

Máster Universitario en Ingeniería de Telecomunicación
2017 - 2018

Trabajo Fin de Máster

Análisis de prefijos más específicos en trazas BGP

Ignacio Gutiérrez Díez de los Ríos

Tutor: Alberto García Martínez

Lugar y fecha de presentación prevista: Universidad Carlos
III de Madrid, Septiembre de 2018



Esta obra se encuentra sujeta a la licencia Creative Commons
Reconocimiento – No Comercial – Sin Obra Derivada

Abstract

Nowadays, Internet is the most important communication technology in global terms. It presents an old technology, but it is not easy to know how operates due to its distributed constitution. Internet is constantly evolving and growing. It is essential to ensure connectivity and reachability among connected devices. This task is carried on by routing protocols. BGP is a standardized exterior gateway protocol (EGP).

The proposed study analyses data that has been interchanged among a set of nodes. The data used for this project was acquired at the RIPE NCC database. The analysis is focused on more specifics prefixes which have been learnt by individual nodes. For each prefix, it has been extracted the following features: length, deep, number of updates and visibility.

Regarding prefixes distribution, not every node learns the same number of prefixes. Those nodes that depict providers give a good perspective about protocol operation. These nodes are composed mostly of more specific prefixes (around 51%). In addition, most of these prefixes have a length greater than /22s, although most of them are /24s. For each root prefix, there is a mean of 8 more specific prefixes. Furthermore, the most typical deep is 8 and most of more specific prefixes have a single level of disaggregation, which indicates that it is being advertised only a little fraction of prefixes for each root prefix.

As for visibility, more specifics represent the least stable prefixes in contrast with the rest of the prefixes. There is a relationship between prefix lengths and visibility, which is reflected in more specifics such as network filters or connectivity loss.

Keywords: Routing protocols, BGP, prefixes, more specifics, longest prefix match, visibility, RIPE NCC, Python, Tableau.

Resumen

Actualmente, Internet es la tecnología de comunicación más importante a nivel global. Se trata de una tecnología veterana, pero su forma de operación no es fácil de conocer debido a su propia naturaleza distribuida. La evolución y el crecimiento de este sistema es evidente en los tiempos que corren. Por ello, resulta imprescindible garantizar la conectividad y alcanzabilidad entre los dispositivos conectados. Esta tarea se lleva a cabo por una serie de protocolos de encaminamiento. BGP es el estándar de facto de enrutamiento EGP (Exterior Gateway Protocol).

El estudio propuesto analiza la información intercambiada por un conjunto de nodos obtenida de la base de datos de RIPE NCC. El análisis se centra en el estudio de los prefijos más específicos aprendidos por cada nodo, de los cuales se extraen las características de longitud, profundidad, número de updates y visibilidad.

En lo que se refiere a la distribución de prefijos, no todos los nodos presentan el mismo número de prefijos aprendidos. Los nodos que representan a los proveedores proveen una buena visión de cómo opera el protocolo. En estos, se puede apreciar que la mayoría de los prefijos corresponden a más específicos (alrededor del 51%). Además, la mayoría de estos prefijos presentan una longitud de prefijo mayor o igual a /22s, siendo mayoritariamente /24s. Por cada prefijo raíz hay una media de 8 prefijos más específicos. Asimismo, la profundidad más típica es de 8 y la mayoría de prefijos más específicos presentan un único nivel de desagregación; lo que indica que es anunciada solo una parte de los prefijos desagregados para cada prefijo raíz.

Para la visibilidad los más específicos presentan el tipo de prefijos menos estable. Existe relación entre la longitud de prefijos utilizada y la visibilidad ya que los más específicos se ven afectados por los filtrados de la red o la pérdida de la conectividad.

Keywords: Protocolos de encaminamiento, BGP, prefijos, más específicos, longest prefix match, visibilidad, RIPE NCC, Python, Tableau.

Agradecimientos

En primer lugar, me gustaría mostrar mi agradecimiento a mi tutor Alberto de la Universidad Carlos III de Madrid. Alberto siempre ha intentado solventar mis dudas dándome las herramientas necesarias para identificar el problema. Gracias a él ha sido posible el desarrollo de este proyecto, intentando redirigirme en la dirección correcta siempre que lo ha considerado oportuno.

También me gustaría agradecer a Geoff Huston jefe de operaciones en APNIC como experto del problema. Estoy muy agradecido sobre sus artículos escritos sobre el protocolo BGP y en concreto el enfocado a los más específicos por haber podido contrastar los resultados.

Por último, expresar mi agradecimiento a mis padres y a mi hermana por su apoyo incondicional y su continua motivación durante mis años de estudio. Este logro no hubiera sido posible sin ellos. Gracias.

INDICE

ABSTRACT	I
RESUMEN	II
AGRADECIMIENTOS	IV
LISTA DE FIGURAS	VI
LISTA DE TABLAS	VIII
1 INTRODUCCIÓN	1
1.1 DESCRIPCIÓN DEL PROBLEMA	1
1.2 OBJETIVOS DEL PROYECTO	2
1.3 ESTRUCTURA DEL PROYECTO	3
2 ESTADO DEL ARTE	4
2.1 INTRODUCCIÓN A BGP	4
2.1.1 PREFIJO IP	4
2.1.2 LONGEST PREFIX MATCH	5
2.1.3 SISTEMAS AUTÓNOMOS	6
2.1.4 BREVE DESCRIPCIÓN DE BGP	7
2.2 INFRAESTRUCTURA DE MEDIDAS RIPE	9
2.3 TABLEAU	13
2.4 TRABAJO RELACIONADO	13
3 ESPECIFICACIONES DEL ANÁLISIS	16
3.1 CLASIFICACIÓN DE PREFIJOS	16
3.2 ADDRESS SPAN	19
3.3 NÚMERO DE UPDATES POR PREFIJO	21
3.4 VISIBILIDAD POR PREFIJO	22
3.5 PROFUNDIDAD POR PREFIJO	23
4 DISEÑO Y EJECUCIÓN DEL SISTEMA DE ANÁLISIS DE RUTAS	24
4.1 DISEÑO DE UN PROYECTO DATA SCIENCE	24
4.2 DISEÑO DEL SOFTWARE	26

4.2.1	ENTORNO DE EJECUCIÓN	26
4.2.2	NIVEL PRINCIPAL DE EJECUCIÓN	29
4.2.3	CÓDIGO A NIVEL DE COLECTOR	29
4.2.4	COMBINACIÓN DE LOS DATOS	39
5	ANÁLISIS DE LOS RESULTADOS	40
5.1	CARACTERÍSTICAS DE LOS DATOS DE ENTRADA	40
5.2	ANÁLISIS DE DISTRIBUCIÓN DE LOS PREFIJOS	40
5.2.1	DISTRIBUCIÓN POR TIPO DE PREFIJO	40
5.2.2	DISTRIBUCIÓN DE LOS MÁS ESPECÍFICOS POR MONITOR	45
5.2.3	PROFUNDIDAD DE LOS MÁS ESPECÍFICOS POR MONITOR	46
5.2.4	LONGITUD DE PREFIJOS POR MONITOR ATENDIENDO AL TIPO	47
5.2.5	ADDRESS SPAN DE LOS MÁS ESPECÍFICOS	48
5.2.6	ANÁLISIS AGREGADO POR LONGITUD DE PREFIJO	49
5.2.7	ANÁLISIS AGREGADO DE LA VARIACIÓN DEL TIPO DE PREFIJO	50
5.3	ANÁLISIS DE VISIBILIDAD	51
5.3.1	ANÁLISIS DE VISIBILIDAD DE PREFIJOS EN PROVEEDORES	52
5.3.2	ANÁLISIS DE VISIBILIDAD POR TIPO DE PREFIJO	56
5.3.3	ANÁLISIS DE LA VISIBILIDAD POR LONGITUD DE PREFIJO ATENDIENDO AL TIPO DE PREFIJO	59
5.4	COMPARATIVA DE LOS RESULTADOS OBTENIDOS CON EL TRABAJO RELACIONADO	65
6	PRESUPUESTO	66
6.1	HORAS INTERNAS	66
6.2	AMORTIZACIONES	66
6.3	GASTOS	67
6.4	COSTES INDIRECTOS	67
6.5	IMPREVISTOS	67
6.6	RESUMEN	68
7	CONCLUSIONES	69
7.1	DIFICULTADES ENCONTRADAS	70
7.2	TRABAJO FUTURO	70
8	REFERENCIAS	71

LISTA DE FIGURAS

FIGURA 1: ESCENARIO BGP (FUENTE: BLOG THOUSAND EYES)	8
FIGURA 2: GLOBAL INTERNET RESOURCES ADMINISTRATION (FUENTE NRO.NET)	9
FIGURA 3: MODELO PARA LA RECOLECCIÓN DE DATOS RIPE (FUENTE: RIPE.NET)	11
FIGURA 4: DETALLE BACK-END RED RIPE (FUENTE: RIPE.NET)	11
FIGURA 5: PREFIX TRIE - DEPENDENCIA ENTRE PREFIJOS	17
FIGURA 6: CLUSTERING POR TIPO DE PREFIJO	17
FIGURA 7: CLUSTERING DE PREFIJOS MÁS ESPECÍFICOS	18
FIGURA 8: CLUSTERING GENERAL DE PREFIJOS	18
FIGURA 9: JERARQUÍA DE PREFIJOS – ESPACIO DE DIRECCIONAMIENTO ANUNCIADO	19
FIGURA 10: ESPACIO DE DIRECCIONAMIENTO ANUNCIADO POR LOS MÁS ESPECÍFICOS	20
FIGURA 11: ESQUEMA DE CÁLCULO VISIBILIDAD POR PREFIJO	22
FIGURA 12: JERARQUÍA - PROFUNDIDAD DE PREFIJOS	23
FIGURA 13: ENTORNO DE TRABAJO - CONEXIÓN SSH (FUENTE BOOLEANWORLD.COM)	26
FIGURA 14: JERARQUÍA DE DIRECTORIOS PARA LOS RESULTADOS DEL ANÁLISIS	28
FIGURA 15: JERARQUÍA DE LOS DATOS DE ENTRADA AL ANÁLISIS	28
FIGURA 16: DIAGRAMA DE FLUJO DEL CÓDIGO A NIVEL DE COLECTOR	30
FIGURA 17: EJEMPLO DE LECTURA DE ARCHIVO DE UPDATES CON BGPDUMP -M	31
FIGURA 18: PASO 1 - CARGA DE LOS UPDATES CORRESPONDIENTES AL EXPERIMENTO	31
FIGURA 19: PASO 4 - CONCATENACIÓN DE RIB DATA CON UPDATES	33
FIGURA 20: PASO 5 - DIVISIÓN DE LOS DATOS SEGÚN VERSIÓN IP	34
FIGURA 21: DIAGRAMA DE FLUJO PARA LA EJECUCIÓN DEL ANÁLISIS DE PREFIJOS	36
FIGURA 22: EXTRACCIÓN DE NÚMERO DE UPDATES Y DE LA VISIBILIDAD	37
FIGURA 23: RESUMEN DE LA TRANSFORMACIÓN DE LOS DATOS PARA EL ANÁLISIS DE MÁS ESPECÍFICOS	38
FIGURA 24: PROCESO DE COMBINACIÓN DE DATOS DE VARIOS COLECTORES	39
FIGURA 25: NÚMERO DE PREFIJOS VISTOS POR MONITOR	40
FIGURA 26: NÚMERO DE PREFIJOS POR PROVEEDOR (ORDENADOS POR EL NÚMERO DE PREFIJOS) – DIAGRAMA DE BARRAS	41
FIGURA 27: NÚMERO DE PREFIJOS POR PROVEEDOR - NORMALIZADO (ORDENADO POR EL NÚMERO DE PREFIJOS)	42
FIGURA 28: NÚMERO DE PREFIJOS PARA CADA TIPO POR PEER (ORDENADO POR EL NÚMERO DE PREFIJOS) – DIAGRAMA DE BARRAS	43
FIGURA 29: DISTRIBUCIÓN NORMALIZADA PARA CADA TIPO DE PREFIJO POR PEERS (ORDENADO SEGÚN EL PORCENTAJE DE MÁS ESPECÍFICOS)	43
FIGURA 30: RATIO MÁS ESPECÍFICOS ENTRE MENOS ESPECÍFICOS POR PROVEEDOR	45
FIGURA 31: NÚMERO DE MÁS ESPECÍFICOS CON UN ÚNICO NIVEL POR PROVEEDOR (NORMALIZADO POR EL NÚMERO DE MÁS ESPECÍFICOS)	45
FIGURA 32: PROFUNDIDAD DE MÁS ESPECÍFICOS POR MONITOR (ORDENADOS SEGÚN EL NÚMERO DE PREFIJOS)	46

FIGURA 33: AGRUPACIÓN DE PROFUNDIDADES POR MONITOR (NORMALIZACIÓN POR NÚMERO DE PREFIJOS).....	46
FIGURA 34: PORCENTAJE DE LONGITUDES DE PREFIJOS POR MONITOR (NORMALIZADO POR EL NÚMERO DE PREFIJOS).....	47
FIGURA 35: ESPACIO DE DIRECCIONAMIENTO DE MÁS ESPECÍFICOS POR PROVEEDOR.....	48
FIGURA 36: NÚMERO DE PREFIJOS POR MÁSCARA - ANÁLISIS AGREGADO	49
FIGURA 37: ANÁLISIS DE LA VARIABILIDAD DEL TIPO DE PREFIJO.....	50
FIGURA 38: VISIBILIDAD DE PREFIJOS POR PROVEEDOR – DIAGRAMA DE BARRAS.....	52
FIGURA 39: PREFIJOS QUE HAN RECIBIDO UPDATES DURANTE EL EXPERIMENTO POR MONITOR (NORMALIZADO POR NÚMERO DE PREFIJOS).....	53
FIGURA 40: VISIBILIDAD PARA PREFIJOS QUE HAN RECIBIDO ALGÚN UPDATE DURANTE EL EXPERIMENTO POR MONITOR (NORMALIZADO POR EL NÚMERO DE PREFIJOS).....	54
FIGURA 41: ANÁLISIS AGREGADO POR TIPO DE LA VISIBILIDAD PARA LOS PREFIJOS QUE HAN RECIBIDO ALGÚN UPDATE DURANTE EL EXPERIMENTO	55
FIGURA 42: ANÁLISIS DE VISIBILIDAD POR TIPO (NORMALIZADO POR EL NÚMERO DE PREFIJOS).....	56
FIGURA 43: PORCENTAJE DE PREFIJOS QUE HAN RECIBIDO ALGÚN UPDATE POR TIPO DURANTE EL EXPERIMENTO (NORMALIZADO POR EL NÚMERO DE PREFIJOS).....	57
FIGURA 44: VISIBILIDAD PARA PREFIJOS QUE HAN RECIBIDO ALGÚN UPDATE DURANTE EL EXPERIMENTO POR TIPO (NORMALIZADO POR EL NÚMERO DE PREFIJOS).....	58
FIGURA 45: ANÁLISIS AGREGADO POR TIPO DE LA VISIBILIDAD PARA LOS PREFIJOS QUE HAN RECIBIDO ALGÚN UPDATE DURANTE EL EXPERIMENTO	59
FIGURA 46: NÚMERO DE PREFIJOS VS PORCENTAJE DE PREFIJOS QUE RECIBEN UPDATES POR LONGITUD DE PREFIJO - MÁS ESPECÍFICOS.....	60
FIGURA 47: ANÁLISIS DE VISIBILIDAD DE LOS PREFIJOS QUE RECIBEN ALGÚN UPDATE POR LONGITUD DE PREFIJO – MÁS ESPECÍFICOS.....	61
FIGURA 48: NÚMERO DE PREFIJOS VS PORCENTAJE DE PREFIJOS QUE RECIBEN UPDATES POR LONGITUD DE PREFIJO – ÚNICOS.....	61
FIGURA 49: ANÁLISIS DE VISIBILIDAD DE LOS PREFIJOS QUE RECIBEN ALGÚN UPDATE POR LONGITUD DE PREFIJO - ÚNICOS.....	62
FIGURA 50: NÚMERO DE PREFIJOS VS PORCENTAJE DE PREFIJOS QUE RECIBEN UPDATES POR LONGITUD DE PREFIJO - MENOS ESPECÍFICOS.....	62
FIGURA 51: ANÁLISIS DE VISIBILIDAD DE LOS PREFIJOS QUE RECIBEN ALGÚN UPDATE POR LONGITUD DE PREFIJO - MENOS ESPECÍFICOS	63
FIGURA 52: MENOS ESPECÍFICOS VISIBLES MENOS QUE EL 70% DEL TIEMPO POR MÁSCARA	63
FIGURA 53: NÚMERO DE MENOS ESPECÍFICOS VISIBLES UNA FRACCIÓN DEL TIEMPO MENOR AL 70% DEL EXPERIMENTO.....	64

LISTA DE TABLAS

TABLA 1: EJEMPLO LONGEST PREFIX MATCH.....	6
TABLA 2: COMPARATIVA CON EL TRABAJO RELACIONADO	15
TABLA 3: COMPARATIVA DE RESULTADOS DEL ANÁLISIS CON EL TRABAJO RELACIONADO.....	65
TABLA 4: HORAS INTERNAS	66
TABLA 5: AMORTIZACIONES.....	66
TABLA 6: GASTOS.....	67
TABLA 7: RESUMEN DEL PRESUPUESTO.....	68

1 Introducción

1.1 Descripción del problema

Actualmente, Internet es la tecnología de comunicación más importante a nivel global. Se trata de una tecnología veterana, pero su forma de operación no es fácil de conocer debido a su propia naturaleza distribuida. La evolución y el crecimiento de este sistema es evidente en los tiempos que corren. Por ello, resulta imprescindible garantizar la conectividad y alcanzabilidad entre los dispositivos conectados. Esta tarea se lleva a cabo por una serie de protocolos de encaminamiento.

Debido a que Internet se compone de centenares de millones de destinos resulta imprescindible el escalado de la red. Los routers se agrupan en regiones conocidas como Sistemas Autónomos. De esta manera se logra simplificar la red. Por lo tanto, existen dos niveles de enrutamiento: uno interno (IGP) y otro externo (EGP). BGP es el estándar de facto de enrutamiento EGP.

Internet está compuesto por unos 58.000 Sistemas Autónomos, estos son variados tanto en número de nodos como en cobertura. BGP, es el protocolo encargado de mantener la conectividad entre estos componentes. La idea principal es que mediante el uso del protocolo los routers que ejecutan el protocolo intercambien una serie de mensajes formando una visión de cómo evoluciona las componentes adyacentes.

Tenemos a nuestra disposición trazas de información de encaminamiento de distintos nodos de la red. Estas trazas permiten analizar el comportamiento de la configuración del protocolo BGP. Sobre estas trazas se han realizado una gran variedad de estudios que han tratado diferentes problemáticas que surgen por la forma de operación del protocolo [1,2].

En este documento se presenta una metodología seguida para el análisis de estas trazas con un propósito particular, el uso de prefijos más específicos en Internet. El estudio propuesto analiza la información intercambiada por un conjunto de nodos. Estos presentan routers frontera de sistemas autónomos. El análisis se centra en el estudio de los prefijos más específicos aprendidos por

cada nodo, de los cuales se extraen las características de longitud, profundidad, número de updates y visibilidad. El desarrollo del cálculo de la visibilidad se combina con el resto de las características extraídas para cada prefijo lo que permite una mayor exploración de los datos debido a que la mayoría de los prefijos presentan una gran estabilidad en términos de visibilidad.

Para el desarrollo de software se ha utilizado el lenguaje de programación Python tanto para la obtención de los datos como para el tratamiento de los mismos. Asimismo, se ha desarrollado un script en Shell para la ejecución del análisis en paralelo para distintos puntos recolectores de datos. Una vez obtenidos los resultados se ha utilizado la herramienta Tableau para la exploración y el análisis de los resultados. Además del desarrollo de la herramienta software, gran parte del trabajo ha consistido en la forma de visualizar los datos junto con el análisis crítico que se hace de los mismos.

1.2 Objetivos del proyecto

En este proyecto, se propone una metodología para el análisis de trazas BGP mediante la implementación de un software fácilmente extensible en el futuro. El análisis se centra en el estudio de la distribución de prefijos para una serie de recolectores conectados a varios nodos en un intervalo de tiempo determinado. Tanto los recolectores como el intervalo de tiempo son parámetros de entrada del software. Permitiendo así, el estudio de una franja temporal determinada.

El desarrollo propuesto analiza los prefijos aprendidos a nivel de monitor (speaker BGP). A su vez, se contempla el análisis conjunto de los datos de forma agregada pudiendo proveer una visión más general sobre el comportamiento de los datos.

El análisis de las trazas se centra en la extracción de información sobre la dependencia entre los prefijos aprendidos por cada nodo. Esto permite la clasificación de los prefijos pudiendo diferenciar los prefijos más específicos del resto. Resulta de gran interés estudiar la comparativa de estos prefijos en contraste con sus correspondientes prefijos agregados o con los prefijos que no presentan ningún tipo de relación. Para ello, se extraen a nivel de prefijo las características de tipo, longitud y profundidad. Esta última solo tiene sentido para

el tipo más específico. También se analiza el espacio de direccionamiento anunciado para los más específicos. Además, se ha definido el concepto de visibilidad como el porcentaje de tiempo que un prefijo se encuentra visible para el intervalo de tiempo en cuestión.

1.3 Estructura del proyecto

Este proyecto se ha organizado de la siguiente manera. La sección 2 introduce el funcionamiento de BGP, presenta la infraestructura de recolección de los datos que han servido de entrada al análisis, así como menciona y justifica la utilización del software Tableau para el análisis de los datos. Asimismo, expone los detalles de un análisis de prefijos más específicos a partir del cual surge la idea del desarrollo del proyecto.

En la sección 3, se caracteriza el análisis presentando las características que van a ser calculadas y analizadas posteriormente apoyándose en gráficas que simplifican la comprensión de las mismas.

En cuanto a la estructura y la organización del software, esta es presentada a lo largo de la sección 4 donde es explicada cada parte del código a alto nivel. Este código genera el fichero de resultados correspondiente a la entrada del software Tableau.

En la sección 5 se presenta el análisis mediante la herramienta Tableau, el cual, se encuentra subdividido en estudios más pequeños con la intención de proporcionar al lector una mejor visión del masajeo sobre los resultados.

La sección 6 presenta un desglose del presupuesto del proyecto en diferentes partidas que justifican los gastos para el desarrollo del mismo.

Finalmente, la sección 7 presenta las conclusiones que se han extraído del desarrollo del proyecto. Así como, las dificultades encontradas y las líneas de trabajo futuro

2 Estado del arte

2.1 Introducción a BGP

2.1.1 Prefijo IP

Para el direccionamiento en Internet es imprescindible la definición de bloques de red. Las direcciones IPv4 consisten en un bloque de 32 bits agrupados en 8 octetos (bytes), que se expresan en notación binaria ej: 11000000.10101000.00000001.00000001.

Para facilitar el manejo por la 'interfaz humana', estos 32 bits suelen expresarse en notación decimal: 192.168.1.1

Ahora bien, toda dirección IP va acompañada de una máscara de subred que indica el modo en el que el administrador de la red asigna su espacio de direccionamiento ej: 192.168.1.1 255.255.255.240.

Estos segundos 32 bits (255.255.255.240) indican que se utilizan 4 bits para subdividir las redes dentro de la red mayor (192.168.1.0 255.255.255.0).

La máscara de subred es particularmente necesaria en el momento de señalar la red correspondiente a cada subred, y que se encuentra referenciada a la tabla de enrutamiento.

Por lo tanto, el prefijo IP es una forma particular de expresar las direcciones de red a partir de identificar solamente la cantidad de bits que se encuentren a uno en la máscara de subred 192.168.1.0/28 [1].

Cuanto más corto es el prefijo, más general es la ruta. Un prefijo más corto cubre un mayor espacio de direccionamiento y, por lo tanto, una mayor parte de Internet. Los prefijos muy largos cubren pocas direcciones y suelen utilizarse para *routing* local cerca de la dirección de destino. En general no es necesaria la distribución de prefijos muy largos a lo largo de Internet, porque en general, es suficiente con prefijos más cortos para dirigir paquetes hacia la dirección de destino [3].

Con la intención de limitar el crecimiento de Internet, la mayoría de los operadores limita la longitud de los prefijos que propagan.

2.1.2 Longest prefix match

Para resolver y determinar la interfaz correspondiente al siguiente salto para un paquete se usa longest prefix match (la entrada más específica). Longest prefix match es un algoritmo utilizado en los routers IP para seleccionar la entrada de la tabla de encaminamiento [4]. Un router utiliza longest prefix match para determinar el interfaz del egress (router para tráfico saliente) y la dirección IP del siguiente router al que enviará el paquete.

La implementación del algoritmo en un router consiste en:

1. El router recibe el paquete.
2. Mientras procesa la cabecera, el router compara la cabecera, bit a bit, con las entradas de la tabla de encaminamiento.

La entrada la tabla de enrutamiento que presenta el mayor número de coincidencias entre la los bits de red y la dirección IP de destino es siempre la mejor opción.

A continuación, se ilustra el funcionamiento del algoritmo con un ejemplo:

- El router recibe un paquete con dirección IP de destino 192.168.1.33.
- La tabla de enrutamiento contiene las siguientes entradas:
 - 192.168.1.32/28
 - 192.168.1.0/24
 - 192.168.0.0/16

Para la realización del ejemplo y la aplicación del algoritmo es más fácil traducir las direcciones y las entradas a binario. En la Tabla mostrada a continuación se plantea el resultado del ejemplo planteado

Dirección (expresada decimalmente)	Dirección en binario
192.168.1.33.	11000000.10101000.00000001.00100001
192.168.1.0/24	11000000.10101000.00000001.00100000 (mayor coincidencia)
192.168.1.0	11000000.10101000.00000001.00000000
192.168.0.0/16	11000000.10101000.00000000.00000000

Tabla 1: Ejemplo longest prefix match

2.1.3 Sistemas Autónomos

La red se encuentra dividida en unos 58.000 componentes los cuales el protocolo BGP denomina Sistemas Autónomos (AS).

Al fin y al cabo, un sistema autónomo no deja de ser una agrupación de routers y links controlados por un operador/institución. Estos, son variados tanto en tamaño como en cobertura. Para la asignación de un número de AS (ASN) la red tiene que ser **multi-homed**, es decir, tiene que estar conectada a distintas redes.

2.1.4 Breve descripción de BGP

Con el crecimiento de internet surge la problemática de que existen demasiados enlaces para poder intercambiar información de cada uno. La solución consiste en simplificar el encaminamiento agregando (en ASes). Y entre ASes se utiliza BGP.

BGP "Border Gateway Protocol" es un protocolo para el enrutamiento entre sistemas autónomos y se trata de un protocolo EGP. A diferencia de los protocolos IGP no está enfocado en la tolerancia de fallos o en la redundancia para el tráfico saliente, aunque permite la detección de bucles. [5].

Por lo tanto, hay que tener claro que la tolerancia de fallos no es un problema con el que el protocolo BGP tenga que lidiar. Existen una serie de análisis que demuestran que el comportamiento de BGP puede amplificar las anomalías como la caída de un enlace [6] o incluso entrar en bucles buscando rutas alternativas que pueden ocasionar una convergencia lenta del protocolo o en su defecto de la red

BGP es una instancia de Bellman-Ford, se trata de un algoritmo de *routing* de vector de distancias. El protocolo consta de una serie de atributos que determinan la ruta óptima, aunque a veces la elección de las rutas, mejor dicho, la entrada al AS puede ser determinada por las políticas de los ASes atravesados. Esto es conocido como Traffic Engineering.

Lo mencionado anteriormente constituye un quebradero de cabeza típico que muchas veces es el resultado de choques entre políticas internas de los ASes que hacen que el protocolo no funcione de forma eficiente en determinadas situaciones. Al fin y al cabo, la Ingeniería de Tráfico es un tema que permite establecer acuerdos entre ASes, que no deja de ser una serie de acuerdos económicos.

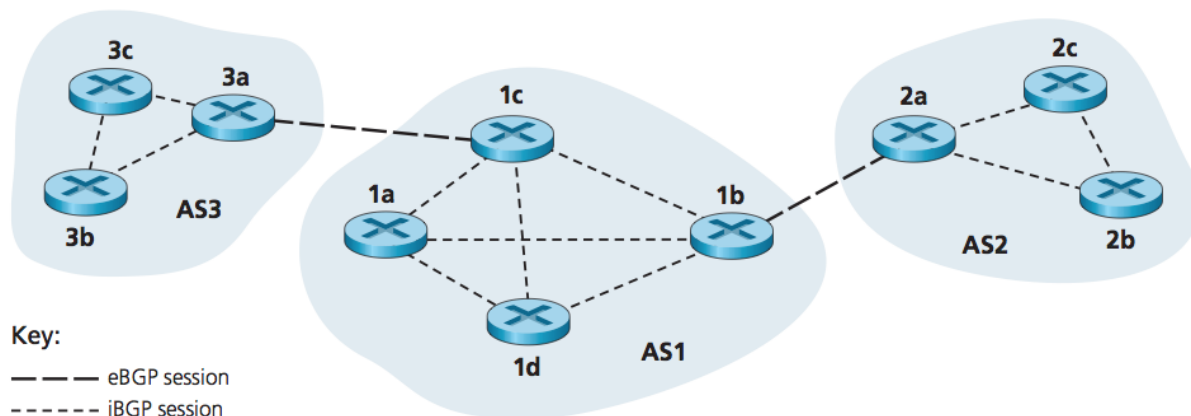


Figura 1: Escenario BGP (fuente: *blog thousand eyes*)

No todos los routers de un AS tienen que correr el protocolo BGP. Los routers que hablan BGP se denominan *speakers* y siempre establecen una sesión entre ellos. Esta puede ser iBGP si es entre *speakers* que pertenecen a un mismo AS o eBGP si se trata de una conexión entre *speakers* que pertenecen a distintos ASes. Los *speakers* conectados mediante sesiones BGP son vecinos inmediatos los unos de otros.

En el escenario presentado en la figura 2 podemos ver como la sesión iBGP constituye una conexión *full-mesh*, es decir, todos con todos. Una vez detallado este escenario se puede entrar más al detalle de cómo funciona el protocolo.

La intención del protocolo consiste en que cada uno de los *speakers* aprenda una topología relativa de la red a la que se encuentra conectado. El funcionamiento básico del protocolo es sencillo: cada *speaker* BGP comunica a todos sus vecinos los prefijos de red aprendidos que alteran su visión local de la red. Cada vez que un vecino informa a un *speaker* sobre un prefijo IP, este compara la nueva alcanzabilidad con los prefijos almacenados provenientes de anuncios recibidos por otros vecinos previamente. Si esta información corresponde un mejor camino hacia el prefijo entonces el *speaker* almacena este nuevo anuncio como preferido y guarda el siguiente salto hacia él. A su vez, anuncia este prefijo a sus vecinos inmediatos, lo que implica que se anuncia a sí mismo como siguiente salto. También existe un proceso de borrado de prefijos, donde un *speaker* declara que no va a ser alcanzable el prefijo a través de una ruta determinada. Cuando un *speaker* recibe el mensaje de borrado lo almacena

y comprueba si el vecino del que lo ha recibido corresponde a la ruta preferida para alcanzar dicho prefijo. De ser así el speaker en cuestión analiza que prefijo almacenado representa la mejor ruta para alcanzar el destino en cuestión. Si existe una ruta alternativa copia la misma a la tabla de *forwarding* y lo anuncia a sus vecinos. En caso contrario, envía el mensaje de borrado a todos sus vecinos.

2.2 Infraestructura de medidas RIPE

RIPE es el Registro de Internet para Europa, Oeste Medio y parte de Asia central, se encarga de la localización y del registro de bloques de Internet para proveedores de servicio de internet (ISPs) y otras organizaciones.



Figura 2: Global Internet Resources Administration (Fuente nro.net)

RIPE NCC recolecta y almacena datos de routing de Internet en varios emplazamientos mediante RIPE NCC Routing Information Service (RIS), desde alrededor de 1999.

Hay 12 colectores activos actualmente: un colector multihop en Amsterdam y 11 colectores locales en Internet Exchange Points (IXPs). La recolección de datos está basada en Quagga y es almacenada en formato MRT. Este formato se encuentra descrito en la RFC6395 [8]. En la sección 4.4 podemos encontrar la versión correspondiente a las trazas BGP4MP.

En cuanto al **back-end**, en sus inicios la infraestructura estaba compuesta por **MySQL** junto con **Zebra**, pero **MySQL** no era capaz de satisfacer el requerimiento del tráfico de datos entrante []. Desde entonces, **Zebra** fue remplazado por Quagga (una nueva versión de **Zebra**) y en cuanto a **MySQL** se adoptó una arquitectura maestro esclavo para intentar satisfacer la gestión del tráfico entrante.

Pero a medida que iba pasando el tiempo el tráfico siguió creciendo y con ello el tamaño de las tablas de enrutamiento BGP. Finalmente, la base de datos fue sustituida por un servidor **Hadoop** tanto para almacenamiento como para procesar la información.

La nueva arquitectura **Hadoop** permite ahora un procesamiento horizontal escalable y un almacenamiento *cluster*. También se ve reflejado en una mejora de la importación de datos, procesado y agregación de datos históricos. La base de datos es ahora gestionada atendiendo peticiones de RIPEstat.

Mientras el problema de almacenamiento y procesamiento ha sido solucionado mediante la arquitectura **Hadoop**, **Quagga** presenta problema de cuello de botella.

La actual versión de **Quagga** es *single-thread* luego no es escalable sobre CPUs con varios núcleos. Esto se traduce en que bloquea los *updates* durante el proceso de volcado de datos y requiere que el volcado de datos termine antes de que expire el temporizador para que la sesión BGP no expire.

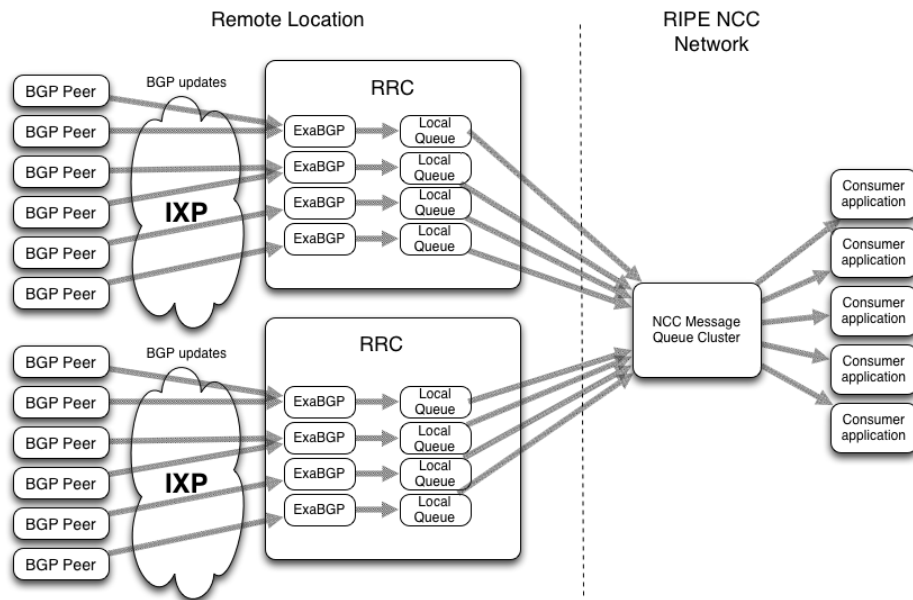


Figura 3: Modelo para la recolección de datos RIPE (fuente: RIPE.net)

Cada colector presenta i conexiones con una serie de *peers*. Cada uno de estos *peers* representan un *speaker* BGP, a partir de ahora vamos a denominar a estos monitores. Los monitores mandan todos los *updates* que mandarían a un vecino, es decir, el colector establece una sesión BGP con cada uno de estos monitores. Existen tres tipos de *updates*: *Advertisement*, *Withdrawn* y *STATE*.

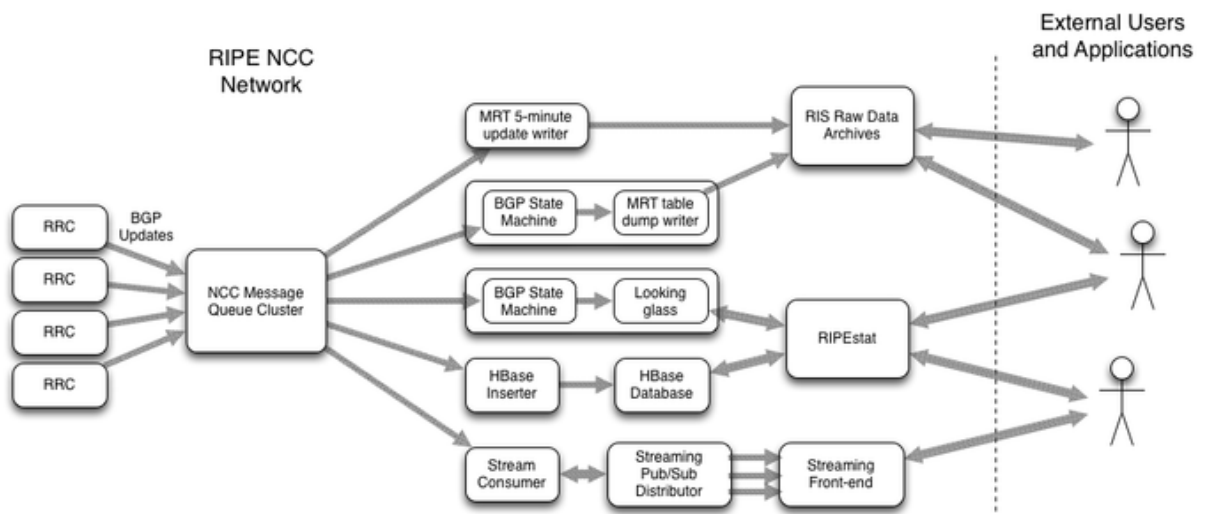


Figura 4: Detalle Back-end red RIPE (fuente: RIPE.net)

Los datos utilizados para el análisis han sido descargados de RIS Raw Data Archives [10].

Para cada colector podemos diferenciar dos conjuntos de datos:

- Todos los paquetes BGP, creados a partir del comando Zebra “*dump bgp all ...*” cuyos ficheros comienzan por *updates* y son creados **cada 5 minutos**.
- La tabla de enrutamiento BGP completa, creada mediante el comando Zebra “*dump bgp routes-mrt ...*”. Estos ficheros son generados **cada 8 horas** y comienzan por *bview*.

Para el volcado de las trazas se ha utilizado la herramienta PyBGPDump [11], la cuál combina la librería de C libbgpdump con la facilidad de Python para parsear mensajes BGP. Para ello, se hace uso del comando *-m*, lo que nos permite volcar una entrada por *timestamp*.

Existen otros sistemas de recolección de trazas BGP como Route Views que es un proyecto de la Universidad de Oregon [12]. La elección de RIPE se basa en que el back-end provee un mejor rendimiento en términos de procesamiento y volcado de los datos. En Route Views existen dos formatos de recolección [13] **Zebra** y **Cisco**. Es en estos últimos donde el procesado es relativamente lento y, por ello el *timestamp* de las trazas puede verse afectado.

2.3 Tableau

La elección de Tableau en la realización de este proyecto se ha debido a que se trata de una herramienta de análisis de datos gratuita para cualquier estudiante.

La herramienta se fundó sobre la idea de que el análisis de datos y los informes subsiguientes no deben de ser actividades aisladas, sino que deben de integrarse en un proceso único de análisis visual: uno que permita al usuario ver rápidamente patrones de los datos y cambiar las vistas al instante para seguir la línea de su pensamiento [14]. Tableau combina exploración de datos y visualización de estos en una aplicación fácil de usar que permite un aprendizaje rápido para cualquier persona familiarizado con Excel o con lenguajes de programación. A su vez permite compartir las visualizaciones y los datos de manera segura a través de servidores.



2.4 Trabajo relacionado

Existe un artículo relacionado con el estudio de los más específicos expuesto por Geoff Huston en junio de 2017 "More specifics in BGP" [15]. El enfoque que realiza el autor está orientado en otra dirección que el presentado en este documento, pero a su vez ha servido de motivación para centrarse en un estudio de estos prefijos.

El objetivo principal del estudio consiste en determinar si los prefijos más específicos presentan una imposición para cualquier router o bien aseguran al usuario el mejor servicio con el mínimo coste. Para ello, se examinan los anuncios de más específicos con profundo detalle, centrándose en el impacto de los más específicos en el crecimiento de los sistemas de rutas y la dinámica de los updates correspondientes a estos prefijos.

Se utiliza una clasificación de los más específicos atendiendo a la comparación de atributos BGP: AS Path y origen frente a su agregado. Ya que no todos estos prefijos tienen el mismo propósito:

1. Hole Punching: distinto origen y distinto AS Path que su agregado
2. Ingeniería de tráfico: mismo origen y distinto AS Path que su agregado
3. Overlay: mismo Origen y mismo AS Path que su agregado

Su análisis utiliza un histórico de **diez años** de un **Sistema Autónomo**. Este Sistema Autónomo se corresponde con un **stub** en la frontera de una red. Por lo tanto, podrían diferir los resultados dados por este análisis de los obtenidos por otro realizado sobre otro stub. El análisis es eBGP luego no tiene en cuenta rutas BGP interiores. El artículo realiza el análisis para las dos versiones del protocolo IP pero al igual que en el estudio presentado en nuestro documento se va a atender a la versión 4.

En un primer, caracteriza la evolución temporal de los prefijos más específicos con respecto al número total de prefijos vistos por el sistema autónomo analizado. Seguido analiza la evolución sobre el porcentaje que los más específicos suponen. Para justificar dicha evolución, analiza las mismas características para los tres tipos definidos.

En segundo lugar, realiza un análisis de la evolución del espacio de direccionamiento que representan los más específicos. También realiza el análisis para los tipos de más específicos.

Hasta aquí, consigue caracterizar, como los más específicos afectan al número de entradas del tamaño de las tablas asociadas a los routers.

En el tercer apartado, analiza los updates correspondientes a los más específicos frente a los recibidos por los prefijos raíz donde determina que estos son ligeramente superiores. El propósito de este apartado es la caracterización de los más específicos en términos de capacidad de procesamiento de los routers. Asimismo, caracteriza en términos de ruido los tres tipos de más específicos determinando la influencia de cada uno.

Finalmente, presenta un análisis sobre la eliminación del tercer tipo de prefijos (overlays). Ya que existen varios estudios que demuestran el impacto negativo que estos suponen.

Se ha realizado la siguiente tabla que presenta una comparativa simple en la diferencia entre el estudio presentado en el presente documento y este otro análisis relacionado sobre los más específicos.

More specifics in BGP	Trabajo realizado
Un único AS	Monitores de distintos ASes
Dimensión temporal	Dimensión espacial
Evolución	Detalle
Estabilidad	Visibilidad

Tabla 2: Comparativa con el trabajo relacionado

3 Especificaciones del análisis

El análisis está enfocado a nivel de monitor, aunque los resultados son extraídos a nivel de prefijo.

Se trata de un análisis que pretende extraer con detalle las siguientes características a nivel de prefijo:

- Clasificación de prefijos.
- Número de updates por prefijo.
- Visibilidad por prefijo.
- Profundidad del prefijo.
- Address Span

A continuación, se explicará con más detalle cada una de las características anteriormente.

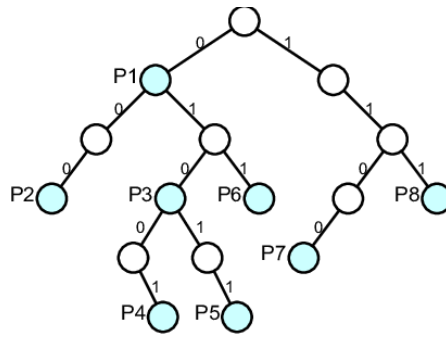
3.1 Clasificación de prefijos

Para cada monitor interesa saber la posición que cada prefijo. Un prefijo puede corresponder a una subred de otro prefijo ya aprendido. También pueden existir prefijos sin relación con los otros prefijos de la tabla de enrutamiento.

Para el desarrollo de esta investigación es imprescindible conocer la relación entre prefijos, que como se explicó previamente, para ello es necesario utilizar un código en el que se implemente una búsqueda en árbol eficientemente.

A continuación, se muestra una jerarquía en árbol que va a ser la utilizada para hacer el *clustering* por tipo de prefijos.

0*	P1
000*	P2
010*	P3
01001*	P4
01011*	P5
011*	P6
1100*	P7
111*	P8



(a)

(b)

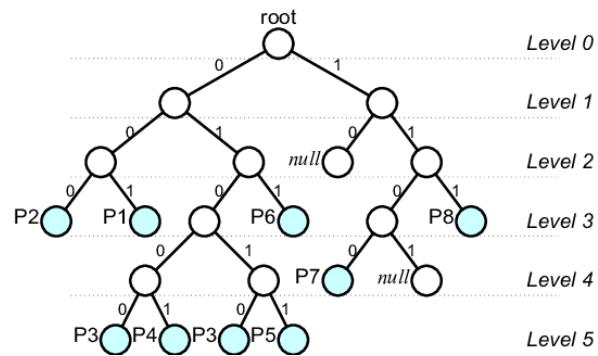


Figura 5: Prefix Trie - Dependencia entre prefijos

Este *clustering* va a consistir en tres niveles principales:

1. **Menos específicos** de los que cuelgan otros prefijos más cortos contenidos en la red representada por dicho prefijo (o prefijos raíz).
2. **Más específicos** son los prefijos colgantes del prefijo menos específico.
3. **Únicos** son los que no presentan ninguna dependencia con otros prefijos.

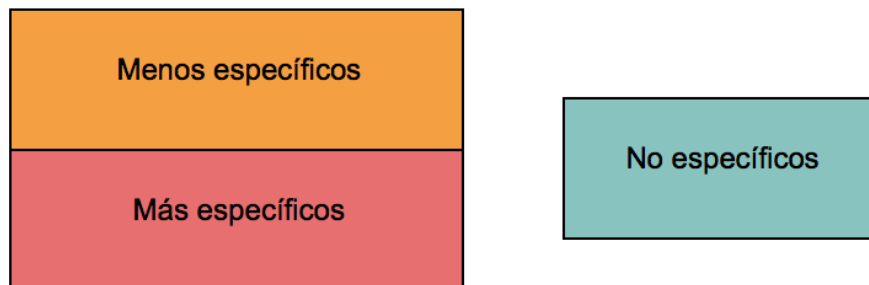


Figura 6: Clustering por tipo de prefijo

Posteriormente se define una clasificación que atiende a mayor nivel de detalle clasificando los más específicos en:

- 1. **Top** son los menos específicos.
- 2. **Más específicos de otros más específicos**, pero sin tener en cuenta los Top.
- 3. **Nivel único** es el caso particular de cuando solo existe un nivel por debajo de los más específicos.

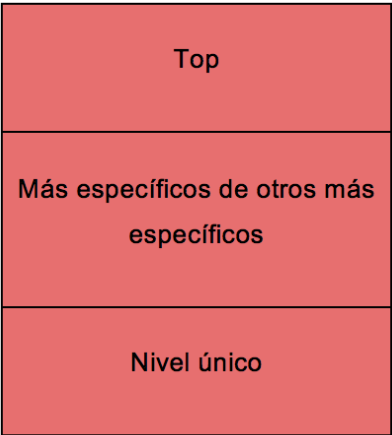


Figura 7: Clustering de prefijos más específicos

En vez de generar dos campos de tipo de prefijo. El código se centra en la clasificación de prefijos al más bajo nivel como se muestra en la imagen mostrada a continuación.

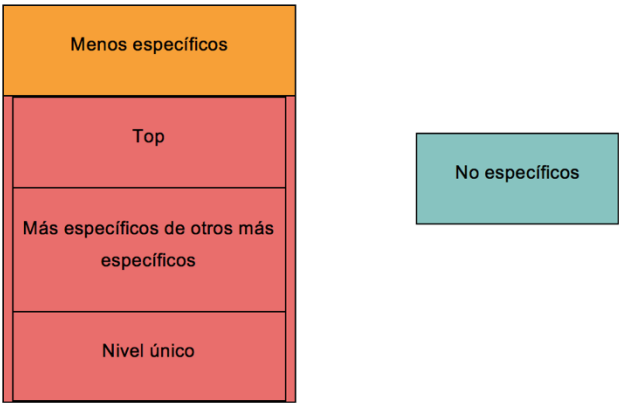


Figura 8: Clustering general de prefijos

3.2 Address Span

El *Address Span* es el espacio de direccionamiento anunciado. Su análisis, solo tiene sentido a nivel de **monitor** y para ello es imprescindible conocer la dependencia entre prefijos que se ha sido expuesta en el apartado 4.1.

Como se ha especificado previamente tanto los prefijos menos específicos como los no específicos están dotados en un solo nivel. Por otro, lado los más específicos están organizados en torno a varios niveles por lo tanto exigen una mayor comprensión de cómo se traduce esto en el espacio de direccionamiento anunciado.

En primer lugar, para un prefijo se define el espacio de direccionamiento anunciado como:

$$\text{Address Span per prefix} = 2^{\text{max prefix length} - \text{prefix length}}$$

Donde *max prefix length* viene definido por la versión de IP utilizada: para IPv4 sería 32 y para IPv6 128. La longitud del prefijo se corresponde con la máscara del mismo expresada en forma decimal. Por ejemplo, para el prefijo IPv4 198.166.123.0/24 el espacio de direccionamiento anunciado sería de 256 direcciones.

A partir de ahora, para la explicación del cálculo del espacio de direccionamiento anunciado se van a utilizar los siguientes acrónimos:

- *Address Span* $\approx AS$
- *Address Span per prefix* $\approx AS_{pref}$

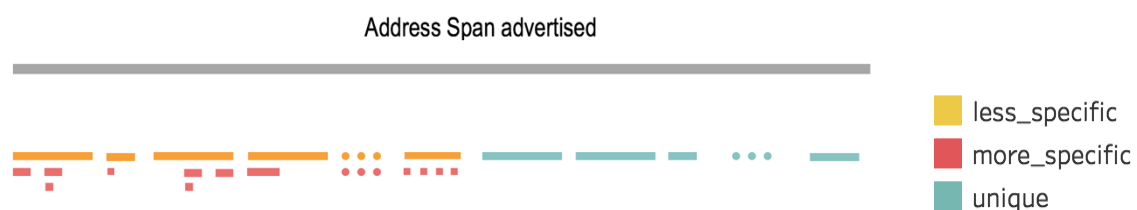


Figura 9: Jerarquía de prefijos – Espacio de direccionamiento anunciado

La figura 9 muestra la jerarquía de prefijos y sirve como base para la explicación del cálculo del direccionamiento anunciado.

En primer lugar, puede apreciarse el cálculo para el espacio de direccionamiento anunciado para los prefijos de un monitor. Para ello es necesario introducir el espacio de direccionamiento anunciado por los prefijos **menos específicos** y el de los **no específicos** que es calculado de la misma manera:

$$AS = \sum_0^N AS_{pref}(i)$$

Siendo N el número de prefijos por tipo.

Por lo que para el cálculo del espacio de direccionamiento anunciado no hay que atender al espacio anunciado por los más específicos (ya que este se tiene en cuenta en los menos específicos), y sería equivalente a:

$$Address\ Span\ Advertised = AS(Uniques) + AS(less\ specifics)$$

Por otro lado, para el espacio de direccionamiento de los más específicos hay que atender a su clasificación a bajo nivel de este tipo de prefijos. Ahora, olvidándose del resto de tipos tendríamos el grafo mostrado en la figura 10.

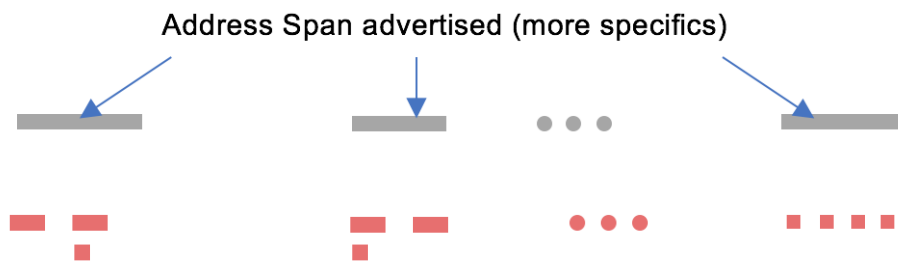


Figura 10: Espacio de direccionamiento anunciado por los más específicos

Contemplando este escenario se puede explicar el cálculo del direccionamiento anunciado por los más específicos atendiendo a la siguiente analogía:

- *Más específicos de otros más específicos \approx Menos específicos*
- *Prefijos Top \approx Menos específicos*
- *Single level \approx no específicos*

Es decir, que el espacio de direccionamiento anunciado por los más específicos se calcula como:

$$\text{Address Span Advertised (more specifics)} = AS(\text{Top}) + AS(\text{single level})$$

Mediante esta característica se extrae más información acerca de los datos ya que la distribución de prefijos por tipo para cada monitor podría parecerse. Es decir, un espacio de direccionamiento similar a lo largo de varios monitores podría significar la visibilidad de los mismos más específicos.

3.3 Número de updates por prefijo

El número de *updates* por prefijo consiste en la cuenta de mensajes de *updates* recibidos por prefijo, es decir, número de borrados + número de anuncios. Esta característica permite identificar los que prefijos reciben updates durante el experimento.

Ya que se tienen en cuenta tanto los *updates* junto con las entradas de la RIB para cada colector es necesario discriminar en la cuenta aquellas entradas de tipo B que son las que representan los prefijos presentes en la RIB.

La clasificación de prefijo es sencilla:

1. **Prefijo no afectado:** Número de *updates* = 0
2. **Prefijo con algún update:** Número de *updates* \neq 0

3.4 Visibilidad por prefijo

La visibilidad del prefijo se define como el porcentaje de tiempo que el prefijo se encuentra visible durante el experimento entre la duración del experimento. Lo que puede expresarse como:

$$\text{Visibilidad} = \frac{\text{tiempo que el prefijo es visible durante el experimento}}{\text{tiempo del experimento}} \times 100$$

Para el cálculo del tiempo en el que el prefijo es visible es necesario tener en cuenta el tipo de cada entrada de datos. El prefijo puede estar visible desde el comienzo del experimento, es decir, se encuentra en la RIB (o es de tipo B) o bien puede aparecer un prefijo a lo largo del experimento siendo anunciado o borrado durante el mismo (tipo A o W respectivamente).

Por lo tanto, hay que segmentar los datos por prefijo y posteriormente ordenar estas por *timestamp*. Una vez hecho esto el escenario de análisis de visibilidad de cada prefijo sería el presentado a continuación.

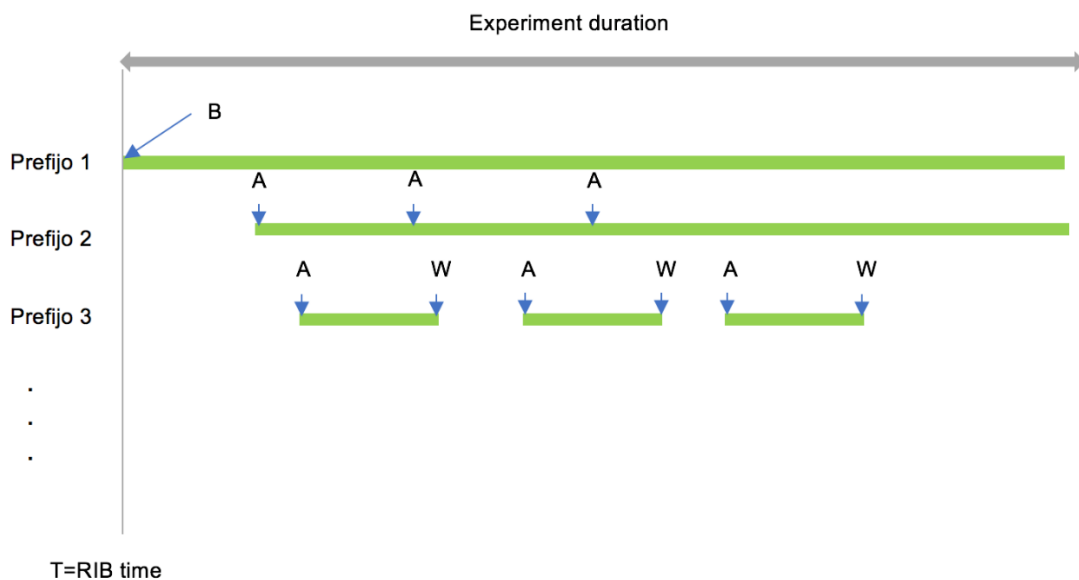


Figura 11: Esquema de cálculo visibilidad por prefijo

3.5 Profundidad por prefijo

La profundidad de un prefijo se define como la diferencia de la longitud de un prefijo más específico con respecto al prefijo menos específico del mismo rango de direcciones. Con lo que podría definirse la siguiente expresión:

$$\text{Profundidad}(\text{pref}) = \text{pref_length}(\text{pref}) - \text{pref_length}(\text{least_specific}(\text{pref}))$$

Siendo *pref* un prefijo más específico. Este cálculo solo tiene sentido para los más específicos luego al resto de prefijos se les asignará una profundidad igual a 0.

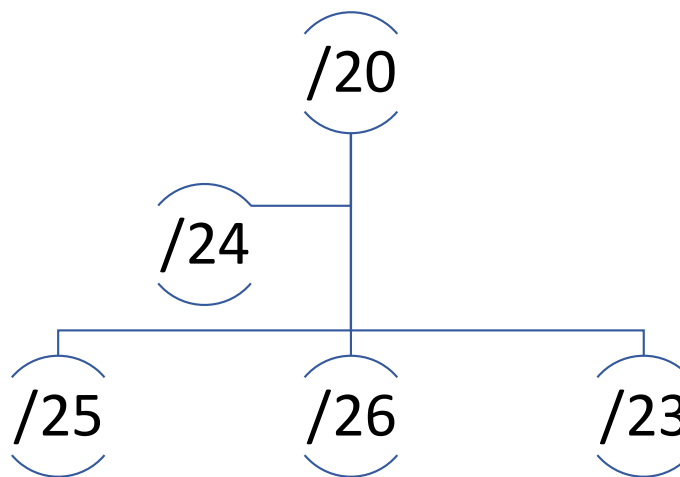


Figura 12: Jerarquía - profundidad de prefijos

Suponiendo que el /20 corresponde al prefijo menos específico para un rango de direcciones dado. El resultado del cálculo para cada uno de los más específicos que cuelgan de este prefijo sería el siguiente:

- $\text{Profundidad}(/24) = 24 - 20 = 4$
- $\text{Profundidad}(/25) = 25 - 20 = 5$
- $\text{Profundidad}(/26) = 26 - 20 = 6$
- $\text{Profundidad}(/23) = 23 - 20 = 3$

4 Diseño y ejecución del sistema de análisis de rutas

4.1 Diseño de un proyecto data science

A continuación, se presenta la estructura del código encargado de obtener el fichero de resultados que constituye la entrada a la herramienta de análisis de datos *Tableau*.

Para el desarrollo de la herramienta software se han seguido en la medida de lo posible las recomendaciones de **PLOS computational biology** expuestas en un artículo sobre una serie de buenas prácticas para proyectos de computo de datos o *data science* [16].

Las recomendaciones que se han considerado más importantes y que, por lo tanto, se han seguido estrictamente se enumeran a continuación. Se sigue la misma organización que en el artículo previamente citado.

1. Manejo de datos
 - a. Dividir el análisis en pasos.
 - b. De cada paso se obtiene un fichero de resultados que corresponde la entrada del siguiente.
2. Software
 - a. Cada fichero de código cuenta con una breve y concisa definición justo al inicio del mismo.
 - b. Descomponer cada fichero en funciones.
 - c. Ser estricto con las duplicidades en el código.
 - d. Utilizar librerías de código fiables.
 - e. Testear librerías antes de confiar en su correcto funcionamiento.
 - f. Asignar a las funciones y a las variables nombres que den un significado a lo que representan.
 - g. No utilizar los comentarios para controlar el comportamiento/flujo del programa.
 - h. Testear las funciones con conjuntos de datos simples, de los cuales se puede deducir el resultado de las funciones y así, depurar las mismas con cierta facilidad.

3. Organización del código

- a. Crear carpetas asociadas al proyecto y a cada paso para depositar los resultados.
- b. Crear carpetas *doc* dentro de la carpeta del proyecto para almacenar los documentos asociados al mismo.
- c. Depositar el código fuente dentro de la carpeta *src*.
- d. Nombrar cada fichero de manera que se entienda su función.

4. Colaboración

- a. Crear una lista enumerando las tareas 'por hacer' para el proyecto.
- b. Hacer que el proyecto sea citable.

5. Realizar seguimiento del código

- a. Usar un control de versiones.
- b. Los cambios han de ser pequeños.

Todas estas recomendaciones permiten que el código realizado sea sostenible y además fácilmente interpretable para la colaboración con diferentes investigadores que puedan verse atraídos por la temática del proyecto. Además, una utilización adecuada del control de versiones supone identificar rápidamente donde se han producido cambios en un determinado momento.

Por ello, es de suma importancia que dichos cambios sean pequeños y que tanto las ramas como los *submits* permitan identificar la tarea realizada.

El proyecto se encuentra ubicado en un repositorio de **GitHub** [17]. También consta de un respaldo en Google Drive.

4.2Diseño del software

4.2.1Entorno de ejecución

Para la ejecución del código se ha utilizado la máquina **zompopo** de la que dispone el departamento de investigación del área telemática de la Universidad Carlos III de Madrid. Esto ha permitido la ejecución del código en paralelo a nivel de colector. Por lo tanto, todo el código está implementado a este nivel.



Figura 13: Entorno de trabajo - Conexión ssh (fuente booleanworld.com)

Debido a la capacidad de esta máquina, tanto en lo que se refiere en memoria como en número de procesadores, ha permitido lanzar el análisis para varios colectores al mismo tiempo.

Dentro del diseño del código pueden separarse en tres niveles:

1. El primer nivel se encarga de lanzar los pasos en los que se organiza el proyecto en paralelo, como se explica posteriormente.
2. El segundo nivel, está representado por la implementación de un código **a nivel de colector**.

Además de estos niveles se ha creado lo que se conoce en los proyectos de investigación como un manifiesto, se trata del fichero ***experiment_manifest.py***.

Este fichero es de suma importancia para la reutilización de código ya que en él se definen las siguientes características del experimento:

- Descripción del experimento.
- Fecha inicial con formato *aaaammdd.hhmm*.
- Fecha final con formato *aaaammdd.hhmm*.
- Directorio principal de resultados.
- Formato de salida de los ficheros intermedios.

El formato de salida solo puede ser csv. En un principio, se pensó que se podría elegir entre csv o xlsx. Pero durante una serie de pruebas se comprobó que la extensión xlsx presenta una limitación de 1.048.576 filas lo que no es suficiente para el almacenamiento de los datos en crudo para el desarrollo de este proyecto. Aunque, el proyecto está implementado de tal manera que se podría incluir otro formato con facilidad.

Los experimentos son definidos mediante un diccionario de Python con lo que es sumamente fácil realizar un análisis que comprenda otra ventana de tiempo. El único requisito es que el directorio del que cuelga el directorio de resultados exista.

La función utilizada en todos los pasos ejecutados a nivel de colector consiste en la carga de los argumentos de entrada, es decir, colector y nombre del experimento. También es realizado un chequeo de si el nombre del experimento existe.

También cuenta con una serie de funciones que realizan operaciones con las fechas de inicio y fin.

Por último, contiene otra serie de funciones las cuales se encargan de gestionar los directorios tanto de los pasos como de los datos descargados para cada colector. En el caso de no existir los crearía otra vez con el requisito de que previamente existiese el directorio del que cuelgan.

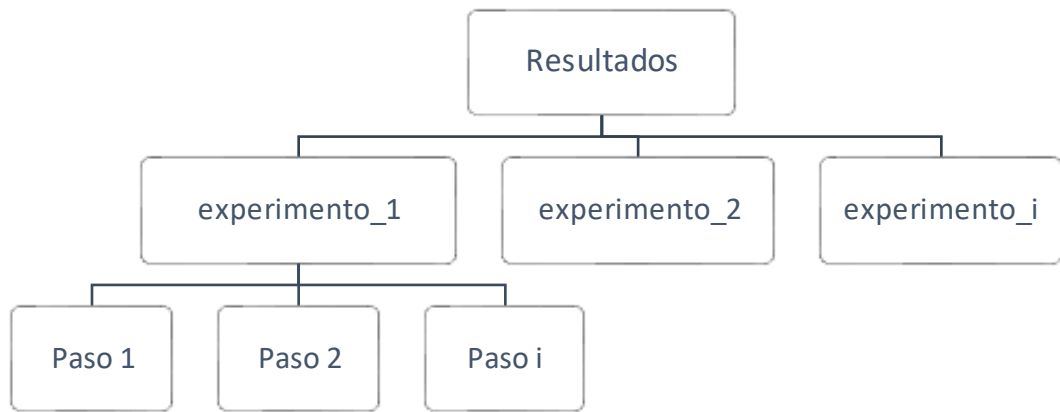


Figura 14: Jerarquía de directorios para los resultados del análisis

La figura anterior resume la jerarquía de los directorios para cada experimento. Del directorio de resultados cuelgan una carpeta correspondiente a cada paso del análisis. Dentro de cada una de estas carpetas se guarda un documento por cada colector con el formato `collector_fromdate-todate.csv` por ejemplo: `rrc07_20180108.0000-20180108.0800.csv`. A partir del quinto paso existirán dos directorios adicionales para separar los resultados de IPv4 de los de IPv6. Finalmente, el último paso solo dispondrá de un fichero que combinará toda la información en uno único.

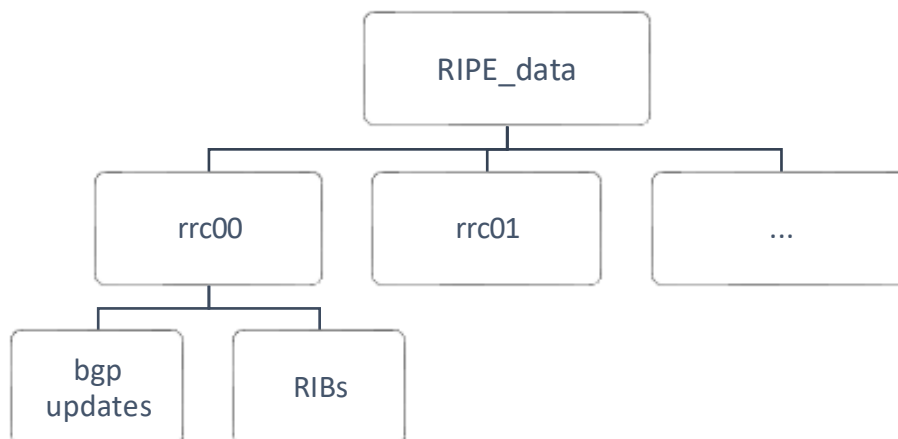


Figura 15: Jerarquía de los datos de entrada al análisis

Para la descarga de los datos la jerarquía no atiende al experimento en cuestión, sino que está organizada a nivel de colector. Para cada colector se almacenará tanto los *updates* de cada uno de los experimentos como la RIB correspondiente al principio del experimento con la nomenclatura que da RIPE NCC por ejemplo: *bview.20180108.0000*, *updates.20180108.0000* o *updates.20180108.0005*.

4.2.2 Nivel principal de ejecución

El código *Shell*, el cuál se denomina ***experiment.sh***, mencionado previamente lanza los códigos Python en los que está organizado el proyecto para una lista de colectores, bien especificados en el mismo código o bien como parámetro de entrada. Otro parámetro de entrada es el nombre del experimento.

El código lanza los pasos en orden indicado a través de terminales independientes, así se consigue la ejecución en paralelo en varios procesadores. La característica principal es que no se lanza el siguiente paso hasta que la tarea que está actualmente en ejecución no haya terminado para todos los colectores.

Cada 5 segundos se lanza el paso en cuestión para un colector. Se establece un máximo de 15 procesos para respetar el uso compartido de la máquina.

4.2.3 Código a nivel de colector

El código está organizado para ejecutar el análisis a nivel de colector. En el siguiente diagrama de flujo se puede observar el análisis completo para un colector.

Por lo tanto, según lo descrito previamente cada uno de los pasos genera un fichero intermedio por colector excepto en el quinto paso que como su nombre indica separa los datos ya que para el análisis de más específicos resulta imprescindible separar los datos en IPv4 e IPv6. El análisis es el mismo, pero no tiene sentido realizarlo con dichos datos mezclados.

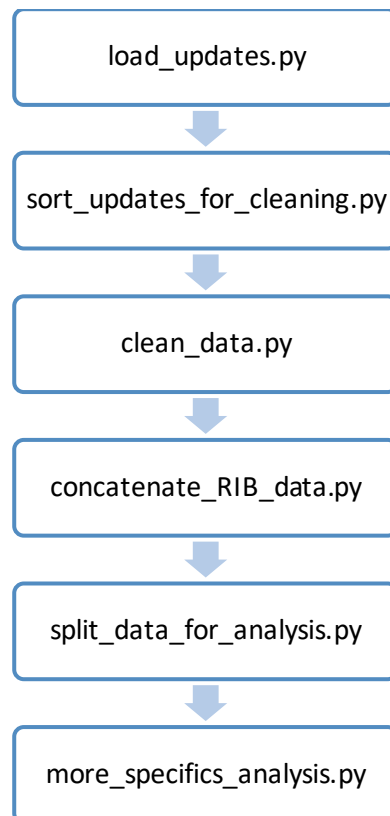


Figura 16: Diagrama de flujo del código a nivel de colector

A continuación, se van a definir brevemente las tareas que realizan cada uno de los pasos definidos en la figura anterior.

4.2.3.1 Paso 1: *load_updates.py*

El primer paso consiste en la descarga y el posterior volcado al tipo de fichero con el que vamos a trabajar en cada paso correspondientes los *updates* comprendidos en la ventana de tiempo del experimento. El código permite la introducción de dos fechas cualesquiera y funcionará correctamente siempre que existan los archivos.

Ha sido necesaria la implementación de un código que permita descargar y descomprimir cada fichero que contiene las trazas BGP (cada 5 minutos). Para contemplar los días de un mes para un año dado se ha utilizado la librería de Python *monthrange*.

Para la lectura de los ficheros se ha utilizado la librería libBGPDump junto con el comando -m. Esto, como se explicó previamente permite separar cada traza en líneas independientes.

```
BGP4MP|1515974695|A|2001:67c:24e4:1::1|57381|2a06:e881:1400::/47|57381 42708 1299 6939 202540 202540 202540|IGP|2001:67c:24e4:1::1|0|0|INAG||
BGP4MP|1515974695|A|2001:67c:24e4:1::1|57381|2620:11b:4004::/48|57381 6939 701 19210|IGP|2001:67c:24e4:1::1|0|0|INAG||
BGP4MP|1515974695|W|2001:67c:24e4:1::1|57381|2a0b:4340:1d::/48
BGP4MP|1515974695|A|2001:67c:24e4:1::1|57381|2407:1300::/32|57381 42708 6939 4788 132850|IGP|2001:67c:24e4:1::1|0|0|INAG||
BGP4MP|1515974695|A|2001:67c:24e4:1::1|57381|2a00:11c0:8::/48|57381 50304 1299 42473|IGP|2001:67c:24e4:1::1|0|0|INAG||
BGP4MP|1515974695|W|2a00:1c10:10::8|50300|2a0a:6840::/29
```

Figura 17: Ejemplo de lectura de archivo de updates con bgpdump -m

Aprovechando la funcionalidad previamente descrita, se obtiene una lista en la que cada elemento presenta una traza. Posteriormente se filtran los datos que queremos obtener en función del tipo de la entrada (STATE, A o W) ya que no es igual el número de datos asociados a cada uno.

Una vez que se ha conseguido una estructura uniforme se vuelca en un archivo del tipo .csv. El fichero contiene para cada entrada los siguientes datos: **timestamp**, **monitor**, **prefijo**, **tipo** y **origen**.

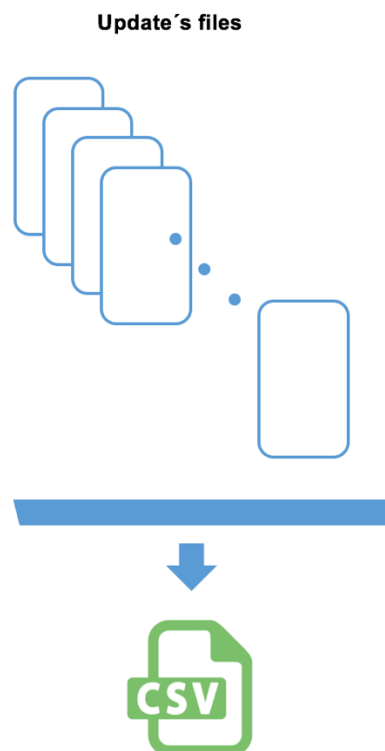


Figura 18: Paso 1 - Carga de los updates correspondientes al experimento

4.2.3.2 Paso 2: *sort_updates_for_cleaning.py*

A partir de aquí es necesario tratar los datos a nivel de monitor por lo que este paso carga el fichero generado por el paso anterior y lo vuelca a un *dataframe*. Este *dataframe* es ordenado por el valor de dos campos: *monitor* y *timestamp*.

Se ha decidido ordenar los datos de esta manera para evitar la generación de ficheros por monitor. Se trata de una manera más eficiente de tratar los datos a nivel de monitor.

4.2.3.3 Paso 3: *clean_data.py*

El criterio de limpieza de las trazas es el siguiente: ‘Si se recibe un *update* de tipo STATE, las trazas recibidas 5 minutos antes o después tienen que ser eliminadas’. El tiempo óptimo de 5 minutos queda demostrado mediante una serie de experimentos realizados [7]. Ha sido necesario convertir el *timestamp* a minutos para realizar la limpieza.

La limpieza de los datos se efectúa a nivel de monitor como se ha indicado previamente en el paso anterior.

El diseño del código consiste en identificar las entradas con el tipo STATE y posteriormente generar una ventana que comprende los mensajes afectados para cada monitor. Finalmente, estas entradas son eliminadas de la estructura de datos y los datos resultantes son salvados en un nuevo archivo.

4.2.3.4 Paso 4: *concatenate_RIB_data.py*

Una vez que las trazas correspondientes a los *updates* han sido limpiadas ya solo quedan los *updates* que se corresponden a anuncios o borrados por prefijos. Por lo tanto, es el momento correcto para cargar los datos correspondientes a la RIB de cada monitor.

Estos ficheros como se ha comentado previamente son grabados cada 8 horas en la base de datos de RIPE.

El código se encarga de cargar la RIB correspondiente a la fecha inicial del experimento; esta ha de ser múltiplo de 8 y para ello se realizan las comprobaciones necesarias sobre esta fecha.

En este apartado se ha podido reutilizar en gran medida el código que se encarga de cargar los *updates* solo que los bucles para recorrer la ventana de tiempo han sido eliminados.

Una vez descargados y descomprimidos los datos, estos son volcados a un *dataframe*. Posteriormente, se cargan los *updates* tras la limpieza correspondientes a la salida del paso anterior y es generada la estructura correspondiente (**RIB + *updates***). Finalmente, se salva la nueva estructura generada que corresponde a la entrada del siguiente paso.

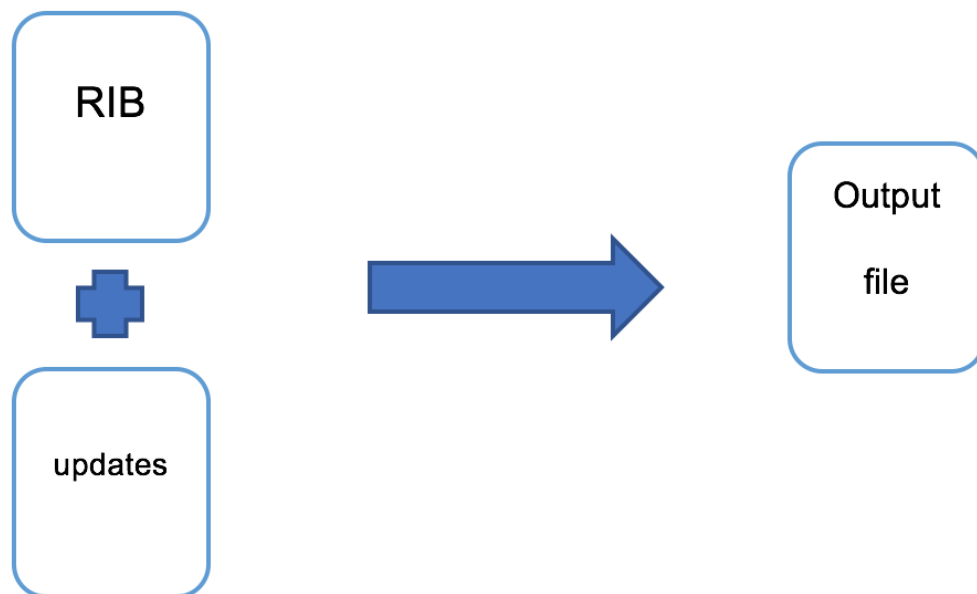


Figura 19: Paso 4 - Concatenación de RIB data con updates

4.2.3.5 Paso 5: *split_data_for_analysis.py*

Como entrada al análisis se ha decidido separar los datos para abstraer al mismo del tipo de versión IP utilizada por cada monitor.

El código carga los datos generados por el paso 4 y los recorre identificando las entradas correspondientes a una y otra versión IP analizando el campo prefijo para cada una de las entradas. El criterio utilizado ha sido el siguiente:

- **IPv6:** El prefijo contiene el carácter ‘:’.
- **IPv4:** El prefijo no contiene el carácter ‘:’.

Una vez identificado los índices de cada una de las versiones la estructura de datos entrante se divide de acuerdo a estos generando dos estructuras y son volcadas a ficheros en directorios independientes.

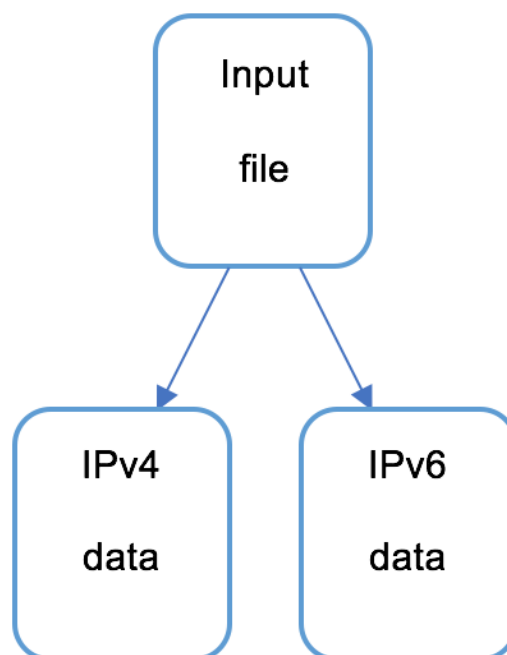


Figura 20: Paso 5 - División de los datos según versión IP

4.2.3.6 Paso 6: *more_specifics_analysis.py*

Una vez que se han cargado, limpiado y separado los datos llega la hora de abordar la extracción de parte de las características a nivel de prefijo mencionadas en el apartado 4 de este documento. Rememorando lo descrito en dicho apartado estas son: tipo - atendiendo a la clasificación, número de updates – estabilidad y profundidad. El *Address Span* se calculará posteriormente atendiendo a la longitud del prefijo junto con el tipo mediante el *software* Tableau.

El código está organizado en funciones donde destaca una función principal que realiza todo el análisis, la cual, recibe un parámetro de entrada que especifica a la versión IP. De esta manera se consigue evitar la duplicidad de código. Además, todas las funciones se han testeado exhaustivamente mediante *tests unitarios* que pretenden contemplar en la medida de lo posible todas las casuísticas.

Esta función principal se encuentra dividida en tres funciones ejecutadas en cascada para conseguir una salida por versión IP:

1. Cálculo de **visibilidad y número de updates**.
2. *Clustering* de prefijos y cálculo de la profundidad atendiendo al nivel más alto de clasificación.
3. *Clustering* de prefijos con detalle de más específicos.

Para el clustering de prefijos se ha utilizado la librería *py-radix* [18] que proporciona la construcción de un árbol de direcciones en formato CIDR.

En la Figura 12 se ilustra la secuencia en lo que comprende a la ejecución de las tres funciones comentadas.

Conviene recordar que como entrada a este paso tenemos todos los datos sin procesar, es decir, por ahora existen varias entradas para un mismo prefijo para un monitor en cuestión. Lo que justifica el orden seguido para la obtención de las características.

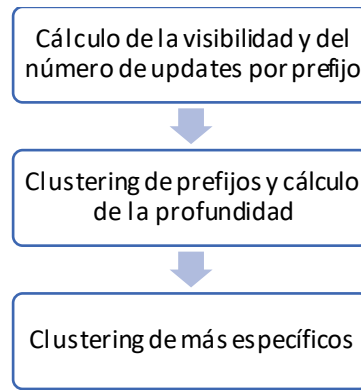


Figura 21: Diagrama de flujo para la ejecución del análisis de prefijos

Para el **cálculo de la visibilidad y el número de updates** conviene ordenar la estructura de datos cargada por monitor, prefijo y *timestamp*. Esto refleja de nuevo una forma práctica de separar los datos sin necesidad de haber generado ficheros nuevos. También evita recorrer el fichero varias veces para la extracción de la visibilidad o del número de prefijos.

Esta estructura de datos se corresponde a la entrada de la primera función del análisis. Llegado a este punto interesa trabajar con **bloques de datos** correspondientes a todos los datos para un prefijo dado por monitor.

Para ello, se ha creado una función que identifica los puntos en los que una columna de la estructura de datos cambia de valor. Esto nos permite separar bloques de datos por monitor y posteriormente por prefijo. Por lo tanto, es necesario recorrer los datos dos veces una para separarlos a nivel de monitor y posteriormente, se recorre cada una de estas para procesar todos los datos correspondientes a un prefijo.

En primer lugar, se obtiene el número de updates por prefijo contando todas las entradas correspondientes al tipo de mensaje excluyendo aquellos de tipo B que representan su presencia en la RIB al comienzo del experimento. Posteriormente, se procesan los datos por prefijo extrayendo la característica de la visibilidad ilustrada en la *Figura 11* de este documento.

Seguidamente se pretende explicar la función que se acaba de describir de manera gráfica para aclarar el funcionamiento de esta subsección del código.

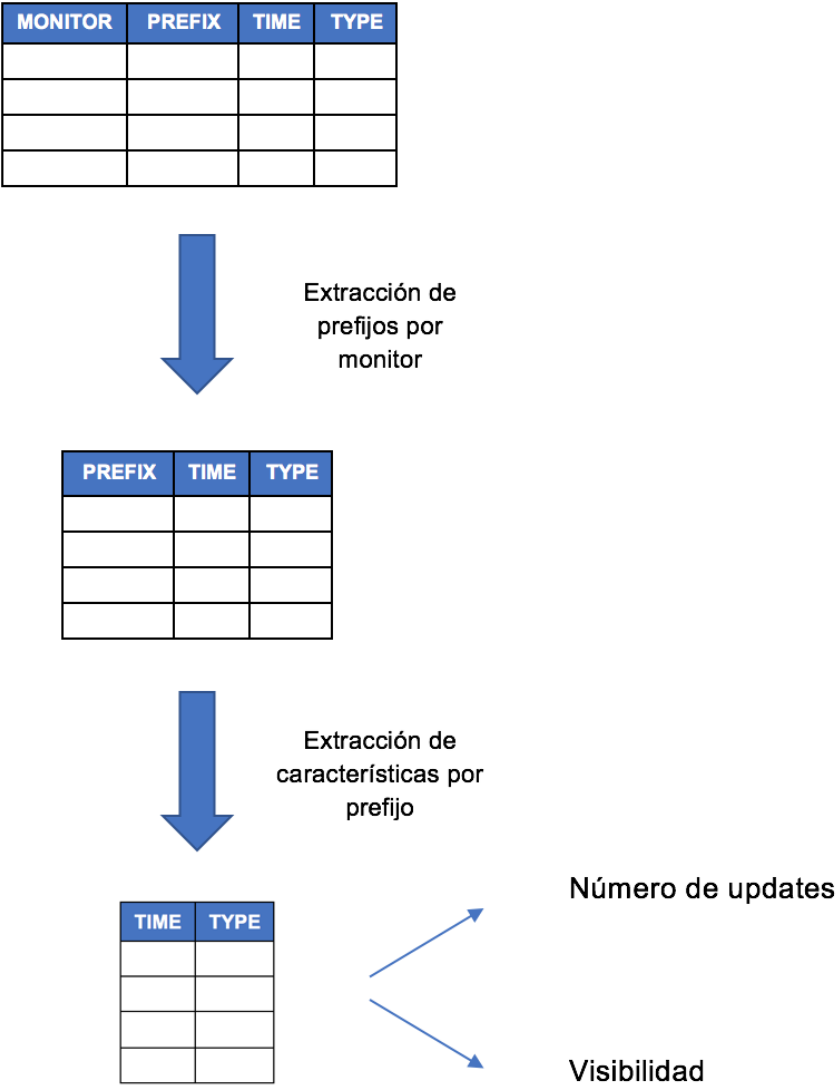


Figura 22: Extracción de número del updates y de la visibilidad

Una vez extraída la visibilidad y el número de updates se ha logrado obtener una estructura que presenta para estas características para cada prefijo visto por monitor. Además, se obtiene una única entrada por cada prefijo visto por monitor lo que corresponde a la entrada para la siguiente función.

Para la clasificación de prefijos y el cálculo de la profundidad también son segmentados los datos de entrada en bloques. Ahora solo es necesaria la separación en **bloques por monitor**.

Para todos los prefijos vistos por cada uno de los monitores se va a generar un **Prefix Trie** como el mostrado en la *Figura 5*. Para lo que se ha testado y posteriormente utilizado la librería de Python [13] cuyas funciones permiten llevar a cabo la tarea de **longest prefix matching** para las dos versiones IP.

Es necesario recorrer la estructura dos veces; para crear el árbol y para clasificar los prefijos. La clasificación de los prefijos es aprovechada para calcular la profundidad en los más específicos con respecto al prefijo menos específico del que depende. Además, ha sido necesaria la implementación de una función para buscar la longitud de prefijo máxima ya que la librería utilizada no contenía esta funcionalidad.

Por último, en cuanto al *clustering* de más específicos se filtran los resultados anteriores para este tipo de prefijos y no se atiende a la profundidad. Se sigue la misma metodología descrita y se clasifican estos con distinto etiquetado.

Ahora ya se tienen todos los datos para el campo tipo se sustituyen las entradas correspondientes a los subtipos calculados por la última función se construye una estructura que consta de **monitor, prefijo, tipo, visibilidad, número de updates** y tipo que presenta todas las características calculadas y se vuelcan a un fichero que ya corresponde el de resultados a nivel de colector.

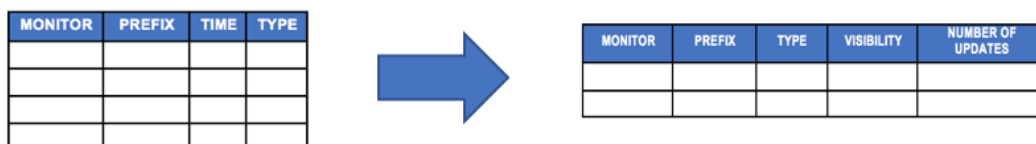


Figura 23: Resumen de la transformación de los datos para el análisis de más específicos

Nota: No confundir el tipo de la entrada de la RIB con el tipo de prefijo.

4.2.4Combinación de los datos

Llegado a este punto ya se ha explicado cómo se han obtenido todas las características del análisis para cada colector. Pero lo realmente interesante es combinar todos estos datos para realizar un estudio de varios colectores. Con este objetivo en mente, se ha desarrollado un último código que agrega todos los datos de varios colectores para las diferentes versiones IP. Este fichero corresponde a la entrada al software *Tableau*.

Hubiese sido posible conectarse independientemente a cada uno de los ficheros generados por colector, pero esto limitaría las funcionalidades de *Tableau* y el análisis conjunto. Además, afectaría al rendimiento de la herramienta de visualización creada.

El código desarrollado se ejecuta como último paso en el fichero *experiment.sh* y como se ha explicado previamente, espera a que todos los resultados por colector mediante el último paso hayan sido extraídos.

Una vez disponibles todos los resultados, se carga para un experimento todos los datos de los colectores especificados y se procede a un proceso de etiquetado en el que se añaden dos columnas nuevas **colector** y **versión IP**.

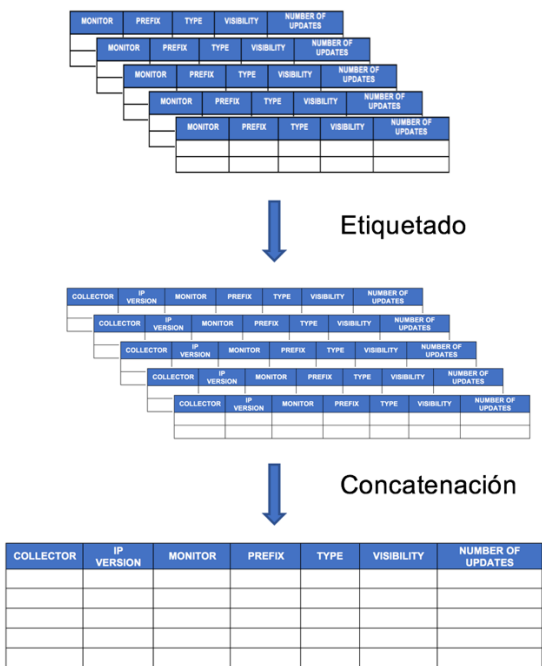


Figura 24: Proceso de combinación de datos de varios colectores

5 Análisis de los resultados

5.1 Características de los datos de entrada

Para el análisis se han tenido en cuenta los colectores rrc00, rrc01, rrc04, rrc05, rrc07, rrc10, rrc11, rrc12 y rrc13 de RIPE. La ventana horaria en la que se ha realizado el experimento ha consistido en las 8 primeras horas del 8 de Enero de 2018. Lo que es equivalente al periodo comprendido entre dos volcados de RIB.

Se han analizado **40.931.180 prefijos** vistos por un conjunto de **185 monitores**. Sin embargo, el **número total de prefijos vistos es de 722.859**.

El análisis se ha dividido en dos secciones. La primera, hace hincapié en cómo es la distribución de los prefijos. A continuación, se realiza un segundo estudio enfocándose más en la característica de la visibilidad de los prefijos. Para ambos estudios los datos son analizados tanto a nivel de monitor como agregando los resultados de cada monitor.

5.2 Análisis de distribución de los prefijos

Todo el análisis se ha centrado en la versión IPv4, aunque en el software generado a partir de Tableau permite el análisis de IPv6.

5.2.1 Distribución por tipo de prefijo

Lo primero que se va a estudiar es la **distribución de prefijos por monitor**. Para lo que se ha creado la siguiente vista que representa el número de prefijos vistos por monitor. Esta, se ha ordenado descendientemente según el número de prefijos.

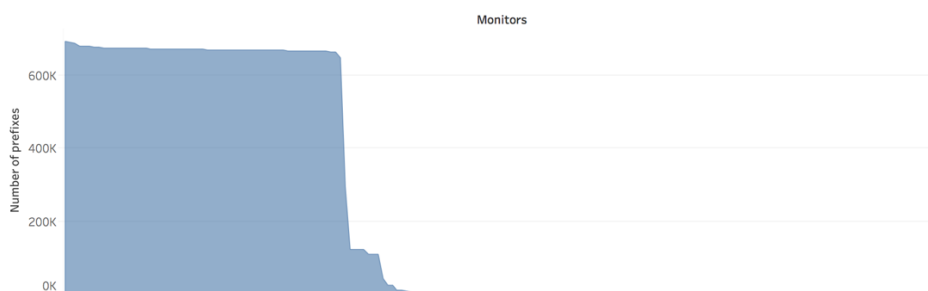


Figura 25: Número de prefijos vistos por monitor

Se puede apreciar que existen una serie de nodos que ven un número de prefijos elevado (alrededor de **670.000**). Estos corresponden a los **proveedores**. Por otro lado, el resto corresponden a **clientes o peers**. Para discriminar entre unos y otros se ha fijado un umbral de 450.000 prefijos.

A partir de ahora, el análisis se va a dividir en vistas para estos dos tipos de nodos. Se cuenta con **59 proveedores** que representan alrededor de 40 millones de prefijos. El resto de los prefijos se encuentra repartido entre los **126 peers** restantes.

Es interesante apreciar como es la distribución de prefijos atendiendo a la clasificación por tipo. Para ello, se han agrupado los tres tipos de más específicos en más específicos mediante el uso de una funcionalidad ofrecida por Tableau.



Figura 26: Número de prefijos por proveedor (ordenados por el número de prefijos) – Diagrama de barras

En la figura 26 se han representado solo los monitores que corresponden a **proveedores (>450.000 prefijos vistos)** ordenados por número de prefijos. Se puede afirmar que la distribución por tipos es estable para cada uno de los monitores. Seguidamente, se da una vuelta a la representación para extraer más conclusiones.

Manteniendo el orden de la gráfica anterior, se ha normalizado el eje Y con respecto al número de prefijos por monitor (eje X). Ahora es más sencillo apreciar cómo se encuentran repartidos los prefijos por monitor.

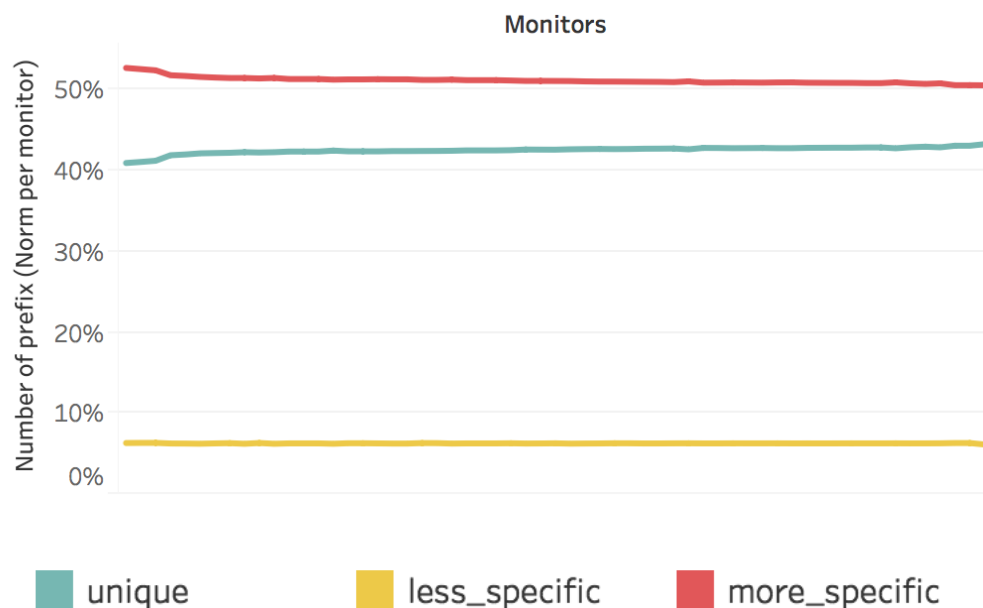


Figura 27: Número de prefijos por proveedor - Normalizado (ordenado por el número de prefijos)

Se puede observar que la mayoría de los prefijos son **más específicos**, además el porcentaje de estos por monitor (~51%) es directamente proporcional al número de prefijos vistos. Sin embargo, no pasa lo mismo para el caso de los únicos que presenta la tendencia contraria, es decir, parece que es indirectamente proporcional al número de prefijos por monitor. Además, estos no difieren tanto en número de los primeros. Por último, los menos específicos presentan la menor parte de los prefijos, pero a la vez son los más estables.

La primera vez que se realizó este análisis tenía **108 proveedores; 49 han sido descartados** para el resto del análisis debido a que tenían configurado el prefijo por defecto (**0.0.0.0/0**) que está definido en la tabla de enrutamiento. Esto anula directamente al análisis ya que este prefijo es clasificado como el 'único' menos específico por monitor mientras que el resto de prefijos son más específicos, lo que podría llevar a falsas interpretaciones. Por el contrario, esta configuración no está presente en los **peers** por lo que se han conservado en su totalidad.

Haciendo el mismo análisis para los **peers**, se muestra a continuación la representación correspondiente a número de prefijos (eje Y) por monitor (eje X).

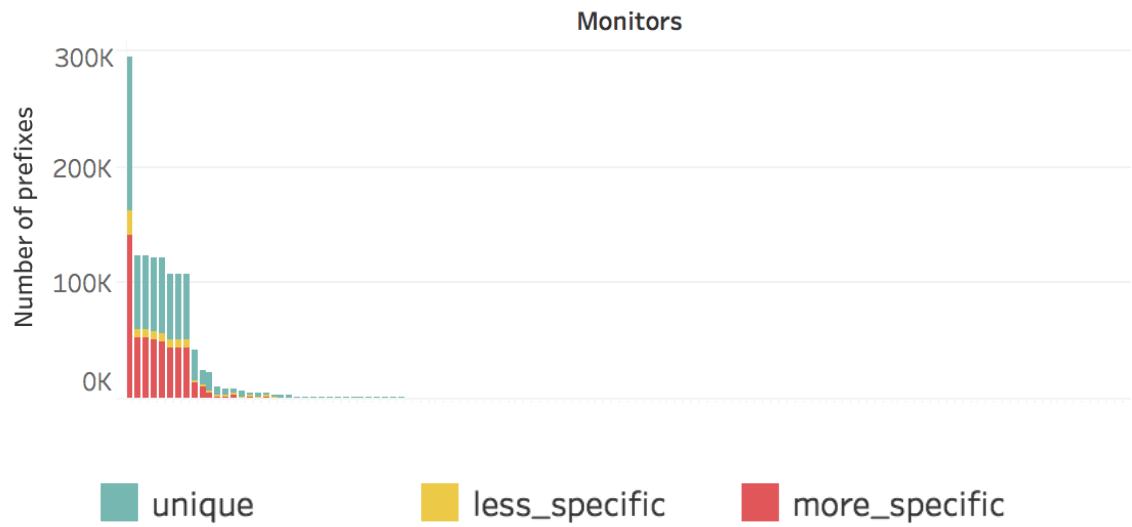


Figura 28: Número de prefijos para cada tipo por peer (ordenado por el número de prefijos) – Diagrama de barras

Debido a que un número reducido de monitores contienen una porción elevada del número de prefijos es necesario realizar la misma normalización que en la figura 27 para comparar la distribución entre tipos de prefijos.

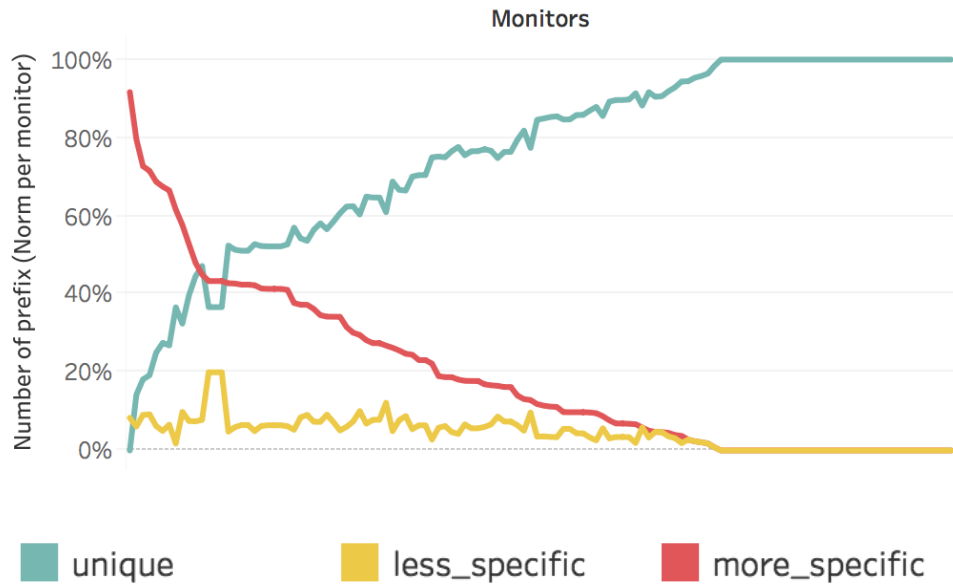
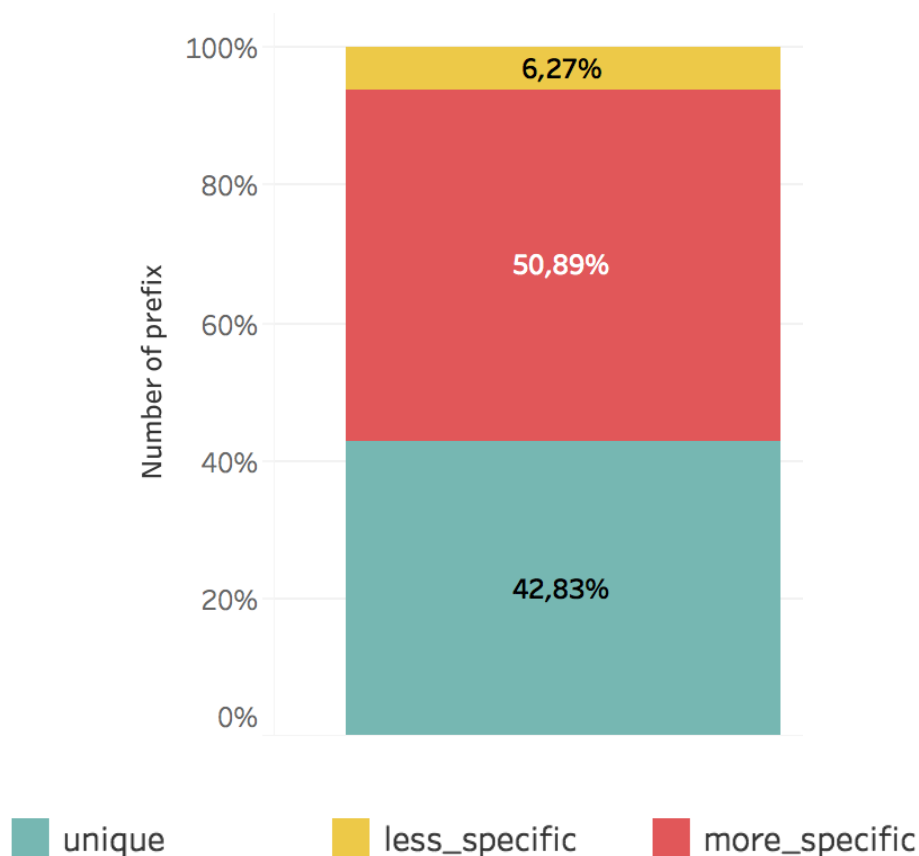


Figura 29: Distribución normalizada para cada tipo de prefijo por peers (ordenado según el porcentaje de más específicos)

En este caso la distribución por tipo de prefijo no es estable por lo que se ha ordenado la gráfica por el **% de más específicos por monitor**. El comportamiento para los **peers** no es predecible como en el caso de los proveedores. A la derecha de la gráfica se puede observar que casi un tercio de los nodos solo ven prefijos únicos. Llama la atención que para la mayoría de los nodos la mayoría de los prefijos son vistos como más específicos.

Debido a la inestabilidad de los **peers** junto con la pequeña fracción de datos que representa. Se ha decidido **centrar el análisis por monitor** exclusivamente para los **proveedores**. Para demostrarlo, se ha realizado la siguiente gráfica que presenta la distribución agregada para todos los monitores por tipo. Se puede apreciar que la influencia de los peers sobre la totalidad de los datos es mínima. Al no estudiarlos se evita el ruido que producen estos monitores en el análisis.



Otro punto de vista de la distribución de prefijos a lo largo de los proveedores es calcular la media de **cuantos más específicos presenta cada menos específico**, es decir, el ratio.

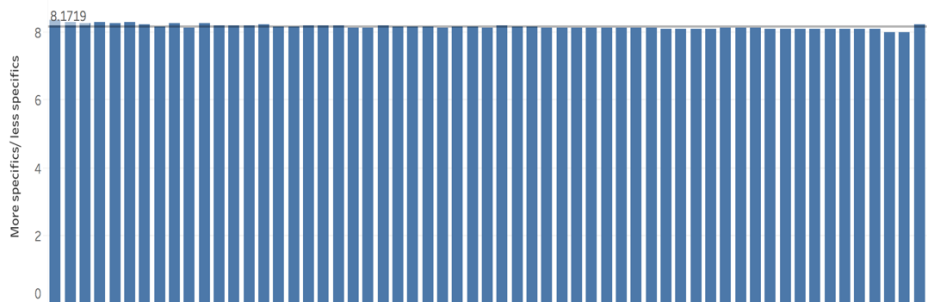


Figura 30: Ratio más específicos entre menos específicos por proveedor

La gráfica muestra que esta comparativa es también muy estable por monitor siendo algo superior a **8 más específicos por cada prefijo menos específico**. Aún es pronto para sacar conclusiones de esto por lo que se pretende aclararlo con los resultados presentados más adelante.

5.2.2 Distribución de los más específicos por monitor

Manteniendo el orden y atendiendo a la clasificación de los más específicos, se ha extraído la característica de que los prefijos más específicos con **nivel único** son los **más frecuentes** a lo largo de los proveedores representando alrededor del **67%** de los más específicos. Luego son estos lo que más influyen en el análisis de las características comentadas para los siguientes análisis del agrupamiento de todos los más específicos.



Figura 31: Número de más específicos con un único nivel por proveedor (Normalizado por el número de más específicos)

5.2.3 Profundidad de los más específicos por monitor

A continuación, se presenta el **análisis de la profundidad** para los más específicos por monitor (ordenados según el número de prefijos). Como se definió en la sección 3 la profundidad es la diferencia entre la longitud del prefijo de un más específico con respecto a la longitud de su raíz (menos específico asociado).

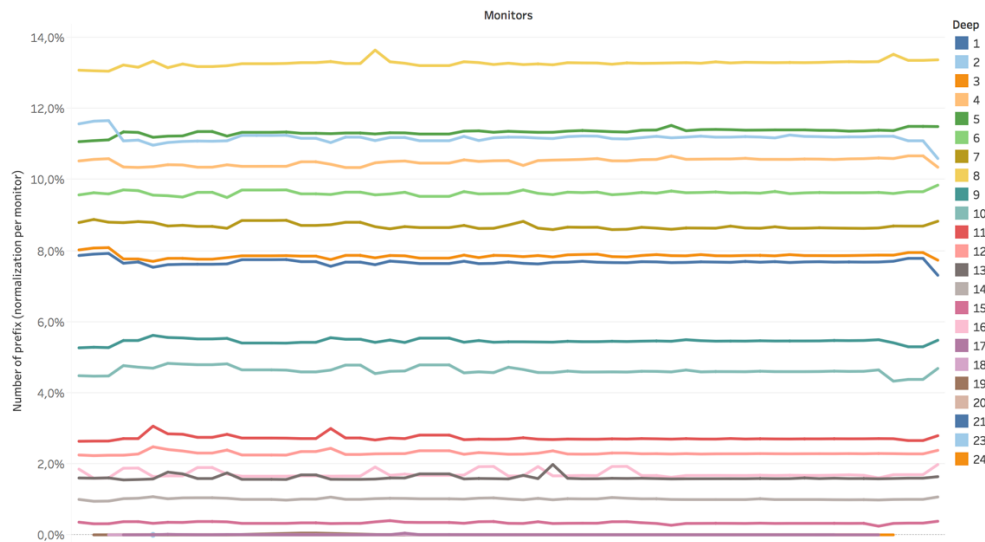


Figura 32: Profundidad de más específicos por monitor (ordenados según el número de prefijos)

La profundidad más típica es de 8. Parece que las profundidades mayores sirven para el reparto de bloques de direcciones, es decir, **podría interpretarse** como un **tema administrativo**. Por otro lado, las profundidades menores sugieren que se está haciendo **Ingeniería de Tráfico**.

Para extraer más conclusiones sobre lo supuesto, se han agrupado las profundidades estableciendo un umbral de **profundidad 4**.

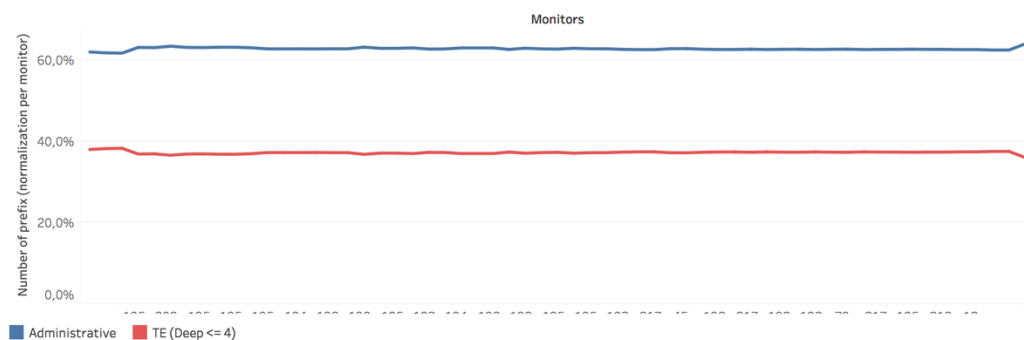


Figura 33: Agrupación de profundidades por monitor (normalización por número de prefijos)

La mayor parte de los más específicos parece estar siendo utilizada para **administración de bloques prefijos** y son inversamente proporcionales al número de prefijos. Por el contrario, los prefijos dedicados a **ingeniería de tráfico** son alrededor del **37%** a lo largo de los monitores. Además, el número de prefijos dedicados con este fin aumenta a medida que aumenta el número de más específicos aprendidos por monitor.

5.2.4 Longitud de prefijos por monitor atendiendo al tipo

A continuación, se representa el **porcentaje de longitudes de prefijo por monitor para cada tipo**. Se ha mantenido el criterio de ordenar los monitores por el número de prefijos por monitor.



Figura 34: Porcentaje de longitudes de prefijos por monitor (Normalizado por el número de prefijos)

Las longitudes de **más específicos** son mayoritariamente **menores que /25** lo que cuadra con los filtrados implementados en la red para prefijos cortos, ya que no es necesario ese nivel detalle para optimizar el enrutamiento, o bien con problemas de conectividad. Con estos filtrados se evita un incremento de la RIB para cada nodo lo que afectaría negativamente al rendimiento de la red.

Comparando los resultados de los **más específicos** con los de los **menos específicos** se pueden sacar varias conclusiones.

En primer lugar, parece que los bloques correspondientes a /16s de menos específicos junto con los /24s de los más específicos tienen una alta dependencia con el resultado de la profundidad más típica 8. Estos menos específicos, no representan la mayoría de los prefijos, aunque también hay que tener en cuenta el hecho de que al ser bloques más grandes existe un menor número de bloques.

Combinando que por lo general los prefijos presentan un único nivel con los valores obtenidos para la profundidad. Se deduce que los menos específicos /22s, /20s, /19s, /21s y /23s, que representan la mayoría de este tipo, contribuyen a la partición de los prefijos de profundidad baja (≤ 4). Por otro lado, los prefijos menos específicos que presentan bloques mayores tienden a partirse en bloques de una profundidad mayor y a la vez anuncian una pequeña parte esos bloques más específicos. Esto permite justificar la media representada en la figura 30 de aproximadamente 8 más específicos por cada menos específico.

Atendiendo a los resultados de los más específicos frente a los únicos se observa que la distribución es parecida. Podría resultar interesante el estudio de este fenómeno, pero el tipo de análisis planteado no tiene una explicación para ello.

5.2.5 Address Span de los más específicos

Habiendo contemplado que tanto el número de más específicos como su distribución por proveedor son parecidas. En la siguiente gráfica se representa el **espacio de direccionamiento de los más específicos** respecto al direccionamiento total visto por cada proveedor manteniendo el orden de los monitores por número de prefijos.

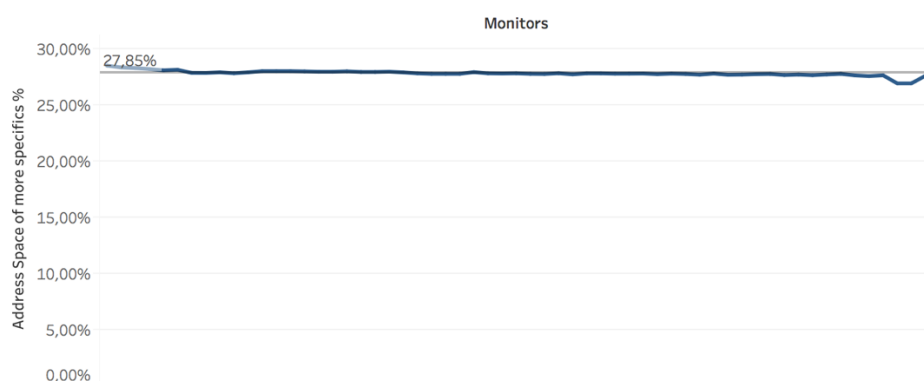


Figura 35: Espacio de direccionamiento de más específicos por proveedor

Se observa que este espacio de direccionamiento es directamente proporcional al número de prefijos vistos por cada monitor y por lo consiguiente al número de prefijos más específicos. Además, bastante estable con valores muy cercanos a la media (**27,85%**). No se puede afirmar que los más específicos vistos por cada proveedor sean los mismos, pero sí que presentan una distribución de longitud de prefijo muy similar como se puede comprobar en la figura 34.

5.2.6 Análisis agregado por longitud de prefijo

Se han agregado todos los datos para el estudio de la distribución de prefijos por máscara. En la siguiente gráfica se puede apreciar que la mayoría de prefijos vistos son /24s siendo más específicos o únicos mayoritariamente.

Debido al número de prefijos que representan los /24s se ha generado la segunda gráfica en la que el eje y se ha escalado para cada columna.

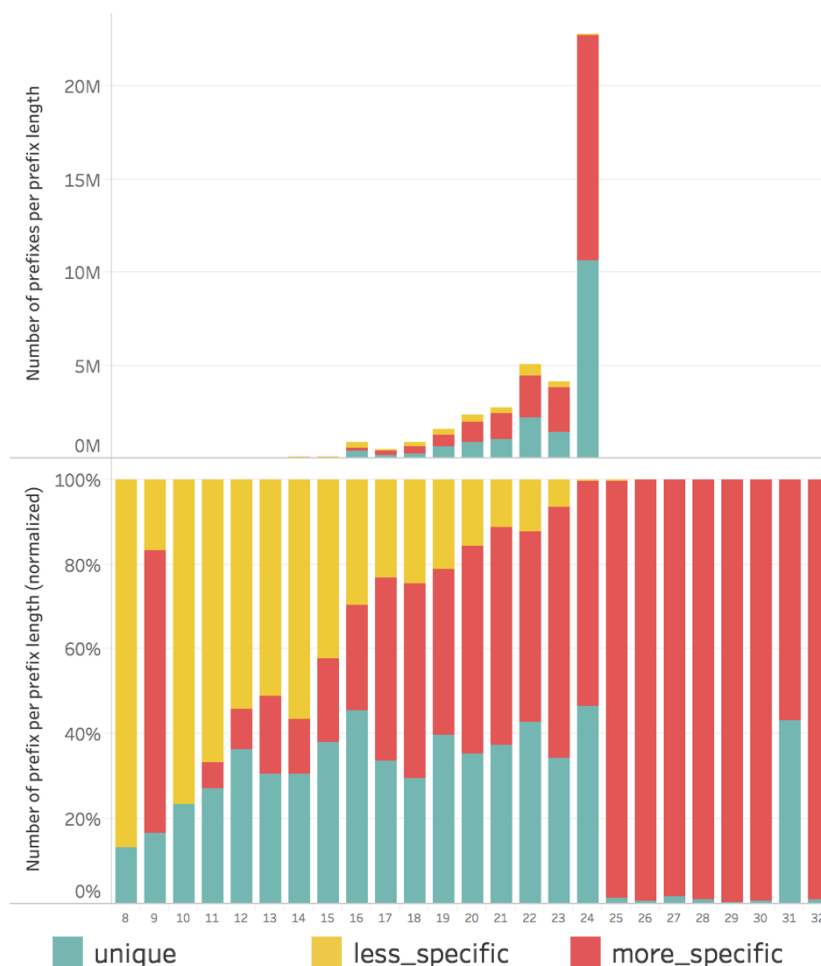


Figura 36: Número de prefijos por máscara - análisis agregado

Lo primero que llama la atención es que **desde los /10s hasta los /24s** se aprecia la tendencia decreciente de los **menos específicos** que muestra como a medida que aumenta la longitud del prefijo disminuye la probabilidad de hacer ingeniería de tráfico, o lo que es lo mismo, de partir estos prefijos en otros más pequeños.

Por, los **prefijos únicos** presentan cierta estabilidad en cuanto a su distribución a partir de los /16s. Estos prefijos reflejan que no hay interés en repartir el tráfico por enlaces distintos.

5.2.7 Análisis agregado de la variación del tipo de prefijo

El último análisis está enfocado a **cómo es visto un mismo prefijo por los distintos monitores**. Para ello se han clasificado los prefijos como:

1. Visto exclusivamente como más específico.
2. Visto exclusivamente como único.
3. Visto exclusivamente como menos específico.
4. Visto como más específico y único.
5. Visto como menos específico y único.
6. Visto como menos específico y más específico.
7. Visto como único, más específico y menos específico

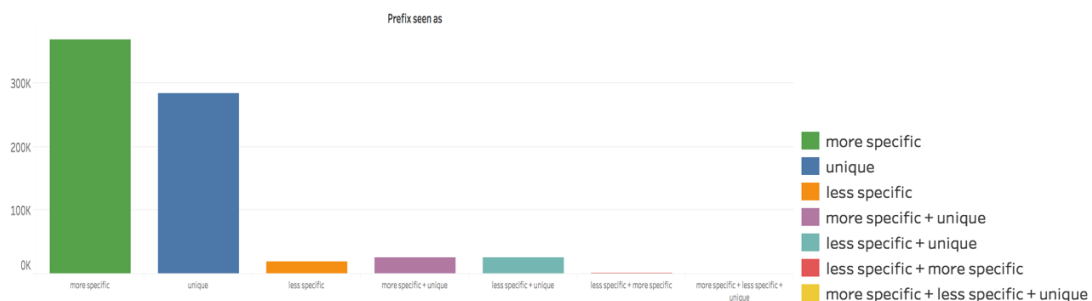


Figura 37: Análisis de la variabilidad del tipo de prefijo

Es interesante el resultado obtenido para los prefijos que han sido vistos de dos formas distintas por el conjunto de monitores. Los que han sido vistos como menos específicos y únicos, sugieren que en unas redes se ve el prefijo y sus más específicos, y en otras redes se han perdido los más específicos. La pérdida puede ser bien por filtrado, o por problemas de conectividad. Lo que refleja que para el haberse visto un menos específico como único se tienen que haber perdido todos sus más específicos.

El razonamiento para los prefijos que han sido vistos como más específicos y únicos es similar. En este caso, para algunas redes se ha perdido el menos específico, pero no es lógico pensar que este se ha perdido por filtrado. Es extraño que el comportamiento de los menos específicos sea parecido al de los más específicos cuando de estos no se espera un filtrado. Pero por alguna razón esto está ocurriendo y sería un punto que convendría estudiar con más profundidad en un futuro.

5.3 Análisis de visibilidad

Para el análisis de la visibilidad se ha subdividido en dos análisis el primero a nivel de colector (solo para los proveedores) y el segundo teniendo en cuenta el tipo de prefijo para la agregación de todos los datos (incluyendo los peers).

Además, para comprender mejor los datos se ha clasificado variable de la visibilidad para cada prefijo atendiendo a los rangos: 100%, 99-90%, 90-80%, 80-70%, 70-60%, 60-50%, 50-40%, 40-30%, 30-20%, 20-10% y <10%. Esto ha permitido trabajar con variables discretas.

También se ha tenido en cuenta el concepto de estabilidad para poder ver con más detalle cómo se comportan los prefijos que resultan ser menos visibles a lo largo de todo el conjunto de datos.

5.3.1 Análisis de visibilidad de prefijos en proveedores

Como primera aproximación se ha representado la visibilidad por proveedor. El eje x se corresponde a los monitores mientras que el eje Y se encuentra normalizado con respecto al número de prefijos por monitor.



Figura 38: Visibilidad de prefijos por proveedor – Diagrama de barras

Puede extraerse como conclusión que la mayoría de los prefijos se encuentran 100% visibles a lo largo del todo experimento. La media de estos prefijos totalmente visibles a lo largo de los monitores representa el **99,4%**. Por lo tanto, se aprecia que la mayoría de los prefijos que ven los proveedores no cambian.

Para conocer un en más detalle el comportamiento de la visibilidad por monitor, se han separado los prefijos estables de los inestables según la definición en la sección 3.3.

Refrescando lo que esto significa: los **prefijos no afectados** son aquellos que se encuentran en la RIB cargada al comienzo del experimento y no reciben ningún *update* durante el tiempo que comprende el análisis. Luego estos prefijos no se verán alterados a lo largo del experimento presentándose **100% visibles**. Por otro lado, están los prefijos que han recibido algún *update* durante la ventana de tiempo analizada.

A continuación, se muestra una gráfica que presenta el **porcentaje de prefijos que han recibido updates por monitor**. Este porcentaje se calcula normalizando el número de prefijos por cada columna. Los monitores que se corresponden con el eje x y han sido ordenados descendientemente por el valor del porcentaje de prefijos que han recibido algún update.

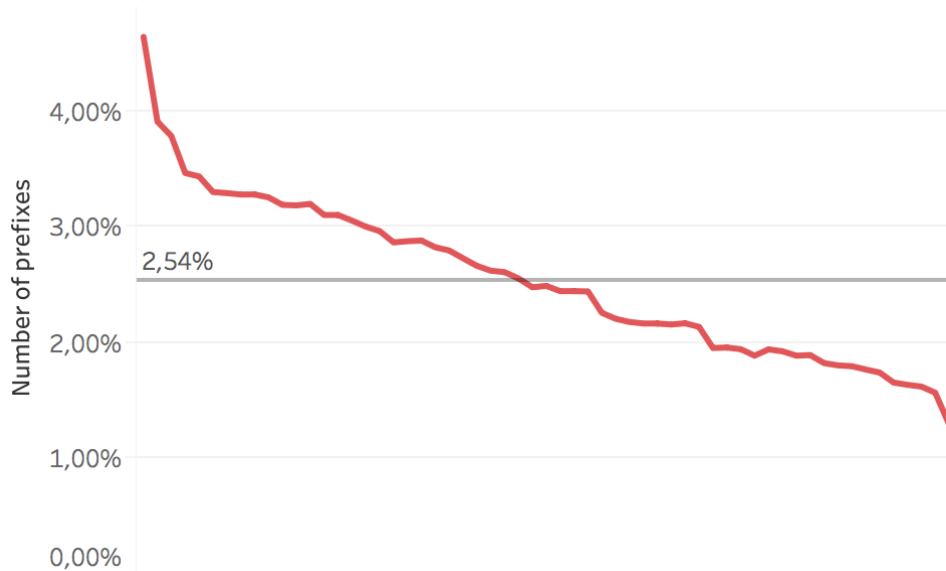


Figura 39: Prefijos que han recibido updates durante el experimento por monitor (Normalizado por número de prefijos)

La tendencia de esta curva parece representar algún tipo de distribución estadística. Para asegurar que esto se trata de una distribución específica se requieren unos análisis estadísticos muy concretos a los que no está enfocado este trabajo. Pero es presenta otro análisis interesante de realizar en un futuro.

La media de prefijos inestables por monitor se corresponde con un **2,54%** de los prefijos por monitor. Estos resultados no reflejan que los prefijos correspondientes al porcentaje dado por monitor presentan una visibilidad <100%.

Para extraer más conclusiones sobre la distribución de la visibilidad a nivel de monitor se va a analizar la visibilidad para los prefijos que han recibido updates durante el experimento.

En la siguiente gráfica se observa la visibilidad para los prefijos que han recibido algún *update* durante el tiempo de observación. La gráfica se encuentra normalizada por el número de prefijos que reciben algún *update* por monitor y además se ordena descendientemente según el porcentaje de prefijos que reciben *updates*.

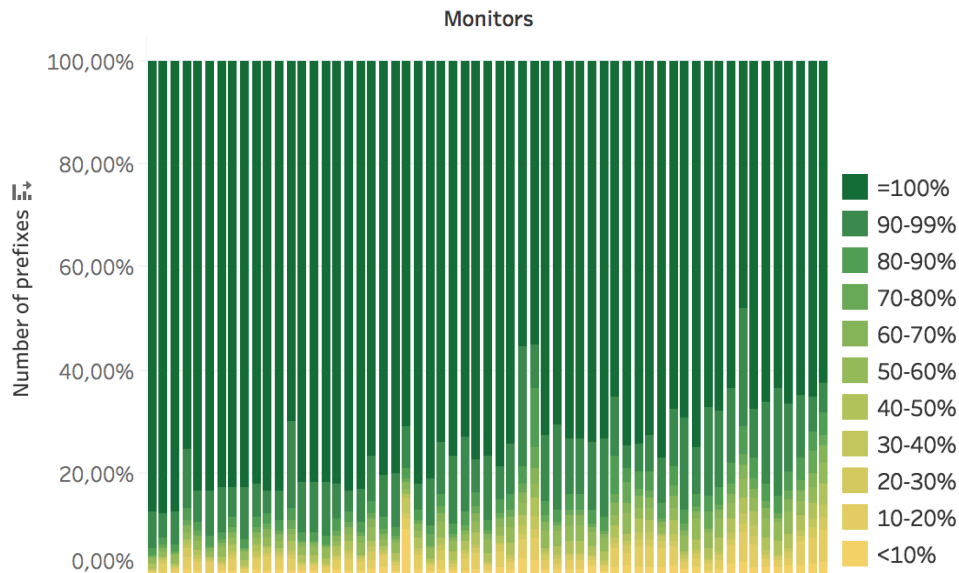


Figura 40: Visibilidad para prefijos que han recibido algún update durante el experimento por monitor (Normalizado por el número de prefijos)

A nivel de monitor se puede extraer la siguiente conclusión: “A medida que la visibilidad aumenta conforme aumenta el porcentaje de prefijos que reciben *updates*”.

Esto, significa que según aumenta el porcentaje de prefijos que reciben algún *update*, aumenta el número de anuncios. Es decir, da la impresión que se está intentando **garantizar la visibilidad de algunos prefijos de la RIB** enviando el anuncio del prefijo a través de caminos distintos.

También se ha realizado un análisis agregado de los datos filtrados por el número de updates recibido por algún prefijo. A continuación, se muestra la distribución agregada de la visibilidad para todos los proveedores.

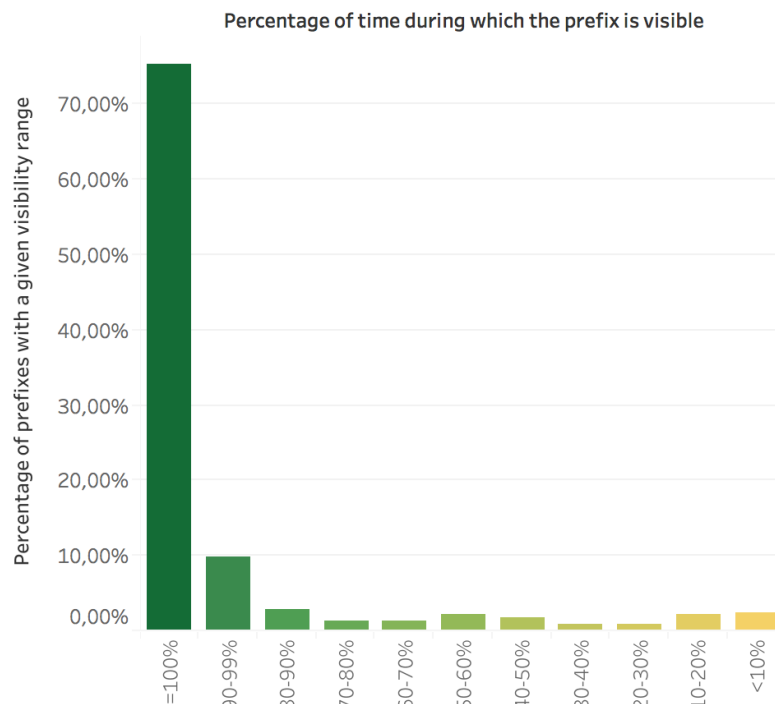


Figura 41: Análisis agregado por tipo de la visibilidad para los prefijos que han recibido algún update durante el experimento

De la gráfica anterior se extraen varias conclusiones. Por un lado, el hecho de que el prefijo reciba algún update no quiere decir que se haya borrado en el algún momento su entrada, ya que el **75,29%** de los prefijos están totalmente visibles a lo largo del experimento.

Es decir, el hecho de que se haya recibido algún update y este corresponda a un mensaje de borrado sucede aproximadamente para una media del 24,71% del 2,54% de los prefijos. Lo que se corresponde a un 0,6% de los prefijos, lo que obviamente se complementa con 99,6% de los mensajes totalmente visibles del conjunto total de los datos para todos los proveedores que se extrajo al comienzo de este análisis.

5.3.2 Análisis de visibilidad por tipo de prefijo

En este apartado se va a analizar la visibilidad dividiendo el conjunto de todos los prefijos por su clasificación según cada monitor. El orden de este análisis sigue el mismo orden que el mencionado en el análisis por monitor.

Lo primero, se observa la visibilidad por todos los prefijos correspondientes a algún tipo. En la gráfica siguiente se observa la visibilidad por tipo según los rangos previamente definidos. El eje y se ha normalizado respecto al número de prefijos por tipo. Esta normalización se hace indispensable debido a la desproporción del número de prefijos entre tipos.

Se aprecia que los todos los tipos son muy visibles durante el tiempo del experimento. Además, los prefijos menos específicos presentan el tipo más visible. Por el contrario, los más específicos son los menos visibles, aun así, siguen presentando un porcentaje muy alto de prefijos visibles.

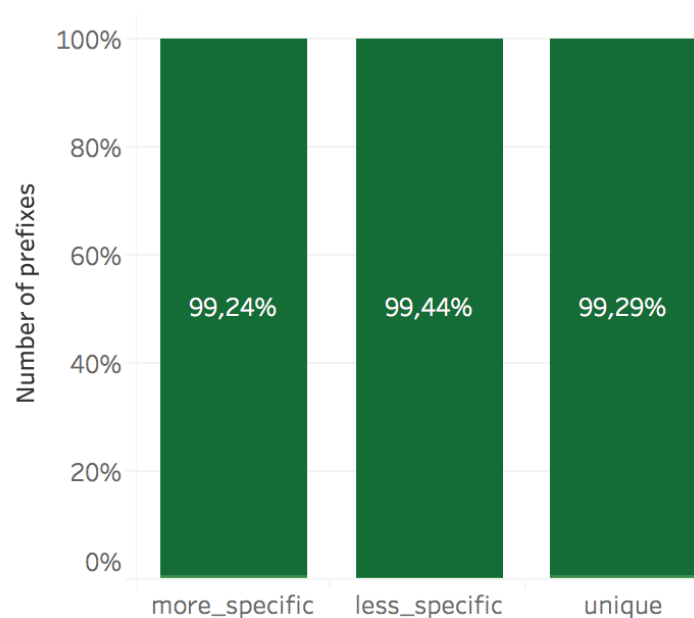


Figura 42: Análisis de visibilidad por tipo (Normalizado por el número de prefijos)

Siguiendo con la estructura del análisis se pretenden extraer más conclusiones analizando los prefijos que han recibido algún update para cada tipo de prefijo.

A continuación, se muestra una gráfica que presenta el **porcentaje de prefijos que han recibido updates por tipo**. Este porcentaje se calcula normalizando el número de prefijos por cada columna. El tipo de prefijo se corresponde con el eje x y ha sido ordenado descendientemente por el valor del porcentaje de prefijos que han recibido algún update.

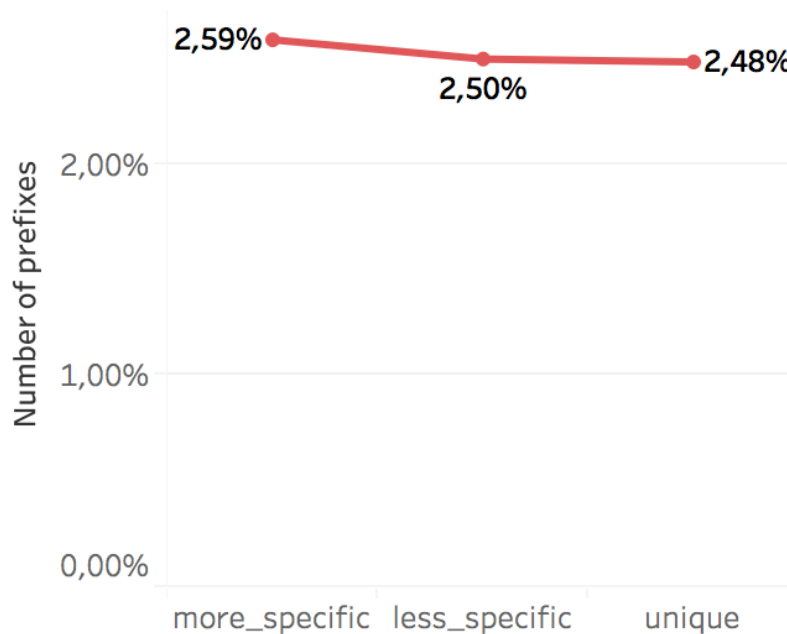


Figura 43: Porcentaje de prefijos que han recibido algún update por tipo durante el experimento (Normalizado por el número de prefijos)

El porcentaje de prefijos que reciben algún es muy parecido atendiendo al tipo. Siendo los más específicos los que presentan un mayor número de updates.

Para obtener un mayor nivel de detalle de cómo es la visibilidad para estos prefijos que reciben algún update se ha realizado la siguiente gráfica. Se ha normalizado el eje respecto al número de prefijos que han recibido algún update durante el experimento. También se mantiene el orden según estos para cada tipo.

De esta gráfica no se puede extraer una relación entre el porcentaje de prefijos que han recibido algún update y la visibilidad como se pudo reflejar en el análisis por monitor. Para el caso de los menos específicos el hecho de que sean más estables puede deberse a que debido a la longitud de prefijos se ven menos afectados por los filtrados. Por el contrario, los más específicos y los únicos podrían verse afectados por estos filtrados además cuadra con su comportamiento es parecido y cuadra con su semejanza en cuanto a la distribución según la longitud de los prefijos.

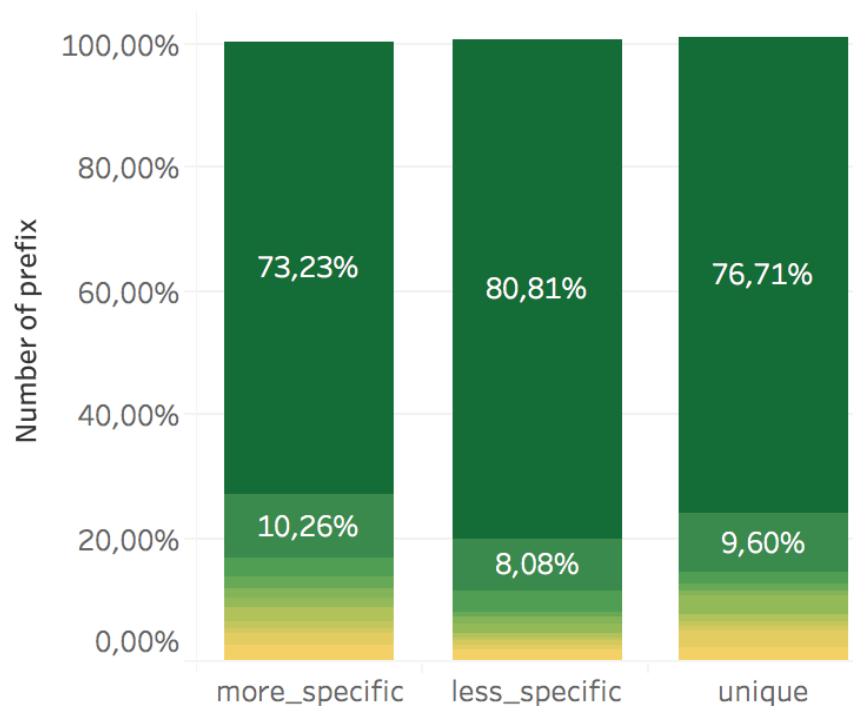


Figura 44: Visibilidad para prefijos que han recibido algún update durante el experimento por tipo (Normalizado por el número de prefijos)

Todo ello, ha sugerido un análisis más profundo sobre la visibilidad dependiendo de la longitud de prefijos para cada tipo. Esta sección se presenta como una subsección independiente al final del análisis de visibilidad y, a su vez, ha contribuido para aclarar otros resultados que habían sido obtenidos en la sección de distribución de distribución por tipo de prefijo.

Por último, se da ha dado una a la representación de la figura 44 para obtener una perspectiva más clara.

Esta gráfica cambia el eje x, que previamente representaba el tipo de prefijo, por los de visibilidad establecidos. Asimismo, los datos han sido desglosados mediante la leyenda de colores que ha sido utilizada hasta el momento. Con la gráfica mostrada a continuación puede concluirse que no existe una diferencia importante en términos de visibilidad para cada tipo de prefijos del conjunto agregado.

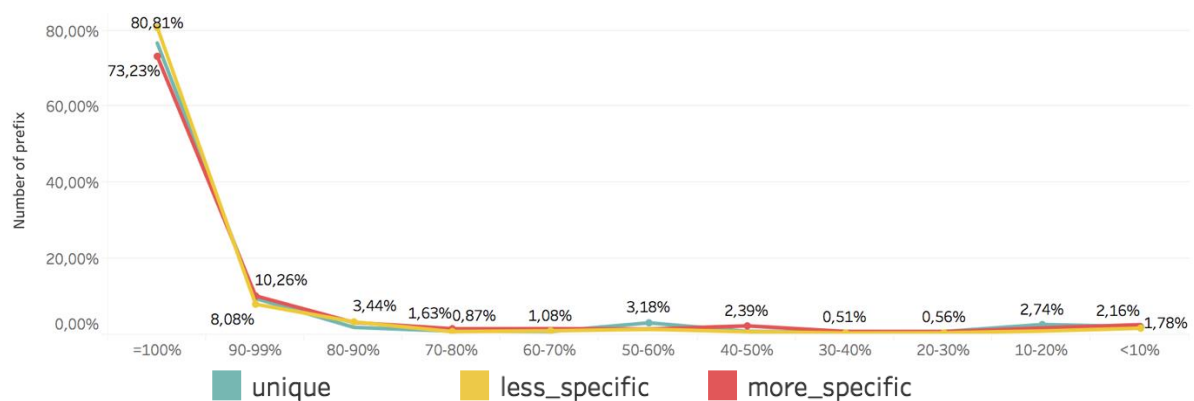


Figura 45: Análisis agregado por tipo de la visibilidad para los prefijos que han recibido algún update durante el experimento

5.3.3 Análisis de la visibilidad por longitud de prefijo atendiendo al tipo de prefijo

En esta sección no se analizan todo el conjunto de los datos para cada tipo en términos de visibilidad ya que está tarea no presenta novedades con respecto al análisis que ha sido realizado previamente.

Por lo tanto, se enfoca el análisis al estudio de la visibilidad de los prefijos que han recibido algún update atendiendo al tipo y a la longitud del prefijo.

Para cada tipo de prefijo se ha seguido el mismo orden de análisis. Primero representan conjuntamente el número de prefijos y el porcentaje de prefijos que han recibido algún update. Después se analiza la visibilidad para los prefijos que han recibido algún update asumiendo que estos prefijos suponen una mejor representación del detalle de la visibilidad por prefijo. La idea es parecida a la presentada en la sección anterior, pero con más profundidad sobre que prefijos son afectados atendiendo a la máscara.

Comenzando con el análisis para los **más específicos** se aprecia lo ya comentado previamente en cuanto al número de prefijos por longitud de prefijo. Este asciende exponencialmente has los /24s y después decae drásticamente reflejando el filtrado a partir de los /25s. También, atendiendo a los datos que presentan la mayoría de los prefijos para este tipo se puede afirmar que el promedio al porcentaje de prefijos que reciben algún update es relativamente estable desde los /14s hasta los /24s presentando un pico importante en los /20s.

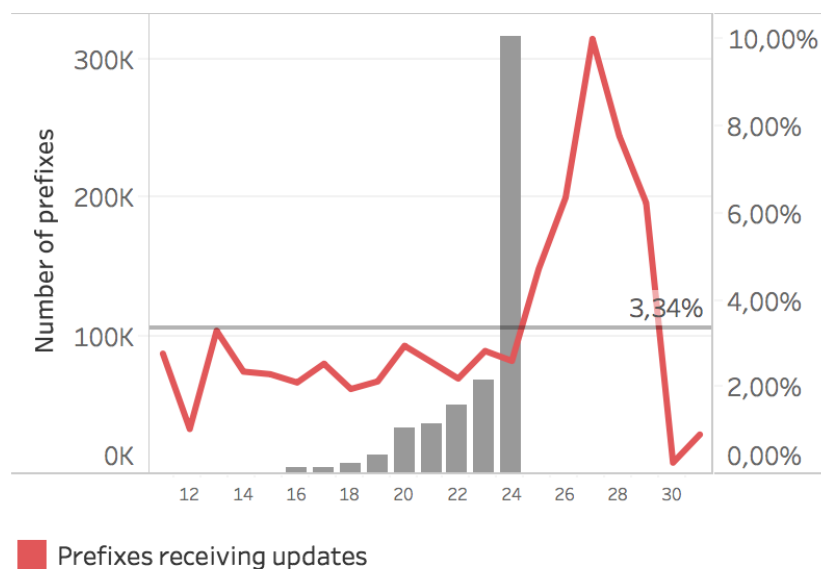


Figura 46: Número de prefijos vs porcentaje de prefijos que reciben updates por longitud de prefijo - más específicos

En cuanto a la visibilidad, el análisis conduce a una interpretación muy acorde con el filtrado de los prefijos mayores a /24 que presentan en su mayoría una visibilidad menor al 10%. Llama la atención que se ven más /32s que /30 mientras que no se ha visto ningún /31 como más específico. Además, presentan una mayor visibilidad, aunque obviamente sigue siendo muy baja.

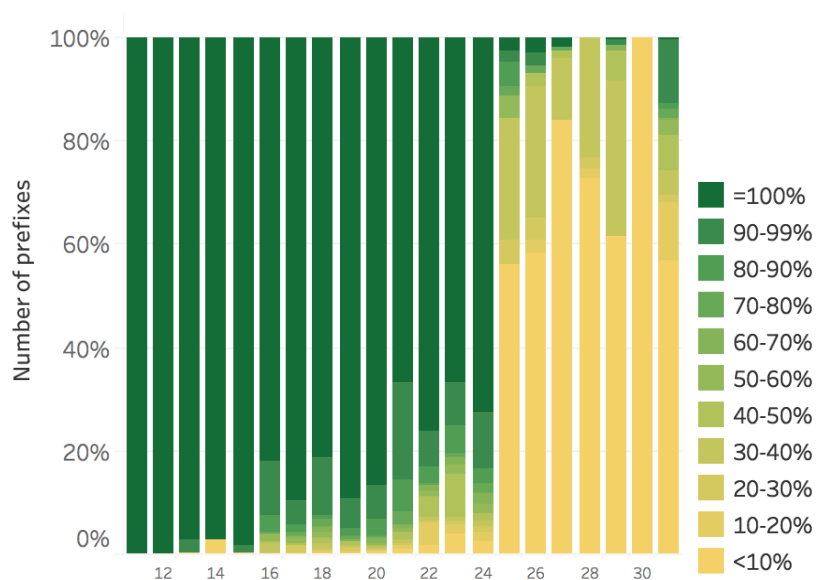


Figura 47: Análisis de visibilidad de los prefijos que reciben algún update por longitud de prefijo – más específicos

Para el caso de los **únicos**, la interpretación del número de prefijos es parecida con la excepción de los /22s que presentan un pico. En lo que al porcentaje de prefijos que reciben updates la tendencia es relativamente ascendiente desde los /12s hasta los /25s presentando picos en los /17s y en los /22s.

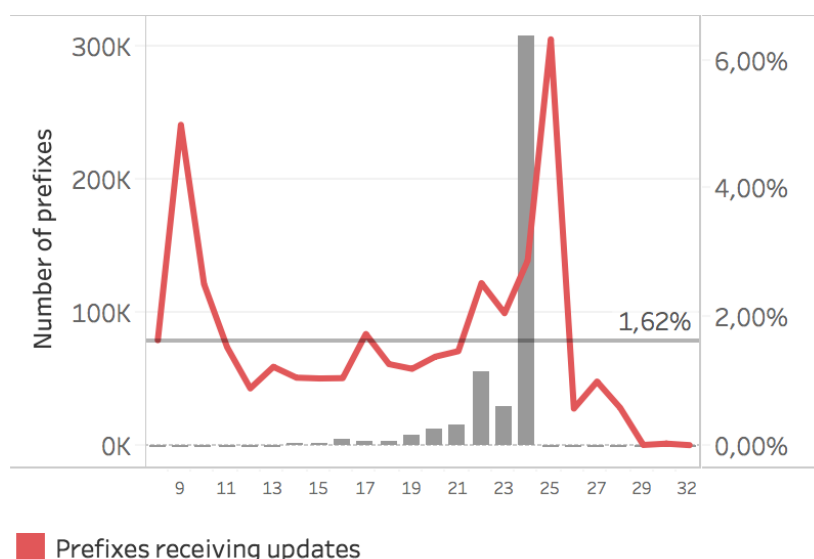


Figura 48: Número de prefijos vs porcentaje de prefijos que reciben updates por longitud de prefijo – únicos

Para la visibilidad, contrastando de nuevo los resultados contra los de los más específicos, no existen demasiadas diferencias, pero si se puede apreciar una menor visibilidad para los prefijos mayores de /25 y una mayor visibilidad para los menores a dicha longitud de prefijo.

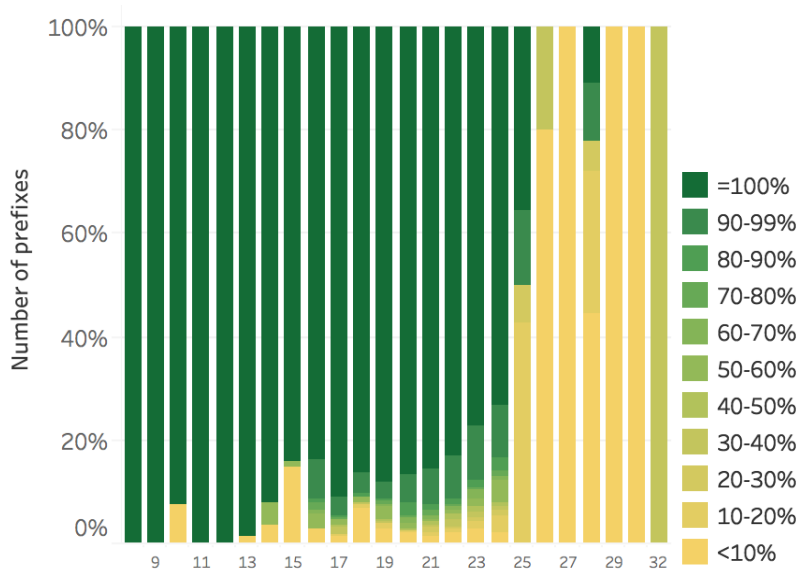


Figura 49: Análisis de visibilidad de los prefijos que reciben algún update por longitud de prefijo - únicos

Nada que ver tienen los resultados obtenidos para los **menos específicos**. Donde el porcentaje de prefijos crece exponencialmente con respecto al número de prefijos.

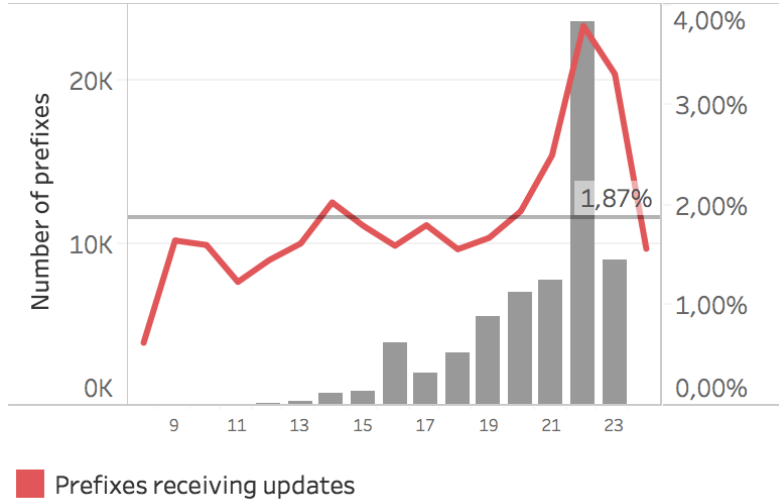


Figura 50: Número de prefijos vs porcentaje de prefijos que reciben updates por longitud de prefijo - menos específicos

Además, el mayor porcentaje de prefijos cambiante corresponde con la longitud de prefijos más frecuente correspondiente a los /22s.

En cuanto a la visibilidad cuenta con un número de longitudes más visibles del 100% del tiempo que para los dos tipos comentados anteriormente.

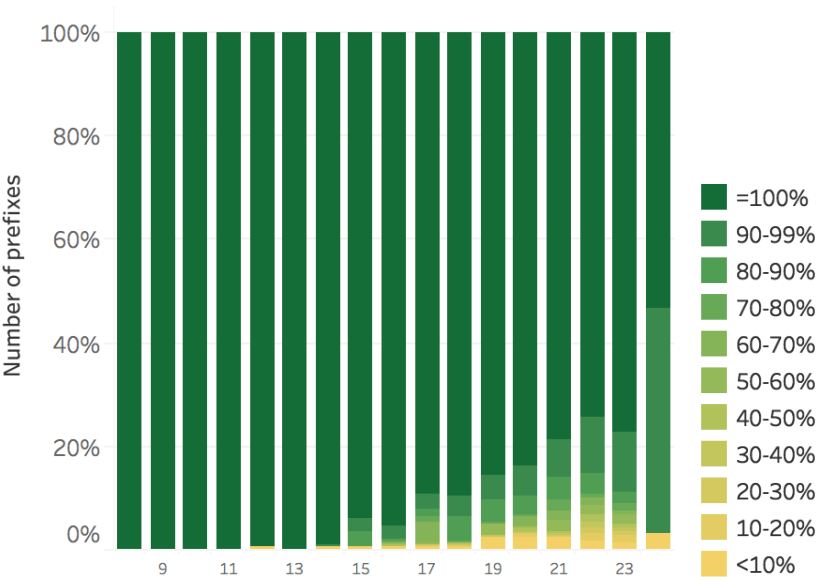


Figura 51: Análisis de visibilidad de los prefijos que reciben algún update por longitud de prefijo - menos específicos

A partir de los menos específicos que son visibles para los rangos menores al 70% del tiempo del experimento se van a hacer una serie de suposiciones. Presentemos en un primer lugar la distribución de estos por máscara.

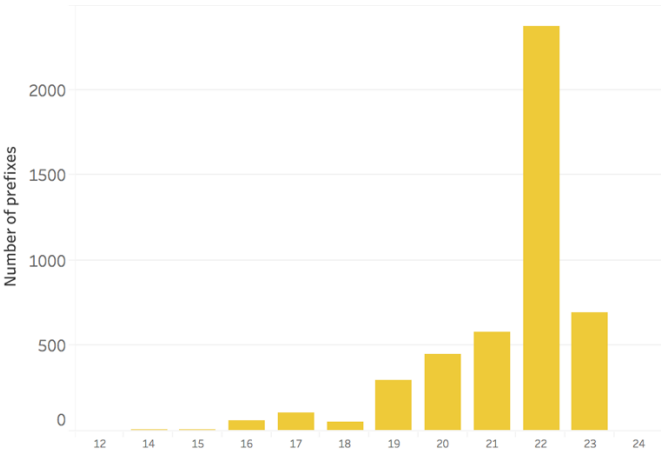


Figura 52: Menos específicos visibles menos que el 70% del tiempo por máscara

Agregando todos los datos de cada longitud de prefijo obtenemos 4.632 menos específicos visibles una fracción de tiempo menor al 70%.

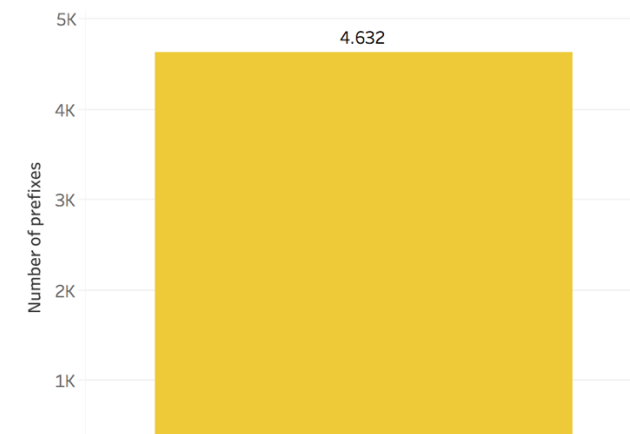


Figura 53: Número de menos específicos visibles una fracción del tiempo menor al 70% del experimento

Atendiendo a los resultados de la sección 5.2.7 se pueden extraer varias conclusiones. El punto de vista de que un más específicos se vea como único sugiere la pérdida de su menos específico. Teniendo en cuenta que el número de más específicos por menos específicos es 8,17 y el número de más específicos que en algún momento son vistos como únicos es de 25.757. Se deduce que para ello han tenido que perderse $\frac{25.757}{8,17} = 3.152,6 \cong 3.153$. Asumiendo que estos prefijos han sido borrados en algún momento del experimento y que aproximadamente un $\frac{3153}{4632} \times 100 = 68\%$ de estos menos específicos es borrado al cabo del experimento se puede hacer una suposición que cuadra bastante con esos resultados de pérdida de los menos específicos.

Esta última conclusión no deja de ser más que una suposición para aclarar en la medida que el experimento lo permite ese resultado anómalo de la sección 5.2.7.

5.4 Comparativa de los resultados obtenidos con el trabajo relacionado

En esta sección se van a comparar parte de los resultados obtenidos para los proveedores respecto a los resultados del análisis relacionado expuesto en la sección 2.4.

Medidas	AS 131.072	Media proveedores
Número de prefijos	680.000	671.808
Número de más específicos	350.000	343.938
% más específicos	51%	51,19%
Espacio de direccionamiento (más específicos)	28%	27,85%
% más específicos-TE	34%	37,2%

Tabla 3: Comparativa de resultados del análisis con el trabajo relacionado

Los resultados de las medidas comparables entre los dos análisis presentan una gran coherencia. Incluso para la suposición que se ha hecho para el caso de las profundidades parece tener bastante sentido teniendo en cuenta que los resultados de Geoff representan una fecha previa al experimento realizado. Con lo que podría tener relación con la limpieza de los prefijos overlays.

6 Presupuesto

En este apartado se determina el coste total del proyecto, determinando cada partida que compone el presupuesto por separado.

6.1 Horas internas

En esta partida se presentan los gastos en recursos humanos, es decir, los sueldos de los trabajadores que se han visto involucrados en el desarrollo del proyecto.

Nombre	Ocupación	Tasa Horaria (€/h)	Horas (h)	Total (€)
Alberto García Martínez	Ingeniero	25	50	1.250
Ignacio Gutiérrez Diez de los Ríos	Ingeniero	10	680	6.800
Subtotal				8.050

Tabla 4: Horas internas

6.2 Amortizaciones

Esta partida contiene gastos de la utilización de bienes que sirven para otros proyectos.

Concepto	Precio unitario (€)	Vida útil (h)	Horas de utilización(h)	Coste (€)
Licencia IDE PyCharm	499	8.760	530	30,19
Ordenador portátil	1.300	10.000	680	88,4
Subtotal				118,59

Tabla 5: Amortizaciones

6.3 Gastos

Se incluyen los gastos que no pueden ser reutilizados en otros proyectos futuros.

Concepto	Coste (€)
Material de oficina	10
Cursos	120
Servidor	600
Repositorio Tableau	10
Subtotal	740

Tabla 6: Gastos

El servidor tiene un coste de 200€ al mes y se ha utilizado durante aproximadamente 3 meses luego aproximadamente unos 600€.

Los gastos dedicados al repositorio remoto se han estimado ya que hay que pagar una licencia mensual de 99€ con la que dispones de una gran cantidad de espacio.

6.4 Costes indirectos

Los costes indirectos son aquellos que no son imputables a ningún proyecto, en ellos se incluyen gastos como la luz o el agua. En los proyectos suelen calcularse como un pequeño porcentaje de los costes directos, en este caso se va a tomar un 2%.

6.5 Imprevistos

A lo largo del desarrollo del proyecto pueden producirse gastos no tenidos en cuenta a la hora de elaborar el presupuesto. Para hacer frente a estos gastos se incluye la partida de imprevistos, que en este caso es de un 5% del presupuesto.

6.6Resumen

Concepto	Coste (€)
Horas internas	8.050
Amortizaciones	118,59
Gastos	740
Subtotal 1	8.908,59
Costes indirectos (2%)	178,17
Subtotal 2	9.086,76
Imprevistos (5%)	454,33
Total	9.0541,1

Tabla 7: Resumen del presupuesto

7 Conclusiones

En este proyecto se ha contribuido al desarrollo de una herramienta de análisis en el ámbito de las Telecomunicaciones. La separación del desarrollo software en módulos facilita la actualización y el futuro desarrollo. Además, permite el testeo de estas unidades discretas de forma independiente.

En lo que se refiere a la distribución de prefijos, no todos los nodos presentan el mismo número de prefijos aprendidos. Los nodos que representan a los proveedores proveen una buena visión de cómo opera el protocolo. En estos, se puede apreciar que la mayoría de los prefijos corresponden al tipo de más específicos (alrededor del 51%). Además, la mayoría de estos prefijos presentan una longitud de prefijo mayor o igual a /22s, siendo mayoritariamente /24s. Por cada prefijo raíz hay una media de 8 prefijos más específicos. Asimismo, la profundidad más típica es de 8 y la mayoría de prefijos más específicos presentan un único nivel de desagregación; lo que indica que es anunciada solo una parte de los prefijos desagregados para cada prefijo raíz.

Por otro lado, analizando cómo son vistos los prefijos por cada monitor, la mayoría son vistos como el tipo anunciado por todos los monitores. Sin embargo, se observa que algunos prefijos raíz/menos específicos se ven como no específicos de ningún prefijo lo que indica que debido a filtrados o fallos de conectividad se han perdido todos los prefijos más específicos. Todavía llama más la atención el hecho de que para algunos casos es el prefijo menos específico el que se ha perdido, pasando los más específicos a ser vistos como no específicos.

Para la visibilidad los más específicos presentan el tipo de prefijos menos estable, aunque la diferencia con respecto a los otros tipos no es tan notable como podía esperarse. Existe relación entre la longitud de prefijos utilizada y la visibilidad ya que para los más específicos y los únicos se observa cómo los filtrados de la red o la pérdida de la conectividad afectan a los prefijos más cortos.

7.1 Dificultades encontradas

En cuanto a las dificultades encontradas destaca el manejo de una gran cantidad de datos que supone una tarea meticulosa. Para un análisis de un gran volumen de datos resulta imprescindible una buena limpieza del dato que evite una mala interpretación de los resultados. Esto supone un proceso iterativo durante el tratamiento de los datos.

Por otro lado, la implementación software de la herramienta exige un testeo y una refactorización constante del código haciendo hincapié en lo que al rendimiento se refiere. Siendo indispensable el uso de librerías que trabajen de una forma eficiente o, en caso contrario, desarrollando las funcionalidades necesarias como ha sido el caso del procesado en bloques. El testeo está enfocado a contemplar diferentes casuísticas logrando el desarrollo de un código robusto.

Por último, la familiarización con los diferentes conceptos que trata el proyecto para llegar a una conclusión sólida de los resultados.

7.2 Trabajo futuro

El desarrollo de una herramienta que estudie la convergencia de los resultados en función del número de ficheros de updates cargados para el experimento. Lo que serviría para concretar el tamaño óptimo de ventana para la realización del análisis.

Asimismo, el estudio de un intervalo temporal que contemple los datos de un intervalo de tiempo anterior. Por ejemplo, el análisis 5 años antes para poder observar cómo han cambiado las políticas de los nodos BGP.

Además, la realización del mismo análisis para los prefijos correspondientes a la versión IPv6.

Finalmente, sería interesante la comparativa de resultados entre monitores pertenecientes a un mismo Sistema Autónomo.

8 Referencias

- [1]. R. Oliveira, B. Zhang, D. Pei, R. Izhak-Ratzin, L. Zhang 'Quantifying Path exploration in the Internet' [en línea]. Disponible en: <https://www2.cs.arizona.edu/~bzhang/paper/06-imc-pathrank.pdf>
- [2]. Visibility of prefix lengths in IPv4 and IPv6 [en línea]. Disponible en: <https://labs.ripe.net/Members/dbayer/visibility-of-prefix-lengths>
- [3]. O. Gerometta 'Prefijos IP' [en línea]. Disponible en: <http://librosnetworking.blogspot.com/2010/09/prefijos-ip.html>
- [4]. 'Longest-match (Static routes) explanation' [en línea]. Disponible en: https://www.juniper.net/documentation/en_US/junos/topics/reference/configuration-statement/longest-match-next-hop-edit-static-routing-options.html
- [5]. 'Alexa An Amazon.com company' [en línea]. Disponible en: <https://www.alexa.com/>
- [6]. Aaron Balchunas 'Border Gateway Protocol' [en línea]. Disponible en: <https://www.routeralley.com/guides/bgp.pdf>
- [7]. Geoff Huston The ISP column 'BGP in 2017' Disponible en: <https://www.potaroo.net/ispcol/2018-01/bgp2017.html>
- [8]. RFC6396 'Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format' [en línea]. Disponible en: <https://tools.ietf.org/html/rfc6396#section-4.4>
- [9]. C. Petrie 'Updates to the RIPE NCC Routing Information Service' [en línea]. Disponible en: https://labs.ripe.net/Members/colin_petrie/updates-to-the-ripe-ncc-routing-information-service
- [10]. RIPE NCC raw data [en línea]: Disponible en: <https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data>
- [11]. J. Oberheide 'pyBGPDump' [en línea]. Disponible en: <https://jon.oberheide.org/pybgpdump/>
- [12]. University of Oregon Route Views Project [en línea]. Disponible en: <http://www.routeviews.org/routeviews/>
- [13]. More about Route Views data [en línea]. Disponible en: <http://www.routeviews.org/routeviews/index.php/archive/>

- [14]. M. Reuter, Ellie Fields 'Descripción general de Tableau' [en línea]. Disponible en: <https://www.tableau.com/es-es/learn/whitepapers/tableau-enterprise>
- [15]. Geoff Huston 'more specifics in BGP' [en línea]. Disponible en: <http://www.potaroo.net/ispcol/2017-06/morespecs.html>
- [16]. PLOS Computational Biology '*Good enough practices in scientific computing*' [en línea]. Disponible en: <http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005510>
- [17]. Código del proyecto 'more specifics analysis through BGP traces' [en línea]. Disponible en: https://github.com/nachodelos/more_specifics_analysis_through_BGP_traces
- [18]. Python Tree implementation for IPv4 and IPv6 prefix matching [en línea]. Disponible en: <https://github.com/mjschultz/py-radix>