como por ejemplo OSPF).
- ***AS_PATH:*** es un atributo bien conocido y obligatorio que se utiliza para indicar, mediante una cadena de texto, el camino y los AS que ha atravesado el mensaje BGP. Este atributo es uno de los más influyentes en la toma de decisiones de rutas de BGP. Para llevar a cabo su funcionamiento, cada AS por el que circula el mensaje debe actualizar el valor de este atributo incluyendo su ASN al final de la cadena. Este atributo también se utiliza para evitar bucles en el proceso de transmisión. Para ello, se obliga a cada AS por el que circule el mensaje a comprobar si su ASN ya está contenido en la cadena, evitando la propagación del mensaje si esto ocurre.
- ***NEXT_HOP:*** este atributo de carácter bien conocido y obligatorio se utiliza para indicar la dirección IP del encaminador frontera que debe ser utilizado como siguiente salto.
- ***LOCAL_PREF:*** este atributo es bien conocido y no obligatorio e indica la preferencia local hacia rutas externas. Este atributo de valor numérico indica la preferencia ante rutas, por lo que un valor más alto indicaría la preferencia o selección de la ruta en cuestión. Además, como es un atributo local, se propaga únicamente en iBGP.
- ***MULTI_EXIT_DISCRIMINATOR:*** es un atributo opcional no transitivo que se utiliza para determinar una preferencia de ruta ante la posibilidad de que existan dos AS con más de una conexión entre sí. De esta forma, el sistema que originó el mensaje puede sugerir una de esas conexiones mediante este atributo.
- ***ATOMIC_AGREGATE:*** es un atributo bien conocido y no obligatorio que se utiliza para indicar que existe un agregado de rutas. Esto sucede cuando un encaminador recibe varios anuncios y necesita elegir una red que englobe a todas si estas se encuentran solapadas.
- ***AGGREGATOR:*** este atributo opcional y transitivo indica el último ASN que formó el agregado de rutas, así como la IP del encaminador que lo realizó. Esto es utilizado con el motivo de dar a elegir al receptor si aceptar dicho anuncio o no.

Desde el punto de vista de este trabajo, uno de los atributos que cobrará más importancia será el `AS_PATH`. Después de que el mensaje de actualización se propague completamente, este atributo contendrá el ASN de cada uno de los nodos atravesados, facilitando al protocolo de un medio para averiguar el camino óptimo entre los nodos de la red. Por ejemplo, como este atributo es manipulable, cualquiera de los nodos participantes durante el flujo del anuncio podría indicar más de una vez su propio ASN, haciendo que la característica de camino óptimo se viese alterada (`AS_PATH` *prepend*).

El aspecto anterior es controlable mediante comandos de filtrado de rutas, de forma que con el uso de expresiones regulares sobre el atributo `AS_PATH` se puede identificar cuándo un AS está realizando una de estas acciones. Además, mediante otros atributos como el `LOCAL_PREF` o el `MULTI_EXIT_DISCRIMINATOR`, se puede establecer una preferencia sobre las rutas recibidas. Las capacidades que ofrecen estos atributos son utilizadas por BgpRS para crear las recomendaciones que son proporcionadas al usuario. Este aspecto se verá más adelante en el capítulo 5.

3.3. Proceso de decisión BGP

Cuando se reciben varios anuncios de ruta para un mismo destino se necesita decidir cuál de ellos se acepta y se usa para encaminar el tráfico de red. Para hacer esto posible, el RFC 4271 [22] define un proceso de decisión implementado en el protocolo BGP. Este proceso de decisión se utiliza para seleccionar las rutas locales y las rutas anunciadas al exterior. Además, opcionalmente proporciona características para realizar agregados de rutas y para reducir la información de encaminamiento.

Este proceso de BGP consiste en tres fases distintas, cada una de ellas desencadenada por un evento diferente. La primera fase se encarga de calcular el grado de preferencia para cada una de las rutas recibidas, utilizando para ello el atributo `LOCAL_PREF`. De esta forma, se procede a descartar ciertas rutas que no son consideradas para la siguiente fase.

La segunda fase se encarga de elegir la mejor ruta entre aquellas que no han sido descartadas. Esta fase y la tercera, comparten el objetivo de realizar la última criba de rutas, utilizando los atributos `NEXT_HOP` y `AS_PATH` para evitar destinos inalcanzables, posibles bucles y descartar rutas no óptimas.

Después de todo este proceso, aun así pueden existir varias rutas hacia un mismo destino. De esta forma, en caso de empate, el RFC 4271 incorpora diferentes pasos que deben tomarse ordenadamente para la elección de la ruta prioritaria. El proceso de decisión consiste, de manera resumida, en los siguientes pasos:

1. Se elige aquella ruta con mayor `LOCAL_PREFERENCE`.
2. Si existe igualdad en la preferencia local de rutas, se elige aquella ruta que fue generada localmente (iBGP).
3. En caso de que todas las rutas entre las que se quiera decidir se hayan originado de manera local, se prefiere aquella que tenga el `AS_PATH` más corto, siendo éste, en teoría, el óptimo.
4. En caso de que el número de AS del `AS_PATH` sea el mismo, el proceso de decisión contempla el atributo `ORIGIN`, prefiriendo aquellos anuncios que se realizaron de manera explícita.

5. Si aun así sigue existiendo más de un anuncio, se elige aquel que tenga menor `MULTI_EXIT_DISCRIMINATOR`.
6. Si el atributo de discriminación no es suficiente para establecer la ruta, se prefiere elegir aquellas rutas que provienen de manera externa (eBGP) que de manera interna (iBGP).
7. En caso de igualdad, se elige aquella ruta que tenga un coste menor IGP al siguiente salto (`NEXT_HOP`).
8. En caso de que la igualdad persista, se elegirá aquella cuyo vecino tenga un identificador BGP menor.
9. Por último, en caso de igualdad se seleccionará la ruta cuyo vecino tenga una dirección IP menor.

Como se puede observar, para elegir el anuncio que finalmente se establecerá en la tabla de rutas se contemplan las configuraciones locales de los AS, es decir, el proceso de decisión que incorpora BGP contempla principalmente los intereses particulares de las entidades y organizaciones al mando.

Esta característica y la posible manipulación de los atributos de BGP hacen que se presenten ciertas situaciones problemáticas. Un ejemplo de ello ocurrió el 28 de junio de 2011 [3]. Por aquel entonces, Egipto se encontraba en una situación de revueltas y protestas contra el gobierno de Hosni Mubarak. Esto desembocó en la orden de suspender los servicios móviles del país y la desconexión de Internet.

Otro hito bastante destacado ocurrió en 2008 [25], cuando el gobierno paquistaní ordenó a los ISP del país que censuraran la plataforma de YouTube con el fin de impedir que su población visualizara un tráiler de una película antislámica. Para conseguir esto, Pakistan Telecom cambió las tablas de direccionamiento de BGP para redirigir a sus usuarios a una página de Internet en la que se les informaría que YouTube estaba bloqueado. El problema realmente se produjo durante la ejecución de esta medida. El ISP anunció accidentalmente la nueva ruta al resto de proveedores que aceptaron el anuncio y lo trasmitieron por todo Internet. Este error no solo provocó la inaccesibilidad de YouTube en gran parte del mundo, sino que también saturó al ISP paquistaní.

Estos hechos son una muestra de cómo las decisiones gubernamentales, los errores de configuración y la flexibilidad del protocolo pueden afectar globalmente sobre Internet. Sin embargo, existen muchos otros casos trascendentales que no han sido documentados públicamente y que repercuten sobre el funcionamiento de Internet. Aunque BgpRS no permite identificar las decisiones gubernamentales que afectan sobre BGP ni los errores de configuración de los AS que producen fallos, sí posibilita obtener recomendaciones para los AS según su reputación y además permite estudiar los eventos producidos desde cualquier país.

Capítulo 4

Información de actualizaciones BGP

A lo largo de este capítulo se procederá a detallar los procesos realizados para la obtención de los diferentes eventos que se producen en BGP, así como la tecnología empleada para su consecución. Por último, se procederán a explicar las razones y las herramientas utilizadas para almacenar dicha información de manera que sea accesible por BgpRS (*BGP Recommendation System*), aplicación implementada para este TFM.

4.1. Obtención de información de actualizaciones

Como se ha visto en el capítulo 3, el protocolo BGP proporciona opciones de configuración muy versátiles. A través del uso de los diferentes atributos vistos en la sección 3.2.2 y con el proceso de selección de rutas de la sección 3.3, este protocolo puede configurarse de tal forma que el *router* BGP siga los intereses de las organizaciones bajo su mando. Este aspecto sería algo positivo si se garantizase que cada una de las organizaciones, pese a sus posibles intereses internos, velase por el correcto funcionamiento de Internet. Sin embargo, esto en la mayoría de las ocasiones no es así, provocando en definitiva comportamientos que podrían desembocar en la pérdida de servicio o la sustracción de información.

Existen multitud de problemas derivados de errores de configuración en BGP, pero este trabajo se centra en los que se consideran más importantes y que, cuando ocurren, implican problemas de mayor rango. Este es el caso de los eventos *Outage*, *Hijack* y *Leak* de BGP. Cabe destacar que los eventos de tipo *Leak* no se han tenido en cuenta al ser mucho menos frecuentes y puesto que la extracción de los mismos fue imposible al ser solo accesibles desde el sitio *web* de BGPStream.

Llegado este punto y conociendo que BGP es un protocolo de encaminamiento que permite que cada sistema autónomo anuncie los prefijos IP que le fueron asignados, se puede comprender en qué consiste un secuestro BGP. Este fenómeno que, se podría considerar como ciberataque, se produce principalmente porque el protocolo BGP no contempla la posibilidad de que un AS propague información falsa a los vecinos a los que está conectado.

Los secuestros BGP son producidos a través de la manipulación de las tablas de encaminamiento posibilitando que un *router* BGP anuncie prefijos que en realidad no le pertenecen. El objetivo de esta manipulación es engañar a los nodos BGP de tal manera que el protocolo identifique una mejor ruta disponible hacia el destino, produciendo una

redirección del tráfico de red hacia el AS atacante y abandonando la ruta original. Esta redirección de tráfico puede afectar a los usuarios finales de tal forma que, mientras el usuario considera que está accediendo a sitios legítimos, en realidad lo está haciendo a sitios donde se produzca una posible sustracción de credenciales, se provoque la descarga de *malware* o se realicen otras actividades perniciosas.

Una redirección del tráfico malintencionada es realmente perjudicial. Además, estos ataques pueden propagarse a nivel internacional. Teniendo en cuenta que las grandes organizaciones mundiales también utilizan los servicios de Internet para su intercambio diario de información, se puede afirmar que los ataques anteriormente mencionados también se dan a este nivel. Este aspecto se ha podido comprobar mediante los ejemplos citados en la ultima parte de la sección 3.3.

Existen extensiones de BGP ideadas para evitar este tipo de ataques. Un ejemplo de ello son las ya mencionadas BGPsec y RPKI que, mediante criptografía, firmas digitales y la sustitución del atributo `AS_PATH` por `BGPsec_Path`[1] verifican la legitimidad de los anuncios de las rutas de los diferentes AS. Sin embargo, estas medidas deben ser adoptadas por cada una de las entidades de Internet para evitar totalmente estos tipos de evento. Por este motivo, todavía se contemplan este tipo de ataques en Internet, aunque son identificables y evitables mediante otros medios, como se verá a lo largo de este trabajo académico.

Las caídas de servicio en BGP son otro fenómeno perjudicial en sí mismo. Estos fallos se producen por problemas en el hardware o por reinicios administrativos de los nodos del sistema autónomo en cuestión. En este aspecto, destacan los problemas de recuperación que tiene BGP. Por este motivo las tablas de encaminamiento albergan métodos de seguridad para mantener enlaces secundarios y así evitar la pérdida completa de servicio. Sin embargo, los problemas comienzan al tratar de recuperar la ruta primaria, ya que las configuraciones realizadas por los administradores y la existencia de enlaces redundantes pueden producir los ya mencionados *Wedgies*.

El seguimiento e identificación de cada uno de los problemas anteriores es verdaderamente complejo, pero mediante el uso de las herramientas que se detallan seguidamente se verá que es posible acceder a esta información que servirá como base para este TFM.

4.1.1. BGPStream

BGPStream es un recurso de carácter público y gratuito que permite el seguimiento de alertas sobre secuestros e interrupciones en el protocolo BGP. Esta herramienta utiliza un proceso automatizado para la identificación de los problemas más relevantes y de mayor repercusión, clasificando en el proceso el tipo de evento y los ASN involucrados. El objetivo principal de esta herramienta es proporcionar la información a los administradores y propietarios de la red perjudicada con el fin de que respondan de la manera que consideren lo más rápido posible.

Para comprender el funcionamiento de esta herramienta primeramente hay que conocer cómo funcionan y cómo se organizan las tablas de encaminamiento BGP. Esto se realiza a través de las RIB (*Routing Information Base*), las cuales clasifican la información de encaminamiento de la siguiente manera:

[1] `BGPsec_Path` realiza las mismas funciones que `AS_PATH`, pero además el *router* que transmite el anuncio, incluye en el atributo el ASN del destino al que va dirigido. Este funcionamiento impide que se encapsulen varios prefijos en una sola actualización, y como el ASN del receptor se encuentra en el anuncio, se debe generar una actualización independiente por cada vecino.

- ***Adj-RIBs-In:*** Consiste en la información de las rutas aprendidas a través de los mensajes de actualización provenientes de los vecinos a los que el *router* BGP se encuentra conectado.
- ***Loc-RIB:*** Se trata de aquellas rutas seleccionadas de la tabla `Adj-RIBs-In` según las políticas locales. El *router* BGP se encargará de instalar dichas rutas en sus tablas de direccionamiento para el envío de paquetes.
- ***Adj-RIBs-Out:*** Son aquellas rutas seleccionadas de la tabla `Loc-RIB` para ser anunciadas al resto de vecinos. De esta forma, el *router* BGP crea una tabla `Adj-RIB-Out` personalizada para cada vecino mediante configuraciones locales.

La obtención de la información contenida en estas tablas permite que BGPStream identifique los problemas producidos en BGP. Como se puede contemplar en el artículo escrito por Orsini et al. [19], algunas organizaciones administrativas publican información de estas tablas para la resolución de problemas, la monitorización de información y para propósitos de investigación. Esta información es accesible mediante los BGP *Looking Glasses* como es Hurricane Electric's [6], RedIRIS [18] o similares. De esta forma y mediante la adquisición de información continua con implementaciones como OpenBMP [24], BGPStream es capaz de obtener los datos para realizar su función.

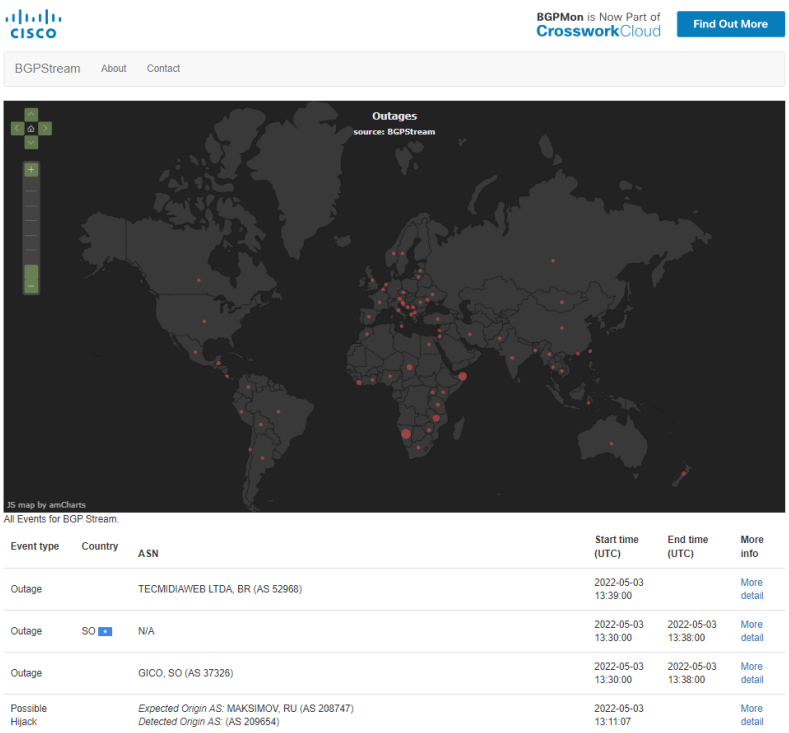


Figura 4.1: Información proporcionada por BGPStream, *fuente:* `https://bgpstream.com/`

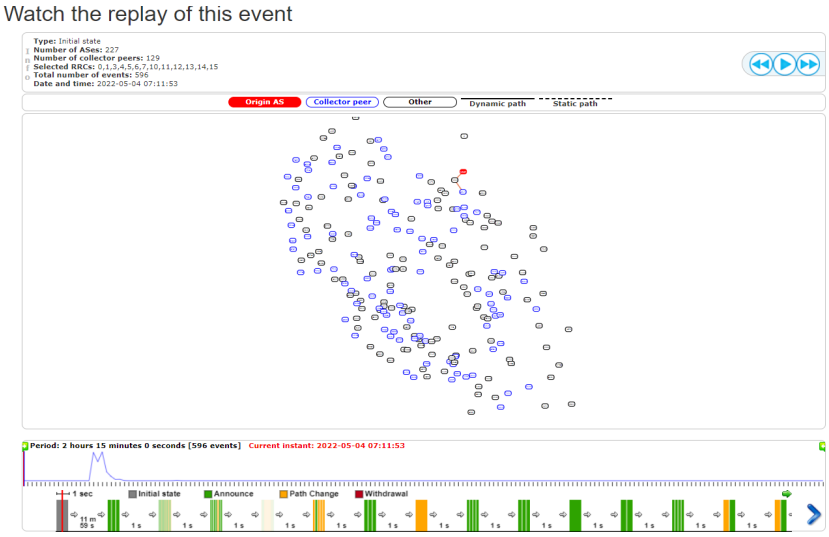
Para que esta información sea accesible de manera pública, BGPStream utiliza Twitter como medio de comunicación, por lo que esta información no solo es visible para los administradores, sino que también es accesible para otros usuarios interesados. Además de esto, BGPStream también proporciona esta misma información en su página *web* como se observa en la imagen 4.1.

4.1.2. BGPMon

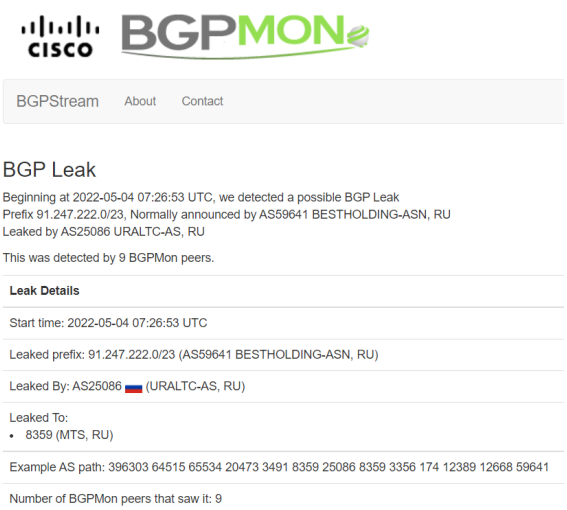
BGPMon es una extensión ligada a BGPStream que se encarga de analizar el gran volumen de mensajes BGP producidos diariamente. Esta herramienta permite identificar aquellos eventos que puedan causar algún daño al correcto funcionamiento de BGP.

Una de las particularidades de BGPMon es que proporciona un grafo interactivo que permite reproducir el momento exacto del incidente, pudiendo observar los mensajes que intercambian cada uno de los AS involucrados en el proceso. Esta función se realiza a través de una herramienta `JavaScript` de *software* libre denominada BGPlay.js. En la figura 4.2a, se representa uno de los grafos formados mediante esta herramienta.

Por otra parte, dependiendo del incidente, también proporciona datos de gran relevancia para BgpRS, como el *AS_PATH*, el número de prefijos afectados o la fecha de inicio y fin del incidente. Un ejemplo de esto se representa en la figura 4.2b.



(a) Ejemplo de grafo BGP interactivo



(b) Información proporcionada por BGPMon

Figura 4.2: Información extraíble a través de la herramienta BGPMon, *fuente:* `https://bgpstream.com/event/290962`

Por último, cabe destacar que existe una forma adicional de extraer información de la *web* de BGPMon mediante la interfaz de inspección `HTML` y `JavaScript` incluida en los navegadores. En la figura 4.3, se observa el resultado de una inspección sobre la página *web* de BGPMon, mediante la que se obtiene el objeto `JavaScript` a la derecha de la imagen. Este objeto es un diccionario que contiene información detallada sobre el evento visualizado en ese momento que permite el análisis de cada mensaje intercambiado desde el inicio hasta el fin del incidente.

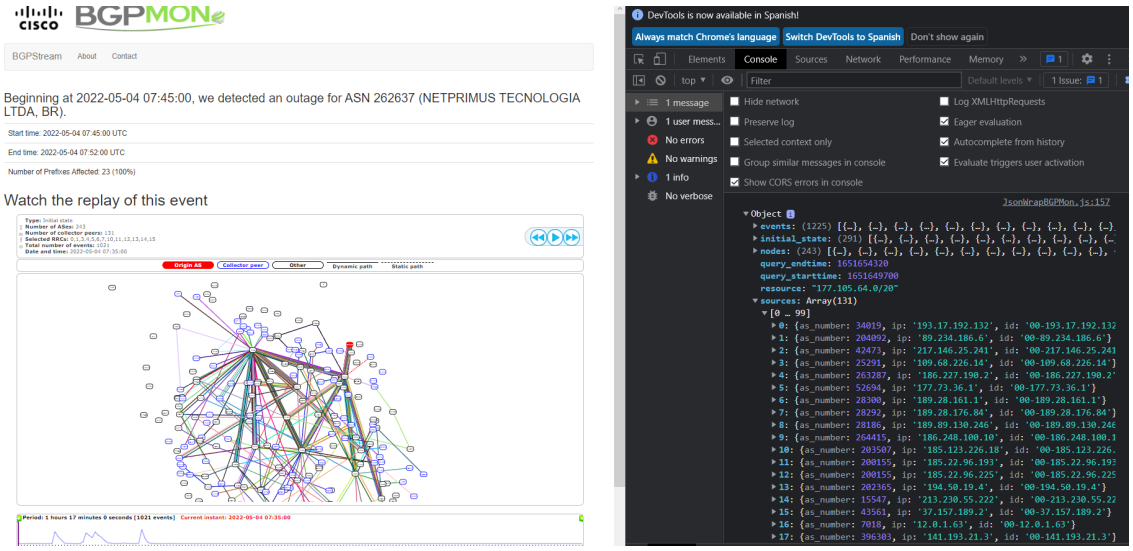
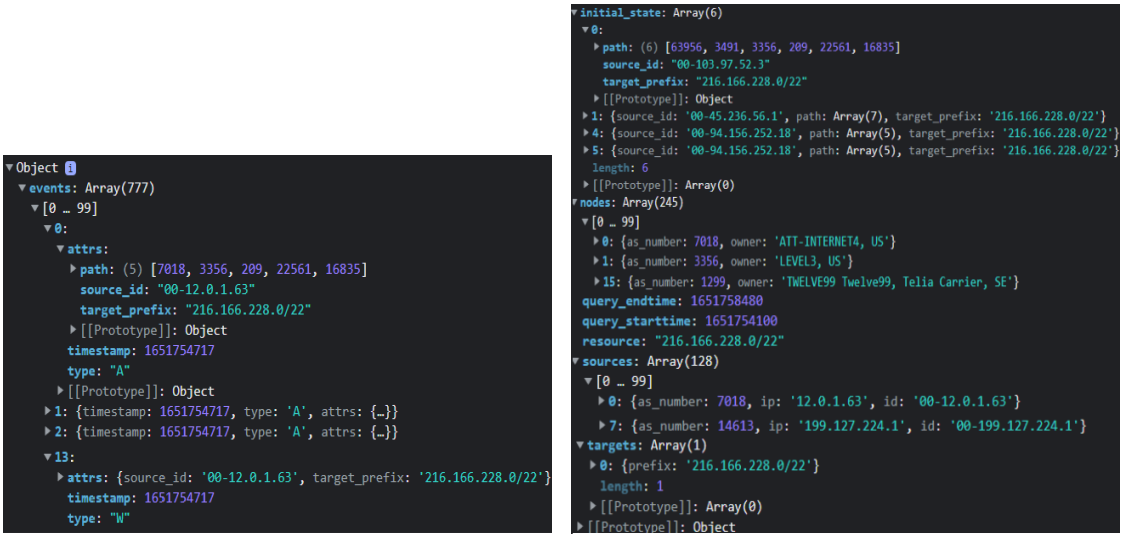


Figura 4.3: Inspección por consola `JavaScript` de la *web* de BGPStream, *fuente: https://bgpstream.com/event/290962*

En el interior del objeto representado en la figura 4.4, se encuentra información de cada uno de los nodos cercanos al AS que originó el problema, las organizaciones a las que pertenecen y los prefijos que fueron anunciados mientras se producía el suceso.



(a) Primer fragmento del objeto. (b) Segundo fragmento del objeto

Figura 4.4: Objeto `JavaScript` visualizable a través de la *web* BGPMon, *fuente: https://bgpstream.com/event/290962*

Teniendo en cuenta el gran potencial de esta información, se le intentó dar uso para dotar de calidad al histórico de datos en construcción. Sin embargo, tras varios intentos de obtención mediante el uso de diferentes librerías como Selenium [2], se determinó que no había forma posible de conseguirlo, al menos a través de las librerías disponibles para Python [27], lenguaje elegido para el desarrollo de la aplicación BgpRS.

4.1.3. Twitter API

Como ya se ha mencionado a lo largo de este capítulo, BGPStream utiliza la red social Twitter para publicar información acerca de los diferentes eventos que tienen lugar en BGP. Esta información se publica a través de la cuenta de Cisco BGPStream (*@bgpstream*) y sigue un formato determinado para cada tipo de evento.

BGPStream identifica y publica información acerca de dos tipos de eventos principalmente. Por una parte, los eventos BGP identificados como *Outages* o interrupciones de servicio, y por otro, los eventos *Hijack* que identifican un secuestro de prefijos. La información referente a estos dos eventos se publica generalmente como se representa en la figura 4.5, donde la imagen 4.5a corresponde a un evento *Outage*, y la imagen 4.5b a uno de tipo *Hijack*.

(a) Ejemplo de mensaje tipo *Outage* publicado en Twitter

(b) Ejemplo de mensaje tipo *Hijack* publicado en Twitter

Figura 4.5: Información extraíble a través de Twitter, *fuente:https://twitter.com/bgpstream*

La información contenida en estos mensajes proporciona datos consistentes de manera resumida, pudiendo estructurar los campos de datos como se detallan a continuación.

- **Formato de mensajes Outage:** en el caso de los mensajes que indican la interrupción en el servicio de un AS, la información podría separarse de la siguiente manera.

  $$|\text{PROT} , \text{TYPE} , \text{ASN} , \text{ORG} , \text{CTRY} , - , \text{MI} , \text{URL} |$$

  En esta trama se identifican dos campos estáticos, estos son PROT (*Protocol*) y TYPE. El primero corresponde al protocolo de encaminamiento sobre el que trata la información a continuación, que en este caso es BGP; el segundo se refiere al tipo de evento producido, que en este caso es *Outage*.

Por otra parte, el siguiente valor que se encuentra en este tipo de mensajes es de tipo numérico. Este número corresponde con el número único que identifica al AS que ocasionó el problema, este valor queda identificado en la trama como ASN.

El siguiente campo es ORG (*Organization*), cuyo valor no siempre se encuentra en el mensaje debido a que Cisco en ocasiones no puede obtener su valor. Su función es identificar la organización a cargo del AS.

De manera similar al campo anterior, se detalla el campo CTRY (*Country*), que puede no encontrarse en el mensaje por el mismo motivo. Este campo sirve para identificar el país asociado a la organización y viene dado generalmente en el formato alpha-2 del estándar 3166-1 [11].

El mensaje continúa con un guión seguido del campo MI (*More Information*), que indica más información acerca del evento OT detectado. Este campo contendrá datos como el número de prefijos afectados por la interrupción del nodo BGP.

Por último, el mensaje terminará con el campo URL que dirigirá al sitio *web* de BGPMon donde se podrán obtener más detalles del evento, con información similar a la representada en la figura 4.2 anterior.

- **Formato de mensajes Hijack:** para los mensajes que informan sobre un posible secuestro de prefijos se utiliza el siguiente formato.

  | PROT , TYPE , INJ , ORG , CTRY , − , CAU , ORG , CTRY , URL |

  De igual manera que en el formato del mensaje de tipo *Outage*, los dos primeros campos de esta trama son estáticos, con la diferencia de que el campo TYPE tomará el valor de HJ, indicando que la información que precede a estos campos contiene datos sobre un posible secuestro en BGP.

  Esta trama contiene dos secciones importantes que se encuentran separadas por un guion. La primera contiene datos acerca del AS perjudicado, indicando los prefijos secuestrados, el ASN que los contiene, la organización y el país de referencia. La segunda detalla la misma información pero para el AS causante del secuestro.

Con la definición del formato de estos mensajes se busca construir el histórico de datos del que se sustentará BgpRS. La API de Twitter proporciona herramientas para que los datos publicados en su plataforma sean obtenidos de manera automática. En este caso, al haberse elegido Python [27] como lenguaje principal de la aplicación BgpRS, el acceso a dicha API se realizó a través de la librería Tweepy [12].

4.1.4. Otras fuentes de información

Para analizar la repercusión de ciertas situaciones internacionales sobre BGP se tomó como ejemplo la actual guerra entre Rusia y Ucrania. Uno de los retos con los que fue necesario enfrentarse fue la búsqueda de un conjunto de datos más antiguo con el fin de poder comparar la situación actual con la anterior a la guerra.

Intentando recabar datos de mayor antigüedad primeramente se investigó si BGPStream proporcionaba algún medio de solicitud del histórico de detecciones a través de su API, pero el resultado fue negativo. Aunque Cisco Crosswork posee multitud de servicios sobre el campo de las redes mediante los cuales se puede obtener información de BGP, el acceso a ellos no está disponible de manera gratuita, siendo éste uno de los requerimientos buscados para el desarrollo de este TFM.

Siguiendo esta línea, se procedió a buscar información en plataformas conocidas en el ámbito *Machine Learning* y similares por contener grandes conjuntos de datos como *Kaggle* o *DrivenData*. Sin embargo, en ninguna de estas plataformas se encontraron eventos BGP clasificados según el criterio necesario para BgpRS. Durante esta búsqueda, se encontró un proyecto sobre el análisis de datos BGP que trabajaba junto con los servicios de Cisco y utilizaba eventos que BGPMon identificaba como alertas con el fin de realizar un estudio analítico sobre BGPStream. De esta forma, tras examinar el repositorio de Arakadakis Konstantinos [1], se encontró un conjunto de datos en formato `JSON` válido para ser utilizado, ya que databa del año 2018 y podría utilizarse como medio de comparación con la situación actual en BGP.

Por razones prácticas y al ser éste el primero de los archivos estáticos que se poseía, se decidió establecer la distribución de claves y valores que utilizaba como estándar para la aplicación. Esto implica que si se busca alimentar con nuevos datos al recomendador de políticas BGP de una forma diferente al uso de la API de Twitter, se deberá realizar a través de un archivo `JSON` que siga la distribución que se detalla a continuación.

Por una parte, los diferentes campos que se admiten en este archivo y que forman parte de la cabecera común de cada fragmento `JSON` vienen detallados en la tabla 4.1. Cabe destacar que existen campos en los que se especifican ciertas restricciones que si no se respetan producirá que los nuevos datos no aporten valor a las funcionalidades de visualización o recomendación.

Clave	Definición
event_type(**String**)	Especifica el tipo de evento que se detalla en los siguientes campos del fragmento `JSON`. Este campo puede tomar los valores: *BGP Leak*, *Possible Hijack* u *Outage*.
starttime_day(**Int**)	Especifica el día de comienzo del evento. Su valor no puede ser nulo, ya que servirá para posicionar el evento en algún momento de tiempo.
starttime_month(**Int**)	Especifica el mes de comienzo del evento. Su valor no puede ser nulo, ya que servirá para posicionar el evento en algún momento de tiempo.
starttime_year(**Int**)	Especifica el año de comienzo del evento. Su valor no puede ser nulo, ya que servirá para posicionar el evento en algún momento de tiempo.
starttime_time(**String**)	Especifica la hora, minutos y segundos de comienzo del evento. Su valor no puede ser nulo, ya que servirá para posicionar el evento en un momento único de tiempo (UTC).
endtime_day(**Int**/**None**)	Especifica el día de finalización del evento.
endtime_month(**Int**/**None**)	Especifica el mes de finalización del evento.
endtime_year(**Int**/**None**)	Especifica el año de finalización del evento.
endtime_time(**String**/**None**)	Especifica la hora, minutos y segundos de finalización del evento.
moredetail(**String**/**None**)	Especifica la *web* de BGPMon donde se puede obtener más información del evento.

Claves con restricciones	
outage asn(**Int/{None}**)	Especifica el ASN del AS que tuvo la interrupción. Este campo se especifica para los eventos de tipo *Outage* y puede ser nulo si el campo ***country*** es especificado.
country(**String/{None}**)	Especifica el país asociado al AS que originó el evento. El campo puede ser nulo si los campos ***outage asn*** **o** ***detected asn*** son especificados.
detected asn(**Int/{None}**)	Especifica el ASN del AS detectado y no esperado para tipos de evento *Hijack o Leak*. Este campo puede ser nulo, siempre y cuando el campo ***country*** sea especificado.

Tabla 4.1: Especificación de campos (cabecera) para alimentación estática de la aplicación de recomendación

Por otra parte, los campos que se admiten para detallar cada tipo de evento se presentan en la tabla 4.2. En este aspecto, cabe destacar que no todos los campos son necesarios en cada fragmento `JSON`, ya que algunos de estos campos son funcionales o no, dependiendo del tipo de evento.

Clave	**Definición**
expected asn(**Int/None**)	Especifica el ASN del AS esperado para tipos de evento *Hijack o Leak*.
detected_aspath(**list[Int]/[]**)	Especifica el conjunto de ASN referentes a los sistemas autónomos que propagaron la actualización del evento *Hijack o Leak*. Este campo puede ser un array vacío.
leaked_to(**Int/None**)	Especifica el ASN del AS del que se sustrajo el prefijo. Este campo se especifica para eventos de tipo *Leak*.
affected_prefixes(**Int**)	Especifica el número de prefijos a los que afectó la interrupción de servicio. Este campo se utiliza para especificar eventos de tipo *Outage*.
expected_prefix(**String/None**)	Especifica el ASN del AS esperado y no detectado para tipos de evento *Hijack*.
detected_prefix(**String/None**)	Especifica el prefijo involucrado en el tipo de evento *Hijack*.
leaked_prefix(**String/None**)	Especifica el prefijo involucrado en el tipo de evento *Leak*.
peers(**Int**)	Especifica el número de vecinos a los que está conectado el AS que produjo el evento, y por tanto propagaron la actualización BGP. Este campo es normalmente utilizado en eventos de tipo *Hijack o Leak*.

Tabla 4.2: Especificación de campos (cuerpo) para alimentación estática de la aplicación de recomendación

Por último, se proporcionan ejemplos sobre cómo construir estos tipos de archivos `JSON`. Como resultado, estos archivos deberán consistir en una lista de diccionarios o fragmentos `JSON` con cada uno de los eventos que se quieran añadir.

- **Formato de alimentación estática para eventos *Outage*:**


```
[{
    "starttime_day": 11,
    "affected_prefixes": 17,
    "endtime_time": null,
    "event_type": "Outage",
    "starttime_year": 2018,
    "outage asn": 263360,
    "moredetail": "https://bgpstream.com/event/145717",
    "endtime_year": null,
    "endtime_month": null,
    "starttime_month": 8,
    "country": null,
    "endtime_day": null,
    "starttime_time": "08:20:00"
},]
```


- **Formato de alimentación estática para eventos *Hijack*:**


```
[{
    "starttime_day": 11,
    "expected asn": 60781,
    "endtime_time": null,
    "event_type": "Possible Hijack",
    "detected_aspath": [],
    "country": null,
    "moredetail": "https://bgpstream.com/event/145711",
    "starttime_year": 2018,
    "endtime_year": null,
    "peers": 196,
    "endtime_month": null,
    "starttime_month": 8,
    "detected_prefix": "136.144.16.0/24",
    "expected_prefix": "136.144.16.0/22",
    "endtime_day": null,
    "detected asn": 39855,
    "starttime_time": "06:21:46"
},]
```


- **Formato de alimentación estática para eventos *BGP Leak*:** Cisco es capaz de identificar fugas de prefijos con sus implementaciones de BGPMon y BGPStream. Dicha compaña únicamente publica esta información a través de su página *web* Cisco BGPMon[2].

  Esto significa que, al menos hasta la fecha de entrega de esta memoria en el año 2022, Cisco no ha realizado ninguna publicación sobre este tipo de eventos en Twitter. Por este motivo, la aplicación BgpRS no trata los eventos de tipo *BGP Leak* con el fin de dar uniformidad a las recomendaciones proporcionadas por la aplicación.

[2]Sitio *web* de BGPMon: `https://bgpstream.crosswork.cisco.com/`

No obstante, se proporciona un medio para suministrar este tipo eventos, dejando su tratamiento como posible trabajo futuro.


```
[{
    "starttime_day": 11,
    "expected asn": 132123,
    "leaked_to": 9498,
    "endtime_time": null,
    "event_type": "BGP Leak",
    "detected_aspath": [],
    "country": null,
    "moredetail": "https://bgpstream.com/event/145696",
    "starttime_year": 2018,
    "endtime_year": null,
    "peers": 68,
    "endtime_month": null,
    "starttime_month": 8,
    "endtime_day": null,
    "detected asn": 58601,
    "leaked_prefix": "103.70.229.0/24",
    "starttime_time": "03:39:10"
},]
```


4.2. Almacenamiento de la información

De la misma forma que se buscaba obtener información, se necesitaba encontrar la manera de almacenarla. Para cumplir con este aspecto, se contemplaron varias opciones. En primer lugar, se contempló la posibilidad de construir una Base de Datos mediante el uso de tecnologías como SQL o MongoDB. Sin embargo, la mutación de los datos y la necesidad de tratarlos para crear un conjunto que alimentara al recomendador, junto con la definición incompleta del alcance de la aplicación, descartó la necesidad de mantener una Base de Datos en el momento actual y propulsó las medidas de almacenamiento que se detallan a lo largo de esta sección.

4.2.1. Google API

Google es una de las plataformas más utilizadas que gracias a los servicios que proporciona a través de su API permite el desarrollo de multitud de aplicaciones. La evolución del *Big Data* y *Machine Learning* ha dado lugar a la existencia de herramientas como la detección de imágenes, servicios de traducción, escritura de texto mediante voz u otros. Sin embargo, BgpRS no necesita de herramientas tan sofisticadas, haciendo uso únicamente del servicio conocido como *Google Drive*.

En el comienzo de la obtención de datos se poseían tan solo 3200 eventos BGP recogidos a través de la API de Twitter, siendo una de las razones por las que se decidió no utilizar una Base de Datos. Sin embargo, no se tenía conocimiento sobre cuánto podría escalar el archivo resultante del filtrado de mensajes, cuyo detalle podrá observarse en el capítulo 6, ni cuánta información estaría disponible para su almacenamiento.

Además, este archivo podía contener información redundante debido a que aún no se podían advertir las posibles necesidades de la aplicación, lo que provocaba problemas para deducir el tamaño del archivo a generar.

La UCM (Universidad Complutense de Madrid) proporciona una cuenta de Google a cada estudiante inscrito en alguno de sus múltiples planes de estudio, brindando la posibilidad de almacenar una gran cantidad de datos en la plataforma de Google Drive. Esta consecución de razones invitaron al uso de esta plataforma como medio de almacenamiento para los archivos que se generasen.

Este servicio se encuentra directamente conectado con la aplicación, por lo que cada actualización de datos que se realice a través de la API de Twitter será automáticamente almacenada en el directorio de Google Drive que sea especificado, dotando a BgpRS de escalabilidad al almacenar la información en la nube y no de manera local.

4.3. Aproximaciones y problemas

Este trabajo tiene el objetivo de realizar un análisis de los eventos BGP con el fin de brindar recomendaciones de configuración. Para hacer esto posible es necesario obtener un gran volumen de datos consistente, para lo cual, como se ha visto a lo largo de la sección 4.1, se decidió utilizar dos fuentes de información. Por un lado, se encuentra BGPStream, que mediante su sitio *web* publica contenido diario acerca de los diferentes eventos identificados. Sin embargo, la información contenida en la página *web* no se mantiene de manera persistente, por lo que no brindaba solución final para la obtención de datos. Por otra parte, se encuentra la API de Twitter, que proporciona los medios necesarios para obtener la información publicada en su plataforma. En este caso, se necesitaba obtener las publicaciones (*tweets*) realizadas por el usuario Cisco BGPStream (*@bgpstream*). Este usuario, que funciona como *bot* fue creado en junio de 2015, por lo que hoy en día posee más de 65000 mensajes sobre los diferentes eventos BGP que detectó desde su implementación (Figura 4.6).

Figura 4.6: Publicaciones de Cisco Systems en Twitter, *fuente:* `https://twitter.com/bgpstream`

La obtención de estos mensajes aporta gran valor para cumplir los objetivos de este trabajo. Con este propósito se decidió utilizar la API de Twitter. El acceso a las prestaciones de esta API es de carácter gratuito en términos generales, pero dada la evolución de la plataforma, la gran cantidad de datos que posee y la importancia de estos, han hecho que Twitter contemple diferentes niveles de acceso, restringiendo ciertas funcionalidades a los niveles más bajos.

La obtención del máximo número *tweets* publicados por el usuario *@bgpstream* se convirtió en la primera tarea fundamental para iniciar la implementación de BgpRS. El acceso a la API de Twitter en su forma más básica, *Essential*, se realizó de manera sencilla, ya que solo se necesitaba un usuario en Twitter para poder acceder a la misma. Sin embargo, el nivel de acceso esencial solo proporcionaba la posibilidad de acceder a un número muy reducido de *tweets*, los 100 *tweets* más recientes en los últimos 7 días, lo cual era insuficiente. Además, no se poseía del tiempo necesario para almacenar la información de manera semanal hasta consolidar un gran volumen de datos.

Por esta razón, se procedió a solicitar acceso a la versión *Academic Research*, aunque no fue tan sencillo, necesitando más de una semana en comunicación con Twitter hasta poder dar comienzo al recabado de información. Esta versión proporcionaba acceso a los 3200 *tweets* más recientes, lo cual sería reforzado de manera diaria mediante la adición de las nuevas publicaciones realizadas por BGPStream, albergando a fecha de entrega de este TFM más de 5000 *tweets* sobre BGP.

Si bien es cierto que el volumen de datos final puede parecer relativamente pequeño, cabe destacar que el objetivo principal de este TFM es proporcionar un medio de recomendación de configuración, por lo que se podría alimentar a la aplicación con datos sintéticos para demostrar su viabilidad. Por otra parte, en cuanto al análisis de datos se refiere, se pretende dar visibilidad al impacto que tienen factores como la actual guerra entre Rusia y Ucrania sobre BGP. Para ello, mediante los datos de 2022 obtenidos con Twitter y los datos de 2018 conservados por Konstantinos [1], se puede analizar comparativamente la actividad de estos países en BGP entre ambos años.

Capítulo 5

Aplicación de recomendación BgpRS

En este capítulo se procederá a explicar la funcionalidad de recomendación de la aplicación BgpRS. Para ello se describirán las posibilidades de configuración que brinda el protocolo BGP con el fin de comprender los resultados que se verán en el capítulo 8.

5.1. Formas de configuración BGP

Internet es una red compleja con multitud de enlaces redundantes que, como se ha visto, requiere de una gran variedad de políticas de encaminamiento diferentes. La configuración manual de cada uno de los nodos en una red pequeña como es una LAN (*Local Area Network*) se puede considerar como una ocupación altamente costosa. Esto se debe a que es realmente sencillo cometer cualquier tipo de error, sobre todo si se realiza la asignación de direcciones y rutas manualmente.

Las dificultades que tiene configurar manualmente una red pequeña se ven aún más pronunciadas cuando se trata de una gran red como es Internet. Por ello, existen implementaciones *software* que proporcionan facilidades para la configuración de redes. En este aspecto, si se busca obtener una *suite* de herramientas que facilite esta labor, se puede optar por opciones de pago proporcionadas por distribuidores como Cisco o Juniper, o en su defecto por opciones gratuitas *open-source* como son BIRD, ExaBGP, Quagga o FRRouting.

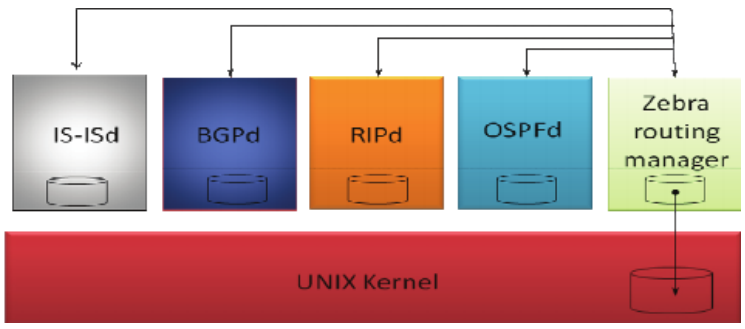


Figura 5.1: Kernel de Unix y Quagga, *fuente:* `https://www.researchgate.net/figure/Quagga-software-architecture_fig1_266970288`

Quagga es un conjunto de herramientas *software* distribuido bajo licencia GPL (*GNU General Public License*) que permite la configuración de aspectos de encaminamiento sobre IP. De esta manera, en dicha herramienta se pueden encontrar implementaciones como OSPFv2, OSPFv3, RIP v1 y v2, RIPng y BGP-4, siendo esta última el contenido principal del presente TFM. Desde el punto de vista de su arquitectura, Quagga funciona como un demonio central (Zebra) que sirve como capa de abstracción al kernel de Unix, por lo que como se observa en la figura 5.1, Quagga en realidad no sustituye el encaminamiento de red realizado por el kernel del sistema operativo subyacente, sino que solo modifica las tablas generadas por el mismo.

Todos estos aspectos hacen que Quagga no sea la mejor de las opciones para realizar una configuración de red en un ámbito profesional. Sin embargo, como los comandos y herramientas que proporciona son transportables a los CLI(*Command-Line Interface*) de Cisco y Juniper, supone un entorno de pruebas ideal para este TFM.

Quagga proporciona los medios necesarios para poder establecer políticas de encaminamiento en BGP. Estas políticas son configuradas principalmente a través de diferentes comandos de filtrado de rutas. Algunos de los comandos más utilizados se detallarán a lo largo de esta sección, ilustrando cómo se definen y se asignan a través de la propia *Shell* VTY (vtysh, *Virtual TeletYpe interface*) de Quagga.

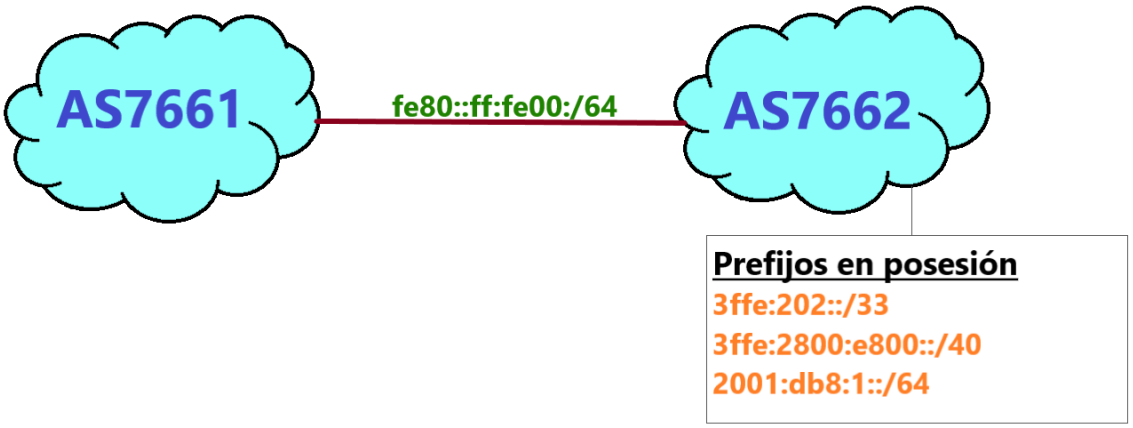


Figura 5.2: Ejemplo de topología de red BGP

5.1.1. Mecanismos de filtrado

Seguidamente se detallan algunos de los mecanismos de filtrado aplicables en Quagga, FFRouting o similares y que serán utilizados por BgpRS.

- `prefix-list`: este elemento permite establecer un filtrado mediante prefijos, pudiendo indicar el permiso o denegación del aprendizaje de rutas que provenga de un vecino particular.

  La topología de la figura 5.2 permite ejemplificar un filtrado de rutas dominado por la política de `prefix-list`. Por ejemplo, el administrador encargado de gestionar el AS7662 puede valorar necesario restringir el tráfico dirigido hacia uno de los prefijos del AS. Para realizar esto, el administrador deberá establecer un filtrado de rutas hacia ese prefijo, pudiendo utilizar para ello la siguiente concatenación de comandos:

```
router bgp 7662
neighbor fe80::ff:fe00:1f0 remote-as 7661
neighbor fe80::ff:fe00:1f0 interface eth0
!
 address-family ipv6
 network 3ffe:202::/33
 network 3ffe:2800:e800::/40
 network 2001:db8:1::/64
 neighbor fe80::ff:fe00:1f0 activate
 neighbor fe80::ff:fe00:1f0 prefix-list peer-out out
 exit-address-family
!
ipv6 prefix-list peer-out deny 3ffe:2800::/32 le 64
ipv6 prefix-list peer-out permit 3ffe::/16 ge 20 le 64
ipv6 prefix-list peer-out permit 2001::/16 le 64
```

En esta porción vtysh se identifican tres secciones divididas por el carácter '!'[1]. La primera de ellas contiene la configuración relativa al encaminamiento donde se define la dirección IPv6 (`fe80::ff:fe00:1f0`) del AS7661. El comando que utiliza esta dirección permite que el AS7662 pueda intercambiar información de encaminamiento con dicho vecino, siendo esta una de las configuraciones básicas y necesarias para el protocolo BGP.

En el tercer fragmento vtysh, se puede observar que el AS7662 mediante la política de `prefix-list`, permite el tráfico hacia dos de los prefijos que posee y deniega el tráfico hacia el prefijo `3ffe:2800::/32`. Por último, en la segunda sección se observa cómo aplicar este comando, asignando al vecino AS7661 las restricciones definidas mediante `prefix-list`.

- `filter-list`: este mecanismo permite utilizar el número único que identifica a cada sistema autónomo (ASN) para filtrar cierta información.

```
router bgp 7661
neighbor fe80::ff:fe00:2f0 remote-as 7662
neighbor fe80::ff:fe00:2f0 interface eth0
!
 address-family ipv6
 neighbor fe80::ff:fe00:2f0 activate
 neighbor fe80::ff:fe00:2f0 filter-list peer-in in
 exit-address-family
!
ip as-path access-list peer-in permit ^7662$
```

Si se sigue la misma topología que la mostrada en la figura 5.2, se puede utilizar la política `filter-list` para permitir el tránsito de tráfico en AS7661 solo cuando el destinatario sea AS7662. Esto se realiza en el fragmento anterior mediante el comando `ip as-path access-list`, donde se utiliza la expresión regular `^7662$` para definir la restricción. Por último, en el segundo fragmento, se observa cómo se asigna esta restricción al vecino del AS7662.

[1] Carácter reservado para indicar el inicio de un comentario dentro del intérprete vtysh.

- `route-map`: esta política de encaminamiento hace uso de la multitud de atributos utilizados en BGP que han sido descritos en la sección 3.2.2, permitiendo la modificación de sus valores y la definición de diferentes filtros de ruta.

  En la siguiente secuencia de comandos, en el último fragmento, se define cómo establecer filtros mediante el comando *route-map*. En este caso, se establecen dos filtros nombrados como `internal-rtm` y `7662-only`. Estos filtros son sencillos y únicamente utilizan la sección `match` del comando para establecer opciones de encaminamiento sobre el AS7662, permitiendo solo el tráfico de salida desde los prefijos que posee y limitando el tráfico de entrada que no haya sido originado vía IGP.

  Este comando contiene opcionalmente una sección `set` que permite actuar sobre atributos como por ejemplo `LOCAL_PREF` o `MED` (*Multi Exit Discriminator*). La aplicación BgpRS se servirá de esto para proporcionar las recomendaciones de configuración.

```
router bgp 7662
neighbor fe80::ff:fe00:1f0 remote-as 7661
neighbor fe80::ff:fe00:1f0 interface eth0
!
 network 3ffe:202::/33
 network 3ffe:2800:e800::/40
 network 2001:db8:1::/64
 neighbor fe80::ff:fe00:1f0 route-map internal-rtm out
 neighbor fe80::ff:fe00:1f0 route-map 7662-only in
!
access-list internal permit 3ffe:202::/33
access-list internal permit 3ffe:2800:e800::/40
access-list internal permit 2001:db8:1::/64
access-list internal deny any
!
route-map internal-rtm permit 10
 match ip address internal
route-map 7662-only permit 10
 match origin igp
```

Como ya se ha mencionado, el intercambio de información de encaminamiento entre cada uno de los *routers* BGP se realiza a través del envío y recepción de diferentes anuncios. De esta manera, cada uno de los encaminadores construye una tabla local de rutas que le permite dirigir el tráfico hacia los diferentes destinos de la red. Por último, como se ha visto en los mecanismos de filtrado, también existen expresiones regulares para aplicar filtros de encaminamiento bajo un criterio particular. En la tabla 5.1 se detallan algunos ejemplos de expresiones regulares que se pueden aplicar sobre el atributo `AS_PATH`. Mediante estas expresiones, los comandos anteriores y a gracias a la posibilidad de asignar los filtros sobre el tráfico de entrada o de salida, cada AS tiene la posibilidad de establecer la ruta más adecuada según sus propios intereses.

BgpRS establece un factor de fiabilidad[2] para cada AS mediante los datos recopilados. Con esta fiabilidad, los comandos disponibles y las expresiones regulares informa sobre cómo actuar respecto al tráfico de entrada o de salida.

[2]Reputación del AS obtenida a través de los datos recopilados en función de la frecuencia en la que un AS se ha visto envuelto en eventos de tipo *Hijack* u *Outage*.

Expresión Regular	Significado
.*	Cualquier carácter.
.+	Uno o más caracteres.
^$	Rutas locales a este AS
^100	Cualquier AS_PATH que comience por 100, p.e. 1001
200$	Cualquier AS_PATH que termine en 200, p.e. 11200
_100$	Rutas con origen en AS100
^100_	Rutas recibidas desde AS100
_100_	Rutas recibidas a través de AS100
_200_100_	Rutas recibidas vía AS100 y luego AS200
_(100_)+	Detectar varios AS100 consecutivos

Tabla 5.1: Ejemplos de expresiones regulares aplicables al filtrado de rutas BGP

5.1.2. Acciones configurables sobre los anuncios de entrada

Los anuncios que recibe un encaminador BGP sirven para establecer la ruta que deberá seguir el tráfico de datos a su salida. Es decir, establecer un filtrado de rutas sobre los anuncios que se reciben repercute directamente sobre el camino que seguirán los datos hacia su destino.

Cada nodo de una red BGP recibe múltiples anuncios de los vecinos a los que esté conectado. En una red tan grande como es Internet es realmente común que se lleguen a recibir diferentes anuncios con información sobre como alcanzar un mismo destino. Esta redundancia es uno de los motivos por los que BGP es configurable. Un ejemplo de esto se presenta en la figura 5.3.

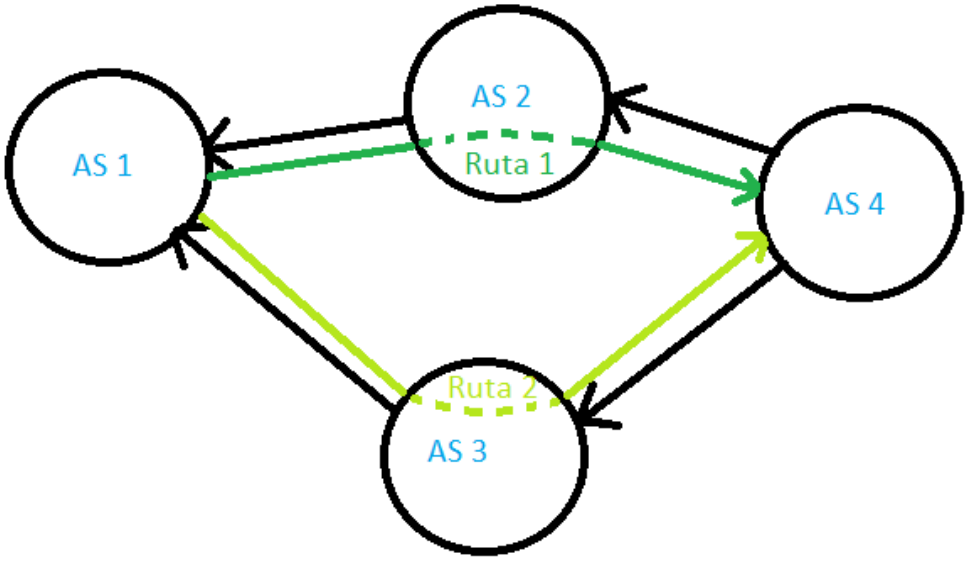


Figura 5.3: Enlaces BGP hacia un mismo destino

En caso de que un nodo BGP contenga más de una ruta hacia un mismo destino, se sigue la toma de decisiones contemplada en la sección 3.3. En esta sección, se ha podido observar que la preferencia local ante una ruta es el primero de los aspectos que se contempla durante el proceso de decisión de BGP.

La modificación del atributo `LOCAL_PREF` es uno de los medios de configuración más utilizado. Una de las acciones de configuración más común sobre este atributo es establecer el orden de preferencia para las diferentes rutas cuando hay varias hacia un mismo destino. De este modo, se puede elegir un enlace prioritario sin descartar los demás y, si por cualquier motivo un enlace deja de funcionar, se puede utilizar el segundo enlace con más prioridad local.

En la figura 5.4 se representan cuatro sistemas conectados de tal forma que AS1 puede alcanzar a AS4 mediante dos rutas diferentes. Siguiendo esta imagen y el filtro `route-map` detallado a continuación se muestra un ejemplo de cómo fijar el valor del atributo `LOCAL_PREF` para establecer la ruta 2 como prioritaria.

```
router bgp 1
    bgp router-id 0.0.0.1
    neighbor 10.0.12.2 remote-as 2
    neighbor 10.0.13.3 remote-as 3
    neighbor 10.0.13.3 route-map Ruta2Preferente in
    neighbor 10.0.12.2 route-map Ruta3Inferior in
!
ip as-path access-list DeAS3 permit ^3_$
ip as-path access-list DeAS2 permit ^2_$
!
route-map Ruta2Preferente permit 10
    match as-path DeAS3
    set local-preference 120
route-map Ruta3Inferior permit 10
    match as-path DeAS2
    set local-preference 80
```

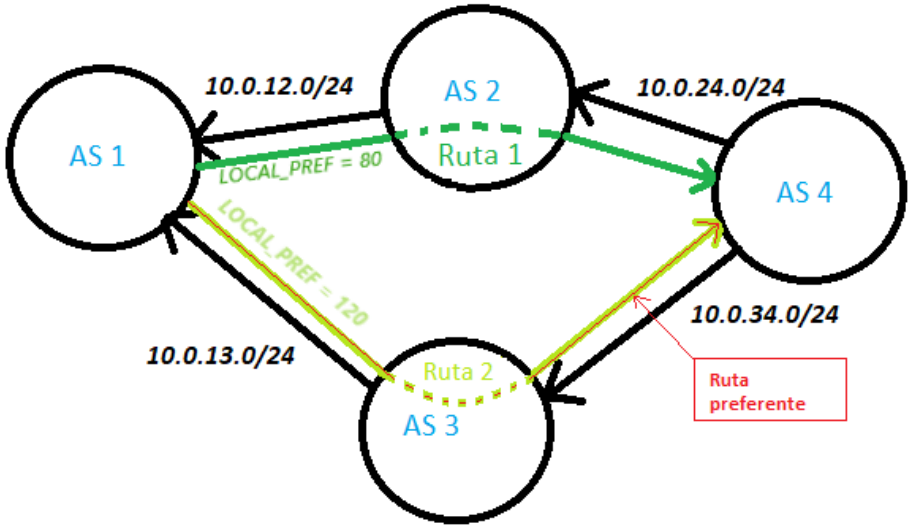


Figura 5.4: Preferencia de ruta mediante el atributo `LOCAL_PREF`

Como se verá más adelante, este es uno de los aspectos sobre los que trabajará la funcionalidad de recomendación de BgpRS, facilitando al administrador comandos similares a este para denegar o restringir el envío de datos hacia los AS a los que esté conectado y que presenten comportamientos problemáticos.

5.1.3. Acciones configurables sobre los anuncios de salida

BGP también permite la configuración sobre los anuncios de salida. Como el sentido de los datos navega en dirección opuesta a los anuncios, actuar sobre los anuncios salientes afecta sobre el tráfico recibido.

Limitar el tráfico hacia alguna de las subredes contenidas en un AS es un posible requerimiento. Un AS puede no estar preparado para convertirse en un sistema de tránsito para la red, por lo que para evitar que esto suceda, el administrador debe decidir a qué vecinos aplica un filtro para restringir el anuncio de rutas que lo utilizan como tránsito.

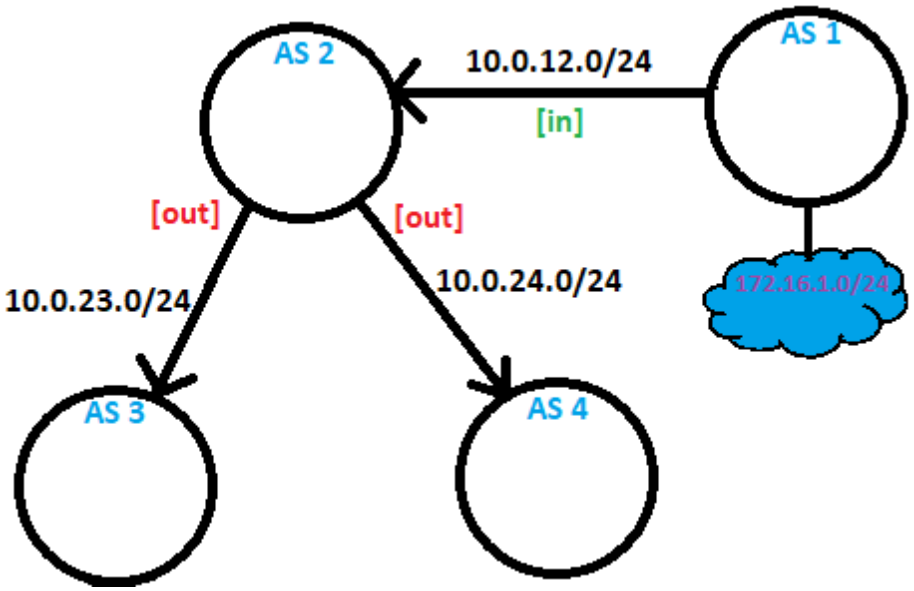


Figura 5.5: Ejemplo de configuración en los anuncios de salida

En la figura 5.5 se observa un posible escenario BGP con anuncios de entrada y de salida donde se pueden aplicar los aspectos descritos anteriormente. En esta imagen, se aprecia cómo el AS2 recibe un anuncio de prefijos proveniente de AS1. En una situación estándar AS2 propagaría el prefijo recibido a los vecinos a los que está conectado, en este caso AS3 y AS4. Sin embargo, AS2 puede necesitar limitar o denegar la propagación de dicho prefijo, optando por un filtrado por prefijos (`prefix_list`) o un filtrado por `AS_PATH`.

En el caso de que el AS2 decida restringir el flujo de datos hacia el prefijo `172.16.1.0/24` este puede tomar medidas como intentar realizar un `AS_PATH` *prepend*. Esta medida consiste en incorporar en el `AS_PATH` de sus anuncios su propio ASN varias veces, tratando de disminuir las probabilidades de que otros vecinos le elijan para el envío de datos. Sin embargo, en la tabla 5.1 se ha visto que es posible detectar cuando un AS está repitiendo su ASN.

En consecuencia, si verdaderamente AS2 no tiene la capacidad de servir de tránsito hacia AS1, deberá restringir por completo el anuncio del prefijo recibido por AS1, manteniendo una configuración similar a la que se detalla a continuación que, en este caso, utiliza el mecanismo de filtrado por `AS_PATH`.

```
router bgp 2
    bgp router-id 0.0.0.2
    neighbor 10.0.12.1 remote-as 1
```

```
    neighbor 10.0.23.3 remote-as 3
    neighbor 10.0.24.4 remote-as 4
    neighbor 10.0.23.3 filter-list trAS1 out
    neighbor 10.0.24.4 filter-list trAS1 out
!
ip as-path access-list trAS1 deny _1$
ip as-path access-list trAS1 permit .*
```

En la sucesión de comandos anterior se puede ver cómo a través de `ip as-path access-list trAS1 deny _1$` se indica que los filtros se aplicarán sobre los anuncios recibidos desde el AS1. Por otra parte, mediante la ejecución de los comandos `neighbor 10.0.23.3 filter-list trAS1 out` y `neighbor 10.0.24.4 filter-list trAS1 out`, se aplica el filtro sobre el tráfico de salida, evitando que AS2 se convierta en sistema de tránsito para AS1. Por último, cabe destacar que es necesario permitir el anuncio de otros prefijos que no provengan de AS1, esto se realiza mediante el comando `ip as-path access-list trAS1 permit .*`.

5.2. Sistema de Recomendación

El sistema de recomendación de BgpRS se sirve de cada uno de los aspectos estudiados a lo largo de este capítulo. El objetivo principal de este trabajo es proporcionar información útil a los administradores de los AS. Por esta razón, se tomó la decisión de facilitar las recomendaciones como opciones de configuración, utilizando para ello comandos presentes en FRR (*FRRouting*), una variante actualizada de Quagga.

5.2.1. Sistemas autónomos con interrupciones de servicio recurrentes

A lo largo de esta memoria, se ha podido observar la repercusión que tiene una caída de servicio. Una interrupción de servicio puntual no supone un gran problema, ya que BGP incorpora los medios suficientes para la recuperación ante este tipo de eventos. Además, aunque pudiesen aparecer problemas inherentes al mismo protocolo como *Wedgies*, al tratarse de una interrupción inusual, podría implementarse una solución de forma manual.

Sin embargo, esta situación cambia cuando las interrupciones de servicio se producen de manera recurrente. En relación con esto, es posible que los administradores encargados de diferentes AS estén interesados en evitar el tránsito del tráfico a través del sistema que produce el problema.

La aplicación BgpRS identifica los sistemas que originan este tipo de problemas y obtiene tendencias según su histórico de datos. De tal forma que si un sistema ha fallado más de tres veces en un periodo de treinta días, la aplicación lo clasifica como un AS poco fiable y recomienda la disminución del valor de preferencia local sobre las rutas que lo atraviesen.

Los sistemas poco fiables que realmente suponen un problema son aquellos que sirven a modo de tránsito. La caída de estos nodos repercute negativamente sobre todo el tráfico que los atraviesa. Aislar por completo un sistema que produce este tipo de problemas, independientemente de su función en la red, no tiene sentido. La interrupción de servicio en un sistema *stub* únicamente incidiría en el transporte de datos hacía el destino que contiene, viéndose perjudicado únicamente el AS en estado *Outage*.

Por otra parte, tampoco aportaría nada positivo dejar de aceptar los anuncios originados por estos AS, ya que solo provocaría la pérdida de una gran cantidad de información de encaminamiento y la posible inaccesibilidad a ciertos destinos de la red.

Por este motivo, la aplicación BgpRS se centra únicamente en aquellos sistemas autónomos que son de tránsito para proporcionar una posible configuración, haciendo uso de filtros y expresiones regulares similares a los explicados en la sección 5.1.

Este aspecto de BgpRS se ve reflejado en la figura 5.6, donde destaca la zona inferior de la imagen que incluye la recomendación que proporciona. Con el fin de analizar esta recomendación en profundidad se exponen a continuación los comandos retornados en mayor detalle.

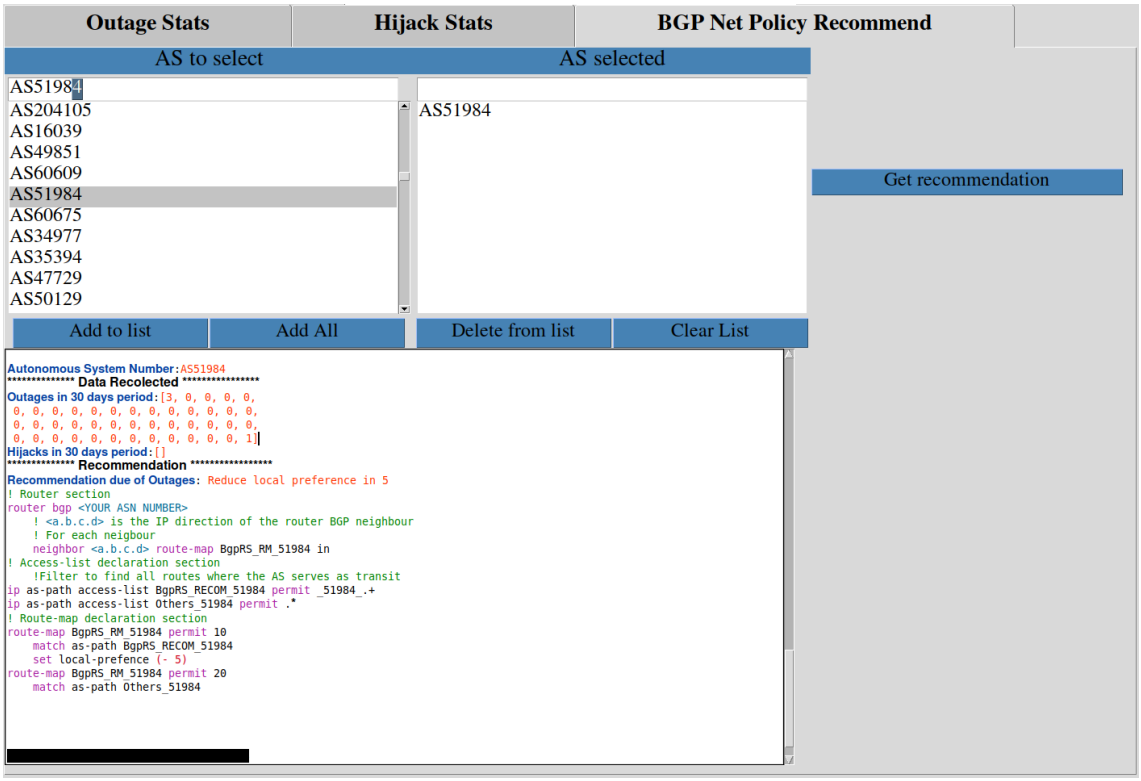


Figura 5.6: Recomendación para sistemas autónomos con interrupciones de servicio recurrentes

```
Autonomous System Number:AS51984
********** Data Recolected **********
Outages in 30 days period:[3, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1]
Hijacks in 30 days period:[]
********** Recommendation **********
Recommendation due of Outages: Reduce local preference in 5
! Router section
router bgp <YOUR ASN NUMBER>
    ! <a.b.c.d> is the IP direction of the router BGP neighbour
```

```
    ! For each neigbour
    neighbor <a.b.c.d> route-map BgpRS_RM_51984 in
! Access-list declaration section
    !Filter to find all routes where the AS serves as transit
ip as-path access-list BgpRS_RECOM_51984 permit _51984_.+
ip as-path access-list Others_51984 permit .*
! Route-map declaration section
route-map BgpRS_RM_51984 permit 10
    match as-path BgpRS_RECOM_51984
    set local-prefence (- 5)
route-map BgpRS_RM_51984 permit 20
    match as-path Others_51984
```

Para mostrar un ejemplo de lo que retorna la aplicación se ha elegido consultar el AS51984. Esto se debe a que este AS ha tenido interrupciones de servicio suficientes para que BgpRS lo clasifique como un posible AS defectuoso. De forma preventiva, para evitar que la recomendación proporcionada sobre el AS en cuestión fuese demasiado drástica, se decidió que el valor de `LOCAL_PREF` para las rutas que atravesaran dicho AS disminuyese de manera acumulativa en función de un factor determinado.

Cuando se establecen las rutas BGP, el atributo `LOCAL_PREF` toma un valor por defecto de 100. Este atributo, toma valores en un rango de 32 bits y no acepta valores negativos, admitiendo números enteros en un rango de 0 a 4294967295.

El factor para disminuir la preferencia local no debía ser demasiado alto, ya que si lo fuese se podría alcanzar rápidamente el límite inferior de este rango. Por todo ello, se decidió separar el histórico de eventos de los AS en ventanas de 30 días, de manera que el valor de este atributo se redujese en 5 cada vez que se produjesen más de dos fallos de servicio en cada una de estas ventanas. Como se observa, este es el primero de los datos que se proporcionan al usuario.

Después de que BgpRS calcule el factor de disminución, la aplicación proporciona acciones en forma de comandos. Estos comandos, sirven para aplicar las reglas que se consideran adecuadas sobre el AS que produjo el problema. En primer lugar, cabe destacar que la aplicación no tiene medio de conocer el ASN del AS sobre el que se aplicarán dichas reglas. Tampoco se conoce la dirección IP del AS que produjo el problema. Por consiguiente, la especificación de este ASN y esta IP se proporcionan al usuario en forma de las variables `<YOUR ASN NUMBER>` y `<a.b.c.d>` respectivamente. Esto se representa en la `Router section` de la salida proporcionada.

Por razones ya mencionadas, las rutas sobre las que se aplicará la modificación del atributo serán aquellas en las que el AS que produjo el problema funciona como tránsito. Para ello, se proporciona un filtro para seleccionar las rutas en cuestión. Esto se realiza a través de la expresión regular `_51984_.+`, que filtra el AS_PATH para encontrar en qué rutas el ASN 51984 es tránsito y no tienen su origen en él.

La última acción que se lleva a cabo es asignar el valor del atributo `LOCAL_PREF` a las rutas filtradas. Para conseguir esto, se hace uso de la sección *set* del comando `route-map`. Para el AS51984, la aplicación ha determinado reducir en 5 la preferencia local de todas las rutas en las que el AS sirva de tránsito. De esta forma, si el usuario aplica las recomendaciones proporcionadas a cada uno de sus vecinos, las rutas que contengan el AS51984 como tránsito serán descartadas antes en el proceso de decisión de BGP.

5.2.2. Sistemas autónomos propensos al secuestro de prefijos

Este aspecto de la aplicación funciona de manera similar a la vista en la sección anterior. La diferencia es que en este caso se hace uso de la información referente al secuestro de prefijos. En cuanto a los datos utilizados, es necesario hacer mención de que no todos los datos que se poseen serán útiles para realizar una recomendación.

Para implementar la funcionalidad de análisis sobre los eventos internacionales en BGP (Capitulo 6) se necesita de la información de los AS causantes y de los perjudicados. Sin embargo, el aspecto de recomendación no necesita la información del AS perjudicado, ya que tomar medidas para este no ayudaría a terceros en ningún aspecto. Por este motivo, el recomendador solo hace uso de la información referente a los AS que hayan sido causantes de un secuestro de prefijos.

Un evento de secuestro de prefijos es detectado cuando un AS anuncia de manera inusual el origen de ciertos prefijos. Normalmente, los prefijos que anuncia el causante pertenecen a un segundo AS, dándose así la posibilidad de que exista un mismo prefijo con dos orígenes distintos, provocando un cambio en el encaminamiento de BGP y añadiendo nuevas rutas hacia el mismo destino en las tablas de otros *routers*.

Partiendo de esta base, para realizar el filtrado de rutas, se decidió que solo se debía tener en cuenta aquellas rutas anunciadas donde el AS causante fue el origen. El último ASN que aparece en el atributo `AS_PATH` de las rutas indica el AS que originó el anuncio, por lo que la expresión regular utilizada se basará en este criterio.

El siguiente fragmento de comandos de configuración muestra la salida devuelta por la aplicación de recomendación al realizar una consulta sobre el AS12479. En el caso de que el AS consultado haya anunciado un número abundante de prefijos falsos, la aplicación clasificará al mismo como sistema no fiable.

```
Autonomous System Number:AS12479
************** Data Recolected ****************
Outages in 30 days period:[]
Hijacks in 30 days period:[13, 2, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 2]
************** Recommendation *****************
Recommendation due of Hijacks: Reduce local preference in 10
! Router section
router bgp <YOUR ASN NUMBER>
    ! <a.b.c.d> is the IP direction of the router BGP neighbour
    ! For each neigbour
    neighbor <a.b.c.d> route-map BgpRS_RM_12479 in
! Access-list declaration section
    ! Filter to find all routes where the AS is origin
ip as-path access-list BgpRS_RECOM_12479 permit _12479$
ip as-path access-list Others_12479 permit .*
! Route-map declaration section
route-map BgpRS_RM_12479 permit 10
    match as-path BgpRS_RECOM_12479
```

```
    set local-prefence (- 10)
route-map BgpRS_RM_12479 permit 20
    match as-path Others_12479
```

Como se puede observar, la recomendación proporcionada es muy similar a la dada para los sistemas con interrupciones de servicio. Sin embargo, en este caso la expresión regular para el filtrado de rutas hace referencia a las subredes anunciadas por el AS, por lo que tiene la forma `_12479$`.

Otro punto a destacar es el factor por el que se reduce el atributo `LOCAL_PREF`. Para este aspecto, se consideró que un anuncio de prefijos falso podía tener mayor repercusión en la red y, por lo tanto, que el factor de disminución debía ser mayor que el utilizado para los sistemas con problemas de servicio, asignado un valor de 10 al mismo. Por último, se proporciona la figura 5.7 donde se refleja comportamiento de BgpRS ante este tipo de situaciones.

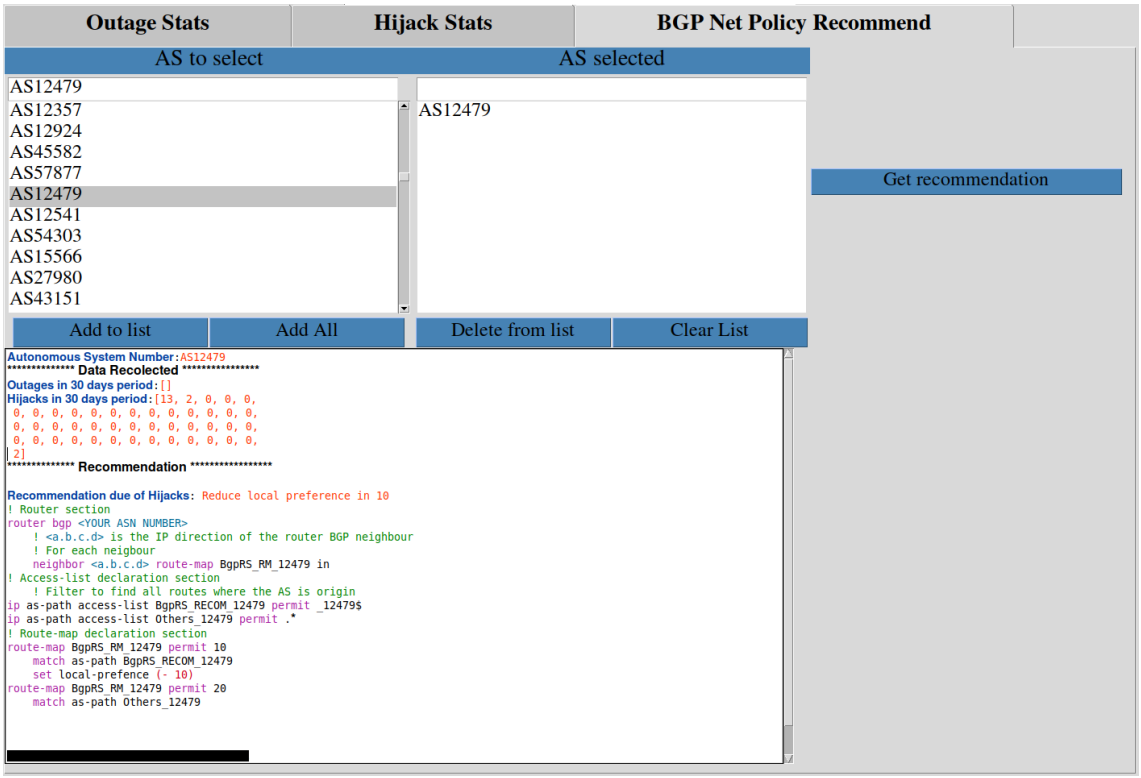


Figura 5.7: Recomendación para sistemas autónomos propensos al secuestro de prefijos

Capítulo 6

Análisis de eventos BGP

En el siguiente capítulo se detallará la funcionalidad analítica de BgpRS. Para ello, se explicarán las tecnologías utilizadas y se aportarán algunos ejemplos ilustrativos de la aplicación.

6.1. Whois

Una vez recogidos los datos sobre los eventos BGP se debía proceder a la limpieza y tratamiento de estos. Para comenzar se debía obtener el país y la organización de los sistemas autónomos envueltos en algún tipo de evento, ya que esta información era necesaria para comprobar el impacto internacional que tienen ciertas cuestiones políticas sobre BGP. En la mayoría de las ocasiones, esta información estaba incluida en los mensajes de Twitter. Sin embargo, como había momentos los que BGPStream no disponía de esta información, podía darse el caso de que no se incluyese. Esta situación producía que los datos disponibles para BgpRS se viesen mermados, afectando negativamente a sus capacidades, razón que propició la búsqueda de una solución.

Existen muchos medios para obtener información acerca los diferentes elementos de Internet. Este es el caso ya mencionado de los *Looking Glasses*, mediante los cuales podrían obtenerse los atributos que se buscaban. Sin embargo, la mayoría de los LG únicamente proporcionan acceso a través de navegadores y sus páginas *web*, lo que dificulta la extracción de estos atributos.

Whois es un protocolo TCP (*Transmision Control Protocol*) definido en 1982 que permite la consulta de cualquier dominio conectado a Internet a través de una base de datos pública. Este protocolo, que es accesible mediante línea de comandos ya sea en el sistema operativo Windows o las diferentes distribuciones de Linux, se caracteriza por poseer una gran gama de servicios que proporcionan información sobre los diferentes dominios de Internet.

La información que proporciona Whois es útil para conocer datos sobre los propietarios de un domino, la empresa responsable de una página web, la disponibilidad de un dominio para su registro o compra, y en el caso de este trabajo, para obtener la identificación de la organización y el país responsable de la administración de un determinado AS.

Registrar un nombre de dominio requiere ciertos requisitos, siendo necesario que las organizaciones, gobiernos, empresas y personas que se den de alta, proporcionen datos de identificación y de contacto, como son: nombre, domicilio, correo electrónico, etc.

La gestión del protocolo Whois es realizada por la corporación ICANN (*Internet Corporation for Assigned Names and Numbers*), una organización sin ánimo de lucro cuyo objetivo es la realización de ciertas tareas encargadas a IANA. En lo que al protocolo Whois se refiere, debe encargarse de implementar medidas para mantener el acceso público y gratuito a sus datos. Por esta razón, cualquier usuario puede usar este protocolo y realizar una búsqueda en su base de datos mediante comandos.

El comando `whois` permite obtener información acerca de los AS de BGP, para ello simplemente se necesita acompañar a dicho comando con el ASN del nodo BGP que se quiera consultar. De esta forma, si se pretende realizar la acción anterior, se deberá usar el comando `whois AS766`, obteniendo la información que se observa en la figura 6.1.

```
% Information related to 'AS766 - AS766'

as-block:       AS766 - AS766
descr:          RIPE NCC ASN block
remarks:        These AS Numbers are assigned to network operators in the RIPE NCC service region.
mnt-by:         RIPE-NCC-HM-MNT
created:        2018-11-22T15:27:08Z
last-modified:  2018-11-22T15:27:08Z
source:         RIPE

% Information related to 'AS766'

% Abuse contact for 'AS766' is 'seguridad@rediris.es'

aut-num:        AS766
as-name:        RedIRIS
org:            ORG-RA6-RIPE
descr:          RedIRIS Autonomous System
descr:          SPAIN
import:         from AS174 accept ANY
```

Figura 6.1: Fragmento de la información obtenida mediante el comando `whois AS766`

En esta imagen se observa que el AS766 pertenece a RedIRIS, la red española encargada de la gestión de recursos informáticos de universidades y centros de investigación. En este caso, mediante una sola consulta se obtiene toda la información necesaria (país y organización). Sin embargo, en múltiples ocasiones para sistemas autónomos de menor rango esto no es así.

También cabe destacar que esta información de contacto no sigue un estándar fijo, por lo que los países pueden estar representados con su nombre completo o con su código alpha-2 (ISO3166 [11]). Por esta razón, durante el proceso de obtención de país fue necesario utilizar una librería que contuviese la correspondencia entre ambas (Tabla 6.1), siendo la librería pycountry [4] la elegida para ello.

Name	Code
Afghanistan	AF
Russian Federation	RU
Ukraine	UA
United Arab Emirates	AE
...	...

Tabla 6.1: Fragmento de correspondencia entre país y código alpha-2

En ocasiones no es posible determinar la nacionalidad de un AS con una única consulta. Un nodo puede ser administrado por más de una persona, y en estas situaciones, `whois` puede retornar más de un país, número de teléfono u otra información de contacto, produciendo cierta incoherencia y dificultades para determinar correctamente el país de un AS.

BgpRS utiliza los países de los AS para dar su funcionalidad de análisis, por lo que la correcta obtención de estos datos era necesaria. Por esta razón, se decidió hacer uso de otro de los servicios de `whois`, la obtención de información sobre organizaciones. En el caso de RedIRIS, como se ha observado en la imagen anterior, se ha obtenido que su organización es ORG-RA6-RIPE, por lo que si se ejecuta el comando `whois ORG-RA6-RIPE` se podrá obtener más información de la misma.

Este servicio también retorna una gran cantidad de información, con la ventaja de que las incongruencias en la obtención del país se ven reducidas. Por este motivo, se decidió utilizar dos consultas, primero una sobre el AS, y posteriormente otra utilizando la organización obtenida, optimizando de esta forma la extracción del país. No obstante, aunque la realización de estas dos consultas permitía deshacerse de gran parte de las incongruencias, podía darse el caso de que estas se siguiesen produciendo.

Para terminar de resolver este problema, se decidió aplicar una versión sencilla de lo que se conoce como TF-IDF (*Term frequency – Inverse Document Frequency*), una medida numérica que indica cuánto de relevante es una palabra en un texto dado, lo que en este caso particular se traduce en obtener el país que más se repite en el texto devuelto por `whois`. De esta forma, con todos los pasos anteriores, se procedió a determinar el país de cada AS.

Por último, cabe destacar que la base de datos de Whois proporciona acceso a la información acerca de la gran mayoría de los AS presentes en Internet. Esto significa que existen ocasiones en las que la obtención del país referente a un ASN no podría efectuarse a través de este comando, afectando negativamente al análisis de información que se pretendía conseguir.

Una vez realizadas todas las tareas anteriores se comenzó a filtrar y construir los datos necesarios para implementar el aspecto analítico de BgpRS. Como ya se poseían datos acerca de los sistemas autónomos que habían generado eventos de tipo *Outage* y *Hijack*, así como de un medio para identificar la procedencia internacional de los mismos, solo faltaba implementar una interfaz sencilla para buscar y visualizar los datos.

6.2. Outages

Los eventos que implican una interrupción de servicio pueden generar problemas de encaminamiento en BGP, produciendo que algunos prefijos de red queden inaccesibles para una parte o la totalidad de los nodos de la red. Estos eventos se producen por fallos físicos del nodo perjudicado o por restablecimientos administrativos que pueden esconder tras de si motivos de carácter político-social, haciendo que conocer el origen o el motivo detrás de una interrupción de servicio sea complicado.

El protocolo BGP no detalla los motivos por los que un conjunto de prefijos dejó de ser inaccesible. La única forma de averiguar la causa de uno de estos eventos es a través de las organizaciones a cargo y solo si estas creen conveniente proporcionar la información. Sin embargo, si se conoce un posible conflicto entre países y se poseen datos en la fecha en

la que se produjeron, mediante BgpRS se puede visualizar el impacto del mismo en BGP.

Para realizar esto, la aplicación pone a disposición del usuario diferentes elementos. Por un lado, la aplicación proporciona un buscador para la selección de los países que el usuario pretenda visualizar. En este aspecto, cabe destacar que puede que no todos los países estén presentes, ya que solo se consideran seleccionables aquellos que tuvieron presencia durante el estudio[1]. Además, como existe la posibilidad de que el comando `whois` no proporcione información acerca de un determinado AS, en el buscador se podrá consultar también el ASN del mismo con el fin de inspeccionarlo.

Por otra parte, BgpRS proporciona un medio para seleccionar el fragmento de tiempo que se pretenda visualizar, obteniendo de esta forma un gráfico con la cantidad de eventos producidos por un determinado país durante el inicio y fin de la fecha seleccionada. Un ejemplo del resultado obtenido se representa en la figura 6.2.

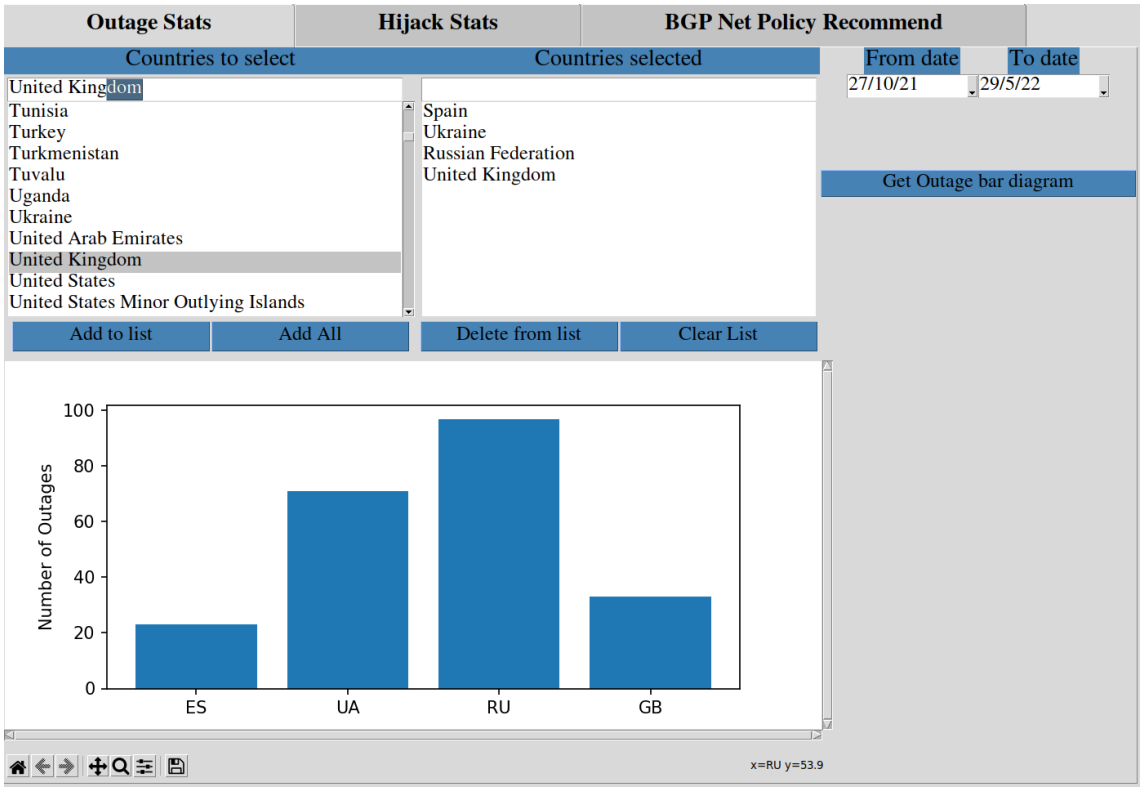


Figura 6.2: Visualización de información *Outage* a través de la interfaz de la aplicación

6.3. Hijacks

Como se ha podido contemplar a lo largo de esta memoria, el protocolo BGP también es susceptible de verse comprometido por fenómenos que impliquen una fuga o secuestro de prefijos. Este tipo de eventos indican un cambio inusual en el encaminamiento BGP, desembocando en un cambio en el flujo de datos y que los mismos puedan ser dirigidos a servidores equivocados.

[1] Aquellos países con sistemas autónomos a su cargo que produjeron algún evento de tipo *Outage* y/o *Hijack*.

Estos factores suponen un riesgo potencial dado que los datos de la red pueden verse comprometidos, siendo un aspecto que cobra importancia añadida desde un punto de vista internacional.

Una de las funciones de BGPmon es identificar los eventos anteriores como *BGP leaks* y *Hijacks*. De estos eventos se decidió tener solo en cuenta los eventos *Hijack*. Esto se debe a que BGPStream solo publica los eventos de tipo *Hijack* y *Outage* a través Twitter, produciendo que los eventos de tipo *Leak* solo sean accesibles a través de su *web* y de su API.

Desde el punto de vista de la interfaz, la aplicación BgpRS aporta características de visualización de datos de manera similar a las detalladas para los sucesos de tipo *Outage*. Sin embargo, presenta algunas diferencias en cuanto al modo y características de la visualización de los datos.

Para cumplir el propósito de la funcionalidad de análisis se consideró necesario descartar ciertos datos. Algunos de los sucesos clasificados como *Hijack* tenían el mismo país como origen y destino, produciendo que las métricas se viesen aumentadas y haciendo que el contenido visualizado no permitiese una comparación entre distintos países.

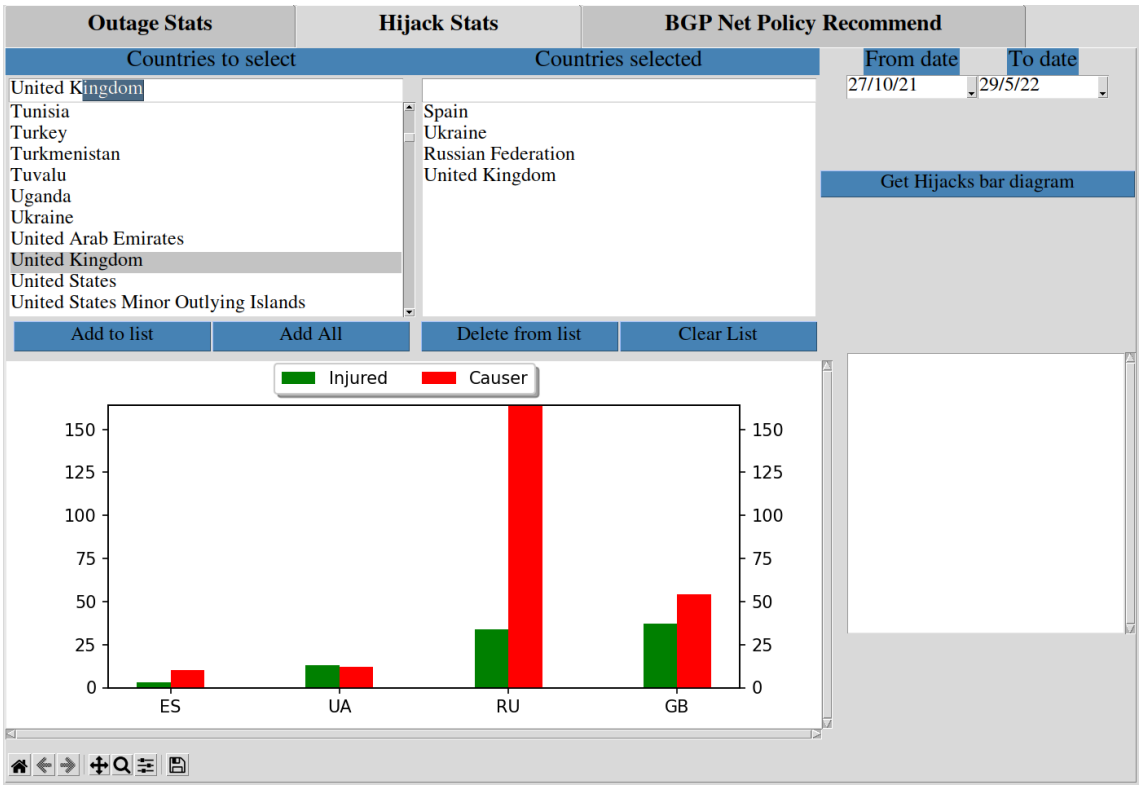


Figura 6.3: Visualización de información *Hijack* a través de la interfaz de la aplicación

La información proporcionada para los sucesos de tipo *Hijack* se ven reflejados en la figura 6.3. Como se puede observar, el gráfico devuelto por BgpRS determina la cantidad de veces que un país ha sido causante y víctima de un secuestro de prefijos durante la ventana de tiempo seleccionada.

Por último, hace falta mencionar que la aplicación también proporciona la posibilidad de obtener más información. En este aspecto, cabe destacar que es condición que el usuario seleccione únicamente dos países para obtener una salida como la reflejada en la figura 6.4.

En la zona resaltada (1) de esta imagen se observa como la aplicación retorna más información relativa a los dos países seleccionados. Esta información tiene el objetivo de proporcionar más datos para una mejor comparación de posibles conflictos entre las dos regiones, mostrando estadísticas sobre la cantidad de veces que cualquiera de los países ha resultado ser causante o perjudicado de un secuestro de prefijos provocado por el otro.

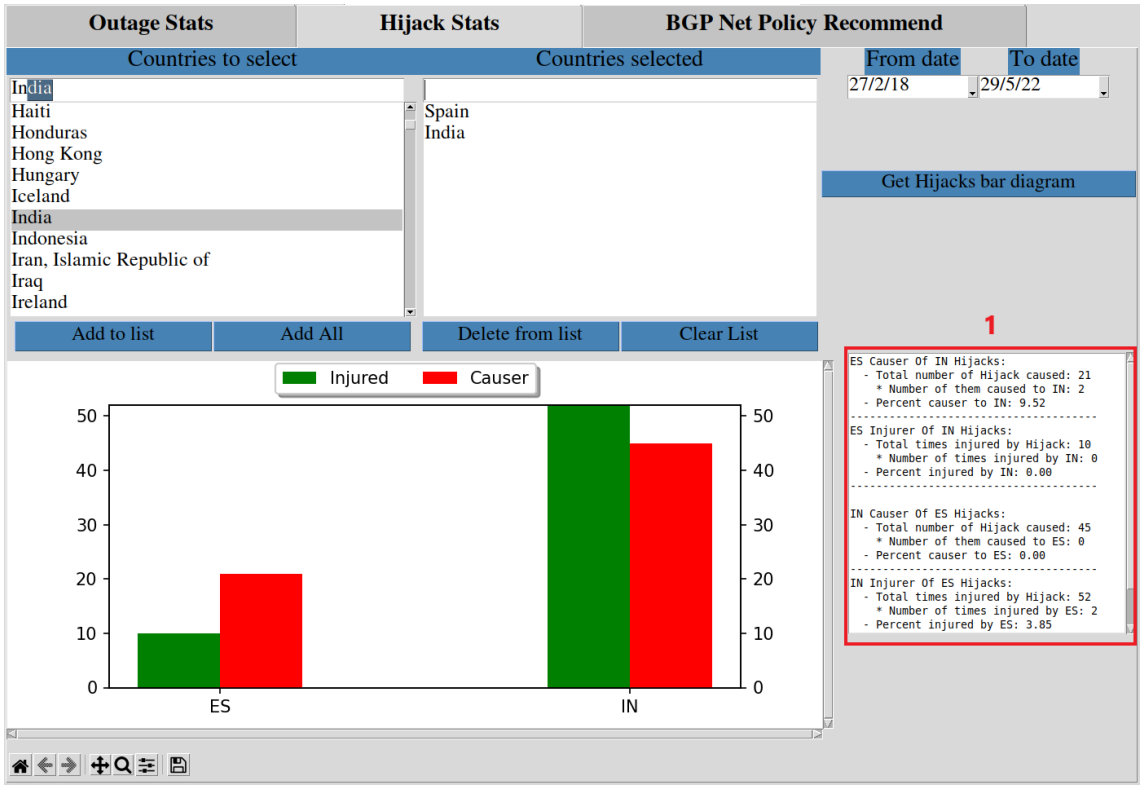


Figura 6.4: Comparación entre dos países a través de la interfaz de la aplicación

Capítulo 7

Puesta en marcha de la aplicación

A lo largo de esta memoria se han explicado las características de la aplicación BgpRS. Por una parte, en el capítulo 4, se han explicado los aspectos relevantes para extraer información de BGP. Además, se ha podido ver que esta información es fundamental para la aplicación y sus funcionalidades vistas en los capítulos 5 y 6. Para hacer todo esto posible, BgpRS utiliza multitud de servicios.

Estos servicios son accesibles a través de diferentes librerías disponibles en Python, por lo que será necesario instalar ciertas dependencias antes de ejecutar BgpRS. En los apéndices de la memoria se proporciona un enlace al código de la aplicación. Dentro de este proyecto se encuentra la carpeta denominada `/dependencies_installation_dir` que contiene el archivo `requirements.txt` con la información de las dependencias necesarias.

Conociendo todo esto y teniendo en cuenta que BgpRS actualmente solo está preparado para funcionar en las distribuciones de Linux, se debe crear un entorno virtual de Python3 para instalar las citadas dependencias. Esto último se puede realizar mediante los pasos enumerados a continuación.

1. **Creación del entorno virtual:**

   ```
   python3 -m venv venv
   ```

2. **Activación del entrono virtual:**

   ```
   source venv/bin/activate
   ```

3. **Instalación de dependencias del proyecto:**

   ```
   pip install -r dependencies_installation_dir/requirements.txt
   ```

De todas las dependencias que se instalarán, podría decirse que las más relevantes son las librerías de Tkinter [8], TweePy [12] y Pydrive [10]. Es importante destacar que estas dos últimas necesitan de permisos y la asignación de credenciales para cumplir con su cometido, siendo necesaria su especificación en el código de la aplicación. Este capítulo se proporciona con el fin de dar a conocer las características a más bajo nivel y de mostrar cómo poner en funcionamiento la aplicación BgpRS.

7.1. Twitter API

Como se ha repetido en múltiples ocasiones, la información obtenida a través de la API de Twitter tiene enorme relevancia para BgpRS. Sin embargo, para que la aplicación utilice las funcionalidades de esta API, es necesario seguir las directrices que se detallan a continuación.

En primer lugar, es necesario puntualizar que BgpRS utiliza funciones de esta API que solo están disponibles con los permisos que otorga el plan *Academic Research* de Twitter. La necesidad de esto se ha podido ver en detalle en la sección 4.3 de esta memoria. Por ello, el primer paso necesario para poner en marcha la aplicación será solicitar acceso al plan mencionado.

Para realizar la solicitud correspondiente, primero se debe tener una cuenta de Twitter para realizar el registro en la plataforma *developer*[1] de esta entidad. Este proceso de solicitud es bastante intuitivo. Sin embargo, como el acceso al plan academico está restringido para ciertos usuarios y se requiere del consentimiento de Twitter, será necesario el intercambio de cierta cantidad de *e-mails*.

Una vez realizado todo esto, será posible crear un proyecto con el fin de dar acceso a los servicios a la aplicación BgpRS. Si todo el proceso se ha realizado correctamente, el nuevo perfil de *Twitter Developer* se asemejará al representado en la figura 7.1.

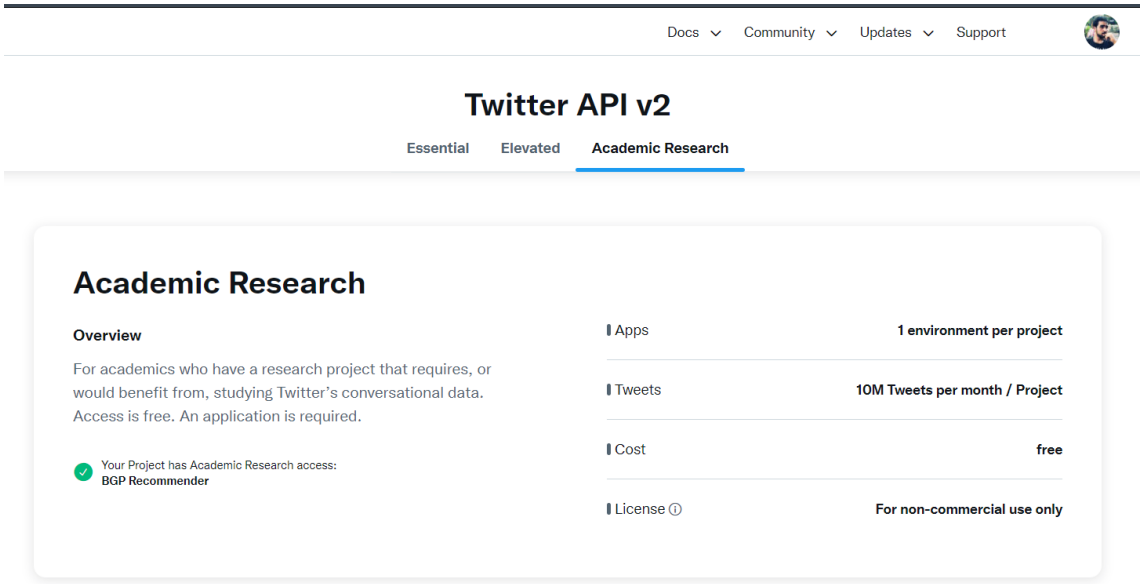


Figura 7.1: Permisos *Academic Research* en Twitter, *fuente:* `https://developer.twitter.com/en/portal/products`

7.1.1. Asignación de credenciales dentro de BgpRS

Una vez obtenido el acceso al plan *Academic Research*, será posible obtener las credenciales necesarias para que TweePy [12] ejecute las solicitudes de BgpRS. El acceso a estas credenciales y su lectura se consigue a través del portal *developer* de Twitter tal y como se muestra en la figura 7.2.

[1]Twitter Developer: `https://developer.twitter.com/en/portal/products`

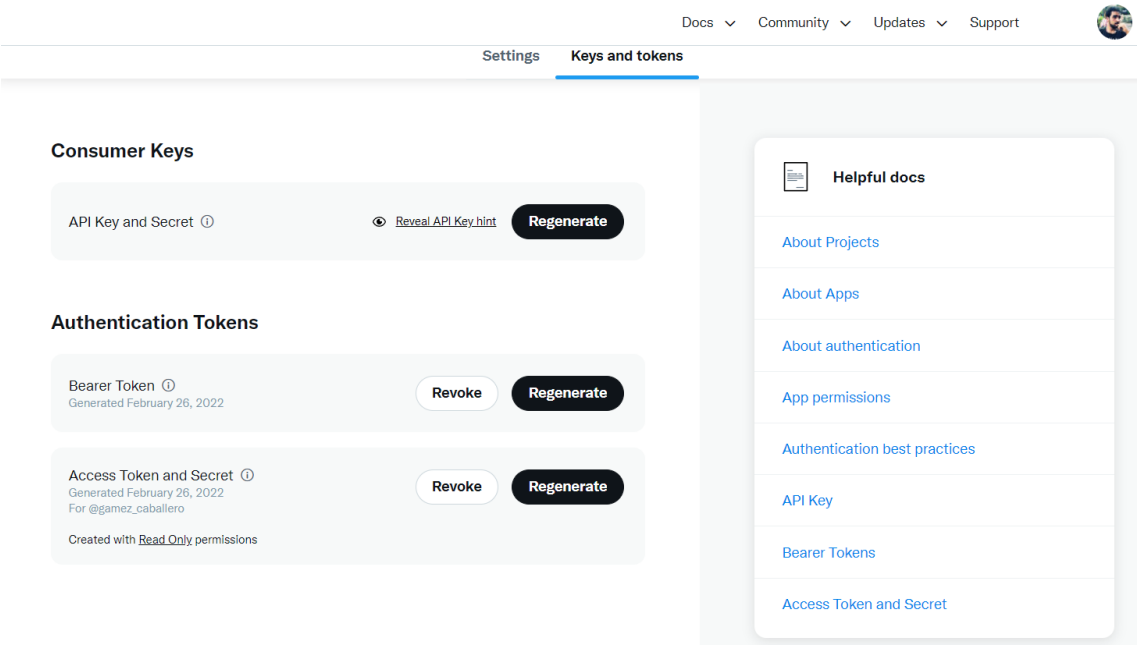


Figura 7.2: Acceso a credenciales de Twitter, *fuente:* ***https://developer.twitter.com/en/portal/projects/***

Con estas credenciales en posesión ya solo se necesita especificar su valor dentro del código de BgpRS. Esto se debe realizar mediante la modificación del archivo situado en la carpeta `/API_classes`, `twitterDevCredentials.py`, donde se deberá asignar valor a cada una de las variables mostradas en la figura 7.3.

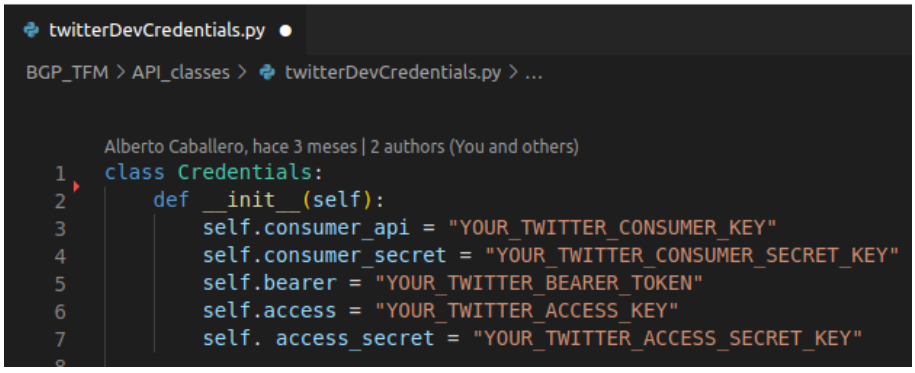


Figura 7.3: Archivo en BgpRS para introducir las credenciales de Twitter

Una vez realizados todos estos pasos, la aplicación será capaz de obtener y actualizar la información de los eventos BGP identificados por Cisco. Sin embargo, para proceder a su almacenamiento, se necesita realizar un proceso similar para la librería de PyDrive [10], el cual se detalla en la siguiente sección.

7.2. Google API

Para ejecutar BgpRS sin errores, es fundamental seguir los pasos que se describen en esta sección. En ella se especifican cuáles son los requerimientos para que la aplicación se

comunique con los servicios de Google Drive. Para obtener las credenciales necesarias, es preciso poseer una cuenta de Gmail y acceder a la consola de *Google Cloud Platform*[2]. A través de esta consola se podrá crear un proyecto para posteriormente activar el servicio correspondiente de Google Drive.

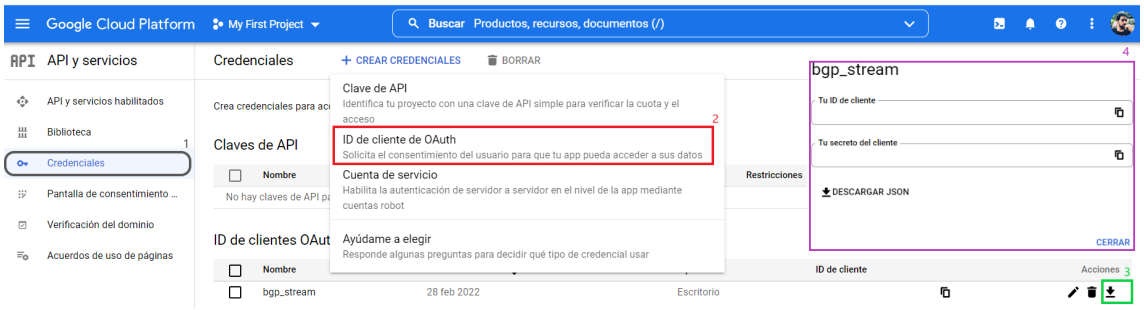
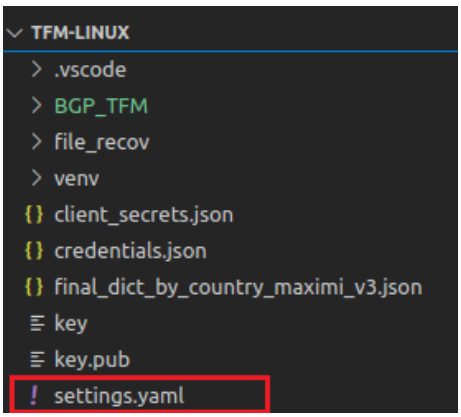


Figura 7.4: Credenciales de Google Drive, *fuente:* `https://console.cloud.google.com/apis/credentials?authuser=1&project=phrasal-clover-312717`

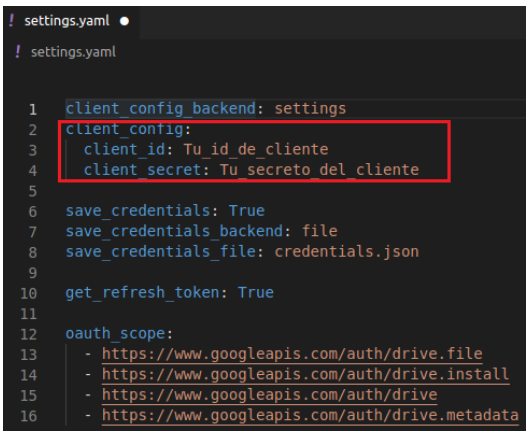
En la figura 7.4 se representa el proceso para crear y obtener las credenciales necesarias para hacer uso de las funcionalidades de PyDrive [10]. En primer lugar, se deberá ingresar en el menú de `API y servicios` y, dentro de este, en el apartado de `Credenciales` para crear unas del tipo `ID de cliente de OAuth`. Para completar el proceso se deberán rellenar los campos solicitados. Si todo se realiza de la manera correcta, se podrá acceder a las credenciales, que en el caso ilustrado son las denominadas como `bgp_stream`.

Siguiendo la misma imagen, si se navega a través del frontal de la consola, se encontrará el botón de descarga resaltado en verde (3). Si presionamos este botón, se desplegará una pestaña como la resaltada en color morado (4) y se podrán obtener las credenciales que necesita BgpRS.

Dentro del proyecto de BgpRs, se deberá crear un archivo de extensión `.yaml`, que deberá denominarse `settings.yaml` y que deberá situarse en la raíz del proyecto (Figura 7.5a). Este archivo, servirá para indicar las credenciales de Google Drive a la aplicación BgpRS, para lo que se deberá cumplimentar la información representada en la figura 7.5b.



(a) Organizacion de directorios



(b) Contenido de `settings.yaml`

Figura 7.5: Archivo de BgpRS para asignar las credenciales de Google Drive

[2]Google Cloud Platform: `https://cloud.google.com/`

En el archivo `settings.yaml` se debe dar valor a los campos `client_secret` y `client_id` con la información de las credenciales de Google Drive[3]. Una vez configurado este archivo, cuando se ejecute la aplicación por primera vez, se abrirá una pestaña en el navegador pidiendo la confirmación de permisos al usuario. Si se aceptan dichos permisos, se generarán automáticamente los archivos `credentials.json` y `client_secrets.json` con la información necesaria para no tener que realizar esta acción en las próximas ejecuciones.

7.2.1. Asignación de identificadores de carpeta dentro de BgpRS

Con los pasos realizados anteriormente, ya solo es necesario crear una carpeta de Drive donde BgpRS almacenará y obtendrá la diferente información. Esta carpeta deberá estar organizada de manera similar a la representada en la figura 7.6. El nombre de estas carpetas no tiene por qué ser idéntico, ya que de estas como se verá más adelante, solo se utilizará su identificador.

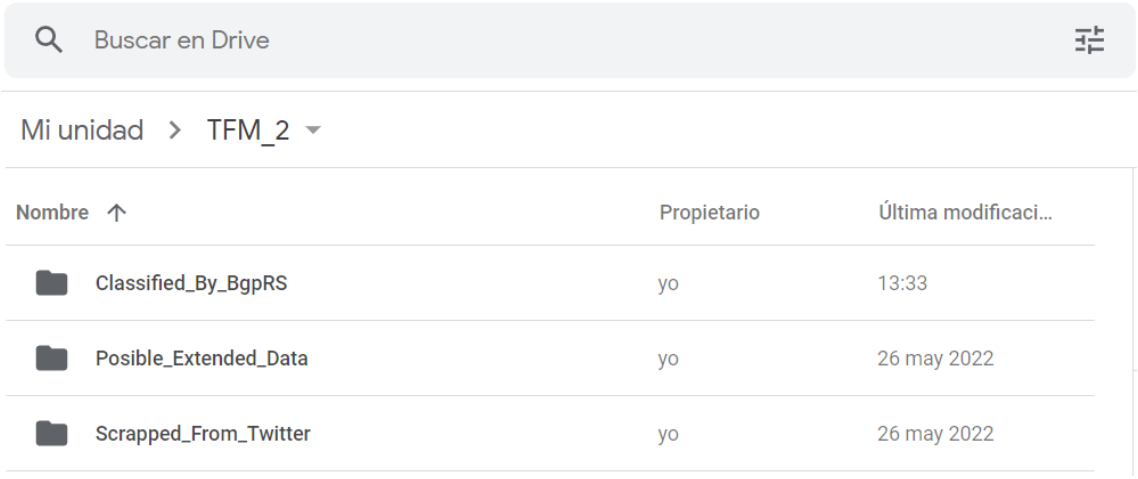


Figura 7.6: Distribución de carpetas en Google Drive, *fuente:* ***https://drive.google.com/drive/u/1/folders***

El contenido de estas carpetas tiene diferente significado para BgpRS. Por un lado, la carpeta `/Scrapped_From_Twitter` contendrá el archivo donde se guardarán y actualizarán los datos obtenidos de Twitter. Este archivo será utilizado por la aplicación para clasificar la información y para generar un archivo analizable que será almacenado en la carpeta `/Classified_By_BgpRS`.

Por último, la aplicación incorpora funcionalidades para aumentar estos datos de manera manual. Con este fin, la aplicación obtendrá la información del archivo almacenado en la carpeta `/Classified_By_BgpRS` y la actualizará con los nuevos datos. Después de realizar todo esto, se generará un nuevo archivo que se almacenará en la carpeta `/Posible_Extended_Data`.

Una vez creadas las tres carpetas se deberán obtener sus identificadores, dado que como se ha mencionado, la aplicación necesita conocer su valor. Esta información se puede extraer desde el mismo navegador, ya que el identificador de cada una de estas carpetas se encuentra en la última parte de la *url*. El proceso de obtención de estos identificadores se representa en la figura 7.7.

[3]Información obtenida en el paso 4 de la imagen 7.4

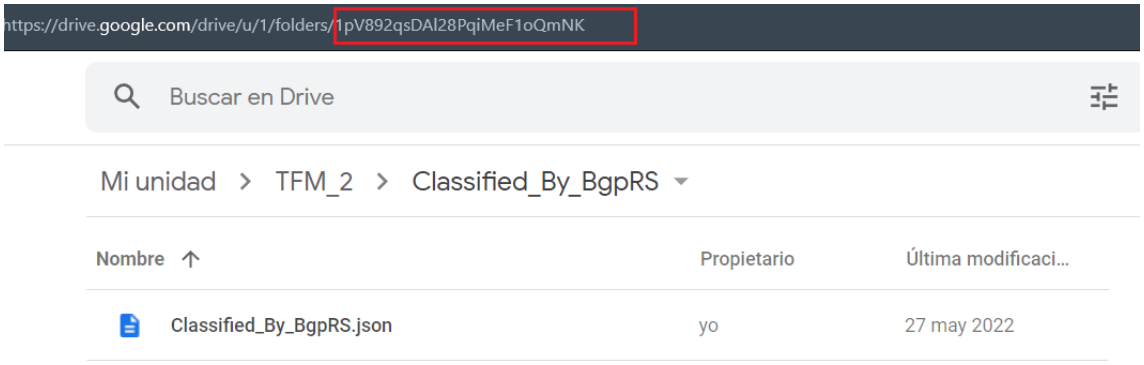


Figura 7.7: Obtención de identificadores de carpeta, *fuente: https://drive.google.com/drive/u/1/folders*

Una vez obtenidos cada uno de los identificadores, solo hace falta indicarlos en la parte correspondiente del código de BgpRS. Esta asignación debe realizarse en el archivo `Data_classifier_class.py` que se encuentra situado dentro de la carpeta `/API_classes`. El valor de estos identificadores de carpeta se deberá indicar sobre las variables que se resaltan en rojo en la figura 7.8.

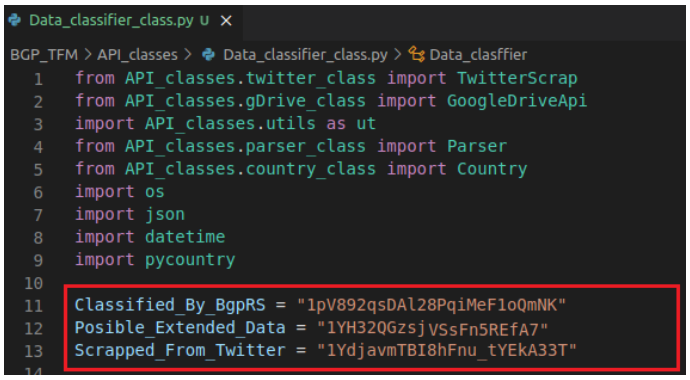


Figura 7.8: Asignación de identificadores de carpeta en BgpRS

Con todos estos pasos, BgpRS estará capacitada para almacenar y obtener la información necesaria para su funcionamiento, proporcionando la posibilidad de obtener, actualizar y analizar los datos al usuario a través su interfaz gráfica.

7.3. Interfaz gráfica

En los capítulos 6 y 5 se han descrito las funcionalidades principales de BgpRS. Sin embargo, el usuario puede realizar otras funcionalidades adicionales a través de la interfaz gráfica.

En el presente capítulo se han explicado cada uno de los aspectos necesarios para que la aplicación pueda utilizar las funcionalidades de obtención y almacenamiento de la información. Para que el usuario pueda ejecutar estas funcionalidades de manera sencilla, se decidió implementar los accesos rápidos de la interfaz que se observan en la figura 7.9. La descripción de cada uno de estos botones se realizará en las siguientes secciones.

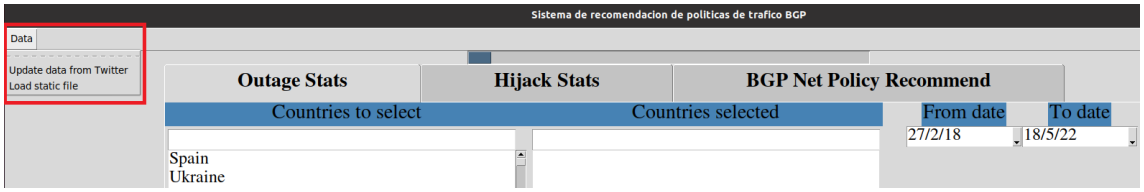


Figura 7.9: Funcionalidades adicionales de BgpRS

7.3.1. Actualización de datos vía Twitter

Una de las posibilidades que se le proporcionan al usuario es la capacidad de obtener y actualizar los datos a través de la API de Twitter. Este proceso está automatizado de tal forma que, si el usuario no posee datos en las carpetas de su Google Drive, se obtendrán los 3200 *tweets* más recientes del usuario *@bgpstream* y BgpRS se encargará de clasificarlos, generando los archivos correspondientes en las carpetas `/Scrapped_From_Twitter` y `/Classified_By_BgpRS` mencionadas anteriormente.

Por el contrario, si el usuario ya posee datos en estas carpetas, la aplicación se encargará de actualizarlos. En este aspecto, cabe destacar que la aplicación puede tardar un tiempo, ya que depende de la cantidad de datos que se obtengan. Además, en ocasiones el comando `whois` no puede obtener el país de cada uno de los eventos BGP y se decidió que la aplicación reclasificara la información que ya contuviese con el fin de mejorar la calidad de los datos, lo que supone un incremento en el tiempo necesario para realizar el proceso de actualización.

Por esta razón, para que el usuario no estuviese esperando infinitamente, se decidió implementar un hilo de proceso independiente para realizar este tipo de acciones, por lo que el usuario podrá examinar los datos ya clasificados mientras espera. El proceso de ejecución sobre el botón `Update data from Twitter` se ve reflejado en la figura 7.10.

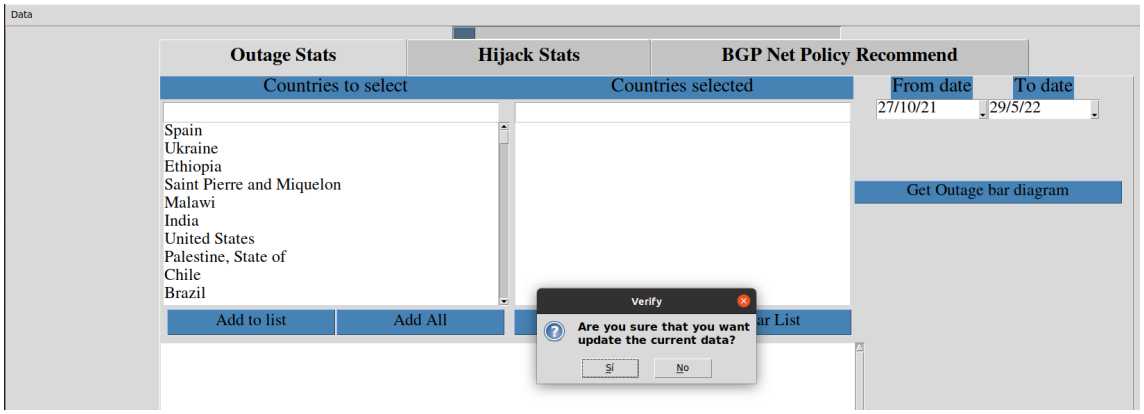


Figura 7.10: Acción del proceso de actualizado de datos

Como se puede observar, en primer lugar se le preguntará al usuario si realmente quiere actualizar los datos. Después de que el usuario de confirmación, el proceso de actualizado comenzará y una vez finalice notificará al mismo a través del mensaje reflejado en la figura 7.11.

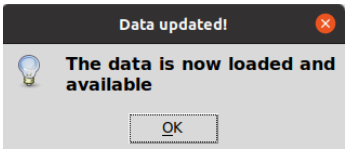


Figura 7.11: Mensaje de notificación sobre el actualizado de datos

7.3.2. Datos sintéticos

A veces es necesario contar con datos generados de manera artificial para poner a prueba la aplicación y entender su uso y utilidad. En la sección 4.1.4 se ha explicado detalladamente cómo construir un archivo de datos para la alimentación estática de la aplicación. El botón de la interfaz, `Load static file`, permite al usuario seleccionar un archivo local, que mantenga dicha sintaxis, para incorporar los datos a BgpRS.

La ejecución de esta funcionalidad se realiza a través de un hilo independiente de la misma manera que para el actualizado de datos de Twitter. Después de que el usuario pulse sobre el botón `Load static file`, la aplicación desplegará un navegador de archivos. Mediante este navegador, el usuario podrá seleccionar el archivo `Json` que desee cargar. Este comportamiento se muestra en la figura 7.12.

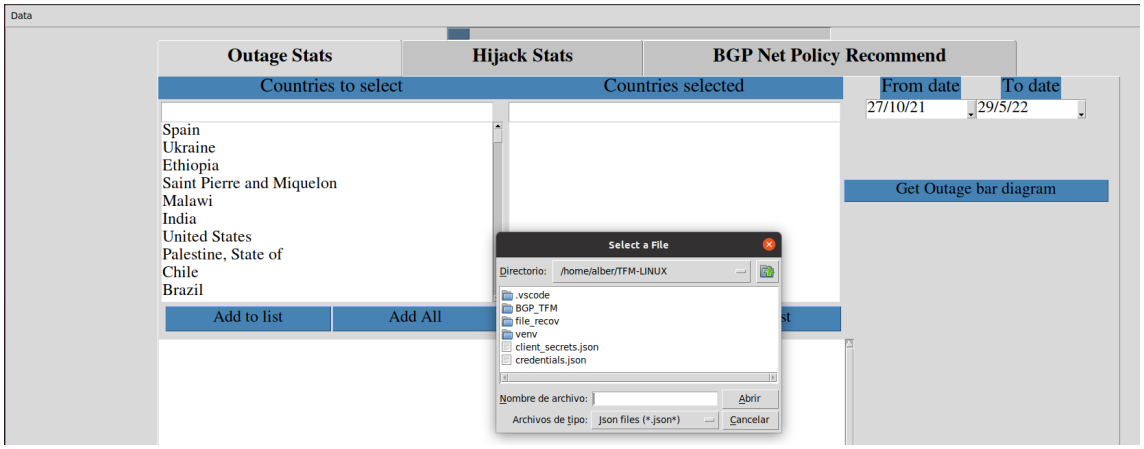


Figura 7.12: Selección de archivo estático para aumentar los datos disponibles

Después de seleccionar el archivo, se solicitará confirmación al usuario mediante el mensaje que se representa en la figura 7.13. Tras la finalización de la clasificación de la nueva información, se generará un archivo adicional que será almacenado en la carpeta `/Posible_Extended_Data`. La finalización de la asignación de contenido en este archivo será notificada a través de un mensaje como el mostrado en la figura 7.11. Este archivo será utilizado en adelante por la aplicación, por lo que si se desea contemplar otros datos deberá eliminarse o modificarse en Google Drive.

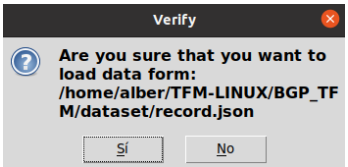


Figura 7.13: Confirmación de cargado de datos estáticos

7.3.3. Limitaciones para el usuario

Para finalizar con este capítulo hace falta mencionar que la aplicación sufre de ciertas limitaciones con respecto a la obtención de datos. Por motivos de simplicidad y para evitar posibles problemas a nivel *software*, se decidió que la aplicación solo pudiese mantener de manera concurrente dos hilos de ejecución. Por una parte, un hilo principal para conservar la aplicación en estado de ejecución, y por otra, otro hilo independiente para realizar alguna de las acciones de modificación de datos.

En consecuencia, si el usuario ya ha emprendido alguna de las acciones disponibles para modificar el conjunto de datos, deberá esperar a que la carga de datos finalice para realizar una nueva acción del mismo estilo. Por esta razón, si el usuario intenta realizar dos acciones de modificación de datos de manera simultánea, la aplicación se encargará de advertirle mediante el mensaje representado en la figura 7.14.

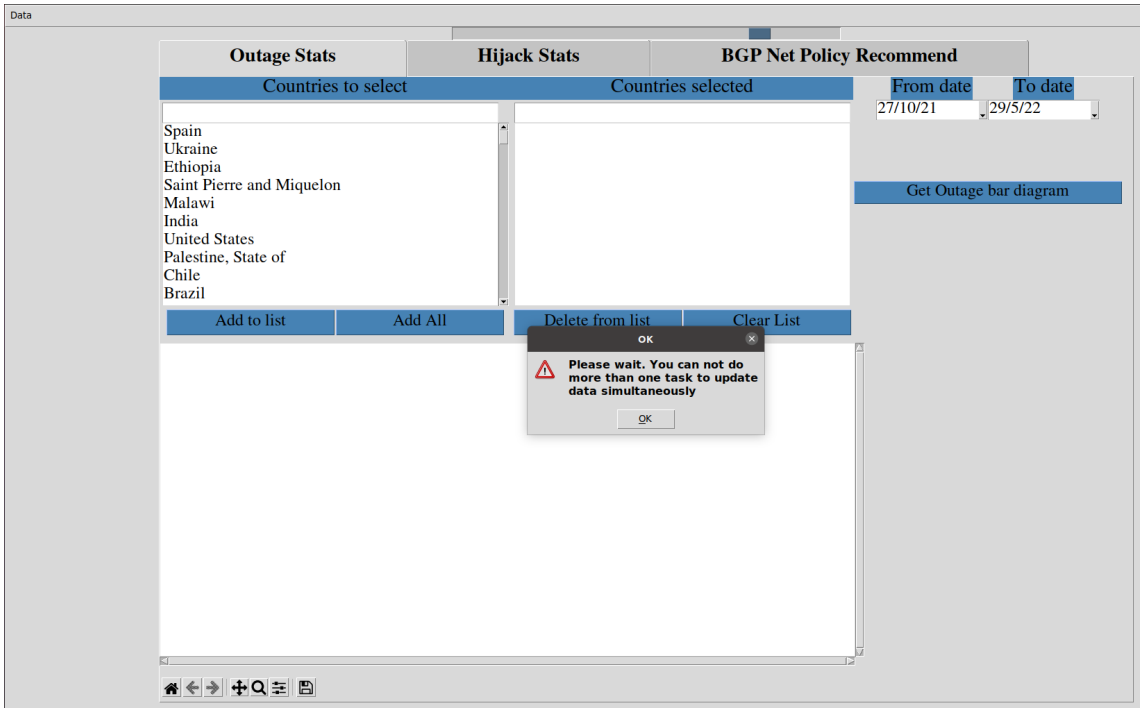


Figura 7.14: Limitaciones del usuario en el actualizado de datos

Capítulo 8

Casos de uso de BgpRS

En este capítulo se proporcionarán algunos ejemplos de uso de la aplicación BgpRS. Para ello se explicará el significado de los datos visualizables a través de BgpRS, justificando el número de eventos *Outage* y *Hijack* clasificados.

Por otra parte, se procederá a exponer las recomendaciones obtenibles a través de BgpRS, representando las consecuencias de tomar las medidas de recomendación que se proporcionan ejecutándolas en un entorno de pruebas.

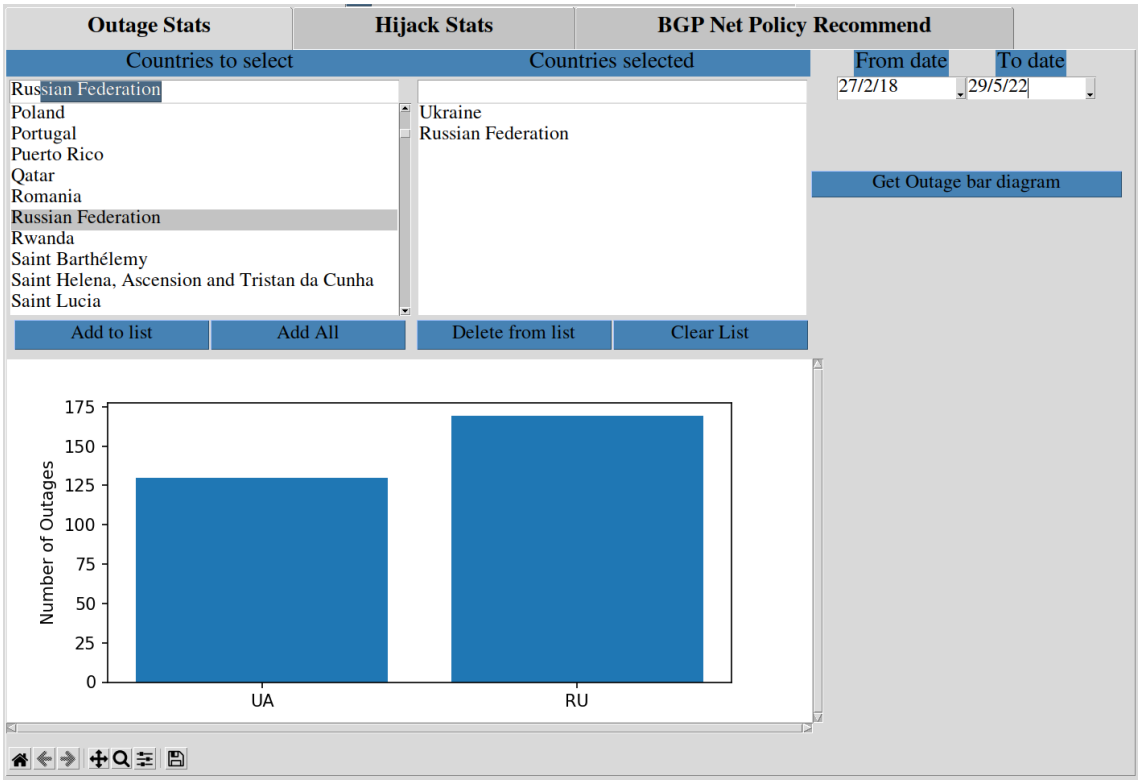
8.1. Análisis de repercusión internacional BGP

Este trabajo ofrece la posibilidad de contemplar la repercusión de acontecimientos históricos de carácter geopolítico sobre el ámbito BGP. La presente guerra entre Ucrania y Rusia puede servir de ejemplo para comprobar este aspecto.

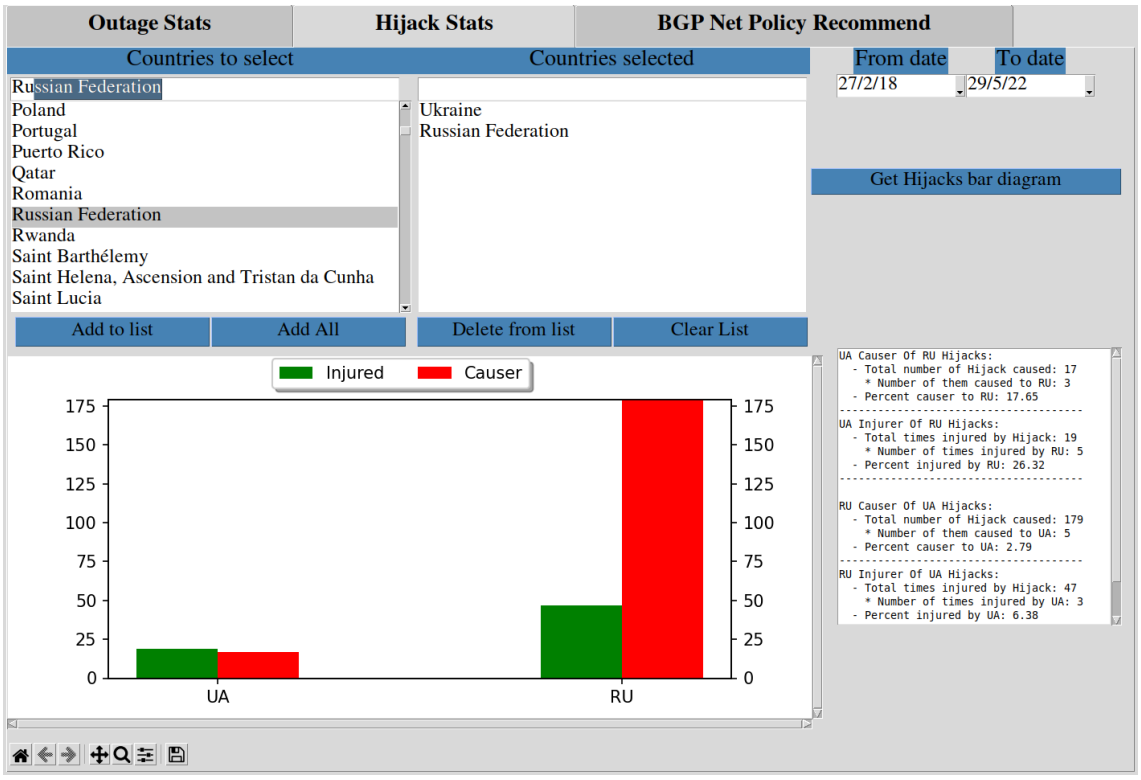
Como ya se ha mencionado, la obtención de eventos de mayor antigüedad a 2022 no se pudo realizar mediante Twitter o BGPMon. Sin embargo, este aspecto se pudo suplir mediante el uso del conjunto de datos de Arakadakis Konstantinos [1] como se ha explicado en la sección 4.1.4. Por ello, cabe destacar que los datos obtenidos solo contienen información acerca de dos años, el actual 2022 y el año 2018, por lo que en realidad los datos no sirven como criterio estricto para analizar la repercusión internacional. Además, BgpRS no utiliza conceptos de estadística compleja para representar la información, sino que muestra información sencilla que representa la repercusión internacional sobre BGP.

Partiendo de la base de los datos obtenidos en los años mencionados, se puede comparar el estado de cualquier región en BGP mediante la observación del número de sucesos *Outage* y *Hijack*. En el caso de Ucrania y Rusia estos datos podrían indicar el impacto de la guerra en BGP, pudiendo visualizar el contenido de los eventos por medio de la aplicación BgpRS.

De esta forma, si se pretende visualizar el total de eventos BGP producidos por Ucrania y Rusia durante estas dos fechas, a través de la aplicación y con los datos que se poseen en el momento de la realización de esta memoria, se pueden obtener los gráficos representados en la figura 8.1.



(a) Eventos *Outage* BGP



(b) Eventos *Hijack* BGP

Figura 8.1: Todos los eventos BGP obtenidos: Ucrania y Rusia

8.1.1. Comparativa en eventos de tipo Outage

En esta sección se realizará una comparativa del número de eventos de tipo *Outage* entre los países Ucrania y Rusia. Con la realización de esto, se busca contemplar el número de caídas de servicio de los diferentes sistemas autónomos a cargo de estos dos países y, por consiguiente, comprobar si la guerra en la que se encuentran aumenta de algún modo el número de eventos de este tipo.

Para corroborar esto, se proporciona la figura 8.2 en la que se selecciona el año 2018 por completo. Como resultado se obtienen el número de eventos referentes a interrupciones de servicio de ambos países, observando que Ucrania tuvo alrededor de 60 interrupciones y Rusia más de 70.

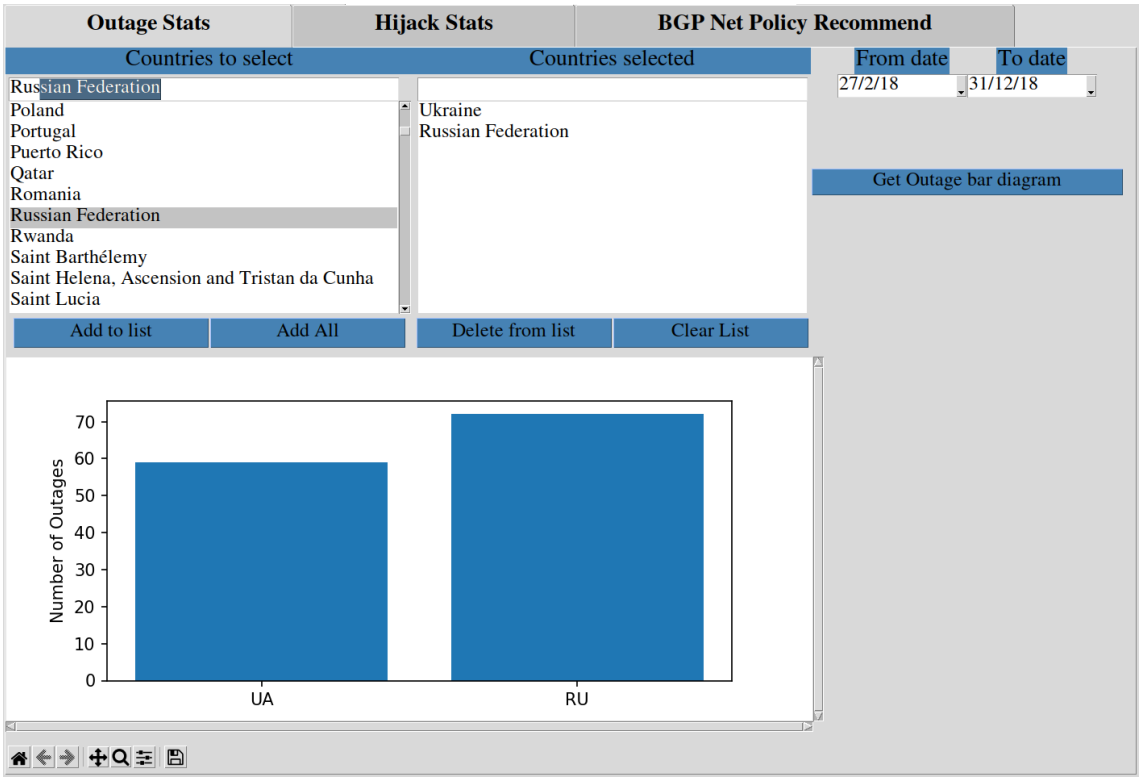


Figura 8.2: Número de interrupciones en 2018

Si de manera similar se seleccionan los datos del año 2022, como se representa en la figura 8.3, se observa que en tan solo la mitad de tiempo Ucrania ya posee más de 50 interrupciones de servicio y Rusia más de 60. Con esta información, se puede concluir que cualquiera de estos países, en 2022, tienen más probabilidades de alcanzar un mayor número de interrupciones de servicio que en el año 2018.

Siendo rigurosos, habría que tener en cuenta el posible aumento en el numero de AS en cada país, a más AS, puede haber más eventos. Sin embargo, observando la imagen representada se atisba que tal vez la guerra entre estos dos países pueda estar afectando directamente sobre el funcionamiento de BGP.

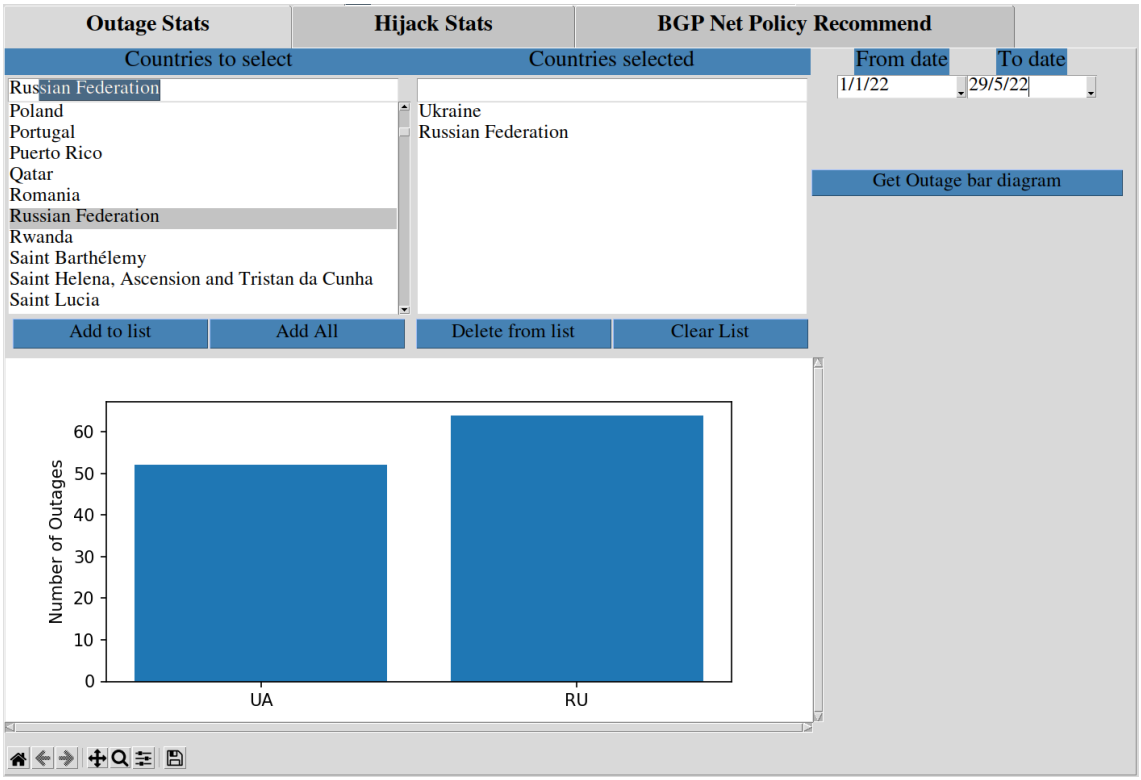


Figura 8.3: Número de interrupciones (Oct 2021 - Jun 2022)

8.1.2. Comparativa en eventos de tipo Hijack

De mano de las interrupciones de servicio, otros de los eventos que suceden en BGP y que pueden estar ligados a la presente guerra entre Rusia y Ucrania, son los secuestros de prefijos. Para poner en observación esta posibilidad, se puede hacer uso de BgpRS. De forma que si se accede a la pestaña de `Hijack Stats` y se seleccionan todos los eventos de este tipo sucedidos en 2018, se obtiene un resultado como el representado en la figura 8.4.

Como se observa en la imagen anterior, la aplicación retorna dos tipos de datos diferentes. Por una parte, en el gráfico se pude ver el total de eventos *Hijack* en los que cualquiera de los dos países ha estado involucrado en 2018.

Si se analiza el diagrama de barras de la imagen, se observa que Rusia ha sido causante de alrededor de 14 secuestros, mientras que Ucrania lo ha sido de 4. Además, en cuanto a las veces que han sido víctimas de un evento *Hijack*, Ucrania suma una cantidad de 6 veces, mientras que Rusia suma 12.

Por otra parte, si se analiza la salida de la derecha proporcionada por la aplicación, se observa que en ninguno de los dos países fue consecuencia del secuestro de prefijos del otro.

Por último, en la figura 8.5 se muestra la salida de BgpRS si trata de obtener este mismo tipo de información, pero esta vez se seleccionando el año 2022. En este caso, se observa como la cantidad de eventos de tipo *Hijack* aumenta considerablemente para ambos países, indicando una mayor actividad o posible presencia en BGP.

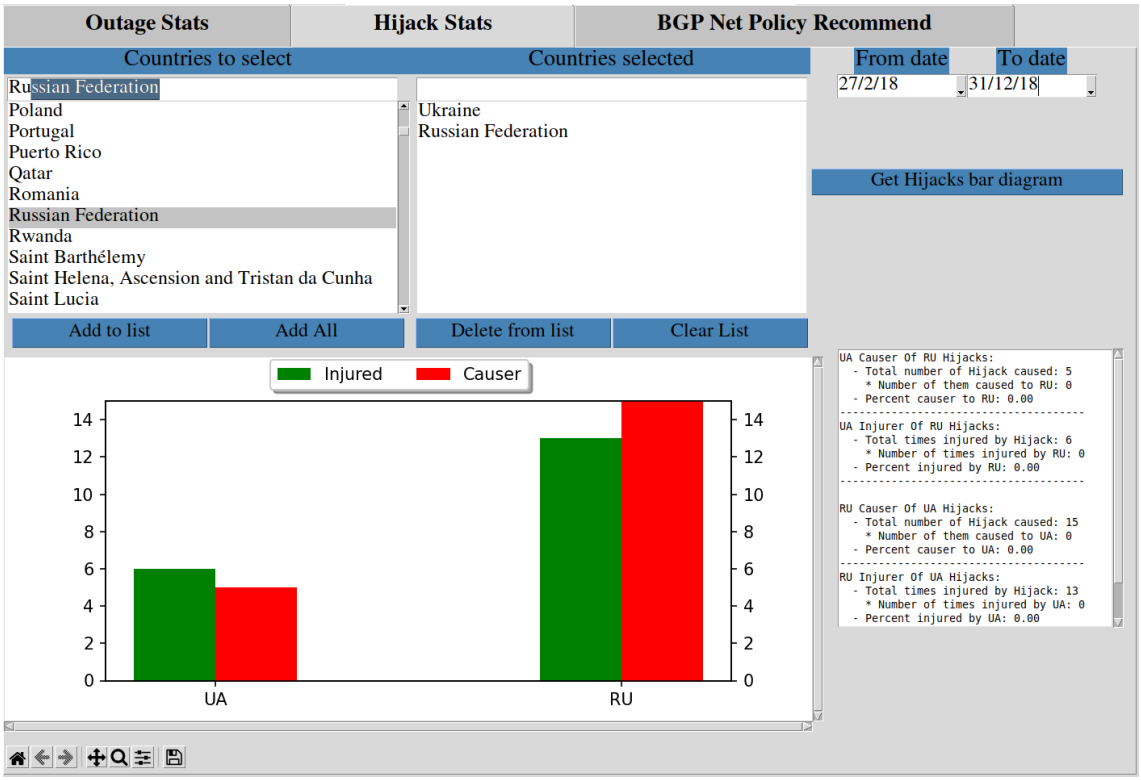


Figura 8.4: Número de secuestros en 2018

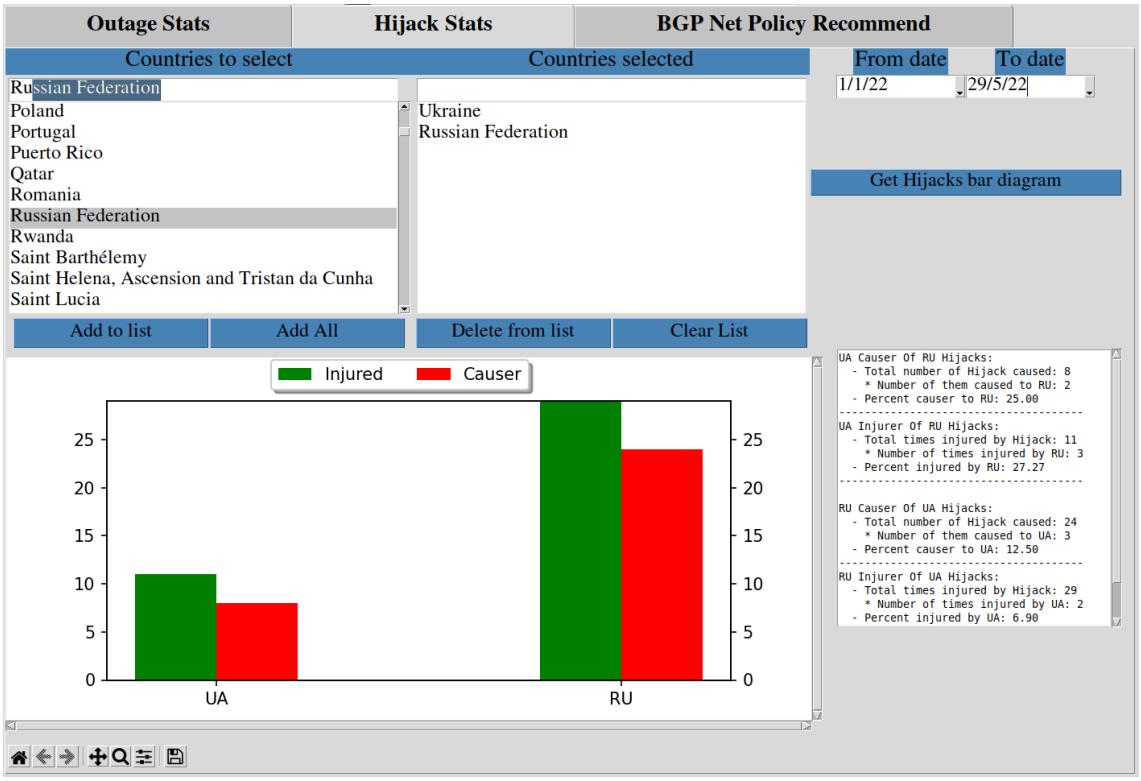


Figura 8.5: Número de secuestros (Oct 2021 - Jun 2022)

De la misma manera, si se observa la salida proporcionada por este tipo de eventos, se puede comprobar que, esta vez, los dos países han sido víctimas del otro ante eventos *Hijack*. Además, también destaca que Rusia no solo ha sido el causante de cierto número de secuestros de Ucrania, sino que ha participado en un gran número de eventos de este tipo, lo que puede significar la representación del conflicto con países terceros a consecuencia de la guerra.

8.2. Sistema de recomendación

BgpRS proporciona mecanismos de recomendación como los observados en el capítulo 5. En esta sección se pretende poner en práctica los dos posibles aspectos de recomendación que proporciona la aplicación.

8.2.1. Recomendaciones Outage

El `AS12479` es uno de los AS de los que se ha obtenido una recomendación debido a la acumulación de distintos tipos de evento. Con esto en mente y con el fin de recrear un posible escenario BGP se proporciona una topología como la representada en la figura 8.6.

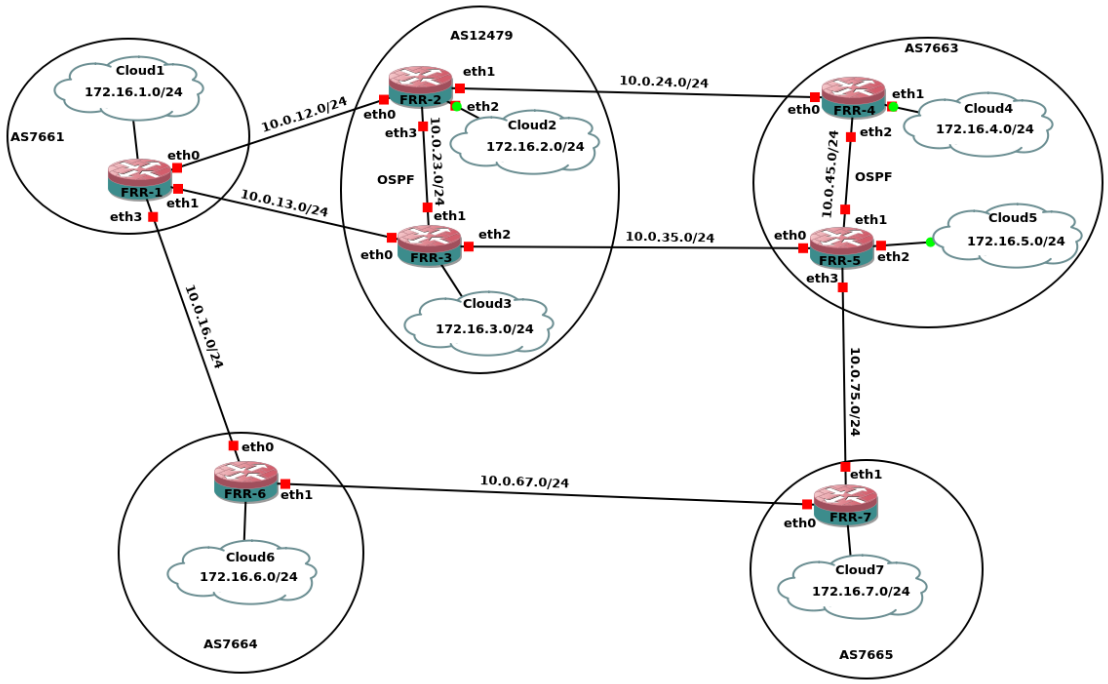


Figura 8.6: Topología BGP para ejemplificar las recomendaciones de BgpRS en eventos *Outage*

Esta topología es trasladable a los diferentes entornos de configuración de redes, donde también son aplicables los distintos filtros que proporciona BgpRS. De esta manera, si se utilizan herramientas como FRRouting o Quagga, la configuración de cada *router* BGP presente en dicha red se realizaría de manera similar a la reflejada en la imagen 8.7. En esta figura se especifican cada uno de los aspectos de configuración necesarios para que el intercambio de mensajes BGP se realice.

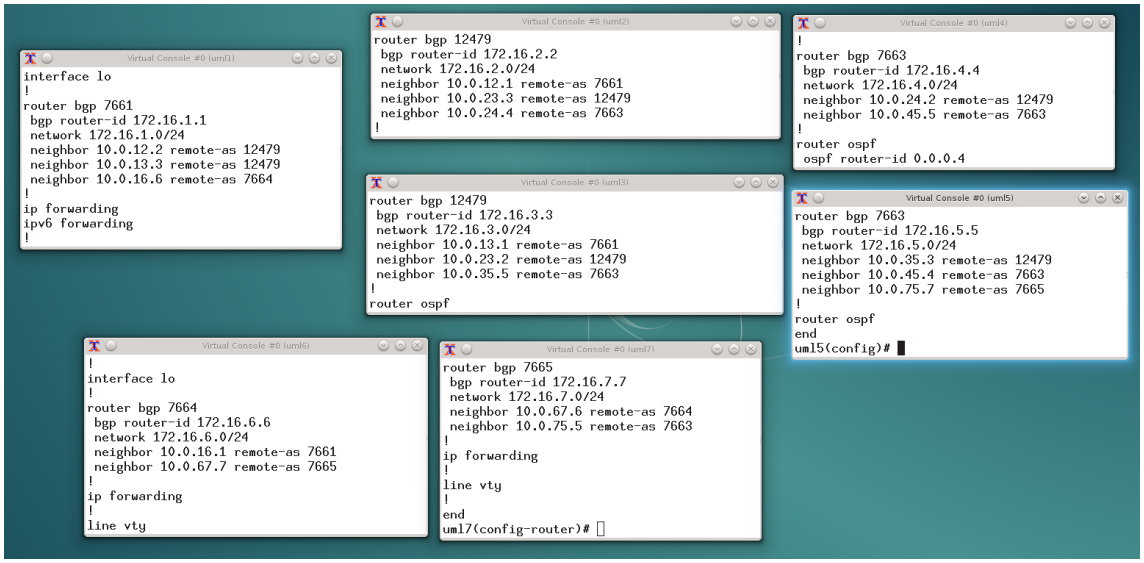


Figura 8.7: Traslado de topología a Quagga/FRRouting

Una vez realizada esta configuración, cada uno de los encaminadores emprenderá su proceso para el rellenado de sus tablas locales de direccionamiento, permitiendo que cada uno de ellos conozca todos los prefijos presentes, los cuales están nombrados como *Clouds* en la topología.

Un extracto del contenido de las tablas de encaminamiento de uno de los nodos de la topología se observa en la figura 8.8. En este caso la tabla representada pertenece a la máquina `FRR-1`. El motivo de elegir esta máquina también se debe a que será la seleccionada para ser configurada mediante los comandos recomendados por BgpRS.

A través de esta interfaz se obtiene información sobre cada uno de los prefijos que conoce `FRR-1` y sobre la ruta preferida para el direccionamiento del tráfico, la cual es indicada mediante el símbolo: `>`. Por otra parte, acompañando a estas rutas se observan campos como `Metric`, `LocPrf` y `Path`, que corresponden con los atributos `MULTI_EXIT_DISCRIMINATOR`, `LOCAL_PREFERENCE` y `AS_PATH` explicados en la sección 3.2.2.

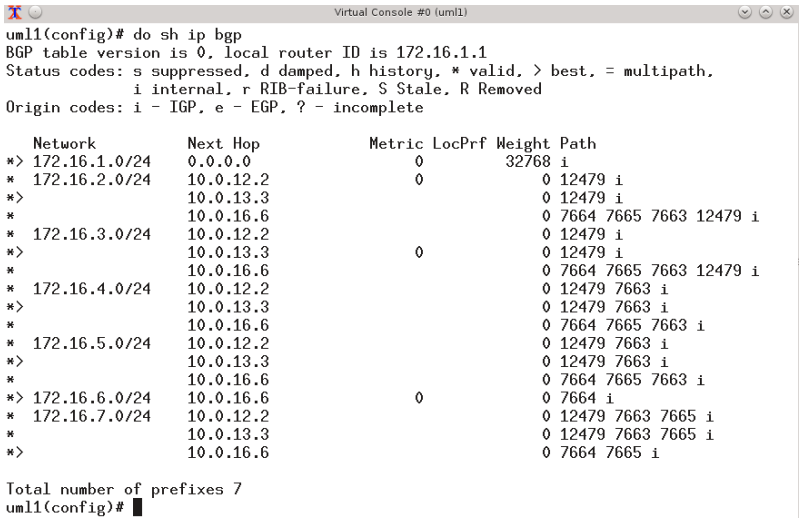


Figura 8.8: Estado inicial de las rutas en `FRR-1`

Llegado a este punto, se pueden realizar las configuraciones proporcionadas por la aplicación. En la imagen 8.9 se observan los comandos basados en la recomendación de BgpRS debidamente introducidos en este entorno de pruebas. En este caso, se utilizan los comandos que son recomendados cuando el `AS12479` ha resultado ser un AS con interrupciones de servicio recurrentes.

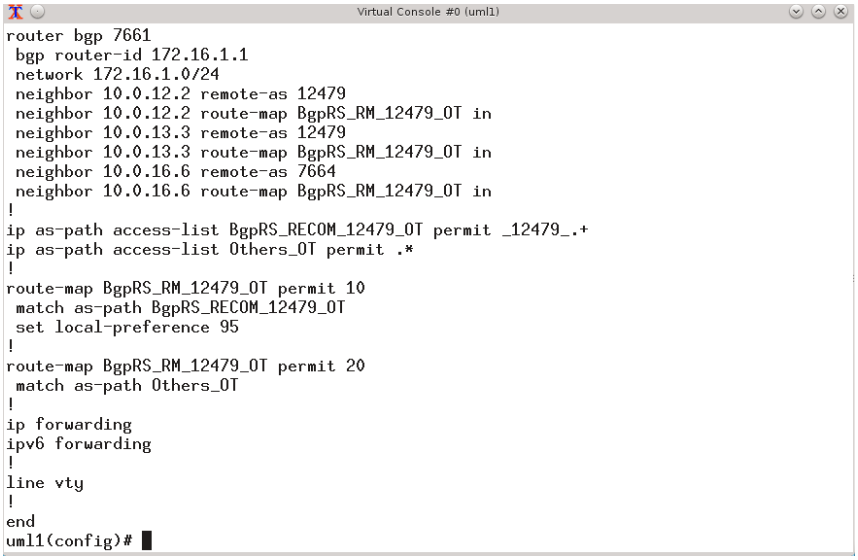


Figura 8.9: Aplicado de políticas recomendadas para eventos *Outage* en `FRR-1`

Una vez aplicados los filtros `access-list` y `route-map` de la imagen anterior, se necesita indicar a `FRR-1` la necesidad de recalcular su tabla de encaminamiento. Esta acción se realiza en Quagga mediante el comando `ip bgp clear`, que permite que el *router* aplique esta vez los nuevos filtros a la hora de recibir de nuevo las diferentes rutas.

Con todo esto, el *router* `FRR-1` cambiará su tabla de encaminamiento como se observa en la figura 8.10, asignando una menor preferencia local en las rutas donde el `AS12479` se comporta como sistema de tránsito, tal y como se explicó en la sección 5.2.1.

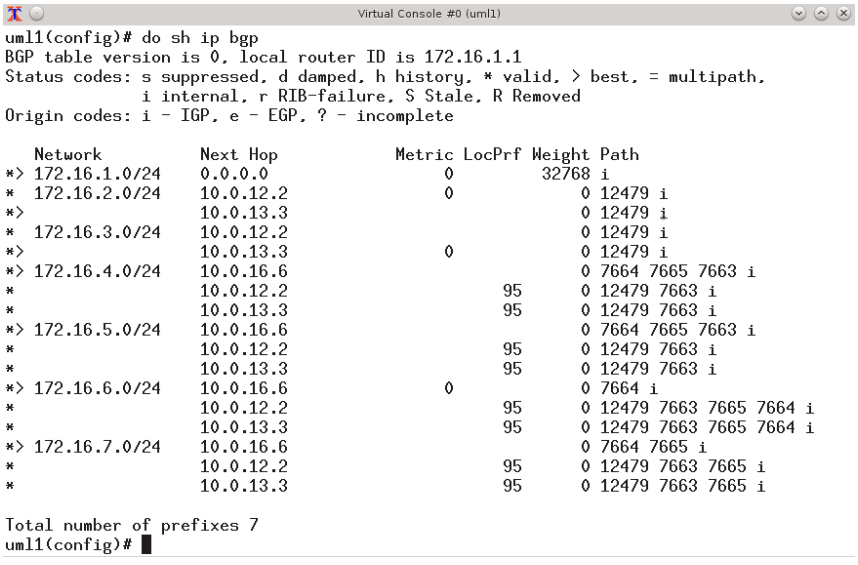


Figura 8.10: Resultado de políticas recomendadas para eventos *Outage* en `FRR-1`

En la figura 8.11 se observa el cambio de rutas como resultado de aplicar la recomendación anterior. La antigua ruta (1) corresponde a la que existía como preferencia hacia el prefijo `172.16.4.0/24` (Figura 8.8) y la nueva ruta (2) corresponde a la preferencia para alcanzar el mismo prefijo (Figura 8.10).

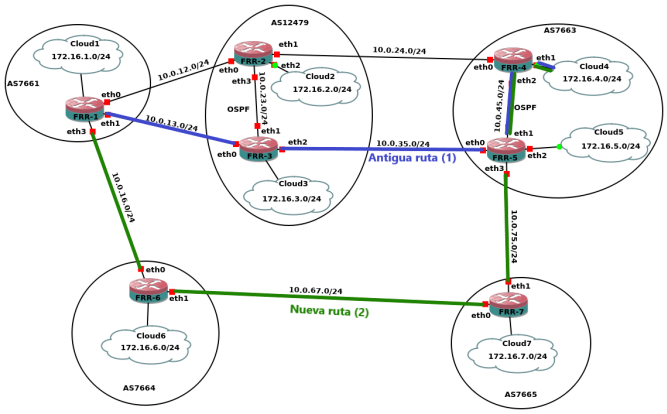


Figura 8.11: Resultado de recomendación sobre la topología (Eventos *Outage*)

8.2.2. Recomendaciones Hijack

De manera análoga, pero esta vez aplicando la recomendación de BgpRS en caso de que el `AS51984` sea propenso al secuestro de prefijos, se proporciona la topología de la figura 8.12. En ésta se observa que el prefijo `172.16.2.0/24` es anunciado tanto por `FRR-2` como por `FRR-4`, siendo este último el origen auténtico de dicho prefijo.

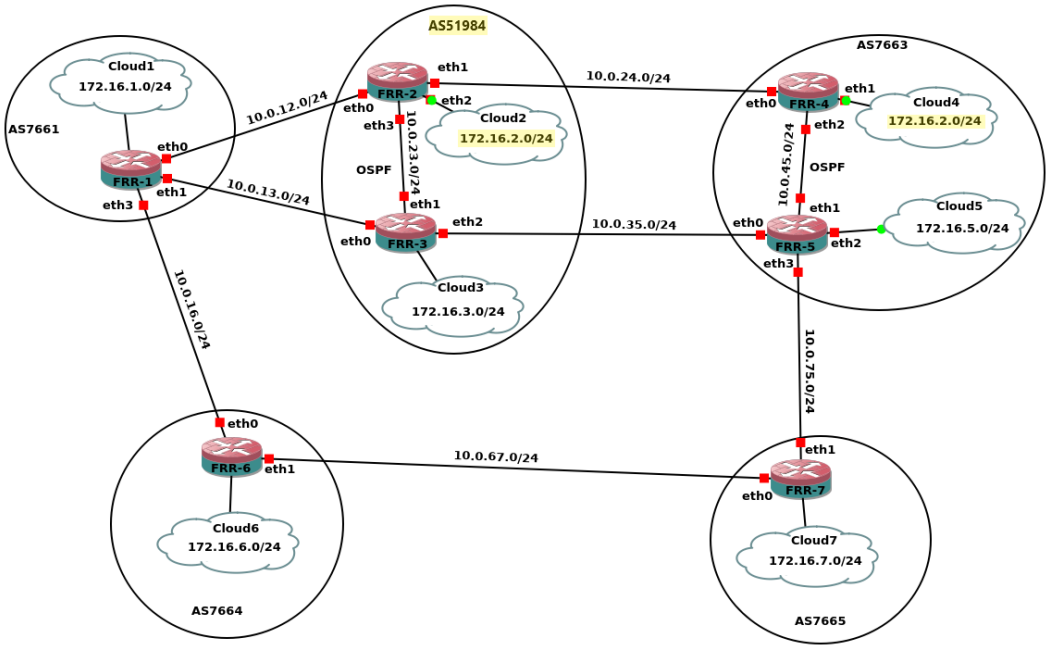


Figura 8.12: Topología BGP para ejemplificar las recomendaciones de BgpRS en eventos *Hijack*

Como el AS51984 ha sido identificado como un sistema propenso al secuestro de prefijos, la recomendación de la aplicación es disminuir la preferencia local de aquellas rutas en las que este AS ha sido origen. En la figura 8.13 se representan los filtros introducidos en la *shell* vtysh.

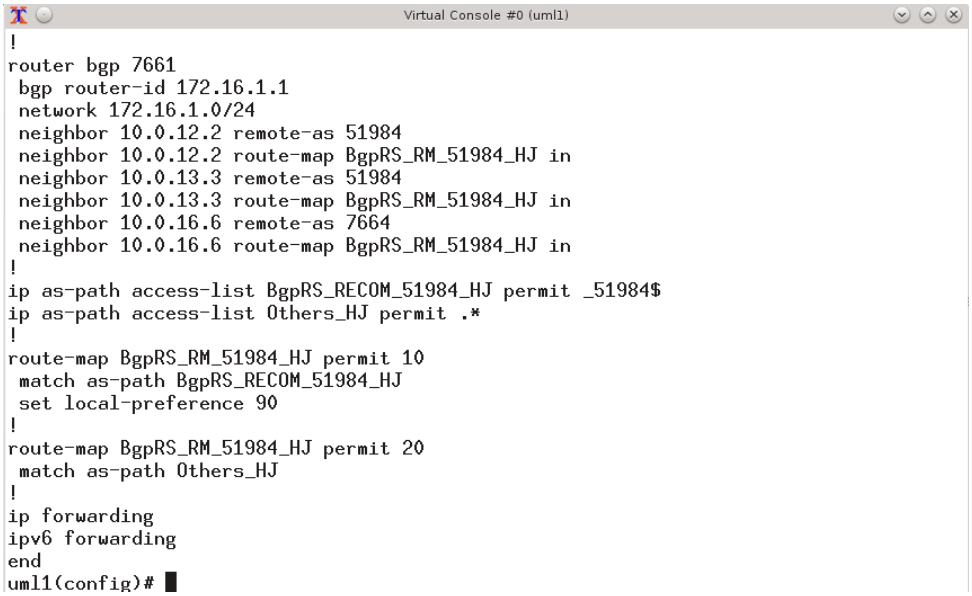


Figura 8.13: Aplicado de políticas recomendadas para eventos *Hijack* en FRR-1

Finalmente, tras la recepción de la información de encaminamiento y habiendo filtrado las rutas según los comandos introducidos, el *router* FRR-1 fija una preferencia de rutas como la observada en la figura 8.14.

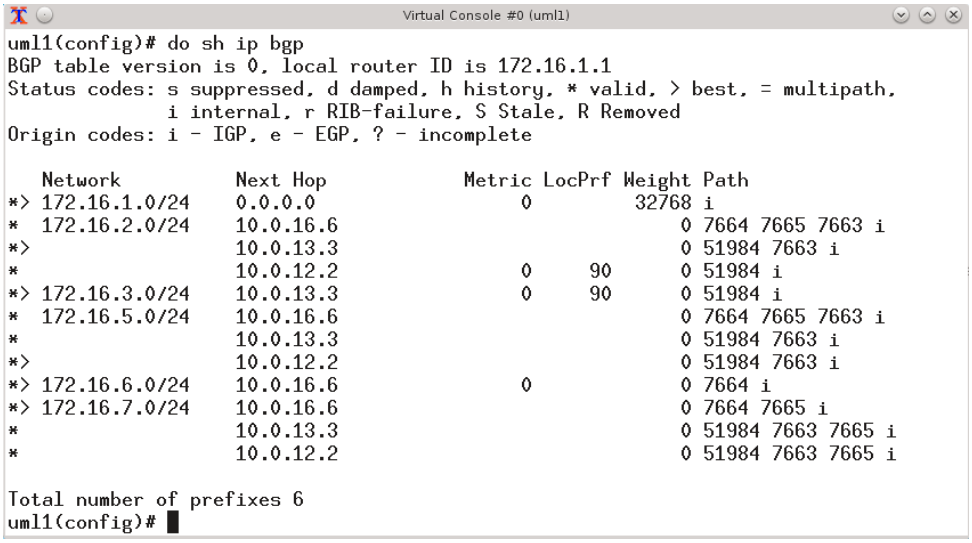


Figura 8.14: Resultado de políticas recomendadas para eventos *Hijack* en FRR-1

Si se compara la información de esta imagen con la de la figura 8.15 se puede observar un cambio en las rutas determinadas, estableciendo para los prefijos originados desde el AS51984 una menor preferencia. Esto se puede ver reflejado en la topología de la figura 8.16.

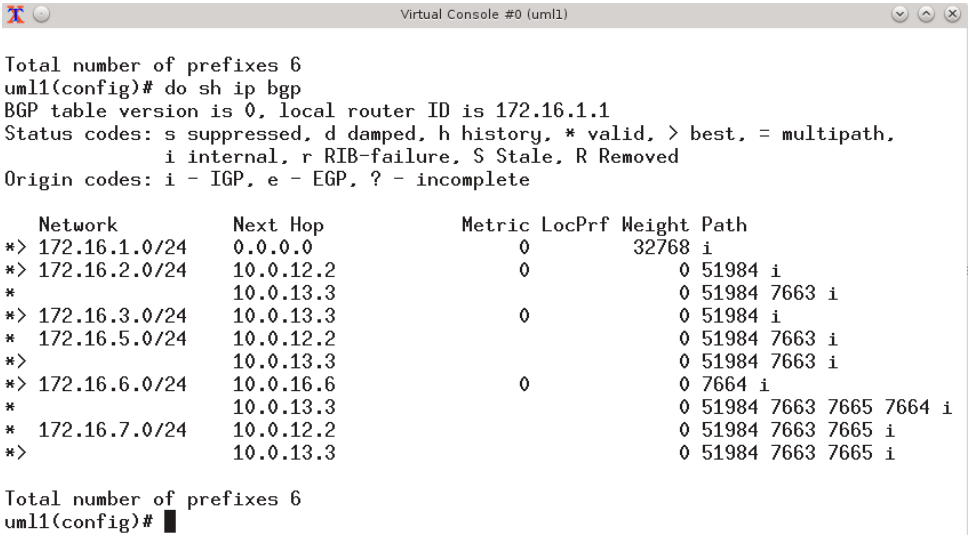


Figura 8.15: Estado inicial de `FRR-1` en la topología *Hijack*

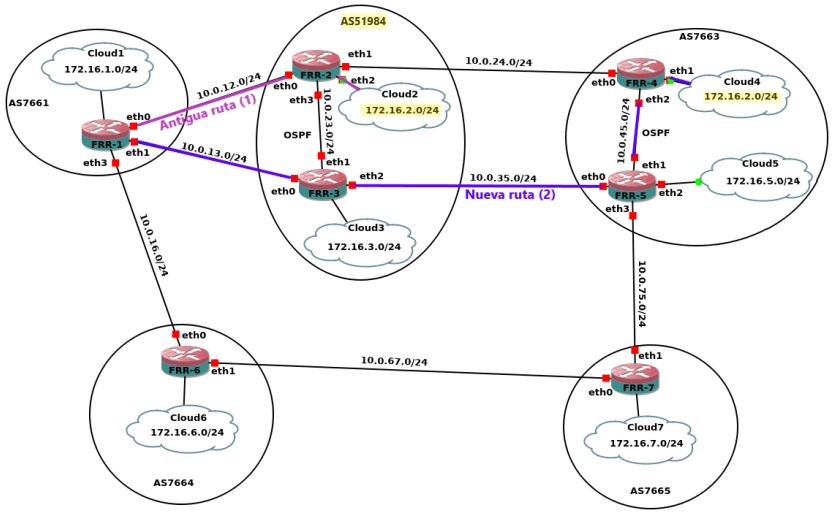


Figura 8.16: Resultado de recomendación en la topología (Eventos *Hijack*)

En esta última imagen se observa que pese a aplicar la recomendación de BgpRS el tráfico sigue transitando a través de `AS51984`. Esta situación implica que el AS puede seguir significando algún riesgo en cuanto a la posible sustracción información. Esto se debe a que la expresión regular `_51984$` solo contempla aquellas situaciones en las que el AS problemático fue el origen y no evita el flujo de datos a través del mismo.

La recomendación proporcionada por BgpRS utiliza únicamente el filtrado por `AS_PATH` mediante políticas de tipo `route-map`. Para evitar que el tráfico siga transitando por un AS con eventos *Hijack* recurrentes bastaría con añadir en la política `route-map` un filtrado por prefijos y una expresión regular adicional utilizando el `AS_PATH`.

El filtrado por prefijos servirá para reducir únicamente el `LOCAL_PREFERENCE` sobre los anuncios que vayan dirigidos al prefijo que se ha visto comprometido, evitando de esta manera que el tráfico dirigido hacia otros prefijos, solo accesibles desde el AS problemático, se vean perjudicados.

En este ejemplo el prefijo comprometido es `172.16.2.0/24`, por lo que el comando a añadir en la configuración debería ser similar a `ip prefix-list BgpRS_prfx permit 172.16.2.0/24`. Además, el AS que aplique la recomendación deberá añadir en la política, `route-map BgpRS_RM_51984_HJ`, la condición `match ip prefix-list BgpRS_prfx`, filtrando de esta manera únicamente las rutas destinadas al prefijo secuestrado.

Por otra parte, el filtrado no solo debe aplicarse en las rutas en las que el `AS51984` es origen, sino que también se debe realizar sobre las rutas en las que este actúa como tránsito hacia el prefijo `172.16.2.0/24`. Para cumplir este objetivo se debe añadir una expresión regular adicional con la forma `_51984_$` e incorporar el `match` correspondiente en la política `route-map BgpRS_RM_51984_HJ`.

Con todos estos pasos se busca obtener un cambio en las rutas de tal forma que aquellas con destino `172.16.2.0/24` eviten utilizar el `AS51984`, dando como resultado la ruta ideal (3) que se contempla en la figura 8.17.

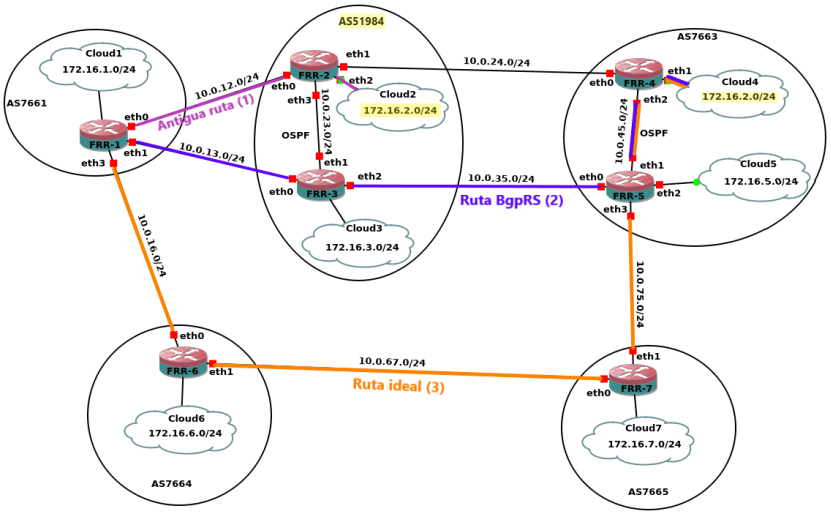


Figura 8.17: Resultado ideal de la recomendación BgpRS en la topología (Eventos *Hijack*)

Capítulo 9

Conclusiones y Trabajo Futuro

En este capítulo se procederá a detallar los resultados obtenidos y las dificultades sobrellevadas en este trabajo. Por último, se describirán ciertas características mejorables de la aplicación y algunas líneas de trabajo que se podrían realizar en el futuro.

9.1. Conclusiones

En el capítulo 4 se ha definido cómo se obtiene la información de los eventos BGP. Twitter solicita multitud de información antes de poder utilizar los servicios del plan *Academic Research* de su API. Esta compañía, para asegurar la identidad del usuario (estudiante o profesor) que realiza la petición de acceso exige datos[1] que, por seguridad, en el caso de ser estudiante de la UCM no son públicos. Además, por motivos desconocidos también exige describir las metodologías que se utilizarán para analizar los *tweets* extraídos, siendo este un aspecto que en el momento de la solicitud todavía no estaba completamente definido. Todos estos factores dificultaron enormemente el acceso a los datos necesarios para BgpRS.

Una vez se consiguió acceder a la API y se extrajeron los primeros datos, la información debía ser tratada para adaptarla de una forma que fuese viable para BgpRS. La necesidad de la limpieza de los datos se debía a que BGPStream, a lo largo de su funcionamiento, había cambiado en varias ocasiones la forma en la que publicaba su información en Twitter. Esto propició la búsqueda de soluciones para completar los datos que faltaban en ciertos *tweets* y para obtener la información de manera correcta.

En los capítulos 5 y 8 se ha podido observar cómo con la extracción y tratamiento de estos datos el sistema de recomendación, BgpRS, proporciona recomendaciones para influir en el encaminamiento de BGP con el fin evitar sistemas autónomos con reputación cuestionable; con lo que se consigue alcanzar el objetivo de este Trabajo de Fin de Máster.

Además, en el capítulo 6 se ha visto que BgpRS puede servir como una herramienta para la monitorización y alerta de eventos BGP, pudiendo visualizar los eventos producidos por los diferentes países. Esto representa una funcionalidad adicional que da un valor añadido a BgpRS, incorporando posibles líneas de desarrollo para este proyecto.

[1] Enlaces hacia el perfil en el directorio de profesores o estudiantes de la institución académica, el sitio *web* del grupo de investigación, laboratorio o departamento, y además un enlace hacia el perfil de Google Scholar.

9.2. Líneas de mejora

Existen muchos aspectos mejorables de la aplicación que no han podido realizarse por falta de recursos y de tiempo. Esta sección trata de dar a conocer algunos de ellos con el fin de proporcionar una guía sobre posible trabajo a realizar en el campo de este Trabajo de Fin de Máster.

9.2.1. Mejoras en rendimiento

BgpRS hace uso de multitud de servicios que, aunque benefician la calidad de los datos que utiliza, repercuten drásticamente en el rendimiento de la aplicación. Por un lado, se encuentra el comando `whois`, el cual trata de extraer información de cada uno de los nuevos eventos que se incorporan. La información que éste devuelve se realiza en forma de cadena de texto, por lo que es necesario limpiarla adecuadamente para que sirva de utilidad para BgpRS. Esto provoca un cuello de botella cuando se tratan de incorporar una gran cantidad de eventos. Además, por problemas propios de `whois`, en ocasiones se tarda demasiado en retornar la información, pudiendo producir excepciones de tipo *time-out*.

Por otra parte, cada uno de los eventos contiene una *url* que se utiliza para obtener más información, en caso de que sea posible, de la página *web* de BGPMon. La aplicación para realizar esto obtiene el contenido `HTML` en su totalidad y trata de extraer la información que pueda faltar, lo que también supone un gran aumento de tiempo por cada elemento que se busque clasificar.

Como consecuencia se podría tratar de estudiar otras alternativas para la extracción de estos datos para mejorar este aspecto de la aplicación o, en su defecto, utilizar técnicas de multiprocesado de datos como son MPI (*Message Passing Interface*) o similares, solución que se trató de adelantar mediante la implementación de un proceso independiente para la carga y clasificación de datos.

9.2.2. Mejoras para el usuario

Existen multitud de líneas para la mejora de la aplicación desde el punto de vista del usuario. En primer lugar, destaca que para iniciar por primera vez BgpRS el usuario debe realizar multitud de tareas que hacen que su despliegue sea demasiado tedioso. Por ello, surge la idea de automatizar este proceso mediante el uso de la ya presente librería PyDrive y un archivo de configuración unificado de tal forma que se proporcionen cada una de las variables necesarias para habilitar los distintos servicios.

Por otra parte, cabe destacar que BgpRS proporciona sus recomendaciones en función de las solicitudes del usuario, siendo necesaria la consulta sobre un AS concreto para recibir las posibles acciones de configuración. Es posible que los AS con baja reputación no sean vecinos inmediatos del sistema autónomo sobre el que se apliquen estas recomendaciones. Por ello, puede que el administrador no conozca el ASN del sistema problemático o si alguna de sus rutas lo utilizan de alguna forma, impidiendo encontrar la recomendación adecuada en BgpRS. Este es un aspecto mejorable de la aplicación donde se podrían ajustar las recomendaciones de BgpRS para que se proporcionen, a la vez y de manera automática, todos los filtros de rutas para cada AS propenso a la interrupción o el secuestro de prefijos.

Otro de los aspectos mejorables es la Interfaz gráfica de la aplicación. Desde el punto de vista del diseño, la aplicación es realmente sencilla y utiliza un medio bastante básico para representar sus elementos, por lo que en un futuro sería interesante mejorar este aspecto con el fin de volver a la aplicación más atractiva. En esta linea, también se podrían agregar medios para representar gráficamente los resultados que arroja la aplicación en forma de texto (Figura 6.4).

Por último, los datos clasificados por BgpRS contienen gran variedad de información que no es utilizada por la versión actual de la aplicación. Por ello, cabe destacar que es posible ampliar sus funcionalidades. Por ejemplo, para contemplar nuevos eventos de tipo *BGP Leak* que no se extraen de BGPMon, pero que sí se pueden proporcionar de manera estática. Esto tiene el objetivo de mostrar más información al usuario acerca de los eventos de un ASN mediante las funcionalidades de recomendación, o de un país mediante las funcionalidades de análisis en BgpRS.

9.2.3. Mejoras en la detección de recomendaciones

Como se ha visto en las secciones 5.2.1 y 5.2.2, si un AS ha producido más de tres eventos de cualquier tipo durante su periodo de funcionamiento, BgpRS clasificará al mismo como problemático. Esta cantidad de eventos está incluida en el código como una variable permanente. No obstante, seguramente este método no sea el más adecuado para determinar la fiabilidad de un AS.

Con el fin de realizar una mejor clasificación para cada AS del conjunto de datos sería lógico calcular la desviación que éste tiene con respecto a los demás. Por este motivo, surge la idea de utilizar técnicas de Aprendizaje Automático para clasificar o agrupar los diferentes AS según su historial utilizando técnicas como Regresión logística o Máquinas de Vector Soporte (SVM, *Support Vector Machines*).

Por otra parte, BgpRS también utiliza medidas simples para determinar cuánto ha de reducirse la preferencia local de una ruta para un AS propenso a eventos problemáticos. Para ello, utiliza un factor global fijo que en el caso de eventos *Outage* toma valor 5 y en el caso de eventos *Hijack* toma el valor 10. Sin embargo, este factor puede ser insuficiente o ser excesivo.

Este factor es dependiente de la configuración local particular de cada AS, ya que el usuario puede haber fijado una preferencia local especifica. Esto puede provocar que la recomendación proporcionada no ejerza repercusión alguna.

La información sobre esta configuración local no es recopilada actualmente por BgpRS. La forma de obtener esta información sería solicitársela al usuario o implementar funcionalidades en BgpRs para que pudiera leer e interpretar la configuración de FRR de manera automática. Cualquiera de estos aspectos mejoraría sustancialmente la herramienta de recomendación.

Chapter 10

Introduction

The Internet is a system that provides a multitude of services to society through various communication protocols. Each of these protocols, with specific functions, makes possible the operation of the biggest source of information in the world. To start from a base, it is important to know that these protocols are classifiable through the OSI model (*Open System Interconnection*) or the TCP/IP model, which contain the same information but use a different layer distribution (Figure 10.1).

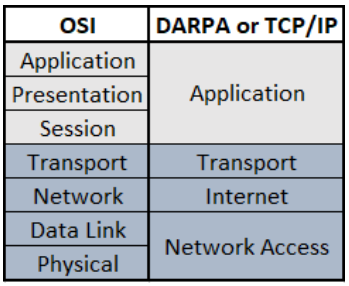


Figure 10.1: OSI Model Layers and TCP/IP Model

It is complex to achieve the coordination of each element present on the Internet, sometimes the protocols in charge of this task include configuration characteristics that can affect its correct operation. The previous models define the different stages that the data must go through to be sent. The network layer is responsible for logical data routing, using protocols such as IP (Internet Protocol) or OSPF (Open Shortest Path First) to determine the best path for optimal sending of data and serve as a basis for routing performed by BGP (Border Gateway Protocol).

The main object of this Master's Thesis is the study of the BGP protocol. This protocol, whose function cannot be classified in a single layer of the OSI model, is responsible for the construction of global Internet routes that define how to reach each of the different destinations present in the network. However, as will be seen later, this protocol also contains some peculiarities that deserve to be studied.

10.1. Motivation

The Internet can be seen as a large graph where the vertices would be in charge of managing and carrying out the exchange of information. The BGP protocol provides mechanisms for the exchange of routing information between the different vertices that make up this network. Within BGP terminology, these vertices or nodes are called Autonomous Systems (AS, Autonomous System) and are identified by a unique number or ASN (Autonomous System Number).

In this way, the protocol is responsible for establishing the optimal routes through which the traffic will be directed. To formalize these routes the protocol incorporates different configuration options that are managed locally by different ASs. However, it remains to define what an optimal route is, since each autonomous system may have different particular interests.

The great versatility is a great benefit in favor of the BGP protocol when it comes to configuring and determining the different routes that network traffic will follow, allowing organizations responsible for routing to do so to suit their own interests. However, an alteration in the configuration of the BGP routing policies determines network traffic and may lead to the theft of sensitive information for the different global geopolitical entities or even its isolation on the Internet.

Throughout the history of BGP many problems have been experienced as a result of misconfiguration. Without going any further, one of these consequences occurred on October 4, 2021 [26], when Facebook, due to a misconfiguration of the BGP routing tables, became inaccessible through the Internet, causing the chain crash of the social networks (Whatsapp and Instagram) most used by users and the interruption of the service of other companies dependent on Facebook as a provider.

For all these reasons, there are many studies that seek to identify when one of these autonomous systems does not work as it should, that is, when an AS does not exchange the appropriate information due to a bad local configuration, whether deliberately or not.

In the same way, there are also tools available to the public that are responsible for monitoring BGP updates, trying to determine a possible fraudulent announcement[1] (Hijack), a possible service drop (Outage) or an improper filtering of prefixes (BGP Leak) during the exchange of information between the different ASs. This is the scope of the BGPStream [19] and BGPMon [28] tools, which are accessible through the Cisco Crosswork Cloud platform [5]. These tools are essential for BgpRS, since through the social platform Twitter they are in charge of publishing relevant information about events that may represent a significant risk in BGP.

10.2. Objectives

In this Master's Thesis a BGP recommendation system is presented in the form of an application. This application, which has been called BgpRS (BGP Recommendation System), based on data published by Cisco Systems provides recommendations for the configuration of BGP routers in order to try to avoid those Autonomous Systems that may mean some risk.

[1] Exchange of routing information between autonomous systems (Section 3.1.1)

In addition, as an additional feature, BgpRS also provides data visualization tools, giving the user the ability to perform a historical study on the events generated by the different countries in BGP.

The current war between the Russian Federation and Ukraine is a possible scenario to prove that BGP is not a simple network protocol. As the AS of these countries can produce Outage and Hijack events daily, if different moments in time before and after the war are selected through BgpRS, and the BGP events of each country are compared, it is possible to visualize the impact of the war on the Internet.

Some BGP routers configuration errors can also trigger events like the ones mentioned above. For this reason, the idea of classifying Autonomous Systems arises according to their incident history. The trends obtained through the data of these events allow the reputation of the ASs to be established and the BgpRS application to inform BGP administrators about the most appropriate way to deal with an AS whose behavior is erratic, using `instructions vtysh` contemplated in Quagga or FRRouting and that are easily applicable to the Cisco CLI.

10.3. Work Plan

During the operation of the BGP protocol numerous events take place: appearance or disappearance of routes, incorporation of new AS, etc. In this project, it is not necessary to use each of these events, since only those that imply a service outage (Outage) or a change in the origin of prefixes (Hijack) are needed.

The extraction of this data is a necessary requirement for the implementation of BgpRS. However, it should be noted that its filtering and classification requires a large number of unavailable resources, which are accessible through mirrors or reflectors of BGP routes (LG, BGP Looking Glasses). For this reason, the data processing will be carried out by the BGPStream and BGPMon tools, which are used internally by LGs to consolidate their information. In this way, by using this information it will be possible to maintain a consistent history that will serve the BgpRS application.

These Cisco tools are in charge of feeding their own historical database. However, access to your data is only possible through their payment API. On the other hand, Cisco also publishes on Twitter for free the information of the events that it identifies through these tools. One of the objectives of BgpRS is to provide a free option for studying the events that occur in BGP. This is done through the extraction of the information available on Twitter, so the API of this social network is an essential tool to obtain much of the necessary information.

Once the information is collected, a classification by country will be made, associating the event with the AS that produced it and the organization in charge. In this way, it will be possible to see if the impact of certain international situations affects BGP.

Finally, we will proceed to build the mentioned BgpRS recommendation functionality through the different data obtained. This system will be able to determine how it is necessary to take measures for a specific AS, providing configuration instructions to avoid redirecting traffic through those ASs that have been considered unreliable.

Chapter 11

Conclusions and Future Work

This chapter will proceed to detail the results obtained and the difficulties encountered in this work. Finally, certain improvable features of the application and some lines of work that could be carried out in the future will be described.

11.1. Conclusions

In chapter 4 it has been defined how the information of the BGP events is obtained. Twitter requests a lot of information before the services of the Academic Research plan of its API can be used. This company, in order to ensure the identity of the user (student or teacher) who makes the access request, requires data[1] which, for security reasons, in the case of being a UCM student, are not public. In addition, for unknown reasons, it also requires a description of the methodologies that will be used to analyze the extracted tweets, this being an aspect that was not yet fully defined at the time of the request. All of these factors made it difficult to access the data needed for BgpRS.

Once the API was accessed and the first data was extracted, the information had to be processed to adapt it in a way that would be viable for BgpRS. The need to clean the data was due to the fact that BGPStream, throughout its operation, had changed the way in which it published its information on Twitter on several occasions. This led to the search for solutions to complete the missing data in certain tweets and to obtain the information correctly.

In chapters 5 and 8 it has been possible to observe how with the extraction and treatment of this data, the recommendation system, BgpRS, provides recommendations to influence BGP routing with the order to avoid stand-alone systems with questionable reputations; with which the objective of this Master's Thesis is achieved.

In addition, in chapter 6 it has been seen that BgpRS can serve as a tool for monitoring and alerting BGP events, it requires means to visualize the trends of the different countries. This represents an additional functionality that gives added value to BgpRS, incorporating possible lines of development for this project.

[1] Links to the profile in the directory of teachers or students of the academic institution, the group's web of research, laboratory or department, and also a link to the Google Scholar profile.

11.2. Improvement lines

There are many improvable aspects of the application that could not be done due to lack of resources and time. This section tries to present some of them, in order to provide a guide on possible work to be done in the field of this Master's Thesis.

11.2.1. Performance improvements

BgpRS makes use of a multitude of services that, although they benefit the quality of the data it uses, have a drastic impact on the performance of the application. On the one hand, there is the `whois` command, which tries to extract information from each of the new events that are incorporated. The information that it returns is in the form of a text string, so it is necessary to properly clean it to be useful for BgpRS. This causes a bottleneck when trying to incorporate a large number of events. In addition, due to problems inherent to `whois`, sometimes it takes too long to return the information, which can produce exceptions of the *time-out* type.

On the other hand, each of the events contains a url that is used to obtain more information if possible from the BGPMon web page. The application to do this obtains the `HTML` content in its entirety and tries to extract the information that may be missing, which also supposes a great increase in time for each element that is sought to be classified.

As a consequence, one could try to study other alternatives for extracting this data to improve this aspect of the application, or failing that, use data multiprocessing techniques such as MPI or similar, a solution that was tried to advance through the implementation of a separate process for data loading and classification.

11.2.2. User enhancements

There are many lines for improving the application from the user's point of view. In the first place, it highlights that in order to start BgpRS for the first time the user must carry out a multitude of tasks that make its deployment too tedious. For this reason, the idea of automating this process arises by using the already present PyDrive library and a unified configuration file in such a way that each of the variables necessary to enable the different services is provided.

On the other hand, it should be noted that BgpRS provides its recommendations based on user requests, requiring a query on a specific AS to receive the possible configuration actions. ASs with low reputations may not be immediate neighbors of the autonomous system over which these recommendations apply. Therefore, the administrator may not know the ASN of the problematic system or if any of its routes use it in some way, preventing it from finding the appropriate recommendation in BgpRS. This is an improvable aspect of the application where the BgpRS recommendations could be adjusted so that all route filters for each AS prone to interruption or prefix hijacking are provided at the same time and automatically.

Another aspect that could be improved is the graphic interface of the application. From a design point of view, the application is really simple and uses a fairly basic means to represent its elements, so in the future it would be interesting to improve this aspect in order to make the application more attractive.

In this line, means could also be added to graphically represent the results that the application throws in text form (Figure 6.4).

Finally, the data classified by BgpRS contains a wide variety of information that is not used by the current version of the application. Therefore, it should be noted that it is possible to expand its functionalities. For example, to contemplate new events of BGP Leak type, which are not extracted from BGPMon, but can be provided statically. This has the objective of being able to show more information to the user about the events of an ASN through the recommendation functionalities, or of a country through the analysis functionalities in BgpRS.

11.2.3. Improvements in detection of recommendations

As seen in sections 5.2.1 and 5.2.2, if an AS has produced more than three events of any type during its period of operation, BgpRS will classify it as problematic. This number of events is included in the code as a permanent variable. However, this method of determining the reliability of an AS may not be the most appropriate.

In order to perform a better classification for each AS in the data set, it would be logical to calculate the deviation that it has with respect to the others. For this reason, the idea of using Machine Learning techniques arises to classify or group the different ASs according to their history using techniques such as Logistic Regression or Support Vector Machines (SVM, Support Vector Machines).

On the other hand, BgpRS also uses simple measures to determine how much to reduce the local preference of a route for an AS prone to problem events. To do this, it uses a fixed global factor that in the case of Outage events takes the value 5 and in the case of Hijack events takes the value 10. However, this factor may be insufficient or excessive.

This factor is dependent on the particular local configuration of each AS, since the user may have set a specific local preference. This may cause the recommendation provided to have no impact.

The information about this local configuration is not currently collected by BgpRS and the only way to obtain it would be to request such data from the user. This aspect could be done through the BgpRS interface by adding more functionality to the application and using the new information to improve the recommendation tool.

Bibliografía

[1] Konstantinos Arakadakis. BGPstream events analysis, 2019. URL: https://gitlab.com/konstantinosarakadakis/BGPstream/-/tree/master.

[2] Muthukadan B. Selenium with Python, 2011-2018. URL: https://selenium-python.readthedocs.io/.

[3] Bob Brown. Vodafone, others ordered to stop cell phone service in Egypt. *Network World*, January 2011. URL: https://www.computerworld.com/article/2512670.

[4] Theune C. ISO country, subdivision, language, currency and script definitions and their translations. URL: https://pypi.org/project/pycountry/.

[5] Inc Cisco Systems. Cisco Crosswork Cloud. URL: https://crosswork.cisco.com/.

[6] Hurricane Electric. Hurricane electric's network looking glass. URL: https://lg.he.net/.

[7] J.C. Fabero Jiménez. Máster en Ingeniería Informática. Encaminamiento externo: BGPv4. Asignatura: Redes de Nueva Generación.

[8] Python Software Foundation. Python interface to Tcl/Tk. URL: https://docs.python.org/3/library/tkinter.html.

[9] Tim A. Griffin and Geoff Huston. BGP Wedgies. RFC 4264, November 2005. URL: https://www.rfc-editor.org/info/rfc4264, doi:10.17487/RFC4264.

[10] JunYoung Gwak, Scott Blevins, Robin Nabel, and Google Inc. PyDrive documentation, 2016. URL: https://pythonhosted.org/PyDrive/.

[11] ISO Central Secretary. Codes for the representation of names of countries and their subdivisions – Part 1: Country code, 2020. URL: https://www.iso.org/obp/ui/#iso:std:iso:3166:-1:ed-4:v1:en.

[12] Roesslein J. Tweepy Documentation, 2009-2022. URL: https://docs.tweepy.org/en/stable/.

[13] Matt Lepinski and Stephen Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480, February 2012. URL: https://www.rfc-editor.org/info/rfc6480, doi:10.17487/RFC6480.

[14] Matt Lepinski and Kotikalapudi Sriram. BGPsec Protocol Specification. RFC 8205, September 2017. URL: https://www.rfc-editor.org/info/rfc8205, doi:10.17487/RFC8205.

[15] K. Lougheed and Y. Rekhter. Border Gateway Protocol (BGP). RFC 1105, June 1989. URL: https://www.rfc-editor.org/info/rfc1105, doi:10.17487/RFC1105.

[16] K. Lougheed and Y. Rekhter. Border Gateway Protocol (BGP). RFC 1163, June 1990. URL: https://www.rfc-editor.org/info/rfc1163, doi:10.17487/RFC1163.

[17] K. Lougheed and Y. Rekhter. Border Gateway Protocol 3 (BGP-3). RFC 1267, October 1991. URL: https://www.rfc-editor.org/info/rfc1267, doi:10.17487/RFC1267.

[18] RedIRIS NOC. RedIRIS Looking glass. URL: https://www.rediris.es/red/lg/.

[19] Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, and Alberto Dainotti. Bgpstream: a software framework for live and historical bgp data analysis. In *Proceedings of the 2016 Internet Measurement Conference*, pages 429–444, 2016. doi:10.1145/2987443.2987482.

[20] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771, March 1995. URL: https://www.rfc-editor.org/info/rfc1771, doi:10.17487/RFC1771.

[21] Yakov Rekhter. BGP Protocol Analysis. RFC 1265, October 1991. URL: https://www.rfc-editor.org/info/rfc1265, doi:10.17487/RFC1265.

[22] Yakov Rekhter, Susan Hares, and Tony Li. A Border Gateway Protocol 4 (BGP-4). RFC 4271, January 2006. URL: https://www.rfc-editor.org/info/rfc4271, doi:10.17487/RFC4271.

[23] Yakov Rekhter and Tony Li. A Border Gateway Protocol 4 (BGP-4). RFC 1654, July 1994. URL: https://www.rfc-editor.org/info/rfc1654, doi:10.17487/RFC1654.

[24] John Scudder, Rex Fernando, and Stephen Stuart. Bgp monitoring protocol (bmp). *Internet Engineering Task Force*, pages 1–27, 2016.

[25] Ryan Singel. Pakistan's Accidental YouTube Re-Routing Exposes Trust Flaw in Net. *Wired*, February 2008. URL: https://www.wired.com/2008/02/pakistans-accid/.

[26] Bruno Toledano. Los culpables de la caída de Facebook, WhatsApp e Instagram: el BGP y las DNS. *El Mundo*, Octubre 2021. URL: https://www.elmundo.es/tecnologia/2021/10/05/615c1d92fc6c8324028b45df.html.

[27] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. URL: https://www.python.org/.

[28] He Yan, Ricardo Oliveira, Kevin Burnett, Dave Matthews, Lixia Zhang, and Dan Massey. Bgpmon: A real-time, scalable, extensible monitoring system. In *2009 Cybersecurity Applications & Technology Conference for Homeland Security*, pages 212–223. IEEE, 2009. doi:10.1109/CATCH.2009.28.

Apéndice A

Título del Apéndice A

A continuación se proporciona un enlace al código del proyecto.

- `https://github.com/alberc01/BgpRS.git`