



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

ANÁLISIS GEOGRÁFICO Y TOPOLÓGICO DE RUTAS Y SELECCIÓN DE
SERVIDOR EN NUBES ANYCAST DNS

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS, MENCIÓN COMPUTACIÓN

FELIPE ALEJANDRO ESPINOZA ROSALES

PROFESOR GUÍA:
JAVIER BUSTOS JIMÉNEZ

MIEMBROS DE LA COMISIÓN:
PATRICIO INOSTROZA FAJARDIN
LUIS MATEU BRULÉ
FEDERICO MEZA MONTOYA

SANTIAGO DE CHILE
2019

Resumen

El sistema de nombres de dominio (*DNS*, por sus siglas en inglés, Domain Name System) ha pasado a ser una parte fundamental en la infraestructura de Internet, permitiendo el acceso a los diferentes recursos disponibles de manera fácil y rápida. Para manejar la alta cantidad de información que estos servicios reciben, se han implementado distintos métodos de balanceo de carga, permitiendo reducir los tiempos de respuesta e incrementar la disponibilidad de estos. Una de estas metodologías corresponde a *anycast*, la cual fue analizada en este trabajo.

Anycast es una metodología de enrutamiento que nos permite realizar el anuncio de la misma dirección IP desde diferentes localizaciones en internet, permitiendo a los *Sistemas Autónomos* intermedios enrutar la información de los clientes al servidor más cercano.

En esta tesis, se buscó desarrollar diferentes metodologías que nos permitan realizar el análisis geográfico y topológico de una nube *anycast DNS*, determinando las áreas de servicio y rutas utilizadas por los clientes para conectarse a cada uno de los nodos que la componen, además de buscar posiciones óptimas en la red según diferentes parámetros.

Este trabajo se dividió en las etapas de *Captura de Datos*, *Análisis Geográfico*, *Análisis Topológico* y *Posicionamiento de Nueva Infraestructura*, donde en cada etapa se propuso una metodología para realizar los análisis propuestos. Cada una de estas metodologías fue aplicada a la nube en producción *a.nic.cl* de *NIC Chile*, analizando cuatro nodos *anycast* posicionados en América Latina, permitiendo así verificar los resultados en una nube altamente utilizada.

La *Captura de Datos* se realizó directamente en los nodos *anycast* analizados, realizando trazados de rutas a diferentes direcciones IP representativas de las redes que componen internet. A través de la información capturada, logramos determinar las áreas de servicio de cada uno de los nodos *anycast* analizados.

Utilizando esta información, se realizó un *Análisis Geográfico* de las áreas de servicio, donde logramos encontrar nodos con un área geográfica mucho más grande que lo esperado. Además de esto, comparamos las distancias geográficas de las redes hasta los distintos nodos que componen la nube *anycast*, encontrando que las asignaciones generadas por la red no se realizan hacia el nodo más cercano geográficamente en la mayoría de los casos.

El *Análisis Topológico* nos mostró resultados que poseen una alta relación con los vistos en el *Análisis Geográfico*, donde además realizamos una comparación de las distancias topológicas en términos de latencia, desde las redes hasta los nodos *anycast*, concluyendo que las asignaciones realizadas no logran seleccionar el servidor con menor latencia en una gran parte de los casos.

Por último, la metodología de *Posicionamiento de Nueva Infraestructura*, nos mostró los efectos de realizar la optimización de diferentes parámetros, cuantificando el efecto de posicionar un nuevo nodo *anycast* en la red. La verificación realizada utilizando la nube *anycast PEERING* fue afectada por la gran conectividad de uno de los nodos, por lo que no fue posible verificar la metodología exitosamente. A pesar de esto, las predicciones resultantes de esta metodología pueden ser utilizadas por los operadores de las nubes *anycast* como una base para seleccionar una posición en la red para realizar la instalación de nueva infraestructura.

Agradecimientos

A mi madre, Marisol Rosales Lecaros, por estar presente en todo momento con su cariño y apoyo.

A todos mis amigos y compañeros, antiguos y nuevos que siempre estuvieron apoyándome para dar un paso más adelante.

A Javier Bustos Jiménez, por brindarme esta oportunidad, además de su apoyo y motivación en todo este proceso.

A todos en el NICLabs, por su apoyo y risas que compartimos en todo momento.

Tabla de Contenido

1. Introducción	1
1.1. Contexto	1
1.1.1. Sistema de Nombres de Dominio	2
1.1.2. Enrutamiento de Paquetes	2
1.1.3. Anycast	3
1.2. Trabajos Relacionados	3
1.3. Hipótesis	5
1.4. Objetivos	5
1.4.1. Objetivo General	5
1.4.2. Objetivos Específicos	5
1.5. Metodología de Trabajo	6
1.6. Fuente de Datos	7
2. Captura de Información	9
2.1. Introducción	9
2.2. Metodología	9
2.2.1. División de Internet	9
2.2.2. Lista de Direcciones IP	10
2.2.3. Captura de Datos	11
2.2.3.1. Metodología Verfploeter	11
2.2.3.2. Modificaciones de Metodología de Captura	12
2.2.4. Verificación Área de Servicio	14
2.3. Resultados	14
2.3.1. Direcciones IP a medir	14
2.3.2. Captura de Datos	15
2.3.2.1. Medición Unicast	15
2.3.2.2. Medición Nube Anycast NIC Chile	16
2.3.3. Verificación Área de Servicio	17
3. Análisis Geográfico	18
3.1. Introducción	18
3.2. Metodología	19
3.3. Resultados	20
3.3.1. Área de Servicio por País	20
3.3.1.1. arica	20
3.3.1.2. merced	21

3.3.1.3.	saopaulo	22
3.3.1.4.	tucapel	23
3.3.1.5.	Análisis de resultados	23
3.3.2.	Sistemas Autónomos por país	24
3.3.2.1.	arica	24
3.3.2.2.	merced	25
3.3.2.3.	saopaulo	26
3.3.2.4.	tucapel	27
3.3.2.5.	Análisis de resultados	27
3.3.3.	Área de Servicio por País Ponderada	28
3.3.3.1.	arica	28
3.3.3.2.	merced	29
3.3.3.3.	saopaulo	30
3.3.3.4.	tucapel	31
3.3.3.5.	Análisis de resultados	31
3.3.4.	Asignación de Nodo Según Localización	32
3.3.4.1.	arica	32
3.3.4.2.	merced	33
3.3.4.3.	saopaulo	34
3.3.4.4.	tucapel	35
3.3.4.5.	Análisis de resultados	35
3.3.5.	Asignación de Nodo según Localización Ponderada por Tráfico	36
3.3.5.1.	arica	36
3.3.5.2.	merced	37
3.3.5.3.	saopaulo	38
3.3.5.4.	tucapel	39
3.3.5.5.	Análisis de resultados	39
3.4.	Conclusiones	40
4.	Análisis Topológico	42
4.1.	Introducción	42
4.2.	Metodología	42
4.3.	Resultados	44
4.3.1.	Redes por Latencia	44
4.3.1.1.	arica	44
4.3.1.2.	merced	45
4.3.1.3.	saopaulo	46
4.3.1.4.	tucapel	47
4.3.1.5.	Análisis de resultados	47
4.3.2.	Sistemas Autónomos por Latencia	48
4.3.2.1.	arica	48
4.3.2.2.	merced	49
4.3.2.3.	saopaulo	50
4.3.2.4.	tucapel	51
4.3.2.5.	Análisis de resultados	51
4.3.3.	Tráfico Ponderado por Latencia	52
4.3.3.1.	arica	52

4.3.3.2.	merced	53
4.3.3.3.	saopaulo	54
4.3.3.4.	tucapel	55
4.3.3.5.	Análisis de resultados	55
4.3.4.	Sistemas Autónomos por Salto	56
4.3.4.1.	arica	56
4.3.4.2.	merced	57
4.3.4.3.	saopaulo	58
4.3.4.4.	tucapel	59
4.3.4.5.	Análisis de resultados	59
4.3.5.	Tránsito por Sistemas Autónomos	60
4.3.5.1.	arica	60
4.3.5.2.	merced	61
4.3.5.3.	saopaulo	62
4.3.5.4.	tucapel	63
4.3.5.5.	Análisis de resultados	63
4.3.6.	Asignación Topológica	64
4.4.	Conclusiones	64
5.	Posicionamiento de Nueva Infraestructura	66
5.1.	Introducción	66
5.2.	Metodología	66
5.2.1.	Metodologías de Optimización	67
5.2.2.	Verificación	68
5.3.	Resultados	68
5.3.1.	Minimizar Latencia	69
5.3.1.1.	Minimizar Latencia Bruta	69
5.3.1.2.	Minimizar Latencia Ponderada	70
5.3.1.3.	Minimizar Latencia Ponderada Limitando Saltos	72
5.3.2.	Minimizar Saltos	73
5.3.2.1.	Minimizar Saltos Brutos	73
5.3.2.2.	Minimizar Saltos Ponderados	75
5.3.2.3.	Minimizar Saltos Ponderados Limitando Saltos	76
5.3.3.	Verificación	78
5.4.	Conclusión	79
6.	Conclusiones y Trabajo Futuro	81
6.1.	Conclusiones	81
6.2.	Trabajo Futuro	83
7.	Bibliografía	85
8.	Apéndice	88
8.1.	Apéndice 1: Resultados Minimizar Latencia Ponderada Limitando Saltos	88
8.1.1.	Minimizar Latencia Ponderada Limitando a 1 Salto	88
8.1.2.	Minimizar Latencia Ponderada Limitando a 2 Saltos	90
8.1.3.	Minimizar Latencia Ponderada Limitando a 3 Saltos	91

8.1.4.	Minimizar Latencia Ponderada Limitando a 4 Saltos	92
8.2.	Apéndice 2: Minimizar Saltos Ponderados Limitando Saltos	93
8.2.1.	Minimizar Saltos Ponderados Limitando a 1 Saltos	93
8.2.2.	Minimizar Saltos Ponderados Limitando a 2 Saltos	94
8.2.3.	Minimizar Saltos Ponderados Limitando a 3 Saltos	95
8.2.4.	Minimizar Saltos Ponderados Limitando a 4 Saltos	96
8.3.	Apéndice 3: Ruteo Entre Sistemas Autónomo	97
8.3.1.	Peering Amsterdam	97
8.3.2.	Peering UFMG	98
8.3.3.	Peering Seattle	99

Capítulo 1

Introducción

1.1. Contexto

El *Sistema de Nombres de Dominio* (*DNS*, por sus siglas en inglés, *Domain Name System*) es utilizado para acceder a los diferentes recursos disponibles en internet por una gran cantidad de personas y software. La resolución de los nombres pertenecientes a este sistema utiliza aproximadamente el 20 % del tiempo de carga de una página web [12], lo cual ha llevado a los administradores de los servicios *DNS* a implementar diferentes métodos para mejorar la velocidad de respuesta, reducir la latencia y mejorar la experiencia de usuario.

Uno de estos métodos corresponde a *anycast* [19], el cual realiza el anuncio de una misma dirección de red en múltiples puntos de internet, permitiendo acercar la infraestructura a los usuarios en términos geográficos y topológicos, reduciendo la latencia e incrementando la resiliencia de los servicios, al ingresar servidores en redes independientes de manera transparente para el usuario.

Dado que la administración de las redes de internet es de carácter privado, obtener información sobre las tablas de rutas utilizadas por las diferentes redes para comunicar a clientes y servidores en internet pasa a ser una tarea difícil, y así observar los efectos generados en estas tablas por el despliegue de nodos *anycast* en diferentes lugares de la red, donde cada red decide de manera independiente cuál es el servidor *anycast* más cercano con el cual se comunicará.

Obtener conocimiento sobre la asociación entre las redes y los servidores de una nube *anycast* nos permitirá evaluar los despliegues actuales y predecir la calidad de servicio percibida por los usuarios, además de permitir optimizar la red actualmente desplegada y realizar una mejor planificación para la instalación de nueva infraestructura.

1.1.1. Sistema de Nombres de Dominio

El *Sistema de Nombres de Dominio* es un sistema distribuido que busca generar un espacio de nombres consistente para referirse a diferentes recursos accesibles en internet o redes privadas. Este sistema hoy es generalmente utilizado para realizar un mapeo entre un nombre de dominio y una dirección IP (por ejemplo, el nombre de dominio *a.nic.cl* y su dirección IPv4 asociada *190.124.27.10*).

Los nombres de dominio se dividen de manera jerárquica en niveles separados por un punto, en los cuales los primeros tres niveles principales corresponden al *Dominio Raíz*, *Dominio de Nivel Superior* (Ejemplo: *com.*, *cl.*) y *Dominio de Nivel Secundario* (Ejemplo: *uchile.* en el dominio *www.uchile.cl*). Cada nivel en esta jerarquía se encarga de administrar la ubicación del nivel siguiente de manera independiente, donde cada consulta comienza desde el *dominio raíz* hacia los niveles más bajos.

Esta arquitectura permite realizar una separación clara de responsabilidades entre los diferentes administradores, permitiéndoles funcionar de manera independiente y distribuir la carga entre los diferentes proveedores de este servicio.

Los servidores DNS han pasado a ser una parte crítica del funcionamiento de la internet, manejando una gran cantidad de consultas generadas por personas y software (uno de los trece servidores raíz recibe como mínimo 70.000¹ consultas por segundo). Para asegurar su correcto funcionamiento, y manejar la gran cantidad de información que estos servicios reciben, se han implementado distintos métodos de balanceo de carga, permitiendo reducir los tiempos de respuesta e incrementar la disponibilidad de estos. Una de estas metodologías corresponde a *Anycast*, sobre la cual nos enfocaremos en este trabajo.

1.1.2. Enrutamiento de Paquetes

Para realizar el enrutamiento de paquetes, la internet pública se divide en diferentes *Sistemas Autónomos* (*AS*, por sus siglas en inglés, *Autonomous System*), los cuales son grupos de routers, redes y rutas IP independientes e interconectadas.

Los *Sistemas Autónomos* se comunican entre sí a través de un protocolo denominado *Border Gateway Protocol* (*BGP*, por sus siglas en inglés) [21]. Este protocolo permite a los *Sistemas Autónomos* intercambiar información sobre la accesibilidad de la red, incluyendo una lista sobre los *Sistemas Autónomos* que esta información ha recorrido, permitiendo generar un grafo sobre la conectividad de esta red, eliminando ciclos que puedan generarse en el ruteo.

A través de este protocolo, es posible realizar el anuncio de la disponibilidad de un *Prefijo de Red*, el cual es una agregación de direcciones IP, definida por una dirección IP base y un número de bits mínimo que deben coincidir desde el bit más significativo de la IP base. Así, un *prefijo de red* tal como *190.127.27.0/24*, define una IP base *190.127.27.0* y *24* bits

¹Estadísticas servidor *DNS* raíz L: <http://stats.dns.icann.org/> [Accedido: 25/06/2018]

significativos, encapsulando las direcciones IP desde *190.127.27.0* a *190.127.27.255*.

Al realizar el anuncio de un *prefijo de red* utilizando el protocolo *BGP*, los *Sistemas Autónomos* vecinos actualizarán sus tablas de rutas, y propagarán a sus propios vecinos la disponibilidad de esta red, agregando su *Número de Sistema Autónomo* (*ASN*, por su sigla en inglés, *Autonomous System Number*) a la ruta anunciada, para así generar el grafo de red y prevenir la generación de ciclos.

Gracias a esto, cada *Sistema Autónomo* puede determinar el camino más corto en termino de saltos a una red IP anunciada en internet. Sin embargo, esto no considera otros factores tales como congestión, velocidad de conexión, o políticas de enrutamiento que pueden cambiar la priorización de las rutas utilizadas. Esto es producido por la independencia en las operaciones de los *Sistemas Autónomos*, donde la información sobre las conexiones que esto poseen es de carácter privado, lo que previene realizar optimizaciones de manera directa.

1.1.3. Anycast

Anycast es una metodología que hace disponible una red *IP* a un sistema de ruteo desde dos o más nodos. El sistema de ruteo realiza la decisión sobre que nodo es utilizado para responder la consulta de un usuario, basado en la topología de la red y el punto en que se genera esta consulta.

Para utilizar la metodología *anycast* en la internet pública, se utiliza el protocolo *BGP* para anunciar un mismo prefijo desde diferentes puntos. Esto genera que los *Sistemas Autónomos* en internet observen diferentes caminos disponibles para un mismo prefijo, y así lograr elegir el que tome la menor cantidad de saltos para acceder a este.

Esto permite generar un balanceo de carga transparente para los usuarios, dado que una dirección IP puede representar más de un nodo a la vez, y según se vayan sumando o removiendo los anuncios en la red, los caminos se rebalancearán automáticamente hacia los otros nodos.

Posicionar nodos de manera distribuida en el mundo involucra observar el origen del tráfico que se desea abarcar, y realizar la instalación de infraestructura de manera cercana a esta, ya sea de manera topológica o geográfica. Muchas veces estas decisiones se realizan de manera geográfica, esperando cubrir el tráfico de un país desde un nodo en específico, sin embargo, los efectos del posicionamiento de un nodo no son del todo claro, dada la naturaleza del protocolo *BGP*, y el desconocimiento de las políticas de ruteo de los *Sistemas Autónomos* vecinos.

1.2. Trabajos Relacionados

Las redes *anycast* han sido un gran punto de análisis para los científicos, especialmente sobre el sistema *DNS*, el cual ha utilizado *anycast* por una gran cantidad de tiempo para

asegurar su disponibilidad y correcto funcionamiento. Hoy el uso de las redes *anycast* se ha expandido, siendo utilizada por *Redes de Distribución de Contenidos* (*CDN*, por sus siglas en inglés, *Content Delivery Network*) para permitir a usuarios obtener la mejor velocidad de transferencia de recursos en internet [6].

Algunos de los análisis desarrollados en estas redes han buscado evaluar a largo plazo la disponibilidad y efectividad de diferentes despliegues, mostrando el efecto de diferentes eventos sobre cambios en la red [16]. Por otro lado, se han realizado estudios sobre la estabilidad de las redes *anycast* [20], donde se ha encontrado que las rutas *anycast* son mayoritariamente estables, permitiéndonos realizar una mejor planificación de los despliegues de estos servidores.

Para observar las áreas de servicios de las redes *anycast*, se han desarrollado distintos métodos, de los cuales podemos apreciar cuatro de los más conocidos a continuación.

- **Análisis de logs y capturas de paquetes:** Este método de análisis permite a los operadores de los servicios *anycast* obtener las fuentes desde donde se genera el tráfico de manera directa. Este método posee la ventaja de entregar información exacta de los lugares en los que sus servicios son utilizados, y los volúmenes de información sobre cada una de las redes [17]. Este tipo de análisis no entrega información sobre la topología de la red sobre la que se trabaja, lo cual no nos permite realizar un análisis a fondo sobre las rutas que se utilizan.
- **Servidores *DNS* Recursivos:** Este método utiliza servidores *DNS* recursivos que se encuentran abiertos en internet, realizando peticiones de resolución a un dominio que se encuentre en la nube *anycast* [9]. Este método permite utilizar un amplio número de puntos de medición para la detección del área de servicio de cada servidor, y nos entrega una visión real del enrutamiento de paquetes *UDP* o *TCP*, el cual puede ser diferente al utilizado por paquetes *ICMP* utilizados en otros métodos. La desventaja de este método se encuentra presente en la obtención, localización y número de servidores recursivos que es posible utilizar, la cual puede ser baja y lejana de los clientes.
- **Análisis de tablas *BGP*:** Este método utiliza bases de datos públicas que contienen las tablas de rutas *BGP* actuales e históricas desde distintos puntos de medición en la internet [13]. Esto permite observar desde diversos lugares las rutas utilizadas por los *AS* para realizar la transmisión de paquetes hacia los servidores. La desventaja de este método corresponde a la cantidad de puntos de observación actualmente disponible, la cual es muy limitada y generalmente no logra abarcar la totalidad de los servidores desplegados en una nube *anycast*, además de no entregar información sobre a qué servidor corresponde la ruta observada.
- **Redes de medición:** Utilizando plataformas con una gran cantidad de puntos de medición, es posible realizar un análisis del estado de una red *anycast* de manera externa. Este tipo de análisis realiza mediciones de valores tales como el *RTT* (*Round Trip Time*) desde diferentes localizaciones [5], trazados de ruta, o utiliza propiedades del software ejecutado en la nube *anycast*, tal como el envío de consultas *CHAOS* a servidores *DNS* para la identificación de estos.

Este tipo de mediciones permite obtener una visión externa de los servicios. Sin embargo, no logran realizar una observación completa sobre los posibles clientes en internet, dado que los puntos de observación no se encuentran presentes en todas las redes des-

plegadas. Plataformas tales como *RIPE Atlas* proveen aproximadamente 10,000 puntos de observación que permiten una buena visión sobre la red [7], sin embargo, generalmente estos se encuentran sesgados a lugares con una gran concentración de usuarios, principalmente Estados Unidos y Europa, lo cual produce que mediciones en lugares tales como Sudamérica y Asia sean poco representativas.

Las últimas investigaciones generalmente utilizan redes de medición sobre nubes *anycast*, para obtener información sobre rutas, áreas de servicio y velocidad de respuesta, y en caso de poseer acceso a la red *anycast*, se han desarrollado mediciones utilizando paquetes *ICMP* para establecer los bloques IP sobre los cuales cada servidor *anycast* trabaja, tal como el método de *Verfploeter* [8].

1.3. Hipótesis

En una nube anycast DNS, el servidor más cercano al cliente en términos de su distancia geográfica y latencia es el que siempre responde las consultas DNS.

1.4. Objetivos

Este trabajo presenta el siguiente objetivo general y los siguientes objetivos específicos.

1.4.1. Objetivo General

Analizar las áreas de servicio de los diferentes servidores en una nube *anycast DNS*, determinando las rutas utilizadas y si estas corresponden a las óptimas en términos topológicos (latencia entre el servidor y sus clientes) y geográficos, además de buscar el posicionamiento de nuevos servidores en la red.

1.4.2. Objetivos Específicos

- Realizar un sondeo para determinar los routers o *Sistemas Autónomos* que se encuentran en el área de servicio de cada servidor en la nube *anycast DNS*.
- Calcular la distancia geográfica de los servidores a los clientes en sus áreas de servicio, comparando esta con la distancia geográfica a otros servidores.
- Calcular la distancia topológica en términos de los *hops* (*saltos*, por su traducción del inglés) y latencia de los servidores hacia los clientes en sus áreas de servicio, comparando esta con la distancia topológica a otros servidores.
- Calcular rutas óptimas teóricas a los clientes en termino de latencia y comparar con las rutas actualmente utilizadas.

- Determinar posiciones óptimas en la red para establecer nuevos servidores *anycast DNS*, minimizando la latencia entre la nube *anycast* a los clientes.

1.5. Metodología de Trabajo

Para desarrollar los diferentes objetivos presentados por esta tesis, el trabajo se dividió en cuatro diferentes etapas, cada una de estas enfocada a una parte específica de este trabajo.

1. **Captura de Información:** En esta etapa, se definirá una metodología que nos permita obtener las diferentes áreas de servicio, en términos de los prefijos de red, con los que cada nodo perteneciente a una nube *anycast* se comunica.

Esta metodología será ejecutada sobre una nube *anycast* en producción, obteniendo estadísticas sobre los resultados generados desde los nodos analizados.

Para realizar la verificación de resultados, se utilizará una plataforma de medición externa, junto con la cual se compararán las áreas de servicio calculadas con los nodos de medición.

2. **Análisis Geográfico:** En esta etapa, definiremos una metodología que nos permita utilizar la información de las áreas de servicio previamente generada, para obtener información geográfica sobre las redes analizadas.

Esto nos permitirá observar las diferentes áreas de influencia geográfica que posee cada nodo de la nube *anycast*, además de permitirnos observar irregularidades en la asignación actual.

Por último, definiremos y ejecutaremos una metodología para realizar la verificación de una de nuestras hipótesis sobre la asignación de nodos en términos geográficos.

3. **Análisis Topológico:** En esta etapa, se desarrollará una metodología que nos permita observar el comportamiento de los nodos *anycast* analizados desde un punto de vista topológico.

Esto se realizará analizando los datos obtenidos según los *Sistemas Autónomos* afectados, latencia y saltos requeridos para acceder a cada uno de estos, contrastándolos con los resultados geográficos que se observaron anteriormente.

Además de esto, definiremos y ejecutaremos una metodología que nos permita verificar la segunda parte de nuestra hipótesis, referida a la selección de servidor en una nube *anycast* según su distancia topológica en términos de latencia.

4. **Posicionamiento de Nueva Infraestructura:** En esta última etapa, buscaremos utilizar la información anteriormente obtenida para calcular el efecto de posicionar un nuevo nodo *anycast* en internet.

Para esto, definiremos diferentes parámetros y métodos que nos permitan observar los efectos del posicionamiento de estos servidores en la nube, mostrando las diferencias de latencia, tráfico y *Sistemas Autónomos* afectados.

Finalmente, utilizando una nube *anycast* pública de pruebas, realizamos la verificación de la metodología propuesta.

1.6. Fuente de Datos

En este trabajo se utilizaron diversas fuentes de información para generar los resultados que serán expuestos en los siguientes capítulos.

Las fuentes de datos utilizadas, en conjunto con una descripción de estas, se presentan en la siguiente lista.

- **Nube *Anycast DNS NIC Chile***: Se utilizó la nube *anycast a.nic.cl* de *NIC Chile* para realizar las pruebas de las metodologías propuestas. Estas se ejecutaron desde los mismos servidores que realizan la resolución de los dominios *.cl*, asegurándonos que los datos obtenidos corresponden a una nube utilizada actualmente.
- **Capturas de paquetes *DNS - NIC Chile***: Se utilizaron capturas de datos realizadas por *NIC Chile* sobre sus servicios, con el fin de extraer las fuentes de tráfico que recibe esta nube, en conjunto con el volumen de información que cada prefijo IP genera.
- **Nube *Anycast PEERING Testbed – University of Southern California y Universidade Federal de Minas Gerais***: Esta nube *anycast* se utilizó para realizar la verificación de la metodología propuesta para posicionar nuevos servidores, realizando el anuncio *BGP* desde tres localizaciones distribuidas geográficamente.
- ***RIPE Atlas - RIPE NCC***: Esta corresponde a una plataforma que permite realizar mediciones topológicas desde una gran cantidad de puntos de observación distribuidas por el mundo. Esta fue utilizada para realizar verificaciones y mediciones de latencia a los diferentes nodos pertenecientes a la nube *anycast* a través de consultas *ICMP* y *DNS*.
- **Capturas *BGP* en formato *MRT - Routing Information Service - RIPE NCC***: Los archivos *MRT* (por sus siglas en inglés, *Multi-Threaded Routing Toolkit*) nos entregan información sobre los anuncios realizados por los diferentes *Sistemas Autónomos* que componen la internet pública. Esto nos permite obtener los diferentes prefijos que cada uno de estos maneja, y así realizar un mapeo entre las direcciones IP capturadas y su *Sistema Autónomo* correspondiente.
- ***RAdb - The Internet Routing Registry***: Esta base de datos contiene información sobre los diferentes registros de *Sistemas Autónomos*, agregando bases de datos como *AFRINIC*, *ARIN* y *RIPE*, entre otras. Esta fue utilizada para complementar el mapeo realizado por las capturas *MRT*, donde existen direcciones IP que no se encuentran anunciadas, pero si se encuentran en los trazados de ruta calculados.
- ***GeoLite2 – MaxMind***: Esta base de datos fue utilizada para realizar la geolocalización de los diferentes prefijos de red capturados desde las nubes *anycast*.

La localización y nombres de dominio de cada servidor analizado se presentan en la tabla 1.1 y la figura 1.1, los cuales componen la nube *anycast a.nic.cl* de *NIC Chile*.

Nodo	Ubicación	Nombre de Dominio
<i>arica</i>	Chile	arica.nic.cl
<i>merced</i>	Chile	merced.nic.cl
<i>tucapel</i>	Chile	tucapel.nic.cl
<i>saopaulo</i>	Chile	saopaulo.nic.cl
<i>praga</i>	República Checa	praga.nic.cl
<i>amsterdam</i>	Países Bajos	amsterdam.nic.cl
<i>elsegundo</i>	Estados Unidos	elsegundo.nic.cl
<i>monterreya</i>	Mexico	monterreya.nic.cl

Tabla 1.1: Esta tabla muestra la ubicación geográfica y el nombre de dominio de los nodos pertenecientes a la nube *anycast a.nic.cl* de *NIC Chile*, la cual fue utilizada en este trabajo.



Figura 1.1: Mapa geográfico con los nodos de la nube *anycast a.nic.cl* presentados en la tabla 1.1. Los nodos destacados en azul corresponden a los nodos utilizados en los experimentos de manera directa, y los rojos solo se interactuó de manera externa.

Capítulo 2

Captura de Información

2.1. Introducción

La metodología *anycast* permite a los usuarios conectarse al nodo más cercano utilizando una única dirección IP, donde cada *Sistema Autónomo* selecciona la ruta más corta a la nube *anycast*.

Esta asignación es difícil de conocer, dado el carácter privado en la operación de los *Sistemas Autónomos* que componen internet, y así las tablas de rutas que estos poseen. Esto produce que optimizar los despliegues *anycast* actuales y planear la instalación de nueva infraestructura sea una tarea mucho más difícil para los operadores de estas.

En este capítulo, se describirá una de las últimas metodologías definidas para realizar la medición del área de servicio de una nube *anycast*, y se propondrán modificaciones para incrementar la información capturada por este proceso, permitiéndonos obtener una vista mucho mejor del estado de una nube *anycast*, observando las rutas utilizadas por los clientes y sus características.

2.2. Metodología

En esta sección definiremos la metodología utilizada para realizar la captura de información sobre los diferentes nodos de una nube *anycast*.

2.2.1. División de Internet

Para realizar el enrutamiento de paquetes, los *Sistemas Autónomos* utilizan el protocolo *BGP* para intercambiar las rutas hacia los diferentes recursos que se encuentran en la red. Estas rutas representan a un conjunto de direcciones IP que pueden ser accedidas desde cada

uno de estos, evitando almacenar las 2^{32} direcciones IP posibles en las tablas de rutas que estos poseen.

Las agrupaciones se generan representando a una dirección IP como una secuencia de bits, donde se selecciona el largo de prefijo deseado, el cual representa todas las direcciones IP que comienzan con este.

Por ejemplo, si indicamos una dirección IP *11.22.33.00* con un prefijo de red de largo 24 bits, este representará a todas las IP que se encuentren en el rango de direcciones *11.22.33.00* a *11.22.33.255*. Los prefijos generalmente se abrevian en la forma *11.22.33.00/24*, lo cual simboliza que nos referimos a una red IP *11.22.33.00* con un prefijo de largo 24 bits.

Para el caso del protocolo *BGP* utilizado en la internet abierta, generalmente se utiliza como un largo de anuncio máximo de 24 bits, para así evitar que las tablas de rutas crezcan de manera incontrolable.

Por este motivo, se decidió dividir el espacio de direcciones IP en agrupaciones con prefijos de largo /24, seleccionando una dirección IP representativa de cada grupo, para así disminuir el tiempo necesario para realizar las mediciones, al reducir el espacio de búsqueda utilizado de 2^{32} a 2^{24} .

2.2.2. Lista de Direcciones IP

Para realizar la elección de los representantes de cada prefijo de red /24, se utilizó el protocolo *ICMP* (*Internet Control Message Protocol*, por sus siglas en inglés) para detectar diferentes clientes activos en internet.

ICMP es un protocolo de control utilizado por diferentes dispositivos de red para indicar mensajes o errores ocurridos en la transmisión de paquetes en internet. Este protocolo posee un tipo de mensaje denominado *Echo*, el cual indica al receptor del paquete que debe responder al emisor del paquete con el mismo contenido que la consulta. Este tipo de mensaje es utilizado para verificar si un cliente se encuentra activo en internet, y además calcular la latencia a este. Además de esto, si el tiempo de vida (*TTL*, por sus siglas en inglés, *Time to live*) de un paquete termina antes de llegar a su destino, el router en el cual este expira generará un código de operación *Time Exceeded* con información de este router, permitiéndonos trazar la ruta de este paquete al enviar múltiples consultas con un *TTL* incremental.

Es importante tener en cuenta, que existen sistemas que no responden o filtran los paquetes *ICMP*, lo cual dificulta el descubrimiento de estos sistemas.

Para elegir a los representantes de cada prefijo de red /24, se realizó un sondeo en toda internet en busca de nodos que respondan a consultas *ICMP Echo*, anotando las direcciones IP que responden satisfactoriamente.

Para acelerar esta medición, este sondeo se realizó enviando las consultas a las direcciones IP de manera desordenada, lo cual permite marcar un prefijo de red /24 como medido, y así prevenir el envío de consultas extra al mismo prefijo, disminuyendo en gran manera el

tiempo de sondeo. La lista resultante con las direcciones IP representante de cada prefijo /24 es denominada como *Hitlist* [10].

2.2.3. Captura de Datos

Para realizar la captura de datos, se utilizará como base la metodología *Verfploeter* [8], realizando modificaciones para aumentar la información capturada, lo cual se describe a continuación.

2.2.3.1. Metodología Verfploeter

Verfploeter es una metodología de mapeo de áreas de servicio de nubes *anycast*, en término de las redes /24 que se comunican con cada nodo que pertenece a esta. Esto se realiza a través de mediciones activas desarrolladas desde los diferentes nodos que pertenecen a la nube *anycast* hacia una lista de direcciones representantes de cada red, tal como fue descrito en la sección 2.2.2.

Para realizar las mediciones del área de servicio, uno de los nodos pertenecientes a la nube *anycast* realiza el envío de un paquete *ICMP Echo* a cada dirección IP representativa que se encuentre en la *Hitlist* utilizada, donde la dirección IP fuente de estos paquetes corresponden a una que se encuentra en la nube *anycast*.

Dado que la dirección de origen de los paquetes enviados corresponde a la nube *anycast*, las respuestas serán enrutadas al nodo más cercano desde el cliente según la topología actual de la red, generando que los nodos solo reciban respuesta desde los prefijos /24 con los que efectivamente se comunica, tal como se puede apreciar en la Figura 2.1.

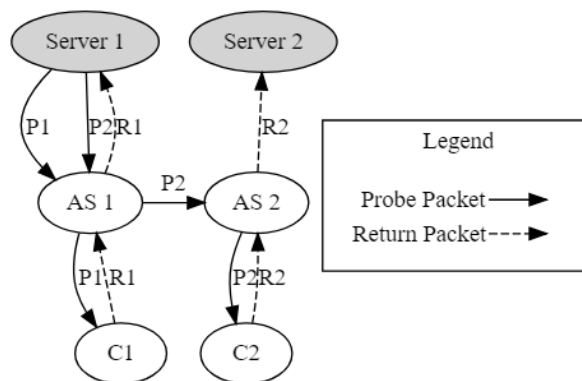


Figura 2.1: Ejemplo de ejecución de medición de área de servicio. El servidor 1 envía paquetes a los clientes 1 y 2, y las respuestas son enrutadas según las tablas de rutas de los AS 1 y 2 a sus respectivos servidores.

Esto permite obtener información sobre las redes y *Sistemas Autónomos* con los cuales cada nodo se comunica de una manera rápida, previniendo el envío de consultas duplicadas al solo utilizar un nodo para realizar el envío de las consultas.

Las desventajas que posee este método, es que no entrega información adicional sobre los clientes consultados, tales como la latencia o las rutas que se están utilizando actualmente por cada nodo para comunicarse con sus clientes, dificultando encontrar las causas de un enrutamiento no deseado, o que puntos se encuentran manejando una alta cantidad de tráfico o latencia.

Por otro lado, al necesitar realizar la ejecución simultánea en todos los nodos para realizar la captura de los resultados, si bien disminuye altamente la cantidad de tráfico generado, genera una dificultad mayor para la ejecución de los experimentos en ambientes más controlados.

Por último, este método no considera falsos positivos que pueden generarse por consultas modificadas por clientes, posibilitando el registro de un área mayor a la real. Esto se mitiga en este trabajo basándose en la realización de mediciones por un periodo de tiempo mucho mayor, esperando que este cliente no se encuentre enviando estos paquetes de manera continua.

2.2.3.2. Modificaciones de Metodología de Captura

Para incrementar la información capturada de los nodos, buscaremos modificar la metodología planteada para buscar las rutas utilizadas, saltos requeridos y la latencia de cada nodo a sus clientes, de manera que el proceso sea fácil de ejecutar por los operadores de estas nubes.

Para lograr obtener las rutas sobre la cual cada nodo analizado trabaja, por cada respuesta a la consulta *ICMP Echo* recibida, es necesario ejecutar un *traceroute* para identificar los routers intermedios desde cada punto de medición.

Realizar *traceroutes* por cada dirección IP de manera tradicional (Incrementando el *TTL* de un paquete *ICMP* desde 1 hasta encontrar el host) aumentaría de una manera muy grande la cantidad de paquetes a enviar y el tiempo de ejecución de este proceso. Si se realizaran estas mediciones de manera directa, realizar la medición de los 3.8M de direcciones planteadas por *Verfploeter*, considerando un promedio de 18 saltos [1] y una velocidad de consulta de 10,000 paquetes por segundo, el tiempo de ejecución aumentaría de 6.3 minutos a 114 minutos, enviado 64.6M de paquetes extra comparados a los 3.8M originales.

El efecto sobre el tiempo de ejecución sería disminuido según la cantidad de nodos desde donde se realizan las mediciones, dado que estos pueden realizarse en paralelo en cada uno de estos. Sin embargo, muchos de los operadores no permiten la ejecución de una medición tan rápida como esta, limitando la velocidad de envío a valores cercanos a los 100 paquetes por segundo. Esto ya aumentaría el tiempo base de medición a 9 horas, y realizar los trazados de ruta aumentarían esto mucho más.

Para disminuir el efecto de ingresar los trazados de ruta, se propuso realizar una predicción de la cantidad de saltos en la que se encuentra cada cliente detectado por un nodo, ejecutando el trazado de ruta desde esta predicción hacia atrás, finalizando en el servidor que ejecuta el trazado de ruta. Para disminuir la cantidad de paquetes enviados, es posible detener este

trazado al detectar un router en el camino que ya se ha visto anteriormente, previniendo la duplicación de las mediciones en rutas ya conocidas. Dado que estas mediciones se focalizan en las redes /24 representativas, es posible además considerar que a los routers que se encuentren en esta misma red como iguales, dado que estos se encuentran en el mismo *Sistema Autónomo* y se espera que sean cercanos, reduciendo aún más la cantidad de paquetes necesarios a enviar. Un ejemplo de este proceso se puede apreciar en la figura 2.2.

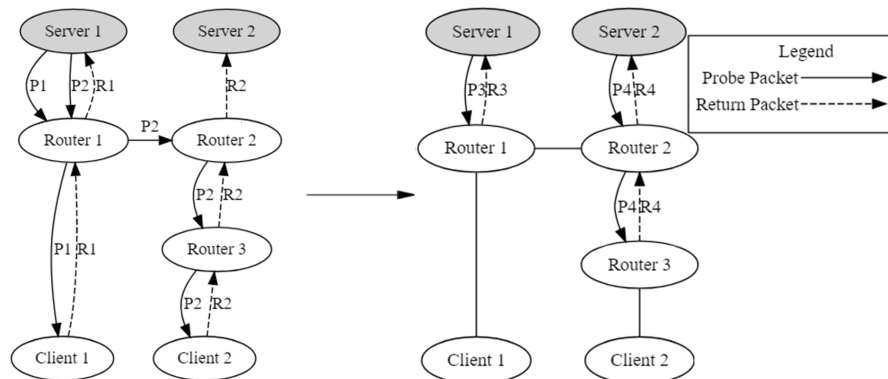


Figura 2.2: Ejemplo del proceso del trazado de rutas. En la primera parte se envía un paquete P1 y P2 a los clientes 1 y 2, respondiendo a su servidor asignado. Luego de esto, cada servidor realiza un trazado de ruta desde sus clientes, donde el Servidor 1 envía un paquete P3 con *TTL* 1 y el Servidor 2 envía P4 con un *TTL* de 2, detectando al Router 1 y 3 respectivamente. Para el caso del servidor 2, si este ya ha visto anteriormente al Router 3 en otros trazados de rutas, se detendrá y no continuará con este, previniendo el envío de paquetes extra en la red.

Esto tiene la desventaja de remover información exacta sobre los caminos recorridos por los paquetes, donde algunos de estos pueden ser balanceados por caminos diferentes según las políticas de enrutamiento de los *Sistemas Autónomos* intermedios. Además, una predicción incorrecta puede hacer invisible algunas redes, o generar una cantidad de consultas mucho mayor si esta fue realizada de manera incorrecta.

La predicción de saltos se realizó observando el *TTL* recibido desde el cliente en la respuesta, tomando como suposición que los clientes generalmente poseen configurado un *TTL* de 64, 128 o 255. Con este *TTL*, es posible ejecutar el traceroute desde el cliente hasta el origen, restando uno de los *TTL* iniciales esperados con el recibido, tal que el valor resultante sea el mínimo mayor a cero. Por ejemplo, si se recibe un paquete con un *TTL* de 116, al restar los tres valores iniciales de 64, 128 y 255, obtenemos los resultados -52, 12 y 139, donde el menor resultado mayor a cero corresponde a 12, distancia que será utilizada para iniciar los trazados de ruta.

Para prevenir falsos positivos que pueden generarse por consultas modificadas por clientes, se generó una llave codificada en los campos *id* y *seq* de un paquete *ICMP Echo*, utilizando la dirección *IP* destino como un parámetro. Esta llave no es posible agregarla al *payload* del paquete *ICMP*, dado que el *payload* no es incluido si la respuesta corresponde a un *ICMP Timeout*, respuesta recibida si se realiza un trazado de ruta.

Estas modificaciones se implementaron en un software llamado *AnyTrace* [15], el cual

realiza la captura de datos según la metodología descrita en esta sección. Este puede ejecutarse de manera independiente en cada servidor *anycast*, o en conjunto a múltiples si es posible.

2.2.4. Verificación Área de Servicio

Para verificar el cálculo de las áreas de servicio calculadas, se utilizó la plataforma de mediciones *RIPE Atlas*, a través de la cual es posible enviar consultas *DNS* desde diferentes puntos de medición distribuidos a lo largo de internet.

Para identificar el servidor asignado a cada uno de estos puntos de medición, se procedió a realizar el envío de consultas *DNS* clase *CHAOS* [14] (consulta *DNS* clase número 3), consultando por un tipo de recurso *TXT* (*Recurso de Texto*, por sus siglas en inglés, *Text Record*) con el valor *version.bind*. La respuesta a esta consulta varía según el software utilizado para resolver las consultas *DNS* y la configuración empleada por los operadores de estos.

Para la nube *a.nic.cl* de *NIC Chile*, la respuesta recibida al realizar el envío de una consulta *DNS CHAOS*, contiene el nombre del nodo que procesó la respuesta, lo que nos permite comparar el nodo obtenido de esta consulta con el calculado por la metodología propuesta en este capítulo.

Esto nos permite configurar la plataforma de medición, de manera que los puntos de medición envíen consultas *DNS CHAOS* a la nube *a.nic.cl*, y así obtener el nodo verdaderamente asignado, permitiendo comparar este con el calculado por la metodología propuesta en este capítulo.

Para verificar la predicción de la distancia en términos de saltos hasta los clientes, se comparó el número de redes detectadas antes y después de aplicar la predicción, donde los números deben ser prácticamente iguales si esta predicción se realizó de manera correcta.

2.3. Resultados

En esta sección, presentaremos los resultados obtenidos al aplicar la metodología expuesta sobre la nube *anycast a.nic.cl* de *NIC Chile*.

2.3.1. Direcciones IP a medir

La captura de direcciones IP representativas produjo un resultado de *3,482,667* direcciones IP únicas, cada una representando un prefijo de red /24.

Estas direcciones representan a *51,897* *Sistemas Autónomos* que realizar anuncios de distintos prefijos de red en internet, representando a más de un 75% del total de *Sistemas Autónomos* registrados en los diferentes registros.

2.3.2. Captura de Datos

La captura de datos se realizó para dos configuraciones, *unicast* y *anycast*, permitiendo comparar la visión de la red desde un cliente normal de internet y la visión desde una nube *anycast*.

Para cada una de estas configuraciones, realizaremos la medición de las siguientes métricas de los resultados obtenidos.

- Destinos en la *Hitlist*: Número de direcciones IP originalmente en la *Hitlist* que respondieron a la consulta *ICMP Echo*.
- Respuestas totales: Número de respuestas totales capturadas que se recibieron, utilizadas para obtener el área de servicio y las rutas hacia los clientes.
- Respuestas traceroute esperadas: Número de respuestas esperadas si se ejecutaran traceroutes de manera tradicional. Esto es calculado como la suma de las distancias en términos de los saltos a cada una de las direcciones IP que fueron medidas satisfactoriamente desde la *Hitlist*.
- Direcciones IP únicas: Número de direcciones IP únicas que se encuentran en la ruta o destino de las mediciones.
- Redes /24 capturadas: Número de prefijos de red /24 capturados.
- Destinos sin predecir distancia correcta: Número de redes a las que la predicción del destino no fue realizada correctamente, o el paquete de consulta fue perdido en el camino.
- Velocidad de medición: Número de paquetes por segundo utilizado para realizar las mediciones.
- Tiempo de medición: Tiempo utilizado para realizar las mediciones desde el nodo indicado.

2.3.2.1. Medición Unicast

La captura de datos *unicast* nos permite observar el estado de la red, y los resultados esperados en caso de realizar una agregación de todos los nodos pertenecientes a una nube *anycast*, además de permitirnos comparar la diferencia de poseer un solo servidor *unicast* a múltiples nodos que utilizan la metodología *anycast*.

Los resultados obtenidos al aplicar esta metodología en un ambiente *unicast* son posibles de observar en la tabla 2.1.

En esta tabla, logramos observar que la gran mayoría de las direcciones IP que se encuentran en la *Hitlist* respondieron satisfactoriamente (95 % del total medido).

Por otro lado, logramos observar que la cantidad de respuestas obtenidas es mucho menor a la esperada si utilizáramos la metodología tradicional para realizar los trazados de ruta, llegando a disminuir un 93 % la cantidad de consultas necesarias, esto a costa de un 15 % de destinos donde la predicción de distancia fue incorrecta.

	Unicast
Destinos en la <i>Hitlist</i>	3,319,488
Respuestas totales	17,403,116
Respuestas traceroute esperadas	239,089,264
Direcciones IP únicas	3,581,753
Redes /24 capturadas	3,318,145
Destinos sin predecir distancia correcta	512,925
Velocidad de medición [pps]	20,000
Tiempo de medición [Minutos]	18

Tabla 2.1: Estadísticas generales de los datos capturados desde una medición *unicast*.

2.3.2.2. Medición Nube Anycast NIC Chile

La captura de datos en la nube *anycast* se realizó desde los nodos en Arica, Merced, Tucapel y Sao Paulo pertenecientes a la nube *a.nic.cl* de NIC Chile, utilizando los nodos que sirven las consultas *DNS* en estos lugares. En cada uno de estos nodos, se realizó la ejecución del software de captura desarrollado, obteniendo los resultados presentados en la tabla 2.2.

	<i>arica</i>	<i>merced</i>	<i>tucapel</i>	<i>saopaulo</i>
Destinos en la <i>Hitlist</i>	8,937	52,045	3,935	815,106
Respuestas totales	61,333	238,294	9,865	2,621,368
Respuestas traceroute esperadas	176,584	623,273	27,660	24,216,172
Direcciones IP únicas	10,217	59,113	4,254	648,902
Redes /24 capturadas	9,551	53,910	4,058	523,685
Destinos sin predecir distancia correcta	0	398	47	320,006
Velocidad de medición [pps]	100	100	100	100
Tiempo de medición [Minutos]	206	275	186	594

Tabla 2.2: Estadísticas generales de los datos capturados desde los servidores *arica*, *merced*, *tucapel* y *saopaulo*, utilizando la dirección *anycast a.nic.cl*.

En esta tabla, logramos observar que el nodo *saopaulo* cubre la mayor cantidad de prefijos comparado con los otros nodos. Esto lo podemos explicar por su lejanía a los otros nodos analizados, los cuales se encuentran concentrados en Chile, disminuyendo su posibilidad de capturar una gran cantidad de clientes.

Por otro lado, logramos observar que para los nodos con un área de servicio pequeña, la predicción de distancia fue muy buena, e incluso los errores encontrados podrían atribuirse a pérdida de paquetes producida por la red. Sin embargo, podemos ver que para el caso de *saopaulo* los errores de predicción son altos, lo que dificulta la observación de las rutas hacia los clientes.

Podemos observar que la cantidad de respuestas recibidas corresponde aproximadamente al 30 % de las esperadas si se realizaran los trazados de ruta de manera directa, indicando un gran ahorro en el tiempo de medición y tráfico generado.

Sobre los datos capturados, se extrajo la cantidad de prefijos /24 que se ven repetidos

en diferentes mediciones, donde se encontró que *3522* prefijos se repiten en al menos dos nodos, y *2456* prefijos se repiten en al menos tres nodos. Estas redes pueden comunicarse con diferentes nodos a la vez, lo cual nos dice que estas redes se encuentran distribuidas. Algunas de estas redes son nubes *anycast* altamente utilizadas, tal como la red de *CloudFlare* (1.1.1.0/24) y *l.root-servers.net* (199.7.83.0/24).

2.3.3. Verificación Área de Servicio

Utilizando la plataforma de mediciones *RIPE Atlas*, se realizaron mediciones utilizando *107* puntos de medición ubicados principalmente en Chile y Brasil, intentando que estos se encuentren cercanos a los diferentes nodos para que las respuestas sean lo más distribuidas posibles.

Las respuestas de las consultas *DNS CHAOS* se compararon con el área de servicio que fue capturada desde los nodos *anycast*, observando si la dirección IP del punto de medición se encuentra en el nodo encontrado por la consulta *DNS*. La comparación entre el resultado esperado y encontrado se puede observar en la tabla 2.3.

Esperado\Encontrado	<i>arica</i>	<i>merced</i>	<i>tucapel</i>	<i>saopaulo</i>
<i>arica</i>	11	0	0	0
<i>merced</i>	0	9	0	0
<i>tucapel</i>	1	0	1	0
<i>saopaulo</i>	0	0	0	87

Tabla 2.3: Resultados de verificación desde RIPE NCC, comparando el nodo resultante al enviar una consulta *DNS CHAOS* y el área de servicio calculada con la metodología expuesta.

En esta tabla, logramos observar que casi la totalidad de las consultas fueron localizadas de manera correcta, siendo solo uno de los nodos localizados incorrectamente. Este error se atribuye al desfase de tiempo entre la medición del área de servicio y la verificación, en la que un cambio en la topología de la red produce un cambio en la asignación de los nodos *anycast*.

Capítulo 3

Análisis Geográfico

3.1. Introducción

Muchas veces los servidores pertenecientes a una nube *anycast* son distribuidos de manera geográfica, esperando que estos capturen el tráfico más cercano a estos según la distancia entre el nodo y sus clientes.

Para verificar los despliegues actuales, los operadores de estas nubes generalmente realizan el análisis de logs sobre los clientes, geolocalizando las direcciones IP de los clientes para averiguar la fuente de tráfico, así ajustando los anuncios *BGP* hasta alcanzar los resultados esperados.

El segundo método utilizado por los operadores corresponde a emplear nubes de medición externas, las cuales entregan la geolocalización del nodo desde donde se realiza la medición, e información sobre la nube objetiva tal como la latencia, ruta, o información específica en caso de los servidores *DNS* como el nombre del nodo en caso de que este sea público. La desventaja de estos servicios es que la distribución de estas nubes se encuentra muy limitada en comparación al número total de redes que existe actualmente, además de que la localización de los nodos pertenecientes a la nube de medición se encuentra muy sesgada.

En este capítulo, se describirá una metodología que nos permitirá obtener una visión completa sobre la distribución geográfica del área de servicio de una nube *anycast*, observando desde qué lugares cada nodo se encuentra recibiendo tráfico, en términos de los prefijos de red y *Sistemas Autónomos* involucrados, comparándolos con el caso ideal de distribución. Esto será aplicado a la nube *anycast a.nic.cl* de *NIC Chile*, desde la cuál realizamos la captura de información sobre su área de servicio en el capítulo 2.

3.2. Metodología

En este capítulo, buscamos obtener la siguiente información sobre la nube analizada:

- Número de prefijos IP /24 por país con el que cada nodo se comunica.
- Número de *Sistemas Autónomos* por país con el que cada nodo se comunica.
- Comparación de la localización de los *Sistemas Autónomos* asignados a cada nodo con la localización de todos los nodos pertenecientes a la nube *anycast*.

Para cada uno de estos conjuntos de información, realizaremos la comparación de la información en bruto y una ponderación por la cantidad de tráfico recibido por estos nodos desde los clientes, permitiéndonos observar las diferencias generadas al darle una mayor importancia a los clientes que actualmente utilizan la nube *anycast*.

Con fin de realizar la medición del área de servicio geográfico de los nodos pertenecientes a la nube *anycast*, es necesario conocer el área de servicio en términos topológicos de los nodos analizados, para lo cual utilizaremos el método descrito en el *capítulo 2*, el cual nos entregará las diferentes redes que pertenecen a cada nodo analizado.

Cada red se mapeó a su *Sistema Autónomo* correspondiente utilizando información sobre las rutas *BGP* capturada por *RIPE NCC* (información *MRT*) [22], sobre la cual podemos mapear el *Número del Sistema Autónomo* objetivo de la ruta al prefijo anunciado. Es importante notar que no todos los routers anuncian su dirección IP, aun cuando estos se encuentren en la red analizada, por lo que fue necesario complementar esta información con la base de datos *RADB*, la cual posee información de registro de *APNIC*, *ARIN*, *RIPE*, entre otras.

Para realizar la geolocalización de las redes obtenidas, se utilizó la base de datos *GeoLite2* de *MaxMind* [18] a nivel país, la cual nos permite identificar la localización de los prefijos de red a través de la dirección IP representativa capturada. Esta base de datos nos permite tener una precisión de un 77.5% [11] a nivel país, generalmente afectado por la presencia de esta red en internet.

Esto se repitió para los *Sistemas Autónomos*, mapeando las direcciones IP representativas y su geolocalización a su *Sistema Autónomo* correspondiente, solo considerando una vez cada país por *Sistemas Autónomos*.

Para realizar la comparación de las distancias geográficas a los diferentes nodos y su asignación real, se utilizó primeramente la base de datos *GeoLite2* de *MaxMind* a nivel ciudad, lo cual, nos permite obtener una localización en términos de la *Latitud* y *Longitud* aproximada de cada una de las direcciones IP representativas capturadas. Esta base de datos nos entrega la precisión en kilómetros de cada medición, por lo que, para asegurar una correcta medición, solo se utilizaron las medidas que poseen una precisión de al menos 500 kilómetros.

La localización de cada IP representativa se comparó con la localización aproximada de todos los nodos pertenecientes a la nube *anycast*, determinando cuál es el nodo más cercano en comparación al realmente asignado.

Por último, los pesos utilizados para realizar la ponderación de los resultados, se obtuvieron desde capturas en vivo generadas en los nodos analizados, desde las cuales se extrajo las consultas por segundo generadas por cada prefijo /24.

3.3. Resultados

Los siguientes resultados fueron obtenidos utilizando la metodología descrita anteriormente, aplicada a la nube *a.nic.cl* de *NIC Chile*. Esto se aplicó a los cuatro nodos correspondientes a Arica, Merced, Tucapel y Sao Paulo analizado en el capítulo 2.

3.3.1. Área de Servicio por País

Los gráficos presentados a continuación, nos muestran el área de servicio por país de los nodos *arica*, *merced*, *saopaulo* y *tucapel*.

3.3.1.1. arica

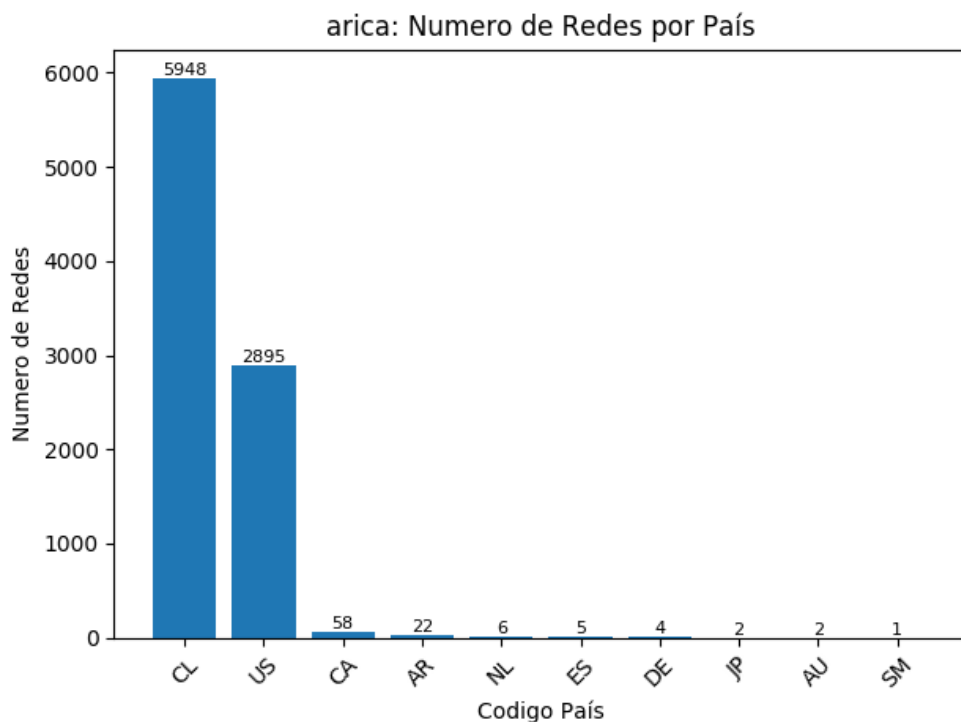


Figura 3.1: Gráfico que muestra el número de redes (prefijos /24) por cada país que se encuentra en el área de servicio del nodo *arica* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.1.2. merced

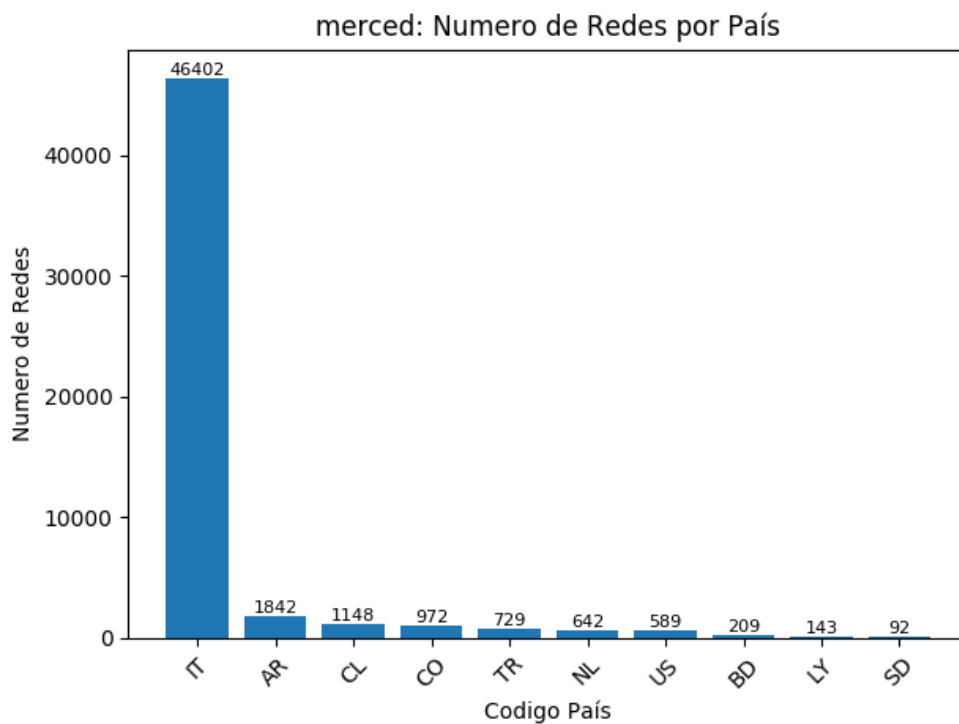


Figura 3.2: Gráfico que muestra el número de redes (prefijos /24) por cada país que se encuentra en el área de servicio del nodo *merced* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.1.3. saopaulo

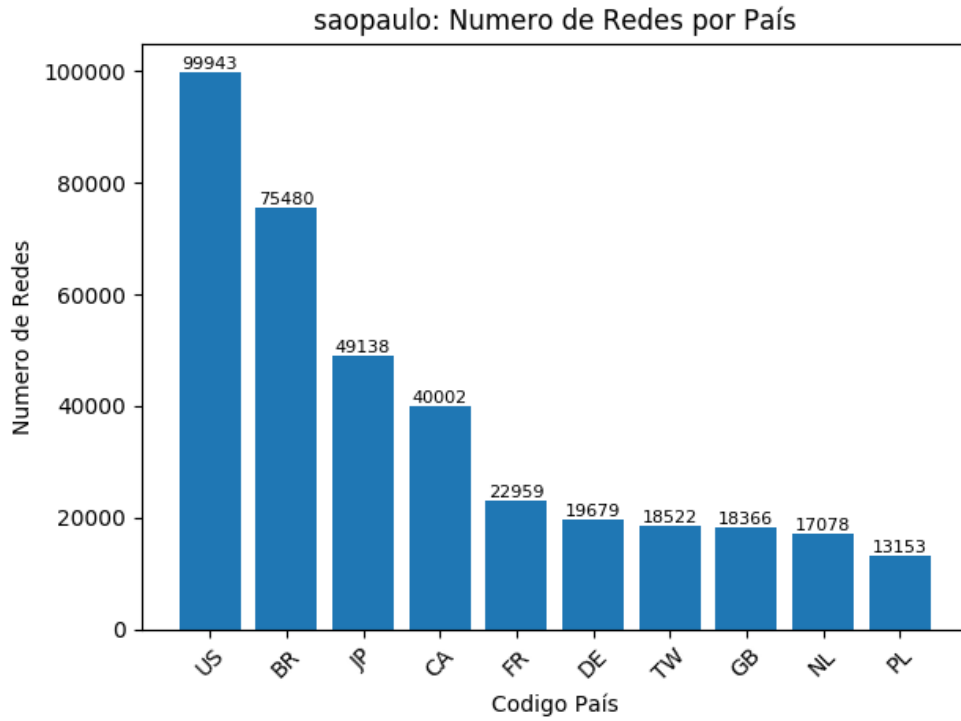


Figura 3.3: Gráfico que muestra el número de redes (prefijos /24) por cada país que se encuentra en el área de servicio del nodo *saopaulo* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.1.4. tucape

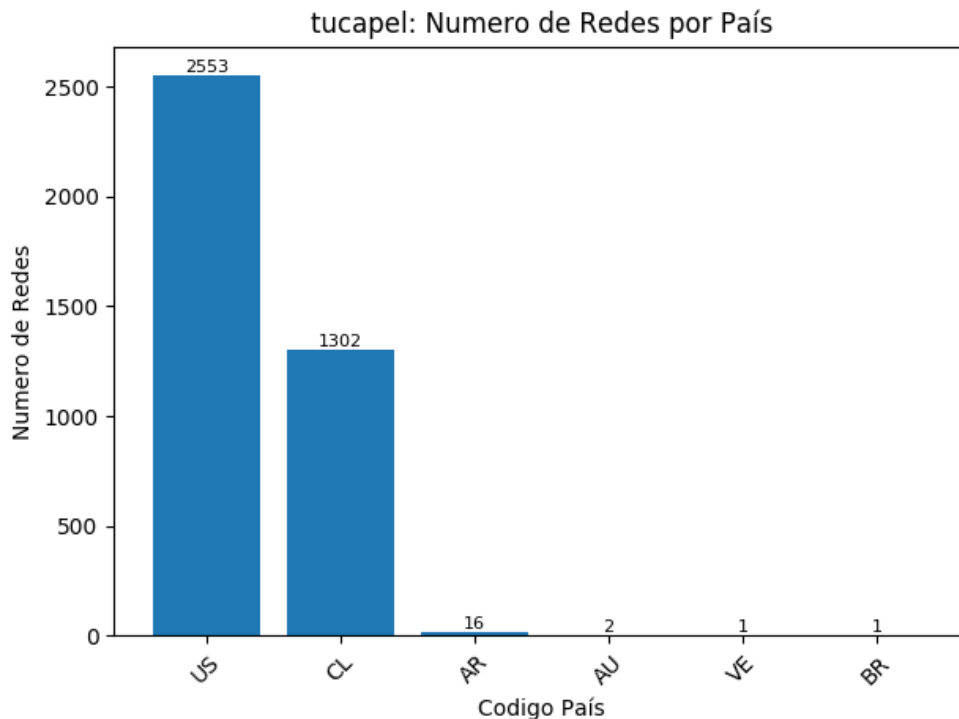


Figura 3.4: Gráfico que muestra el número de redes (prefijos /24) por cada país que se encuentra en el área de servicio del nodo *tucape* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.1.5. Análisis de resultados

Podemos observar que una gran parte de los prefijos de *arica* se encuentran en Chile, seguido por Estados Unidos con aproximadamente la mitad de las redes, y una presencia menor en otros países.

Para el caso del nodo *merced*, podemos ver que existe una gran cantidad de redes que pertenecen a Italia, y no fueron asignados a un nodo diferente tal como el que se encuentra en Ámsterdam o Praga. Para comprobar esto, se verificaron algunas de las rutas utilizadas, donde se encontró que una de estas utiliza la red de *Entel* y la red marítima *Telecom Italia Sparkle* (*Sistemas Autónomos 6471 y 6762 respectivamente*).

El nodo *saopaulo* se encuentra mucho más distribuido, abarcando una cantidad de países mucho mayor. Esto se explica por su lejanía mucho mayor a otros nodos, en comparación a los otros nodos analizados que se encuentran concentrados en Chile. Es importante destacar que existe una gran cantidad de redes que se comunican con Estados Unidos, aun cuando existe un nodo *anycast* en ese país. También podemos destacar la conexión de este nodo con países europeos tal como Francia y Alemania, incluso con la existencia de un nodo en el continente europeo.

Por último, el nodo *tucapel* es uno de los que se encuentran más restringidos, solo hablando con Chile y Estados Unidos, además de una cantidad mínima en otros lugares.

3.3.2. Sistemas Autónomos por país

Los gráficos presentados a continuación, nos muestran los *Sistemas Autónomos* por país que se encuentran en el área de servicio de los nodos *arica*, *merced*, *saopaulo* y *tucapel*.

3.3.2.1. arica

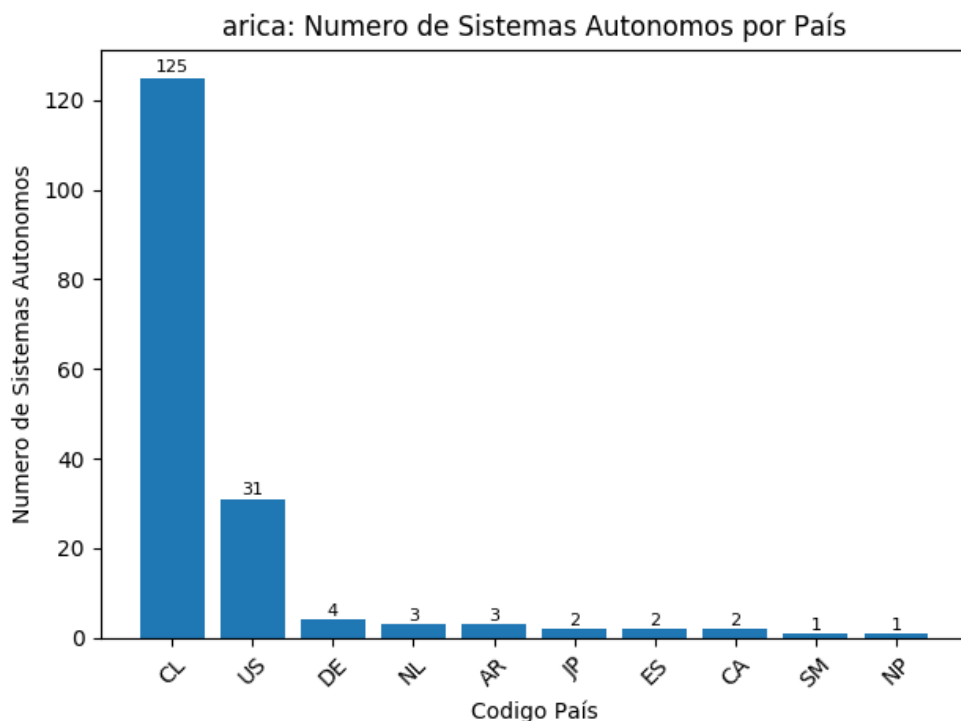


Figura 3.5: Gráfico que muestra el número de *Sistemas Autónomos* por cada país que se encuentra en el área de servicio del nodo *arica* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.2.2. merced

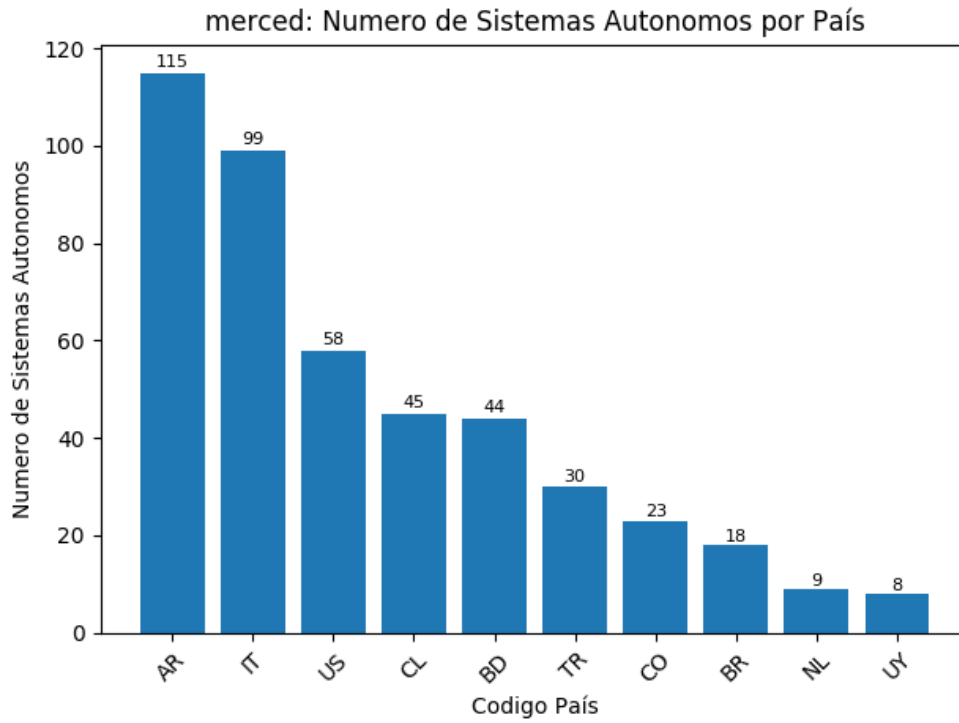


Figura 3.6: Gráfico que muestra el número de *Sistemas Autónomos* por cada país que se encuentra en el área de servicio del nodo *merced* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.2.3. saopaulo

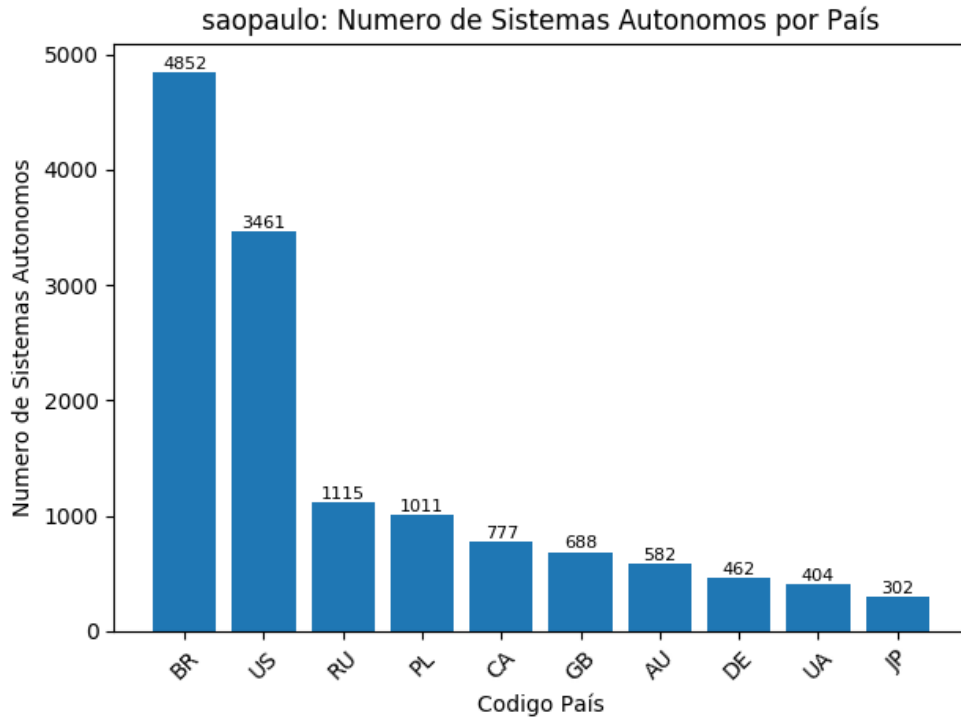


Figura 3.7: Gráfico que muestra el número de *Sistemas Autónomos* por cada país que se encuentra en el área de servicio del nodo *saopaulo* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.2.4. tuapel

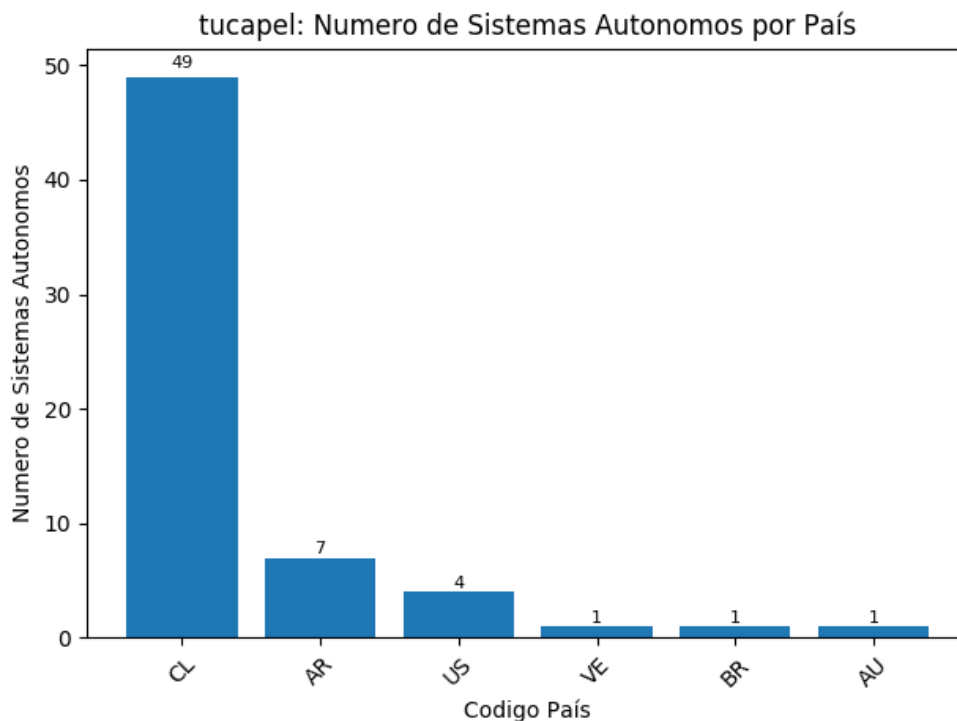


Figura 3.8: Gráfico que muestra el número de *Sistemas Autónomos* por cada país que se encuentra en el área de servicio del nodo *tuapel* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.2.5. Análisis de resultados

Podemos apreciar que en el nodo *arica*, la proporción por país de *Sistemas Autónomos* (figura 3.5) y prefijos de red /24 (figura 3.1) se conserva.

Para el caso del nodo *merced*, podemos ver que la diferencia entre la conexión de Italia y los otros *Sistemas Autónomos* es mucho menor, lo que nos dice que unos pocos *Sistemas Autónomos* abarcan una gran cantidad de redes con la cual este se comunica. Además podemos apreciar como *merced* se comunica con lugares más lejanos, tal como Bangladesh (Asia) y Turquía (a través de *Entel*, *Telecom Italia Sparkle* y *Borusan Telekom*, por los *Sistemas Autónomos* 6471, 6762 y 15924).

El nodo *saopaulo* muestra un cambio en comparación a la figura 3.3, donde ahora existe un mayor número de *Sistemas Autónomos* en el país local Brasil, seguido de Estados Unidos. Además, en esta medición resaltan los países que se encuentran en otros continentes tal como Rusia y Polonia, de los cuales el nodo en Ámsterdam no captura.

Por ejemplo, uno de los caminos hacia Polonia recorre los *Sistemas Autónomos* *GlobeNet Cabos Submarinos Colombia* (Colombia, *Sistema Autónomo* 52320), *AMS-IX* (Ámsterdam,

Sistema Autónomo 1200) y Vectra (Polonia, *Sistema Autónomo 29314*), lo cual resalta en gran manera dado que existe un nodo en la nube *a.nic.cl* desplegado en Ámsterdam.

Por último, el nodo *tucapel* presenta un salto en la cantidad de *Sistemas Autónomos* que se encuentran en Chile, en comparación a la cantidad de redes que se observan en la figura 3.4. Esto nos muestra que existen *Sistemas Autónomos* estadounidenses que poseen una gran cantidad de redes asignadas en comparación a los *Sistemas Autónomos* chilenos, donde se encuentran más distribuidos.

3.3.3. Área de Servicio por País Ponderada

Los gráficos presentados a continuación, nos muestran el tráfico ponderado por país que se encuentran en el área de servicio de los nodos *arica*, *merced*, *saopaulo* y *tucapel*.

3.3.3.1. arica

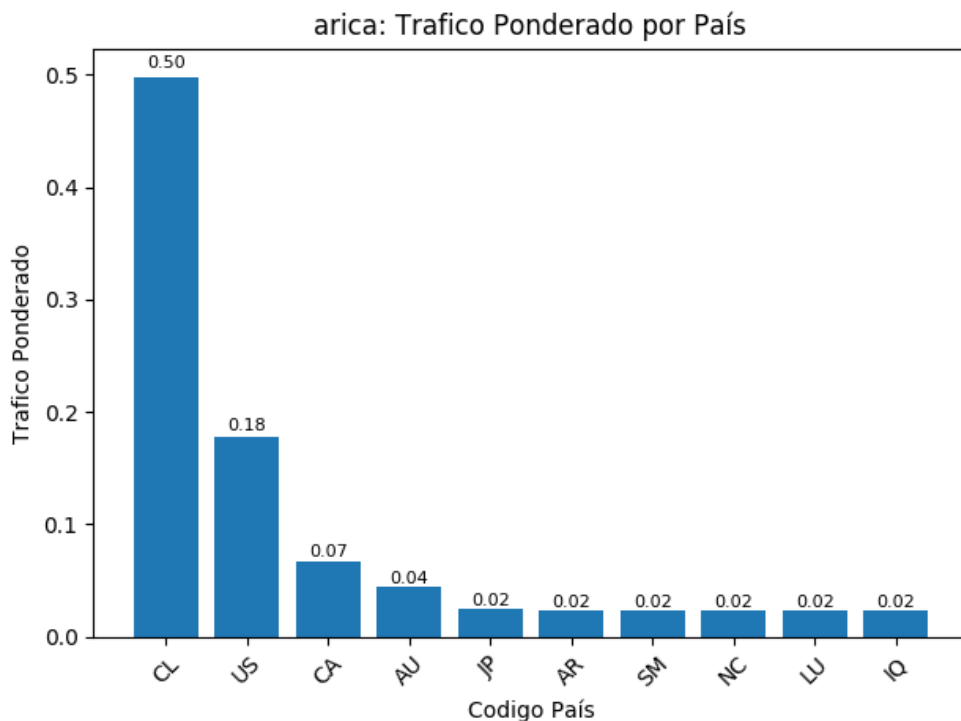


Figura 3.9: Gráfico que muestra el tráfico ponderado por cada país que se encuentra en el área de servicio del nodo *arica* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.3.2. merced

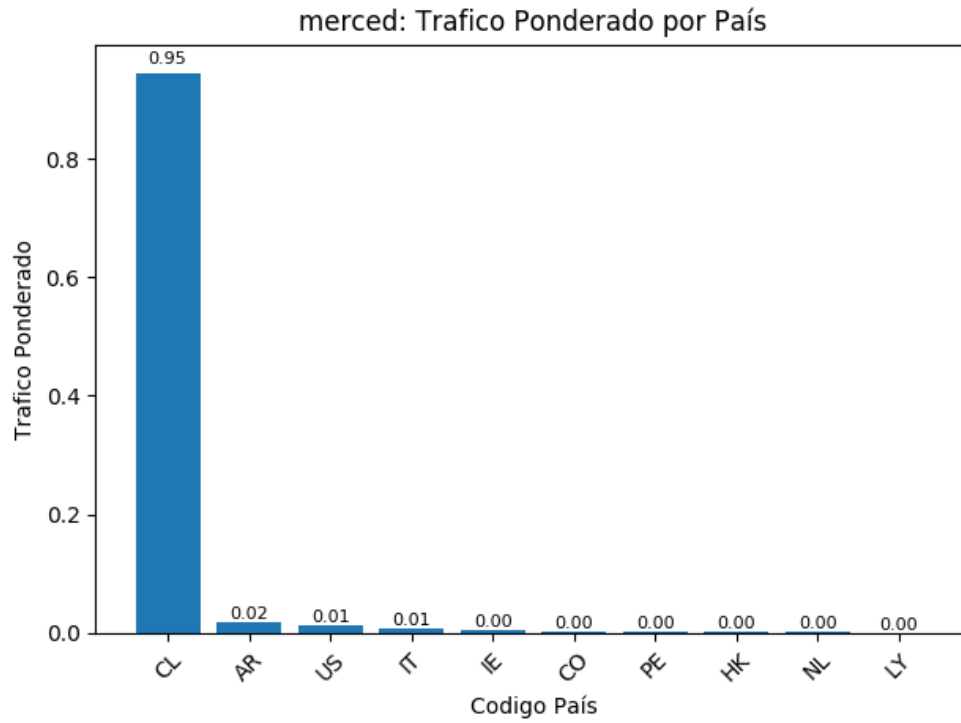


Figura 3.10: Gráfico que muestra el tráfico ponderado por cada país que se encuentra en el área de servicio del nodo *merced* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.3.3. saopaulo

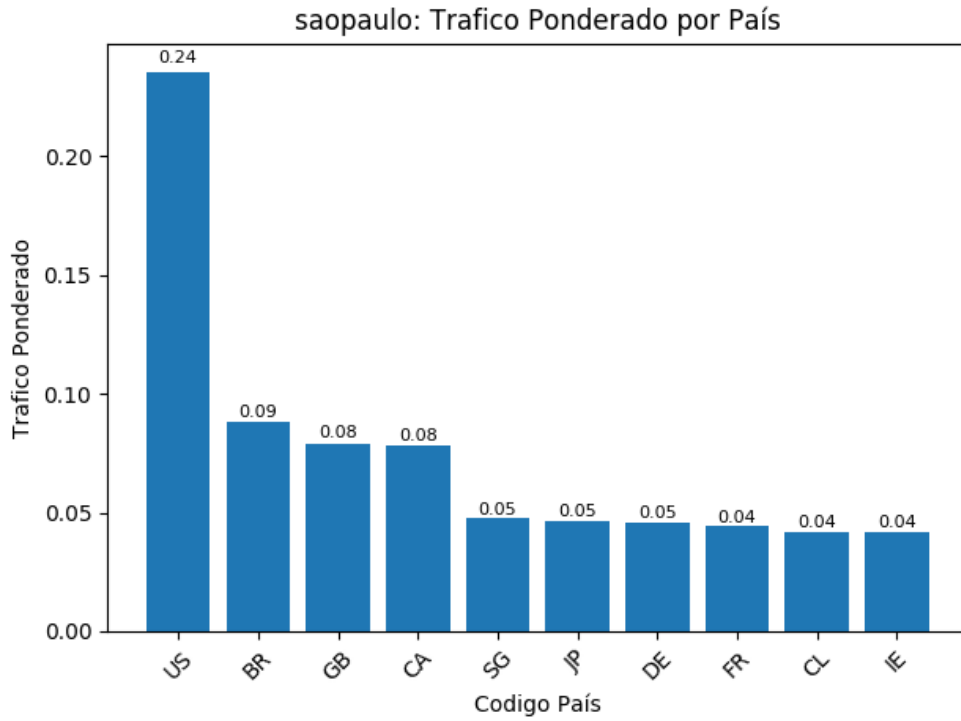


Figura 3.11: Gráfico que muestra el tráfico ponderado por cada país que se encuentra en el área de servicio del nodo *saopaulo* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.3.4. tucafel

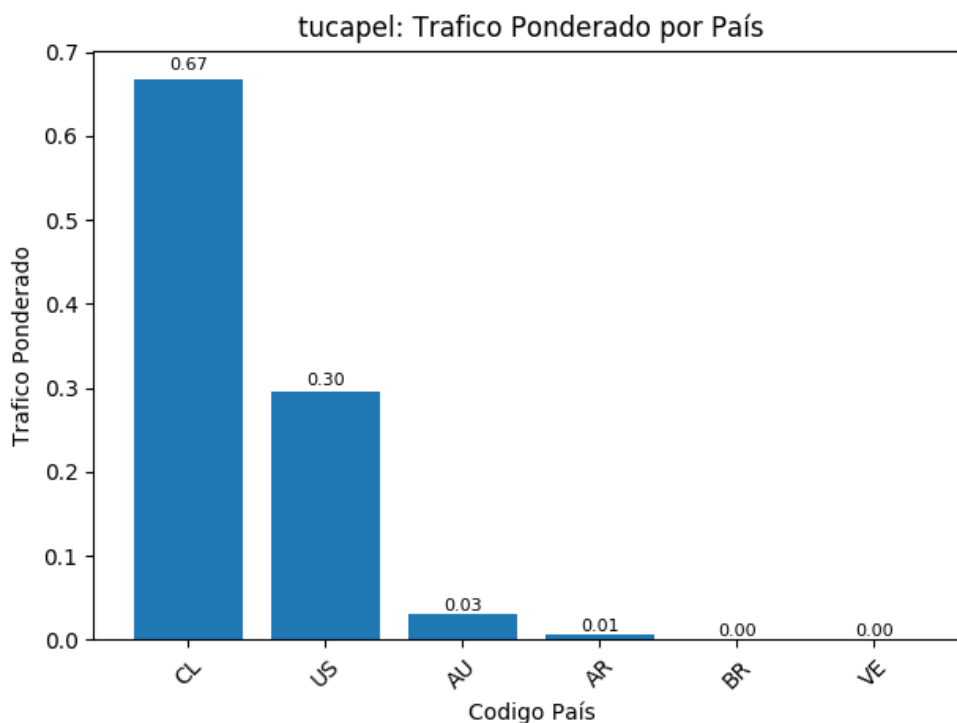


Figura 3.12: Gráfico que muestra el tráfico ponderado por cada país que se encuentra en el área de servicio del nodo *tucafel* perteneciente a la nube *a.nic.cl* de *NIC Chile*.

3.3.3.5. Análisis de resultados

Podemos apreciar que los nodos *arica* y *tucafel* se comportan según lo esperado, cubriendo en mayor manera el tráfico proveniente de Chile, y en menor manera de Estados Unidos, siguiendo las áreas que hemos visto en las secciones anteriores.

Para el nodo *merced*, podemos apreciar que el 95 % del tráfico recibido corresponde a Chile, siendo un porcentaje mucho menor otros países. Esto corresponde a lo esperado, dado que estos servidores administran los dominios chilenos, por lo que este país es el que más genera consultas. Por otro lado, podemos apreciar que existen consultas generadas desde Italia y otros países de Europa, lo que nos dice que el área de servicio calculada es correcta.

Por último, podemos apreciar que el nodo *saopaulo* se comunica mayoritariamente con Estados Unidos, seguido por Brasil, países europeos, e incluso una parte de Chile, lo cual es destacable dado que existen tres nodos en este país que deberían cubrir todo su territorio.

3.3.4. Asignación de Nodo Según Localización

Los gráficos presentados a continuación, nos muestra una comparación entre la asignación geográfica óptima de las redes con su nodo *anycast* más cercano, separando las redes según las áreas de servicio de cada nodo.

3.3.4.1. arica

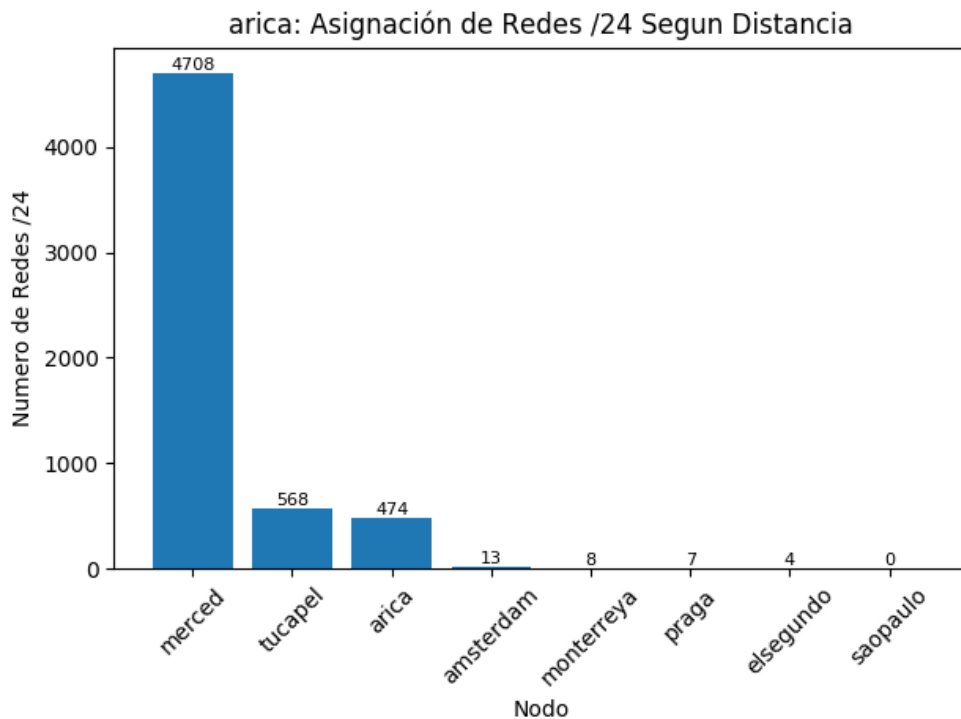


Figura 3.13: Gráfico que muestra la asignación esperada de las redes pertenecientes al área de servicio del nodo *arica*, realizando una comparación de las distancias geográficas a todos los nodos.

3.3.4.2. merced

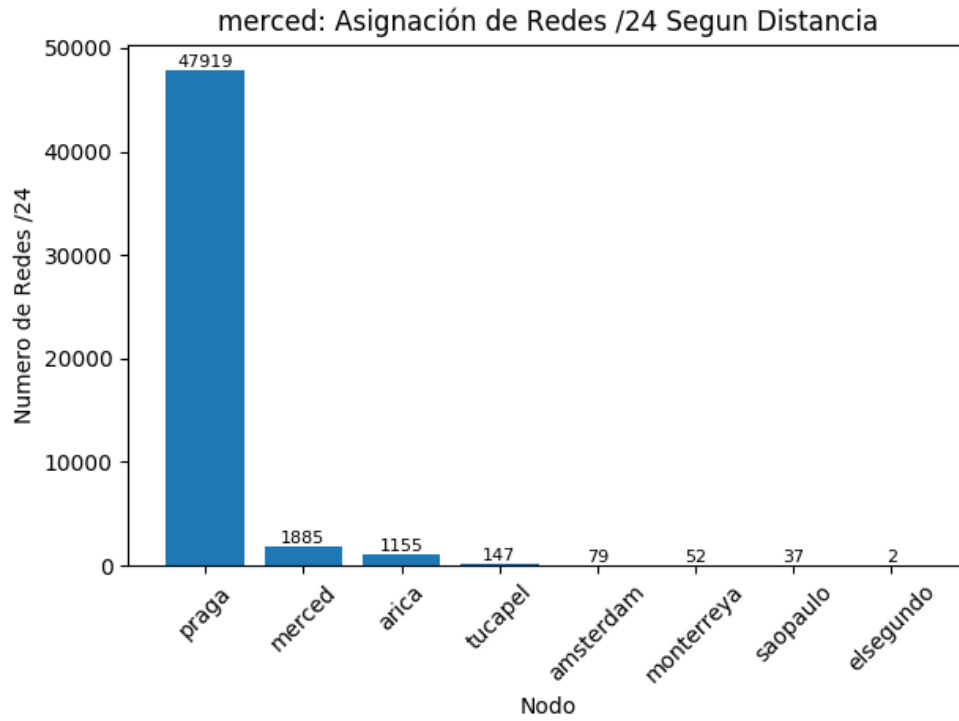


Figura 3.14: Gráfico que muestra la asignación esperada de las redes pertenecientes al área de servicio del nodo *merced*, realizando una comparación de las distancias geográficas a todos los nodos.

3.3.4.3. saopaulo

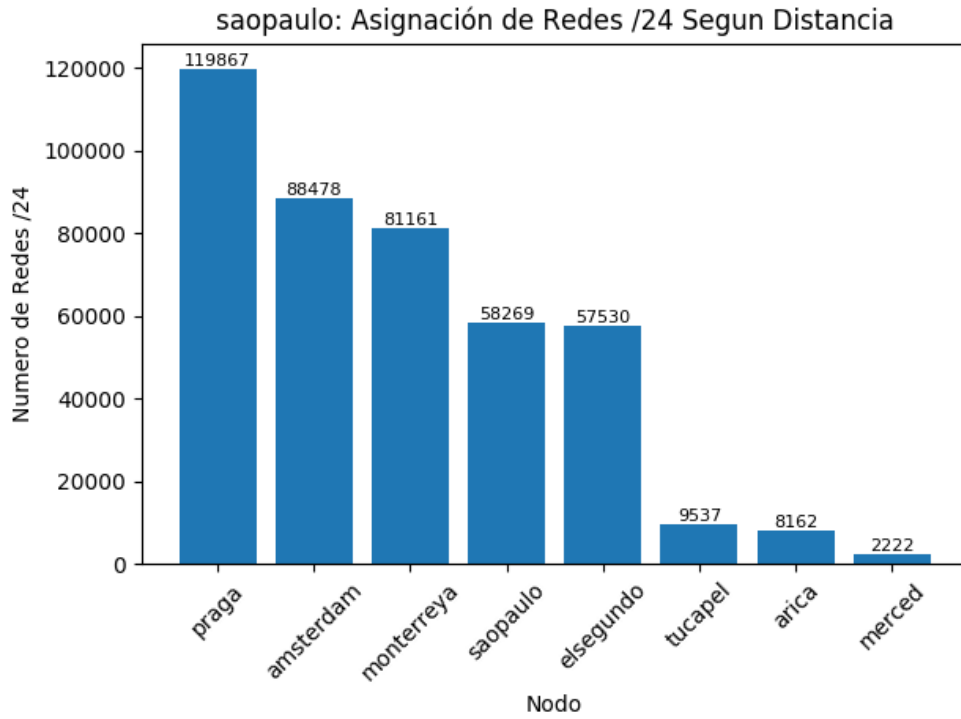


Figura 3.15: Gráfico que muestra la asignación esperada de las redes pertenecientes al área de servicio del nodo *saopaulo*, realizando una comparación de las distancias geográficas a todos los nodos.

3.3.4.4. tucafel

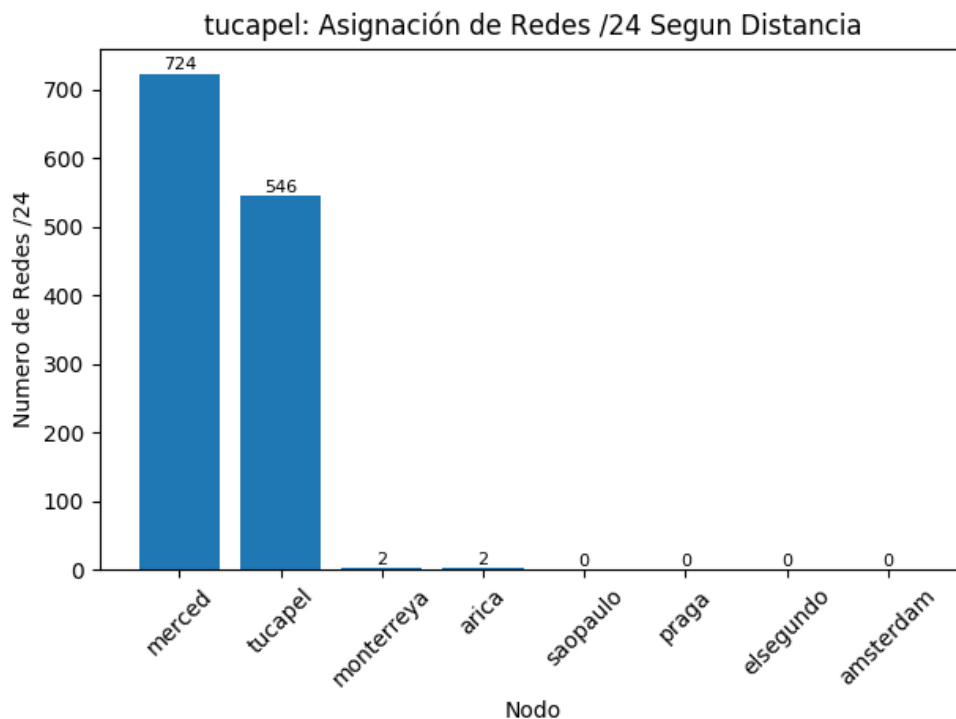


Figura 3.16: Gráfico que muestra la asignación esperada de las redes pertenecientes al área de servicio del nodo *tucafel*, realizando una comparación de las distancias geográficas a todos los nodos.

3.3.4.5. Análisis de resultados

Podemos observar que en el nodo *arica*, muchas de las redes se deberían encontrar asignadas a *merced*, seguido en menor manera por la asignación en *tucafel*. La cantidad correctamente asignada a *arica* corresponde a aproximadamente el 8% del total de redes asignadas a este nodo.

En el nodo en *merced*, podemos apreciar que una gran cantidad de redes debería encontrarse asignadas al nodo *praga*, lo cual podemos explicar por el alto contenido de redes pertenecientes a Europa que encontramos anteriormente. Además de esto, podemos ver que existe una parte importante de las redes asignadas que deberían encontrarse en *arica*, las cuales pueden encontrarse geográficamente en el medio de estos nodos.

Para el caso del nodo *saopaulo*, podemos ver de manera mucho más clara la gran distribución que posee, y como la asignación de redes no privilegia de forma geográfica a los nodos si estos se encuentran muy lejanos unos de otros.

Finalmente, podemos apreciar que las asignaciones del nodo *tucafel* se encuentra dividido entre este y *merced*, lo cual podemos entender por la cercanía de estos nodos, y que su

división territorial según la topología de la red es mucho más difícil, además de afectar en menor manera a los clientes, ingresando una latencia mínima.

3.3.5. Asignación de Nodo según Localización Ponderada por Tráfico

Los gráficos presentados a continuación, nos muestran la asignación esperada del tráfico ponderado de las redes pertenecientes al área de servicio capturado desde los nodos *arica*, *merced*, *saopaulo* y *tucapel*.

3.3.5.1. arica

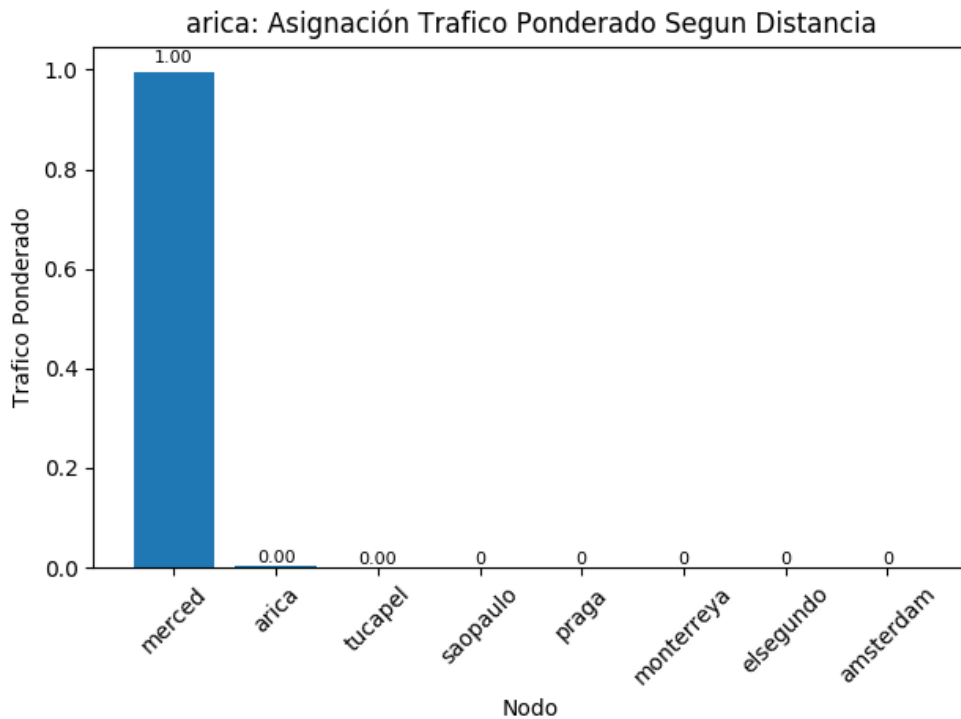


Figura 3.17: Gráfico que muestra la asignación esperada de las redes pertenecientes al área de servicio del nodo *arica*, realizando una ponderación por el tráfico recibido desde las redes analizadas y comparando las distancias geográficas a todos los nodos.

3.3.5.2. merced

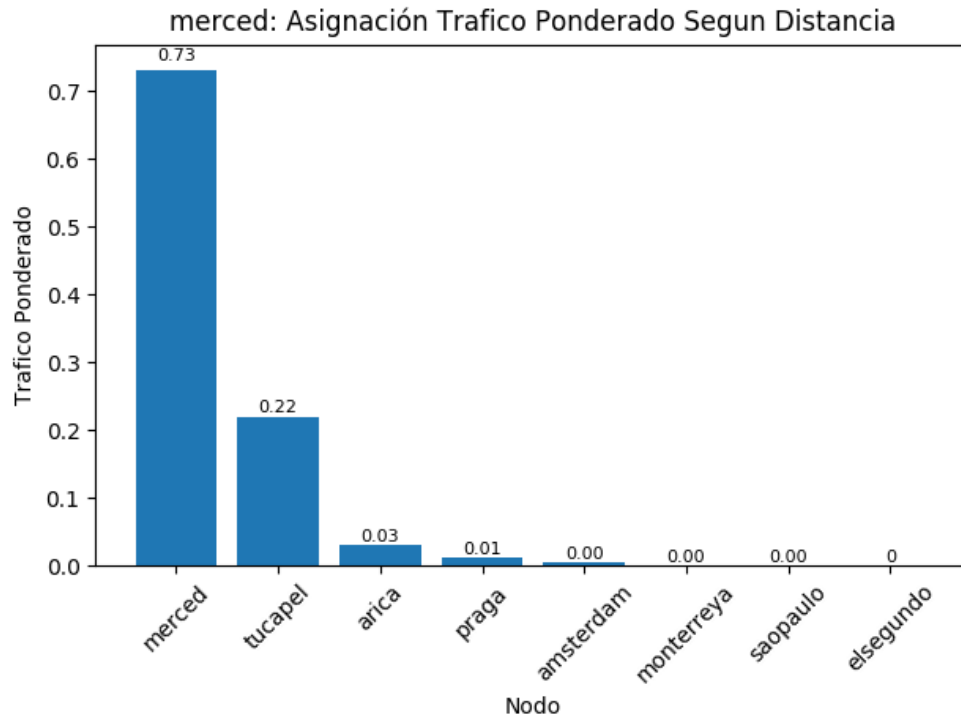


Figura 3.18: Gráfico que muestra la asignación esperada de las redes pertenecientes al área de servicio del nodo *merced*, realizando una ponderación por el tráfico recibido desde las redes analizadas y comparando las distancias geográficas a todos los nodos.

3.3.5.3. saopaulo

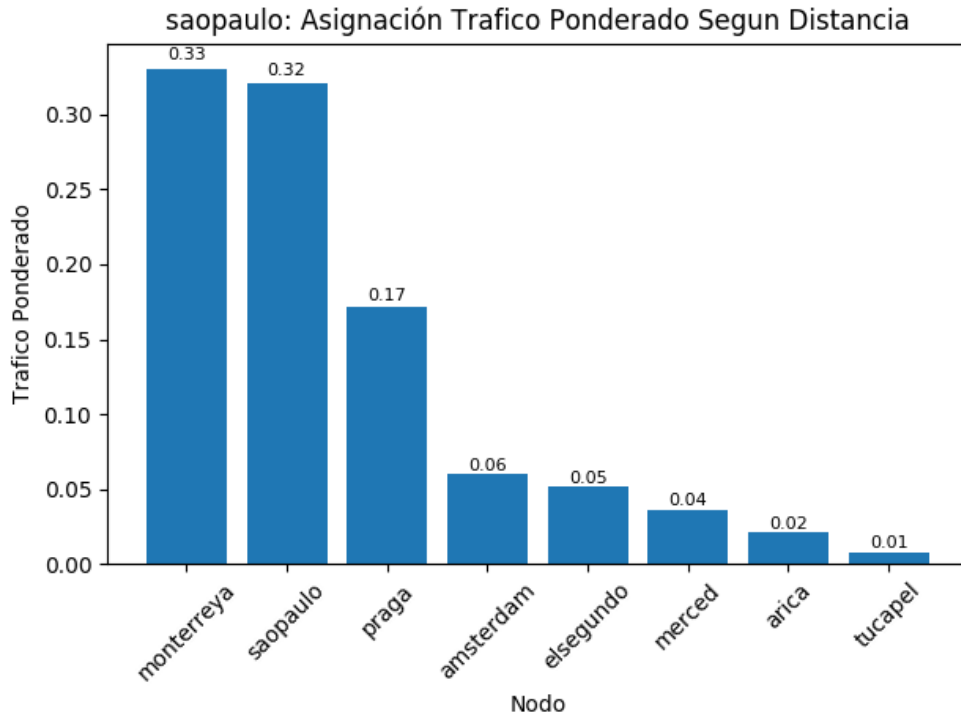


Figura 3.19: Gráfico que muestra la asignación esperada de las redes pertenecientes al área de servicio del nodo *saopaulo*, realizando una ponderación por el tráfico recibido desde las redes analizadas y comparando las distancias geográficas a todos los nodos.

3.3.5.4. tucape

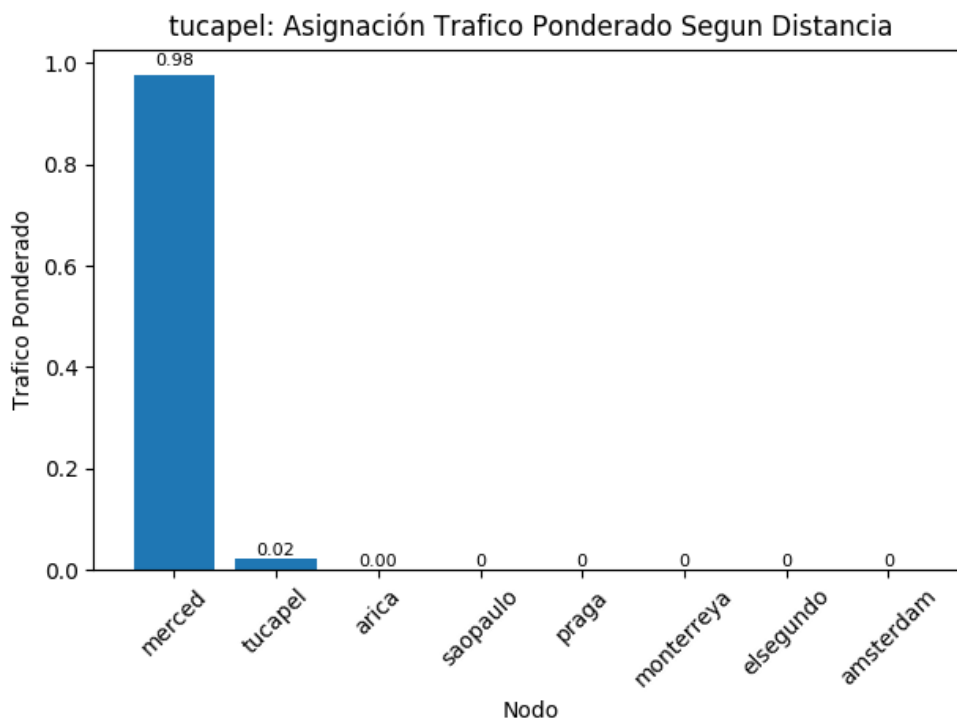


Figura 3.20: Gráfico que muestra la asignación esperada de las redes pertenecientes al área de servicio del nodo *tucape*, realizando una ponderación por el tráfico recibido desde las redes analizadas y comparando las distancias geográficas a todos los nodos.

3.3.5.5. Análisis de resultados

Podemos observar que una gran parte del tráfico recibido por los nodos *arica* y *merced* deberían encontrarse asignados al nodo en *merced*. Esto puede ser explicado dado que *merced* se encuentra en la capital de Chile, donde existe una concentración muy grande de población, en comparación a los otros nodos, generando así la mayor cantidad de tráfico.

Para el nodo *merced*, podemos observar que el 75 % del tráfico se encuentra asignado correctamente, seguido con un 22 % del tráfico que debería ser asignado a *tucape*, y porcentajes mucho menores a *arica* y *praga*.

Por último, podemos apreciar la diversidad de las redes capturadas por el nodo *saopaulo*. Un 33 % del tráfico fue correctamente asignado, seguido de un 32 % del tráfico debería ser asignado a *monterreya*, el cual podemos asociar al área de servicio capturada de la zona centro y este de Estados Unidos. En un grado menor, podemos apreciar que existe tráfico recibido desde Europa, que debería ser enrutado a los nodos en *praga* y *amsterdam*, además del tráfico que se recibe desde Chile.

3.4. Conclusiones

En este capítulo, propusimos y ejecutamos una metodología para observar cómo se comporta la red ante el despliegue de una red *anycast*, observando como esta realiza la asignación de los clientes a los diferentes nodos de la nube *anycast* según sus características geográficas.

Esta metodología fue aplicada la nube *anycast a.nic.cl* de *NIC Chile*, específicamente los nodos *arica*, *merced*, *saopaulo* y *tucapel*.

Utilizando la información capturada sobre el área de servicio de cada nodo extraída en el capítulo 2, logramos geolocalizar a nivel país las diferentes redes que pertenecen a las áreas de servicio de estos nodos, permitiéndonos observar con qué países cada nodo se comunica.

Esta información se contrastó con capturas de datos, que nos permiten comparar la información mostrada con las fuentes de tráfico, permitiéndonos observar los efectos en los clientes de las asignaciones que se están realizando actualmente.

Finalmente, realizamos una búsqueda de la asignación esperada si esta se realizara minimizando la distancia geográfica entre los clientes y cada uno de los nodos. Esto se realizó utilizando las redes en bruto y ponderando por el tráfico que cada red genera, para así observar cuanto tráfico se encuentra bien o mal asignado a cada nodo.

Esto nos permitió observar, que las áreas de servicio de cada nodo llegan a ser mucho más amplias que lo generalmente esperado, donde detectar estas es mucho más difícil utilizando las metodologías tradicionales que solo consideran los clientes actuales. Esto se presentó de manera destacada en el nodo *merced* el cual se comunica desde Chile hasta Europa, aun cuando existen nodos muchos más cercanos a estos.

Por otro lado, logramos apreciar la importancia de considerar el tráfico actual de la nube analizada, el cual puede generar una visión diferente de la efectividad del despliegue *anycast* actual, donde aunque un nodo posea un área de servicio geográficamente amplia, las consultas recibidas desde las redes más lejanas son mínimas.

Obtener las rutas utilizadas para conectarse a los diferentes nodos nos permite analizar las anomalías observadas, encontrando los *Sistemas Autónomos* que no se encuentran enrutando de la mejor manera el tráfico de la nube. Por ejemplo, logramos observar que la red *Telecom Italia Sparkle* genera que el nodo *merced* tenga una presencia mucho más alta en Italia que lo esperado. Esto nos permitirá tomar medidas para mejorar la situación actual con el mejor conocimiento de nuestra nube.

Utilizando esta metodología, logramos completar uno de los objetivos de este trabajo sobre la captura de las distancias geográficas de los diferentes nodos hasta los clientes, realizando una comparación con todos los nodos disponibles. Logramos observar que la asignación al nodo más cercano muchas veces es incorrecta, y generalmente solo se realiza de manera correcta a nivel nacional. Esto puede producirse por las políticas de ruteo de cada *Sistema Autónomo*, que pueden imponer penalizaciones o costos cuando el tráfico salen de las fronteras nacionales, disminuyendo además la visibilidad de la topología de red actual.

Finalmente, con esta información logramos refutar una parte de nuestra hipótesis, donde el servidor más cercano geográficamente al cliente **no** siempre responde las consultas enviadas al servidor *anycast DNS*.

Capítulo 4

Análisis Topológico

4.1. Introducción

Las nubes *anycast* buscan realizar la distribución del tráfico generado por sus clientes, acercando este a sus usuarios lo más posible, permitiendo reducir la latencia percibida y entregando la mejor velocidad de respuesta y resiliencia ante posibles fallos.

Muchas veces, se dice que una nube *anycast* minimiza los tiempos de conexión entre esta nube y sus clientes, sin embargo, realizar demostraciones de esto es una tarea mucho más difícil.

Obtener información sobre la latencia y rutas utilizadas por los diferentes nodos hasta sus usuarios, nos permite observar el estado actual de una nube *anycast*, además de permitirnos identificar diferentes puntos en que nuestra red es más débil, o diferentes áreas que posee una alta latencia.

En este capítulo, se describirá una metodología que nos permita observar el estado de la red actual, identificando la latencia que poseen los clientes al acceder a los diferentes nodos de una nube *anycast*, verificando la suposición sobre la asignación de un cliente a su servidor *anycast* más cercano en términos de latencia. Esto será aplicado a la nube *anycast a.nic.cl* de *NIC Chile*, desde la cual realizamos la captura de información sobre su área de servicio en el capítulo 2.

4.2. Metodología

En este capítulo, buscamos obtener la siguiente información sobre la nube analizada.

- Distribución de latencia por cada prefijo de red /24.
- Distribución de latencia por cada *Sistema Autónomo*.
- Distribución de latencia según el tráfico recibido.

- Número de *Sistema Autónomo* por cantidad de saltos.
- Distribución de tráfico en tránsito en los diferentes *Sistemas Autónomos*.
- Comparación de las distancias topológicas en términos de latencia entre usuarios, su nodo asignado en la nube *anycast*, y todos los otros nodos en esta nube.

Para obtener información sobre los saltos y latencia desde cada uno de los nodos hasta sus clientes, es necesario conocer el área de servicio en términos topológicos de los nodos analizados, para lo cual, utilizaremos el método descrito en el capítulo 2, el cual nos entregará las diferentes redes que pertenecen a cada nodo analizado, en conjunto con la latencia a cada cliente y las rutas utilizadas.

Las mediciones realizadas se agruparon por prefijos de red /24, el cual corresponde al tamaño mínimo que puede ser utilizado en la internet pública para realizar un anuncio a través del protocolo *BGP*.

Sobre cada uno de estos prefijos de red, se realizó el cálculo de la mediana de la latencia, permitiéndonos obtener la latencia esperada por los clientes para acceder al servicio. Para generar una visualización de estos resultados, se realizó una agrupación de la cantidad de redes sobre un rango de latencias.

Este proceso se repitió convirtiendo los prefijos de red /24 a su *Sistema Autónomo* correspondiente, calculando la media total de las mediciones y agrupándolos en rangos de latencia.

Para verificar la efectividad del despliegue actual, se realizará una comparación de la información en bruto y una ponderación por la cantidad de tráfico recibido por estos nodos desde los clientes, permitiendo observar los efectos reales del despliegue actual sobre los clientes.

La información sobre los saltos y tráfico que cada *Sistema Autónomo* maneja, se calculará utilizando los trazados de ruta obtenidos desde el análisis realizado en el capítulo 2. Sobre cada trazado de ruta generado, se obtendrá la cadena de *Sistemas Autónomos* requeridos para llegar a cada uno de los clientes, anotando el salto en que se observa su primera aparición, y la cantidad de tráfico que transita por cada uno de estos hasta llegar a su destino.

Por último, para realizar una comparación de las distancias topológicas en términos de latencia desde un cliente hacia los diferentes nodos, utilizaremos la red de medición *RIPE Atlas* [4] para generar consultas *ICMP Ping* hacia todos los nodos pertenecientes a la nube *anycast* medida, almacenando el tiempo requerido para obtener una respuesta desde estos.

Para obtener el nodo verdaderamente asignado, se realizaron consultas *DNS CHAOS* a la nube *anycast*, consultando por el campo *version.bind*. La respuesta obtenida generalmente posee información única que nos permite identificar el nodo con el cual nos estamos comunicando.

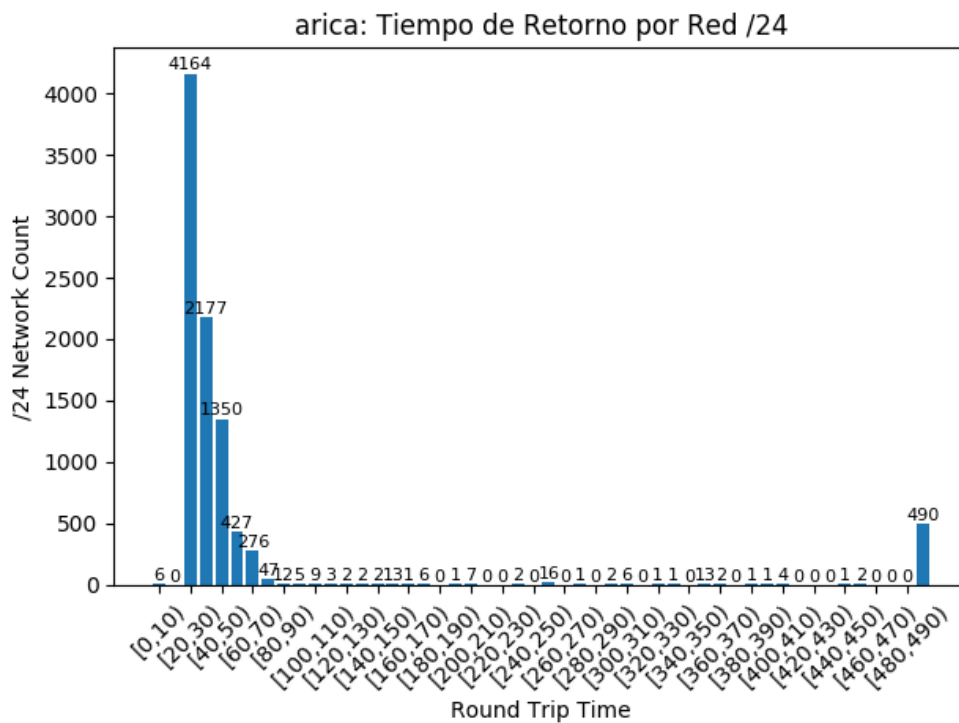
4.3. Resultados

Los siguientes resultados fueron obtenidos utilizando la metodología descrita anteriormente, aplicada a la nube *a.nic.cl* de *NIC Chile*. Esta se aplica a los cuatro nodos correspondientes a *arica*, *merced*, *saopaulo* y *tucapel* analizados en el capítulo 2.

4.3.1. Redes por Latencia

Los gráficos presentados a continuación, nos muestran el número de prefijos de red /24 por conjuntos de latencia dividida cada 10 milisegundos. Esto es analizado desde los nodos *arica*, *merced*, *saopaulo* y *tucapel*.

4.3.1.1. arica



4.3.1.2. merced

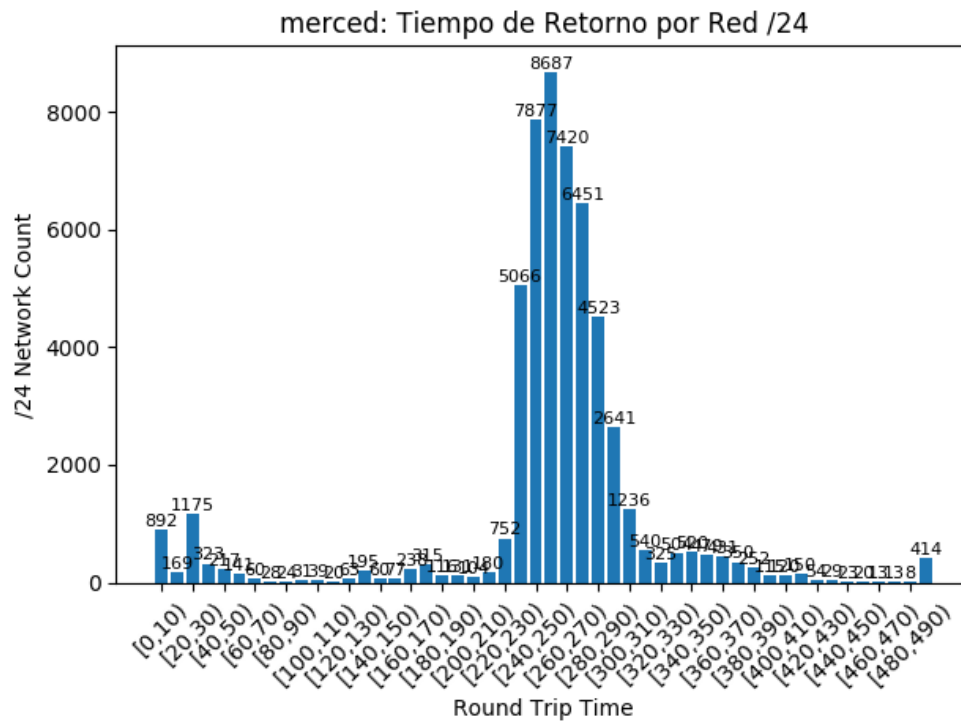


Figura 4.2: Gráfico que muestra el número de redes agrupados por su latencia, dividiendo esta en conjuntos de 10 milisegundos. Las redes presentadas pertenecen al área de servicio del nodo *merced*.

4.3.1.3. saopaulo

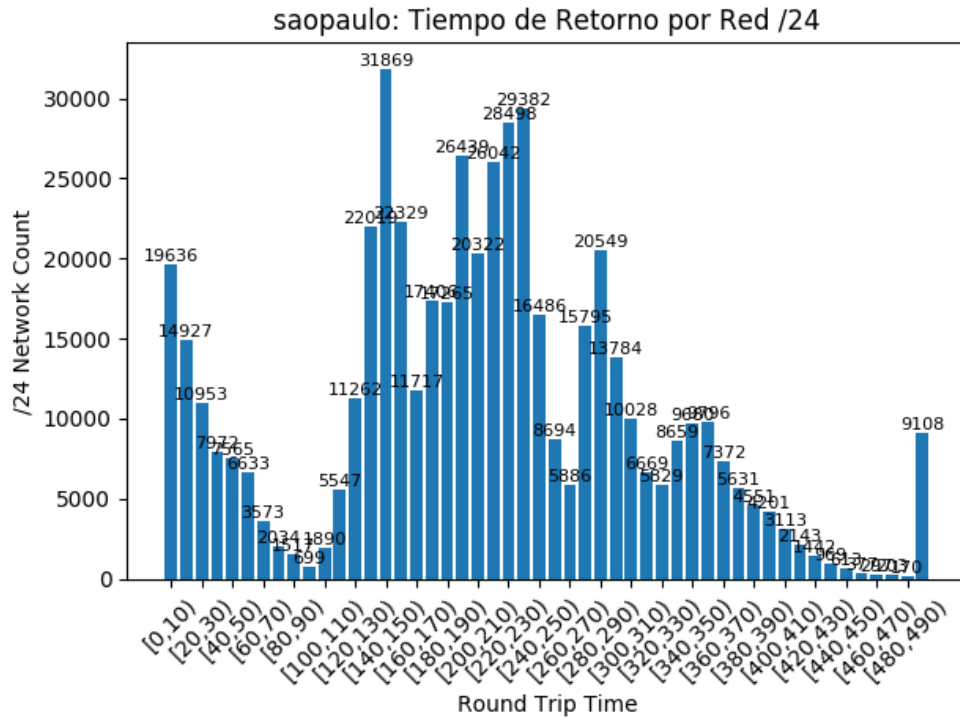


Figura 4.3: Gráfico que muestra el número de redes agrupados por su latencia, dividiendo esta en conjuntos de 10 milisegundos. Las redes presentadas pertenecen al área de servicio del nodo *saopaulo*.

4.3.1.4. tuapel

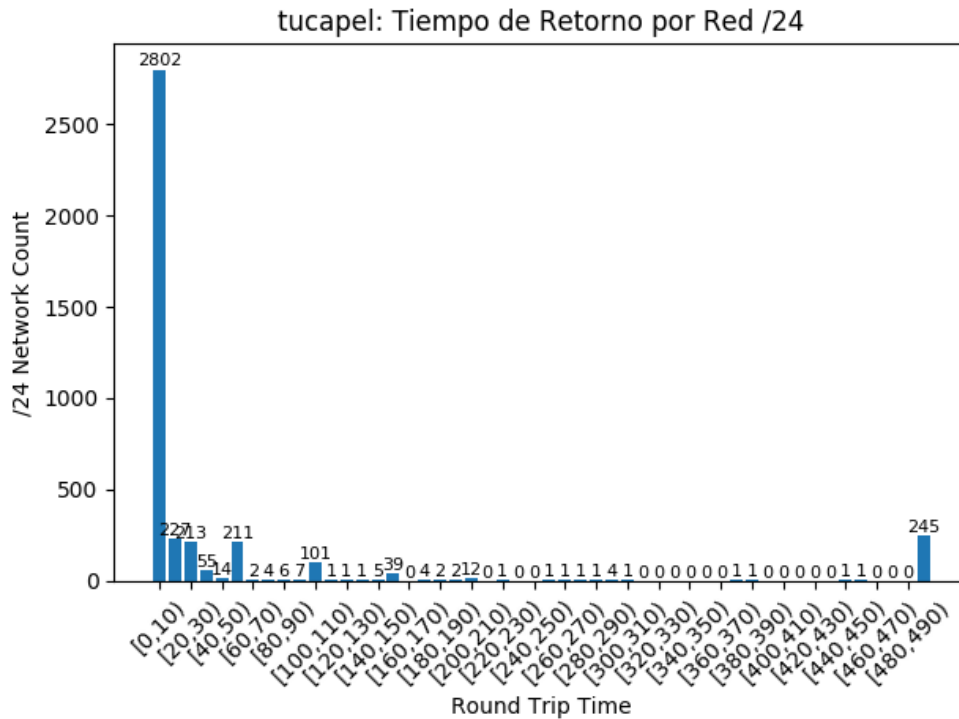


Figura 4.4: Gráfico que muestra el número de redes agrupados por su latencia, dividiendo esta en conjuntos de 10 milisegundos. Las redes presentadas pertenecen al área de servicio del nodo *tuapel*.

4.3.1.5. Análisis de resultados

Podemos observar que la latencia observada en el nodo *arica* comienza desde los 20 milisegundos, con una cantidad muy baja de prefijos de red /24 con una latencia menor. Esto nos muestra que los clientes que se comunican con este nodo no se encuentran tan cerca como se esperaría, y existe una limitante topológica que genera el retraso observado.

En el nodo *merced*, podemos observar que existe una alta tasa de conexión en los primeros 30 milisegundos, y luego se presenta un salto en el rango de los 200 a 300 milisegundos. Este salto lo observamos en el capítulo 3, el cual se conecta hasta Italia a través de la red de *Telecom Italia Sparkle*, lo cual concuerda con la latencia observada hacia estos nodos.

Para los resultados del nodo *saopaulo* podemos observar principalmente cuatro puntos que poseen una gran cantidad de redes. El primero lo podemos asociar a las redes nacionales que se encuentran en Brasil, en el rango de 0 a 60 milisegundos. El segundo lo podemos encontrar en los 90 a 150 milisegundos, el cual corresponde a la latencia entre Sao Paulo y Estados Unidos. El rango de 160 a 230 milisegundos corresponde al área de los países europeos tal como Francia y Alemania. Por último, el rango de 250 a 300 corresponde al continente asiático, con países como Japón.

Por último, podemos observar que el nodo *tucapel* concentra casi todas sus redes en los primeros 10 milisegundos, demostrando que este sirve mayoritariamente a los clientes locales.

4.3.2. Sistemas Autónomos por Latencia

Los gráficos presentados a continuación, nos muestran el número de *Sistemas Autónomos* por conjuntos de latencia dividida cada 10 milisegundos. Esto es analizado desde los nodos *arica*, *merced*, *saopaulo* y *tucapel*.

4.3.2.1. arica

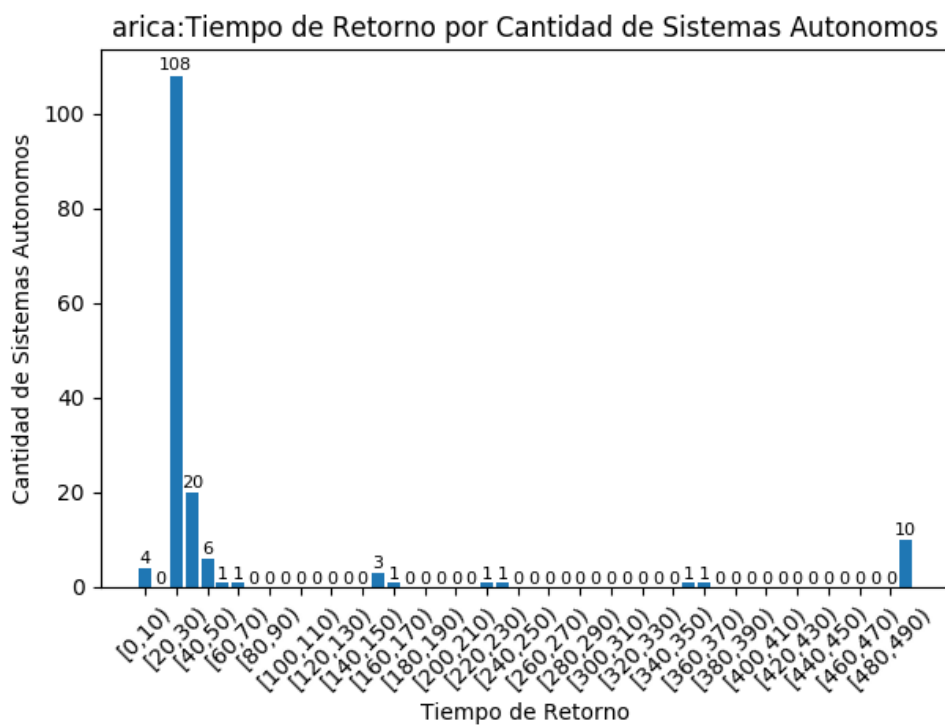


Figura 4.5: Gráfico que muestra el número de *Sistemas Autónomos* agrupados por su latencia, dividiendo esta en conjuntos de 10 milisegundos. Los *Sistemas Autónomos* redes presentadas pertenecen al área de servicio del nodo *arica*.

4.3.2.2. merced

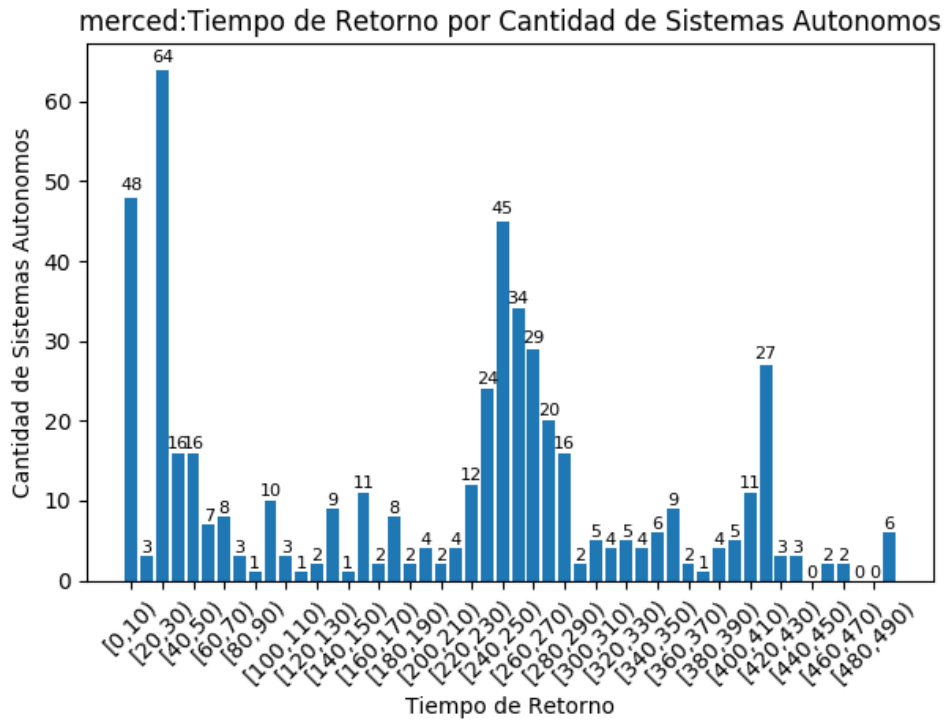


Figura 4.6: Gráfico que muestra el número de *Sistemas Autónomos* agrupados por su latencia, dividiendo esta en conjuntos de 10 milisegundos. Los *Sistemas Autónomos* redes presentadas pertenecen al área de servicio del nodo *merced*.

4.3.2.3. saopaulo

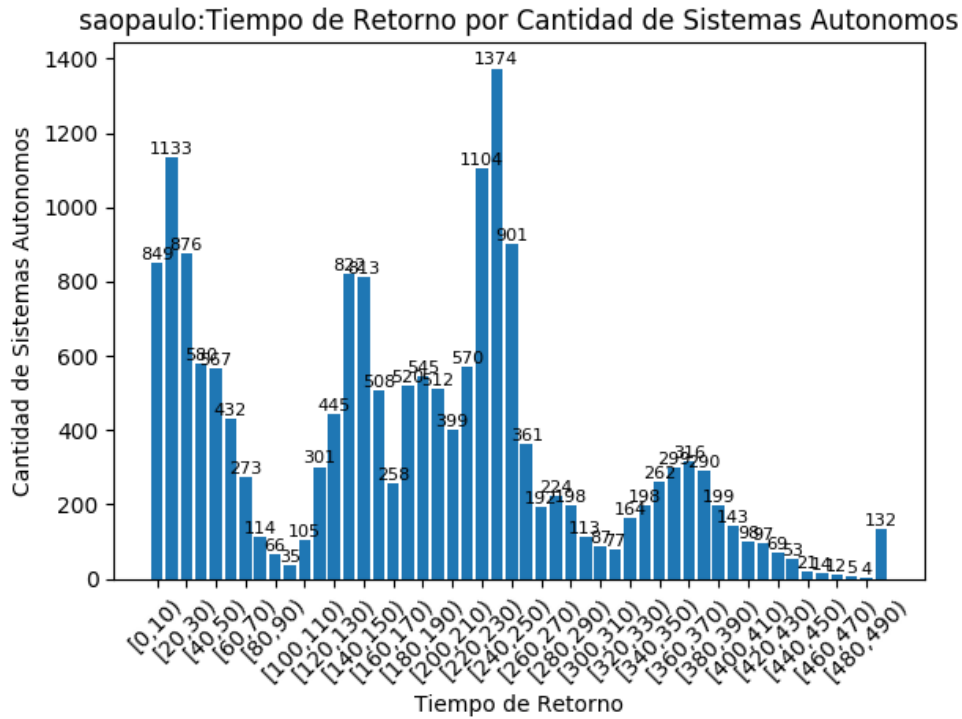


Figura 4.7: Gráfico que muestra el número de *Sistemas Autónomos* agrupados por su latencia, dividiendo esta en conjuntos de 10 milisegundos. Los *Sistemas Autónomos* redes presentadas pertenecen al área de servicio del nodo *saopaulo*.

4.3.2.4. tuapel

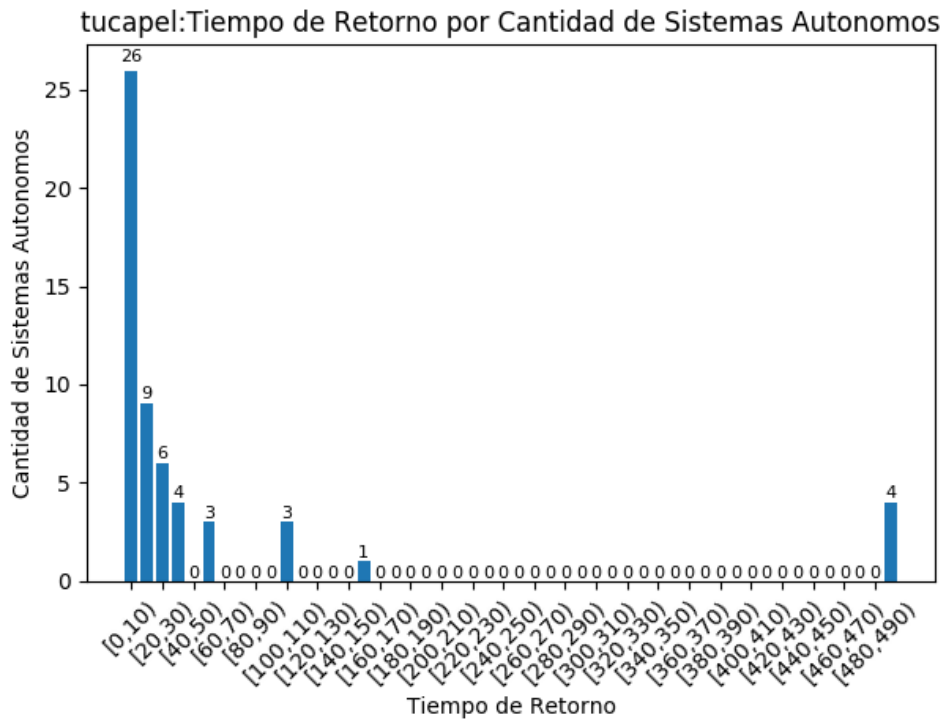


Figura 4.8: Gráfico que muestra el número de *Sistemas Autónomos* agrupados por su latencia, dividiendo esta en conjuntos de 10 milisegundos. Los *Sistemas Autónomos* redes presentadas pertenecen al área de servicio del nodo *tuapel*.

4.3.2.5. Análisis de resultados

Podemos apreciar que los nodos *arica* y *tuapel* se mantienen muy semejantes al análisis de prefijos de red /24 realizado anteriormente, manteniendo la conexión a baja latencia.

Para el nodo *merced*, podemos ver de mejor manera la cantidad de nodos locales, gracias a la agrupación de las redes con alta latencia, que si bien son una gran cantidad, se encuentran contenida en un número menor de *Sistemas Autónomos*. Se puede destacar que el nodo *merced* concentra una gran cantidad de usuarios en los rangos 0-10 y 20-30 milisegundos, con una cantidad mínima de usuarios en el rango 10-20 milisegundos. Esto nos muestra que existe una separación entre la red local y nacional, que involucra un alto costo en latencia.

Por último, el nodo *saopaulo* conserva la misma forma de latencia observada en la sección anterior. En este gráfico destacan de mejor manera los *Sistemas Autónomos* locales, mostrando que existe una gran conectividad con los nodos nacionales.

4.3.3. Tráfico Ponderado por Latencia

Los gráficos presentados a continuación, nos muestran el tráfico distribuido por conjuntos de latencia dividida cada 10 milisegundos. Esto es analizado desde los nodos *arica*, *merced*, *saopaulo* y *tucaapel*.

4.3.3.1. arica

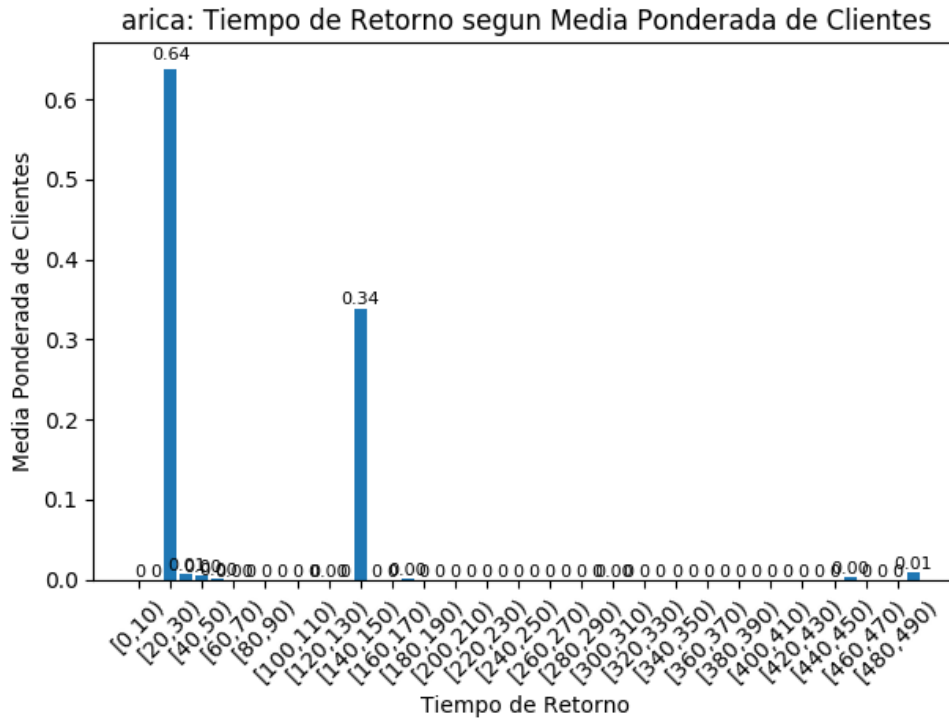


Figura 4.9: Gráfico que muestra el tráfico ponderado agrupados por su latencia, dividiendo esta en conjuntos de 10 milisegundos. Los *Sistemas Autónomos* presentados pertenecen al área de servicio del nodo *arica*.

4.3.3.2. merced

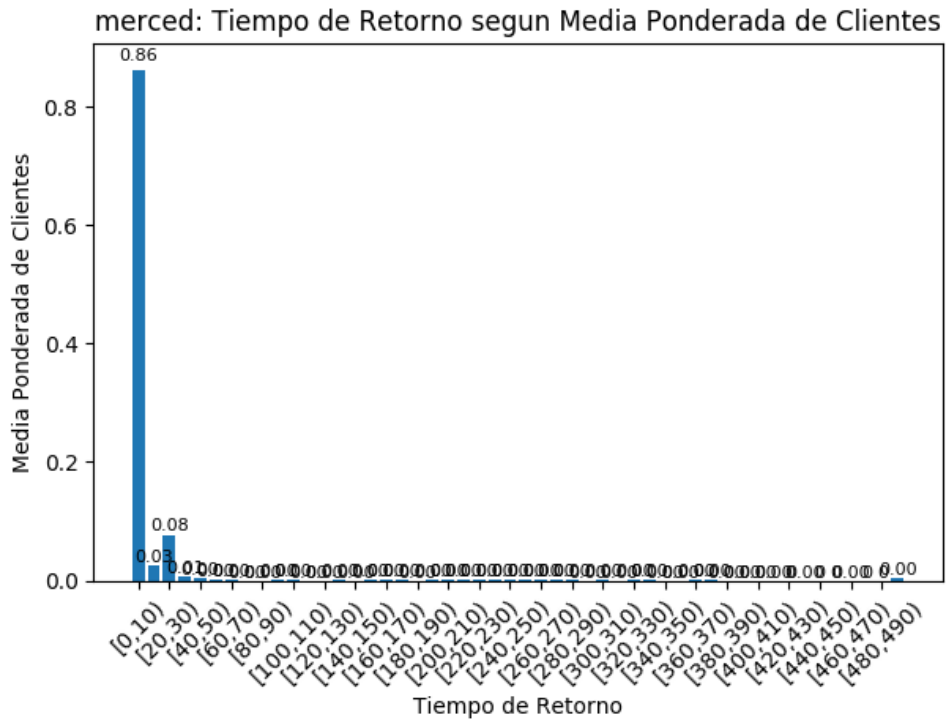


Figura 4.10: Gráfico que muestra el tráfico ponderado agrupados por su latencia, dividiendo esta en conjuntos de 10 milisegundos. Los *Sistemas Autónomos* presentados pertenecen al área de servicio del nodo *merced*.

4.3.3.3. saopaulo

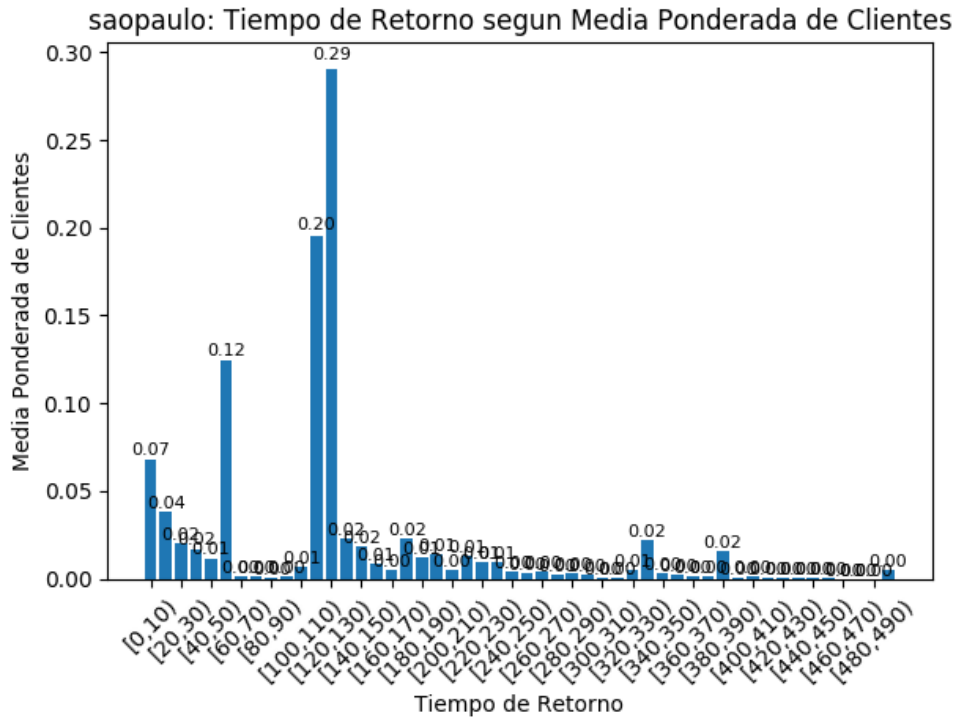


Figura 4.11: Gráfico que muestra el tráfico ponderado agrupados por su latencia, dividiendo esta en conjuntos de 10 milisegundos. Los *Sistemas Autónomos* presentados pertenecen al área de servicio del nodo *saopaulo*.

4.3.3.4. tucapep

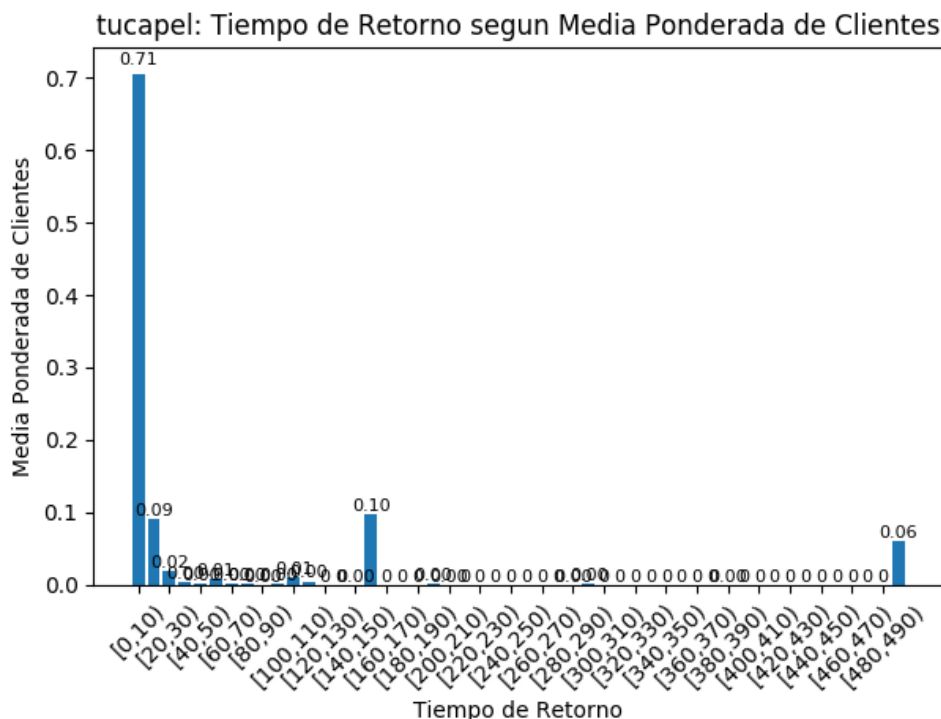


Figura 4.12: Gráfico que muestra el tráfico ponderado agrupados por su latencia, dividiendo esta en conjuntos de 10 milisegundos. Los *Sistemas Autónomos* presentados pertenecen al área de servicio del nodo *tucapep*.

4.3.3.5. Análisis de resultados

En la mayoría de los gráficos, podemos observar que el tráfico se encuentra muy concentrado en pequeños rangos de latencia.

Para el nodo de *arica*, el tráfico se concentra en el rango 20-30 y 120-130 milisegundos, con el 64% y 34% del tráfico recibido respectivamente. El primer rango lo podemos asociar al recibido por la conexión nacional a Chile, y el segundo a las conexiones con Estados Unidos y Canadá.

El nodo *merced* posee la gran mayoría de su tráfico concentrado en el rango 0-30 milisegundos, respondiendo las consultas locales. Además de esto, es posible observar las mínimas consultas recibidas desde el extranjero, que llegan a contar un 1% de las recibidas desde Chile.

Podemos observar que el nodo *saopaulo* recibe una alta cantidad de consultas de manera local y desde el rango de 90-110 milisegundos, el cual corresponde a Estados Unidos.

Por último, el tráfico recibido desde *tucapep* se concentra en el rango de 0-20 milisegundos, mostrando una alta localidad en las respuestas. El segundo punto notorio, es posible verlo

en el rango 130-140 milisegundos, que logramos observar que corresponde al tráfico recibido desde Estados Unidos.

4.3.4. Sistemas Autónomos por Salto

Los gráficos presentados a continuación, nos muestran cuantos *Sistemas Autónomos* se encuentran a una cantidad de saltos determinada, comenzando desde los *Sistemas Autónomos* inmediatamente posteriores al que recibe el tráfico inicial. Esto es analizado desde los nodos *arica*, *merced*, *saopaulo* y *tucapel*.

4.3.4.1. arica

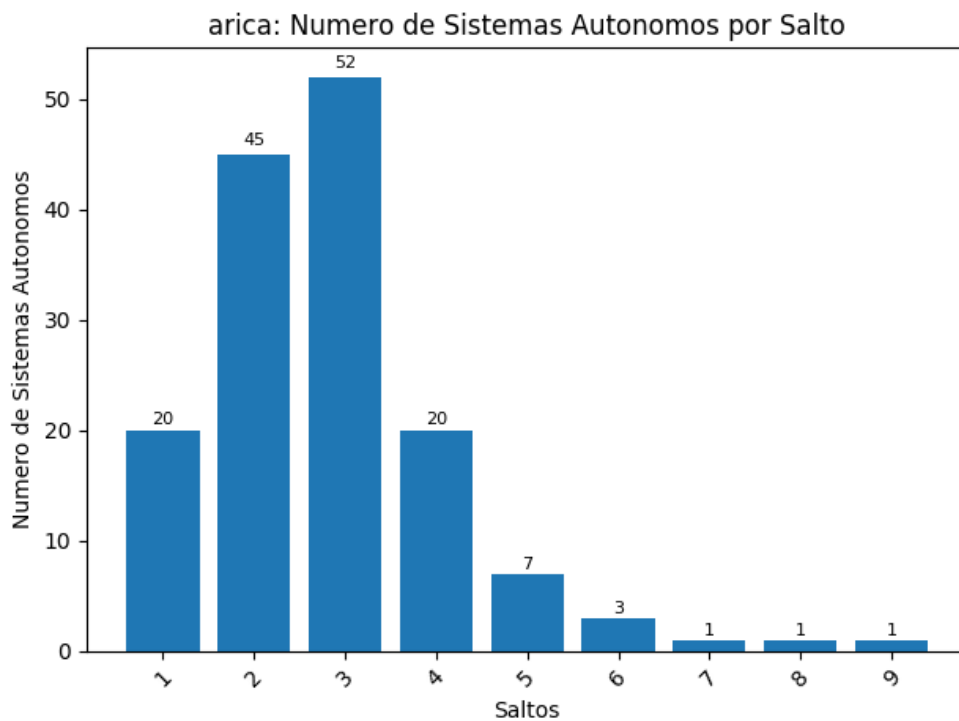


Figura 4.13: Gráfico que muestra cuantos *Sistemas Autónomos* son encontrados en una cantidad de saltos determinada. Los *Sistemas Autónomos* presentados pertenecen al área de servicio del nodo *arica*.

4.3.4.2. merced

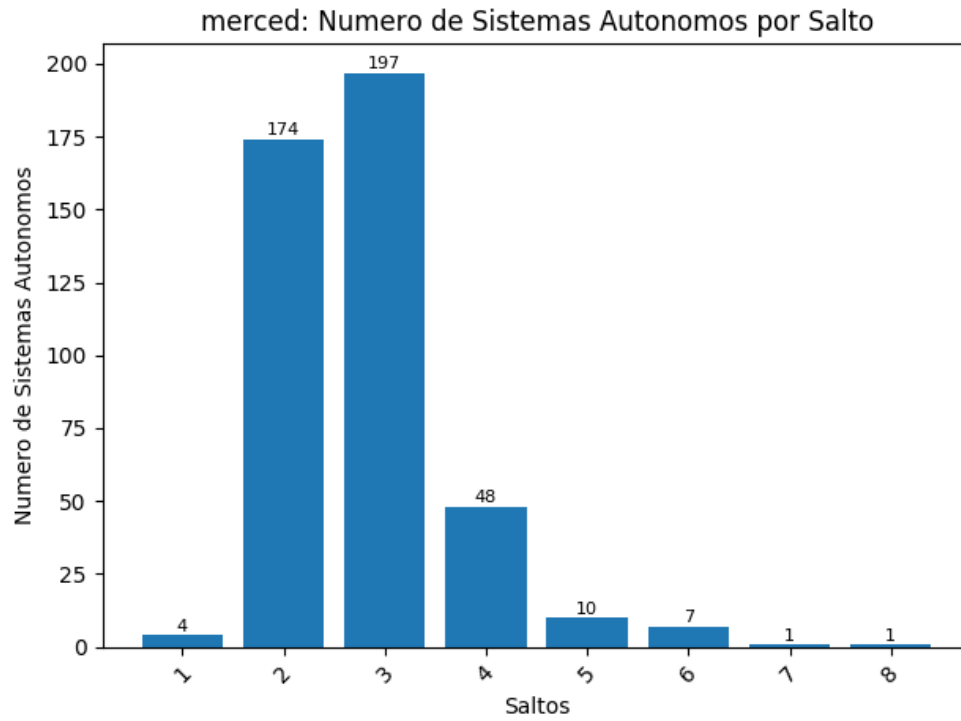


Figura 4.14: Gráfico que muestra cuantos *Sistemas Autónomos* son encontrados en una cantidad de saltos determinada. Los *Sistemas Autónomos* presentados pertenecen al área de servicio del nodo *merced*.

4.3.4.3. saopaulo

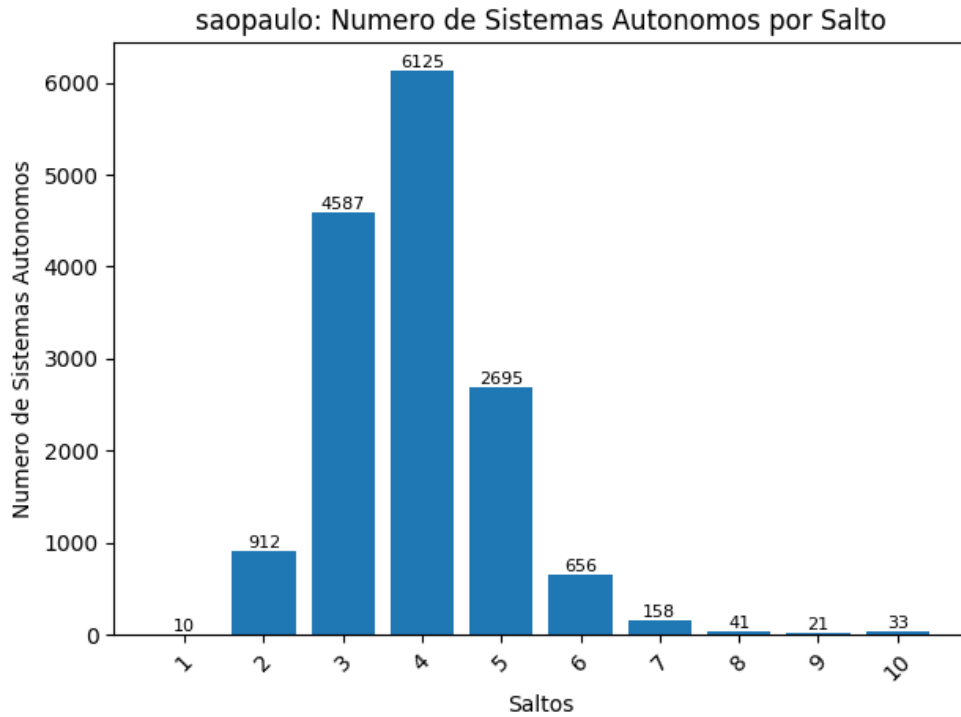


Figura 4.15: Gráfico que muestra cuantos *Sistemas Autónomos* son encontrados en una cantidad de saltos determinada. Los *Sistemas Autónomos* presentados pertenecen al área de servicio del nodo *saopaulo*.

4.3.4.4. tucape1

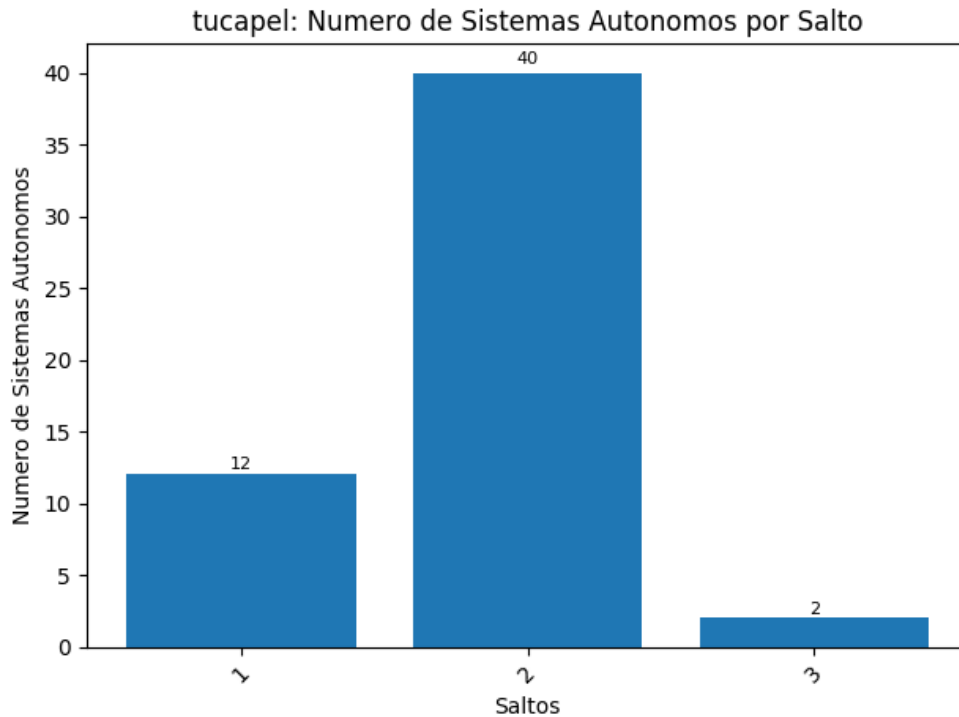


Figura 4.16: Gráfico que muestra cuantos *Sistemas Autónomos* son encontrados en una cantidad de saltos determinada. Los *Sistemas Autónomos* presentados pertenecen al área de servicio del nodo *tucape1*.

4.3.4.5. Análisis de resultados

Podemos apreciar que el nodo *arica* posee una gran conectividad en su salto inicial, comunicándose con 20 diferentes *Sistemas Autónomos*, aumentando al doble en el siguiente salto. La concentración de *Sistemas Autónomos* se puede apreciar entre el segundo y tercer salto.

Para el nodo *merced*, podemos ver que la cantidad de conexiones iniciales es muy baja, comunicándose solo con 4 *Sistemas Autónomos*. Esto nos dice que este nodo puede encontrarse mucho más propenso a fallas dependiendo de la conectividad de estos 4 *Sistemas Autónomos*. La cantidad de *Sistemas Autónomos* realiza un gran salto entre el segundo y tercer salto, el cual podemos asociar con su conectividad internacional directa a través de ENTEL.

El nodo *saopaulo* posee la mayor interconectividad desde el tercer salto en adelante, demostrando la alta conectividad que este posee, dado su lejanía a otros nodos. Podemos destacar, que la cantidad de *Sistemas Autónomos* por salto, es aproximadamente diez veces mayor que los otros nodos, verificando esta alta conectividad que el nodo posee.

Finalmente, el nodo *tucape1* se conecta con una cantidad muy baja de *Sistemas Autónomos*, llegando a sumar 54 en toda su área de servicio. Esto nos muestra, que este nodo se encuentra

especializado en responder el tráfico local, aun cuando el área de servicio posea un bajo porcentaje de lugares extranjeros.

4.3.5. Tránsito por Sistemas Autónomos

Los gráficos presentados a continuación nos muestran los *Sistemas Autónomos* sobre los cuales transita la mayor cantidad de tráfico generado por cada uno de los clientes, presentando el porcentaje del tráfico que cada uno de estos *Sistemas Autónomos* maneja. Esto es analizado desde los nodos *arica*, *merced*, *saopaulo* y *tucapel*.

4.3.5.1. arica

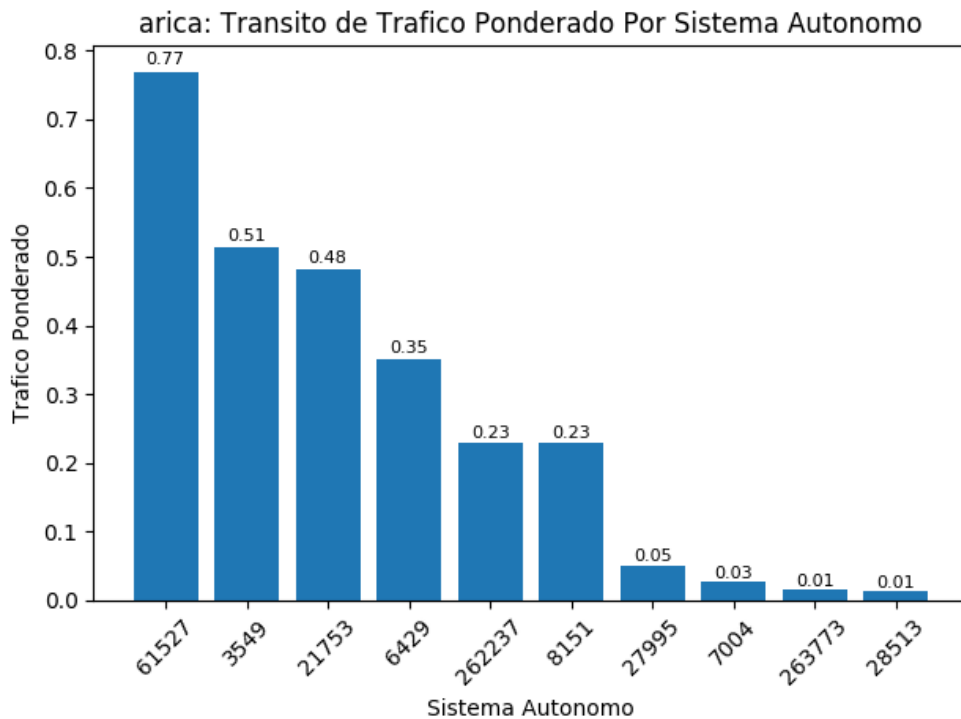


Figura 4.17: Porcentaje del tráfico que transita por los diferentes *Sistemas Autónomos* que se encuentran en el área de servicio del nodo *arica*.

4.3.5.2. merced

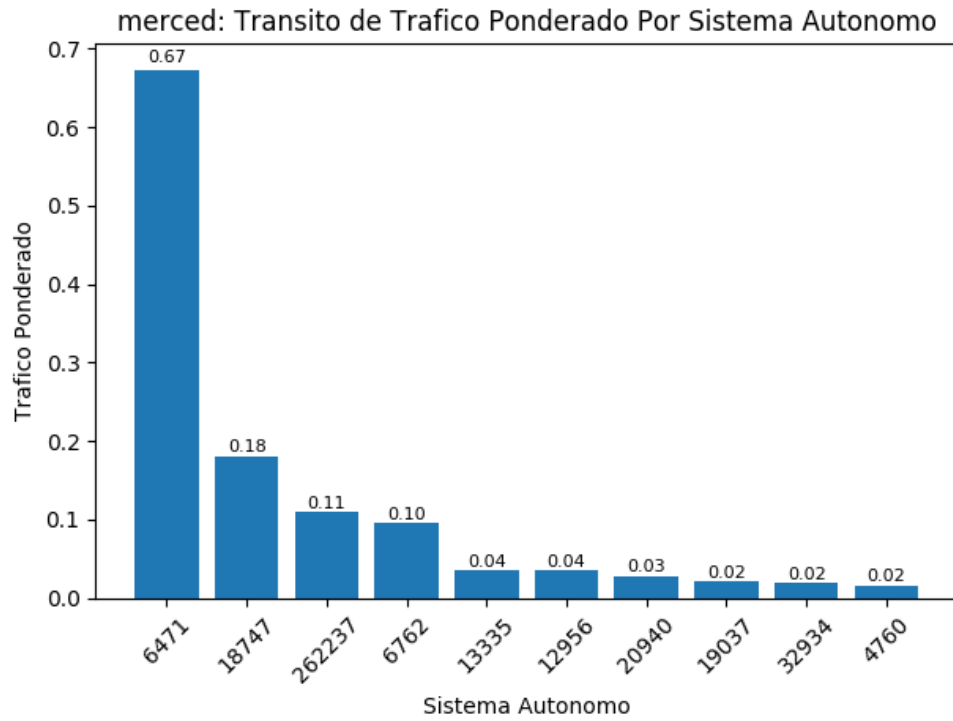


Figura 4.18: Porcentaje del trafico que transita por los diferentes *Sistemas Autónomos* que se encuentran en el área de servicio del nodo *merced*.

4.3.5.3. saopaulo

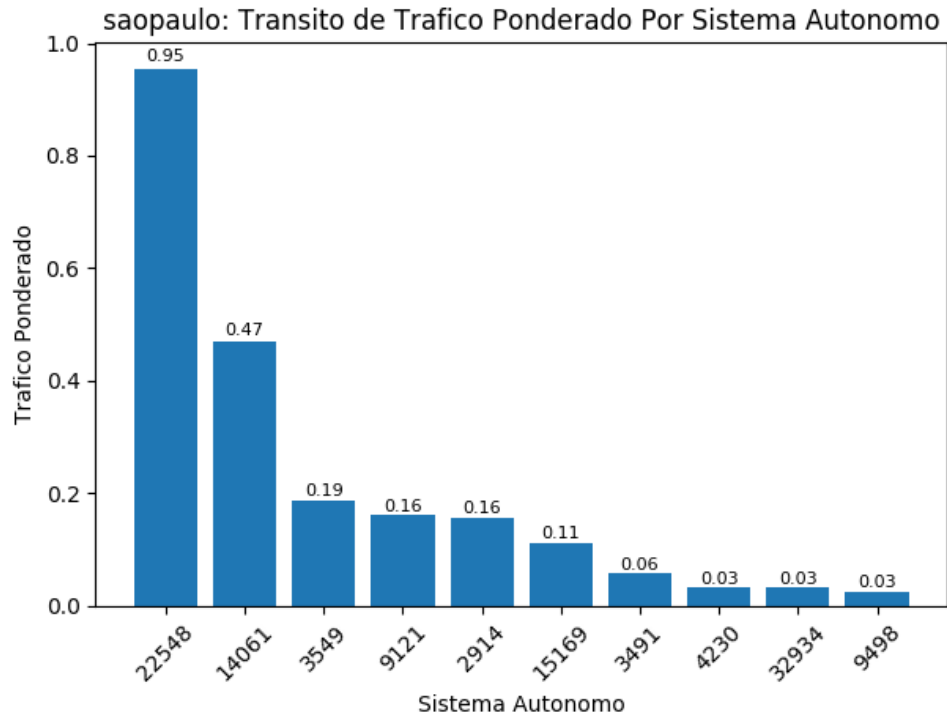


Figura 4.19: Porcentaje del trafico que transita por los diferentes *Sistemas Autónomos* que se encuentran en el área de servicio del nodo *saopaulo*.

4.3.5.4. tucape

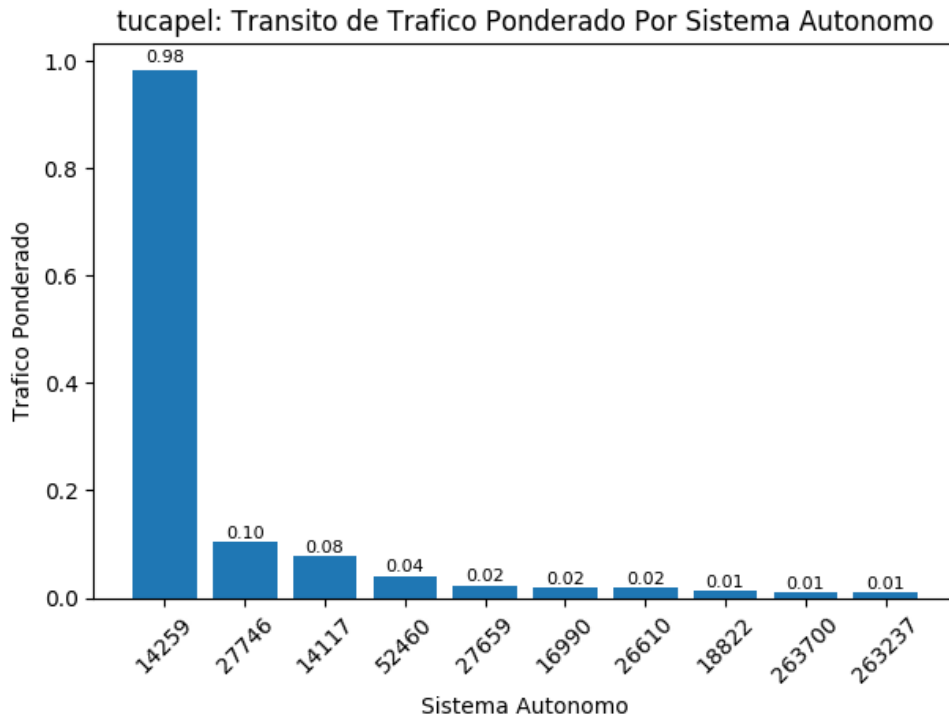


Figura 4.20: Porcentaje del trafico que transita por los diferentes *Sistemas Aut6nomos* que se encuentran en el 6rea de servicio del nodo *tucape*.

4.3.5.5. An6lisis de resultados

Podemos apreciar que cada uno de los nodos posee un *Sistema Aut6nomos* inicial que concentra la mayor cantidad de tr6fico, el cual representa uno de los primeros *Sistemas Aut6nomos* detectados en cada nodo, sobre el cual existe la mayor cantidad de tr6fico.

Para el nodo de *arica*, podemos observar que la mayor comunicaci6n ocurre a trav6s de los *Sistemas Aut6nomos* *PIT CHILE* (61527) y *Level 3* (3549), los cuales corresponden a grandes redes que se encuentran en Chile.

En el nodo *merced*, podemos observar que la mayor1a del tr6fico se encuentra en el *Sistema Aut6nomos* *ENTEL CHILE* (6471), donde los *Sistemas Aut6nomos* siguientes poseen un porcentaje mucho m6s bajo del tr6fico manejado por este nodo. Algunos de estos corresponden a *IFX* (18747), *Orbyta* (262237) y *ITALIA SPARKLE* (6762), los cuales corresponden a grandes redes que no generan un alto tr6fico en relaci6n a su tama1o, esto en comparaci6n al gran tr6fico generado por la red de *Entel*.

Para *saopaulo*, podemos apreciar que el nodo inicial corresponde a *NIC.BR* (22548), donde luego el tr6fico se divide en diferentes redes tales como *Level 3* (3549), *TTNET* (9121) y *NTT America* (2914). Cabe destacar el alto tr6fico recibido desde el *Sistema Aut6nomos*

DigitalOcean (14601), proveedor de servidores virtuales con datacenters al este y oeste de Estados Unidos.

Finalmente, el nodo *tucapel* provee sus servicios principalmente a través de *Gtd Internet* (14259), donde luego existen clientes con una taza mucho menor de tráfico tal como *LAN CHILE* (27746) y *Telefonica del Sur* (14117).

4.3.6. Asignación Topológica

La asignación topológica se calculó utilizando la plataforma *RIPE Atlas*, en la que se utilizaron 687 puntos de medición diferentes, tomando la mayor cantidad de nodos en Latinoamérica para prevenir sesgar los resultados a Europa, dada la distribución de puntos de medición de esta plataforma.

Cada uno de los puntos de medición, realizó una medición de latencia a todos los nodos que pertenecen a la nube *a.nic.cl*, realizando una comparación con el nodo realmente asignado enviando una consulta *DNS CHAOS* a la nube *a.nic.cl*.

Los resultados de estas mediciones podemos observarlos en la tabla 4.1, donde los puntos de medición asignados a cada nodo de la nube *anycast* muestran si este verdaderamente corresponde al más cercano a estos en términos de latencia.

Nodo	Asignado al más cercano	No asignado al más cercano
Arica	0	17
Merced	5	11
Sao Paulo	29	209
Tucapel	0	2
Praga	150	50
Amsterdam	32	121
El Segundo	28	23
Monterrey A	2	8

Tabla 4.1: Tabla que muestra los resultados de las mediciones realizadas a través de la plataforma *RIPE Atlas*. Cada fila muestra la cantidad de puntos de medición que fueron asignados o no a su nodo más cercano según su distancia topológica en términos de latencia.

Esta tabla nos muestra que solo un 35% de los puntos de medición fue correctamente asignado al nodo más cercano, donde el resto debería haber sido asignado a un nodo diferente para minimizar su latencia de acceso a la nube.

4.4. Conclusiones

En este capítulo, propusimos y ejecutamos una metodología para observar diferentes métricas importantes, que nos permitan determinar el estado de un despliegue *anycast*, en conjunto

con un método para determinar si los clientes son asignados al nodo *anycast* más cercano en términos de latencia.

Para observar los resultados generados por esta metodología, aplicamos esta a la nube *anycast* de *NIC Chile*, específicamente los nodos *arica*, *merced*, *saopaulo* y *tucapel*.

Utilizando la información capturada sobre el área de servicio, latencia y rutas utilizadas extraídas desde el capítulo 2, logramos generar información sobre la distribución de tráfico, latencia de las redes y *Sistemas Autónomos* que componen las áreas de servicio de cada nodo analizado.

Para verificar la suposición de que la nube *anycast* minimiza los tiempos de respuesta a todos los clientes, se realizó una medición a través de la plataforma RIPE Atlas hacia todos los nodos, comparando la latencia hacia el nodo asignado con los demás.

Los resultados obtenidos nos permitieron observar que existe una gran relación de los resultados geográficos y topológicos, donde es posible apreciar en los gráficos de latencia, el área general donde se encuentran los *Sistemas Autónomos*, verificando las mediciones realizadas.

Por otro lado, logramos apreciar que el tráfico ponderado para los nodos medidos se encuentra muy concentrado en ciertos rangos de latencia, lo cual nos dice que existen concentraciones de usuarios que realizan consultas a la nube. Esto puede ser utilizado para la planeación de nueva infraestructura, buscando disminuir los grupos con latencia más grande.

Un nodo que podemos tomar como ejemplo es *saopaulo*, el cual posee una gran cantidad de tráfico enrutado desde Estados Unidos, respondiendo con una latencia de más de 90 milisegundos. Esto se puede verificar observando el enrutamiento de tráfico de este nodo, donde logramos encontrar que un 47% del tráfico transita por el *Sistema Autónomo* de *DigitalOcean*, un proveedor estadounidense.

La nube de medición de *RIPE Atlas* nos permitió observar que la correcta asignación de los nodos se genera en aproximadamente un 35% total de las veces, realizando mediciones desde distintos puntos de la red. Esto nos muestra que la red no siempre logra entregarnos la ruta corta en términos de latencia al nodo más cercano de una nube *anycast*, **negando** el segundo punto de nuestra hipótesis.

A pesar de esto, es necesario mencionar que los nodos *anycast* desplegados, logran minimizar la latencia del tráfico que se genera localmente, como vimos en los análisis del tráfico ponderado. Esto nos enseña, que desplegar servidores *anycast* de manera cercana a altas fuentes de tráfico, es un método efectivo de minimizar la latencia global de la nube, esto sin considerar conexiones internacionales que son mucho más difíciles de controlar.

Capítulo 5

Posicionamiento de Nueva Infraestructura

5.1. Introducción

Realizar el posicionamiento de un nuevo servidor *anycast* es una tarea muy difícil, donde muchas veces se realiza observando los lugares geográficos o *Sistemas Autónomos* que generan la mayor cantidad de tráfico.

Estos métodos, generalmente no poseen la suficiente información para tomar una decisión que afecte a la mayor cantidad de clientes, o que desvíe la mayor cantidad de tráfico desde un nodo a otro, permitiendo aliviar el tráfico que recibe en promedio cada uno de ellos.

En este capítulo, definiremos una metodología que nos permita determinar, diferentes posiciones óptimas para realizar la instalación de nueva infraestructura según diferentes métricas, observando los efectos producidos por cada uno de estos, aprovechando la nueva información que nos han entregado los capítulos anteriores.

5.2. Metodología

En esta sección, describiremos diferentes metodologías que nos permitan optimizar el posicionamiento de un nuevo servidor según diferentes métricas.

Luego de esto, definiremos un método para verificar la metodología propuesta utilizando una nube *anycast* de prueba.

5.2.1. Metodologías de Optimización

El posicionamiento óptimo de un nuevo servidor en una nube *anycast* varía según el parámetro que uno desea optimizar, por lo que analizaremos dos parámetros principales.

- **Minimización de Latencia:** Este cálculo involucra utilizar la latencia que cada *Sistema Autónomo* posee para conectarse a la nube, buscando minimizar la suma de la latencia global de internet hacia la nube analizada. Se definió una latencia base de 10 milisegundos desde el *Sistema Autónomo* inicial, propuesto para tomar en cuenta la latencia base de intercambio en el *Sistema Autónomo* mismo, evadiendo presentar una latencia 0 para el tráfico generado en la misma red de este *Sistema Autónomo*, lo cual no es posible si este posee una cantidad de tráfico media o alta.
- **Minimización de Saltos:** Este método busca minimizar la suma de los saltos requeridos por los *Sistemas Autónomos* para comunicarse con los nodos de la nube *anycast*.

Para cada uno de estos parámetros, realizaremos la optimización a través de los siguientes métodos:

- **Optimización Bruta:** Se buscará minimizar la suma de la latencia o saltos de cada *Sistema Autónomo* hasta la nube *anycast*, donde cada *Sistema Autónomo* contará como un cliente independiente.
- **Optimización Ponderada:** Se busca minimizar la suma de la latencia o saltos de cada *Sistema Autónomo*, ponderando cada uno de estos por la cantidad de consultas por segundo que estos generan.
- **Optimización Ponderada Limitada:** Se busca minimizar la suma de la latencia o saltos de cada *Sistema Autónomo*, ponderando cada uno de estos por la cantidad de consultas por segundo que estos generan, limitando la cantidad de saltos máxima por la que este nodo se puede propagar, buscando simular un despliegue que se comporte de manera más similar a las redes actuales o concentrar el tráfico que este nuevo nodo captura.

Para lograr medir los efectos de realizar la instalación de un nuevo servidor conectado a un *Sistema Autónomo* específico, utilizaremos en primer lugar la información de las áreas de servicio capturadas en el capítulo 2 para generar una red inicial, la cual corresponde a la visión que posee la nube *anycast* de internet. Con esta información se generará un grafo con los vínculos iniciales entre los diferentes *Sistemas Autónomos* que fueron encontrados.

Luego de esto, se agregarán nuevas conexiones entre los *Sistemas Autónomos* utilizando una base de datos de trazados de ruta, para lo cual, utilizaremos las capturas diarias generadas por *RIPE Atlas* hacia toda la internet [2]. A través de esta información, podemos encontrar vínculos entre los *Sistemas Autónomos* que no podemos observar directamente dado que los puntos de observación de una nube *anycast* son muy limitados.

Para generar una buena predicción, es necesario filtrar los datos obtenidos desde *RIPE Atlas*, solo utilizando los *Sistemas Autónomos* que fueron capturados utilizando la metodología descrita en el capítulo 2. Esto es importante, dado que si no poseemos una visión completa de la nube *anycast* (por ejemplo, no realizamos la captura de las áreas de servicio

de todos los nodos de la nube *anycast*), existirán *Sistemas Autónomos* sobre los cuales no conoceremos la latencia hacia la nube que estos poseen, por lo que no podremos optimizar estos.

Para calcular efecto del posicionamiento de un nuevo servidor conectado a un *Sistema Autónomo*, utilizaremos el algoritmo de *Dijkstra* para calcular la suma de las latencias o saltos según el método que se está analizando, y se comparará el efecto de cada una de las posiciones tal que el resultado sea el mínimo.

Los pesos a utilizar para realizar la ponderación, se obtendrán desde capturas de datos realizadas en los diferentes nodos, calculando las consultas por segundo que cada *Sistema Autónomo* realiza.

Los resultados presentados mostrarán la mediana de la latencia inicial y final de los nodos afectados, el número de *Sistemas Autónomos* y el número de consultas por minuto que fueron redirigidos al nuevo servidor.

5.2.2. Verificación

Para verificar los resultados, se utilizará la nube *anycast PEERING - The BGP Testbed*, buscando predecir los cambios producidos al agregar un nuevo servidor en esta nube *anycast*.

Para esto, se realizó el anuncio desde dos nodos de esta nube, los cuales se encuentran en *seattle* y *ufmg*. Sobre esta nube, luego de esperar a que se propaguen las rutas por 90 minutos, se realizó la medición de las áreas de servicio de esta nube.

Luego de esto, se repitieron las mediciones ingresando un nuevo nodo en *amsterdam*, calculando nuevamente las áreas de servicio resultante.

Finalmente, se deben aplicar los algoritmos descritos anteriormente y comparar el efecto predicho al agregar un nuevo nodo con las áreas de servicio real calculadas.

5.3. Resultados

Los resultados a presentar fueron obtenidos utilizando la metodología descrita anteriormente, aplicada a la nube *anycast a.nic.cl* de *NIC Chile*.

En estos resultados, solo se consideraron los *Sistemas Autónomos* que se encuentran en el área de servicio de los nodos *arica*, *merced*, *saopaulo* y *tucapel*, descartando la información de los *Sistemas Autónomos* sobre los que no se posee información.

Los resultados presentados deben considerar que la visión de internet utilizada, solo considera el área de servicio de estos cuatro nodos, los cuales se concentran en el continente de América y una parte menor de Europa y Asia, tal como fue descrito en el capítulo 2.

5.3.1. Minimizar Latencia

5.3.1.1. Minimizar Latencia Bruta

Los resultados expuestos en la figura 5.1 nos muestra los diez lugares que minimizan la latencia percibida por los *Sistemas Autónomos*, contando cada uno como un cliente unitario.

Podemos observar que este método demuestra dos resultados de optimización. En primer lugar, este intenta acercar los nodos extremos con una alta latencia, produciendo así una cantidad muy baja de *Sistemas Autónomos* afectados. Por otro lado, cuando la diferencia de latencia es baja, la cantidad de *Sistemas Autónomos* se incrementa.

En ambos casos, el tráfico afectado por estos cambios es muy bajo, donde la variación es aproximadamente 10 consultas por segundo. Esto se produce dado que el algoritmo utilizado no considera este parámetro para realizar la optimización.

A través de la tabla 5.1, podemos observar la localización de los *Sistemas Autónomos* donde esperamos realizar la instalación de nueva infraestructura. Es posible observar, que los resultados sugeridos muchas veces se encuentran en lugares geográficamente lejanos a los nodos analizados, verificando que estos lograrían disminuir la latencia de los *Sistemas Autónomos* presentes en estos lugares.

Sistema Autónomo	Localización	Sistema Autónomo	Localización
25233	Saudi Arabia	28126	Brasil
37100	Mauricio (Africa-Asia)	37662	Mauricio
45437	Australia	136393	Tailandia
21983	Estados Unidos	24086	Vientián
43139	Ucrania	11014	Argentina

Tabla 5.1: Tabla con la localización de los *Sistemas Autónomos* mostrados en la figura 5.1

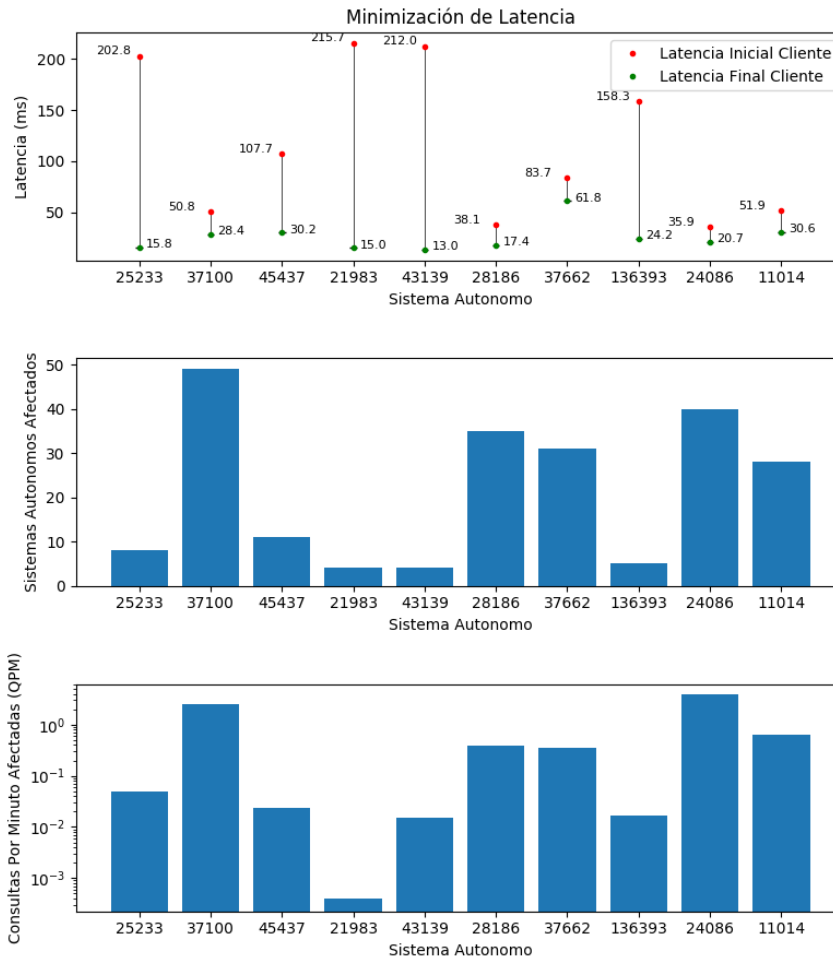


Figura 5.1: Gráfico que muestra la variación de latencia, número de *Sistemas Autónomos* afectados y consultas por segundo afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la ponderación entre el número de *Sistemas Autónomos* y la variación de latencia.

5.3.1.2. Minimizar Latencia Ponderada

Los resultados expuestos en la figura 5.2 nos muestra los diez lugares que minimizan la latencia ponderada por el tráfico generado desde los *Sistemas Autónomos* afectados al realizar la instalación del nuevo nodo en este *Sistema Autónomo*.

Podemos observar que la cantidad de *Sistemas Autónomos* afectados es mucho menor, mostrando en la mayoría de los casos una cantidad menor a 5 *Sistemas Autónomos*. Por otro lado, podemos ver que la cantidad de consultas afectadas es mayor, acercando la instalación a los nodos que generan tráfico hacia la nube.

Podemos destacar al primer nodo, el cual genera cien veces el tráfico comparado con los otros *Sistemas Autónomos*, y aunque la diferencia de latencia no es alta, esta instalación permitiría reducir la carga de los otros nodos en la nube *anycast*.

Dado que ahora nos encontramos minimizando la latencia del tráfico recibido, podemos observar en la tabla 5.2 que la mitad de los *Sistemas Autónomos* recomendados se encuentran en Estados Unidos, el cual vimos en el capítulo 4 que posee un porcentaje alto de tráfico y latencia.

De la misma manera, podemos observar que existen lugares mucho más lejanos recomendados, los cuales presentan una diferencia de latencia lo suficientemente grande como para afectar los resultados, aun cuando el tráfico generado por estos sea bajo.

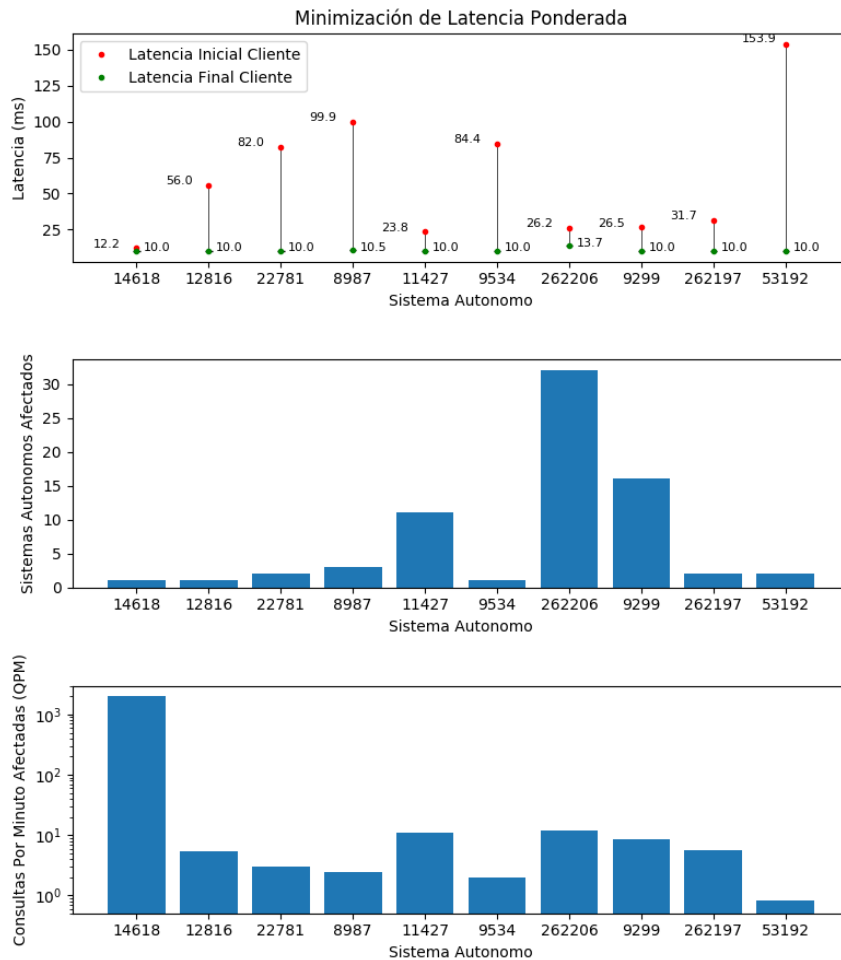


Figura 5.2: Gráfico que muestra la variación de latencia, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autonomo* indicado, optimizando por la ponderación entre el tráfico afectado y la variación de latencia.

Sistema Autónomo	Localización	Sistema Autónomo	Localización
14618	Estados Unidos	9534	Malasia
12816	Alemania	262206	Guatemala
22781	Estados Unidos (Amazon)	9299	Filipinas
8987	Estados Unidos	262197	Estados Unidos
11427	Estados Unidos	53192	Costa Rica

Tabla 5.2: Tabla con la localización de los *Sistemas Autónomos* mostrados en la figura 5.2

5.3.1.3. Minimizar Latencia Ponderada Limitando Saltos

Los resultados expuestos en la figura 5.3 nos muestra los diez lugares que minimizan la latencia ponderada por el tráfico generado desde los *Sistemas Autónomos* afectados al realizar la instalación del nuevo nodo en este *Sistema Autónomo*, limitando la distancia máxima de las rutas generadas por este nuevo nodo en dos saltos. Las figuras que contienen los límites aplicados desde uno a cuatro saltos se pueden observar en el *Apéndice 1: Resultados Minimizar Latencia Ponderada Limitando Saltos*.

Comparando estos resultados con los expuestos en la figura 5.2, podemos observar que los *Sistemas Autónomos* propuestos para realizar la instalación son los mismos, donde la única diferencia encontrada corresponde a la cantidad de *Sistemas Autónomos* que cambian sus rutas a este.

Además, podemos ver que la cantidad de consultas por minuto afectadas prácticamente no posee variación, lo que nos permite verificar que las sugerencias de posicionamiento se realizan de manera muy cercana a la fuente del tráfico, más que instalarse en un punto intermedio.

Sistema Autónomo	Localización	Sistema Autónomo	Localización
14618	Estados Unidos (Amazon)	9534	Malasia
12816	Alemania	262206	Guatemala
22781	Estados Unidos	9299	Filipinas
8986	Estados Unidos (Amazon)	262197	Estados Unidos
11427	Estados Unidos	53192	Costa Rica

Tabla 5.3: Tabla con la localización de los *Sistemas Autónomos* mostrados en la figura 5.3

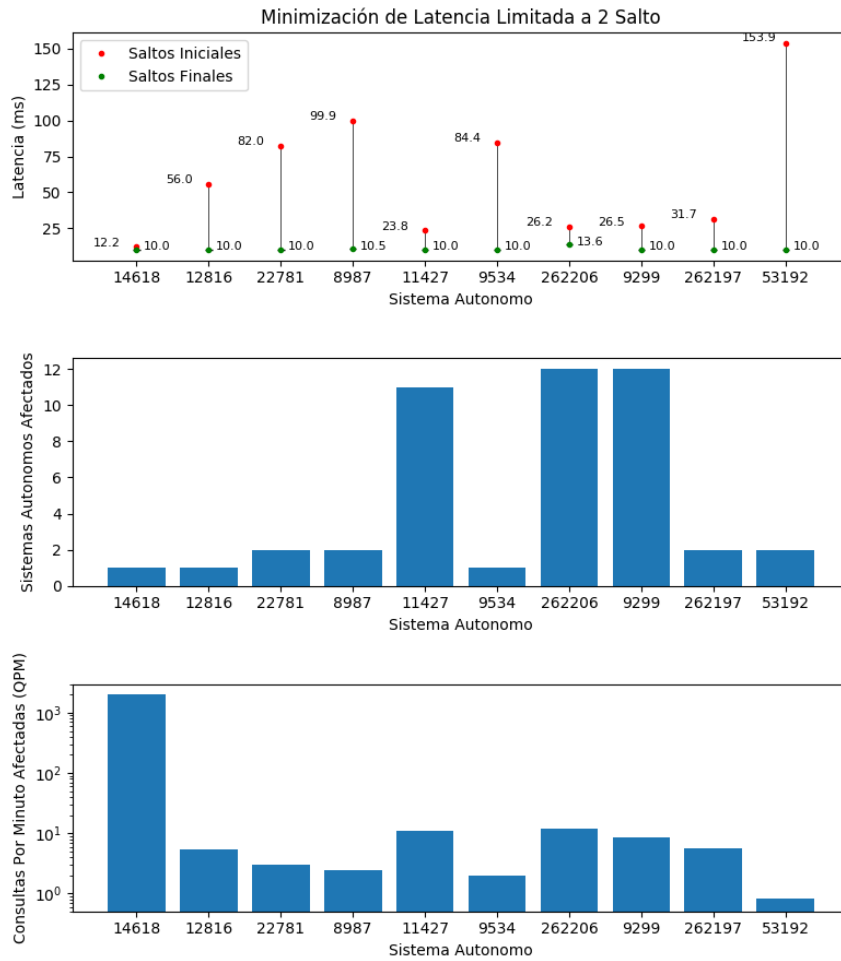


Figura 5.3: Gráfico que muestra la variación de latencia, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la ponderación entre el tráfico afectado y la variación de latencia, limitando el máximo de propagación del área de servicio de este nuevo nodo en dos saltos.

5.3.2. Minimizar Saltos

5.3.2.1. Minimizar Saltos Brutos

Los resultados expuestos en la figura 5.4 nos muestra los diez lugares que minimizan la cantidad de saltos requeridos por los *Sistemas Autónomos* para acceder a la nube *anycast*, al realizar la instalación de un nuevo nodo *anycast* en el *Sistema Autónomo* indicado.

Se puede observar que el posicionamiento propuesto, realiza una disminución en la cantidad de saltos medio de un punto, intentando afectar la mayor cantidad de *Sistemas Autónomos* posibles.

Esto genera que la cantidad de tráfico afectado esté en el rango de las 100 consultas por minuto, uno mucho mayor que el inicialmente analizado con la latencia bruta en la sección 5.3.1.1.

En la tabla 5.4, podemos observar las localizaciones de estos *Sistemas Autónomos* recomendados, los cuales se encuentran mayoritariamente en los Estados Unidos, cubriendo las redes de grandes empresas. Además de estos, podemos destacar la aparición de los *Sistemas Autónomos* 52320 (Colombia) y 9498 (India), donde el primero se encuentra posicionado prácticamente entre los nodos de *saopaulo* y *monterrey*, y el segundo en el continente asiático que fue observado en el área de servicio de *merced*, en el capítulo 3.

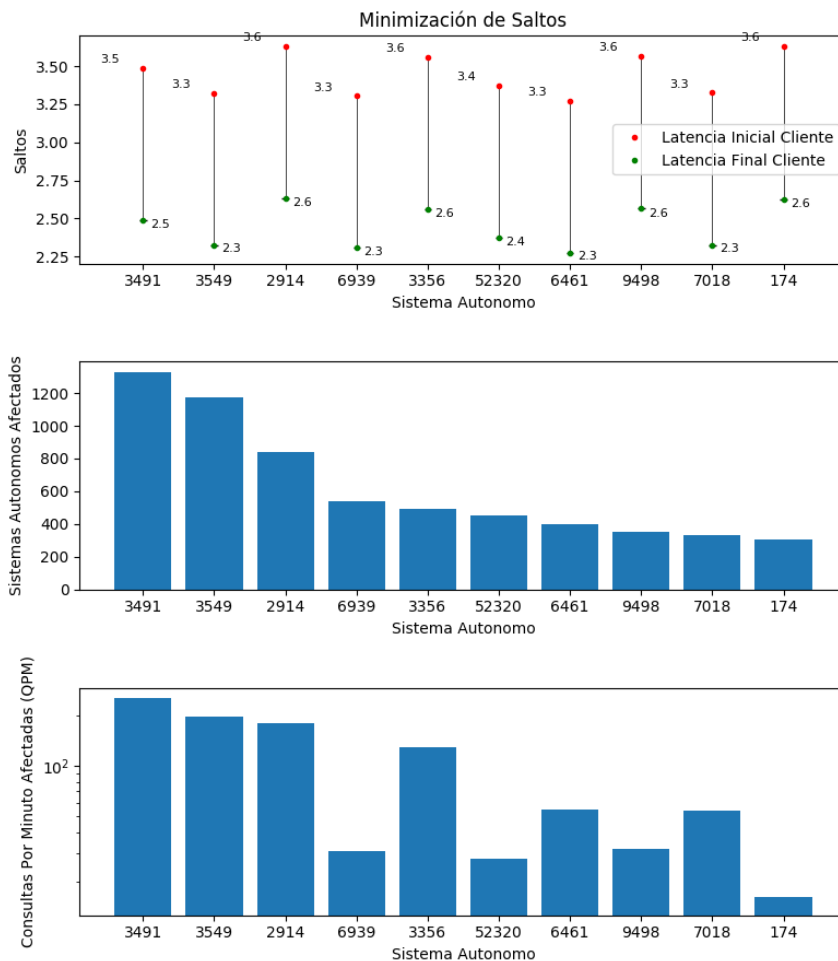


Figura 5.4: Gráfico que muestra la variación de latencia, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la ponderación entre el número de *Sistemas Autónomos* y la variación de latencia.

Sistema Autónomo	Localización	Sistema Autónomo	Localización
3491	Estados Unidos	52320	Colombia
3549	Estados Unidos (Level3)	6461	Estados Unidos
2914	Estados Unidos (NTT)	9498	India
6939	Estados Unidos (Hurricane Electric)	7018	Estados Unidos
3356	Estados Unidos (Level3)	174	Estados Unidos

Tabla 5.4: Tabla con la localización de los *Sistemas Autónomos* mostrados en la figura 5.4

5.3.2.2. Minimizar Saltos Ponderados

Los resultados expuestos en la figura 5.5 nos muestra los diez lugares que minimizan la cantidad de saltos requeridos por los *Sistemas Autónomos* para acceder a la nube *anycast*, ponderando la diferencia de saltos y el tráfico generado por cada uno de los *Sistemas Autónomos* afectados al posicionar el nuevo nodo *anycast* en el *Sistema Autónomo* indicado.

Podemos observar que la variación de los saltos es mucho mayor, disminuyendo los saltos requeridos para acceder a la nube *anycast* por los *Sistemas Autónomos* de una mayor manera, afectando a los que se encuentran a una distancia media de tres saltos.

Por otro lado, la cantidad de *Sistemas Autónomos* afectados es mucho menor en comparación a la optimización bruta, focalizándose en disminuir los saltos desde las fuentes de tráfico más altas.

En la tabla 5.5, podemos observar que la tendencia de seleccionar nodos que se encuentran en Estados Unidos se mantiene, seleccionando a los *Sistemas Autónomos* de grandes compañías que generan un tráfico considerable.

En estos resultados podemos destacar la recomendación de posicionarnos en el *Sistema Autónomo 6471* perteneciente a *Entel Chile*, aun cuando existen dos nodos ya desplegados en este país.

Sistema Autónomo	Localización	Sistema Autónomo	Localización
8075	Estados Unidos (Microsoft)	264959	Brasil
16509	Estados Unidos (Amazon)	15169	Estados Unidos (Google)
14618	Estados Unidos (Amazon)	14061	Estados Unidos (DO ¹)
20940	Estados Unidos (Akami)	16417	Estados Unidos (Cisco)
577	Canada	6471	Chile (Entel)

Tabla 5.5: Tabla con la localización de los *Sistemas Autónomos* mostrados en la figura 5.5

¹Digital Ocean

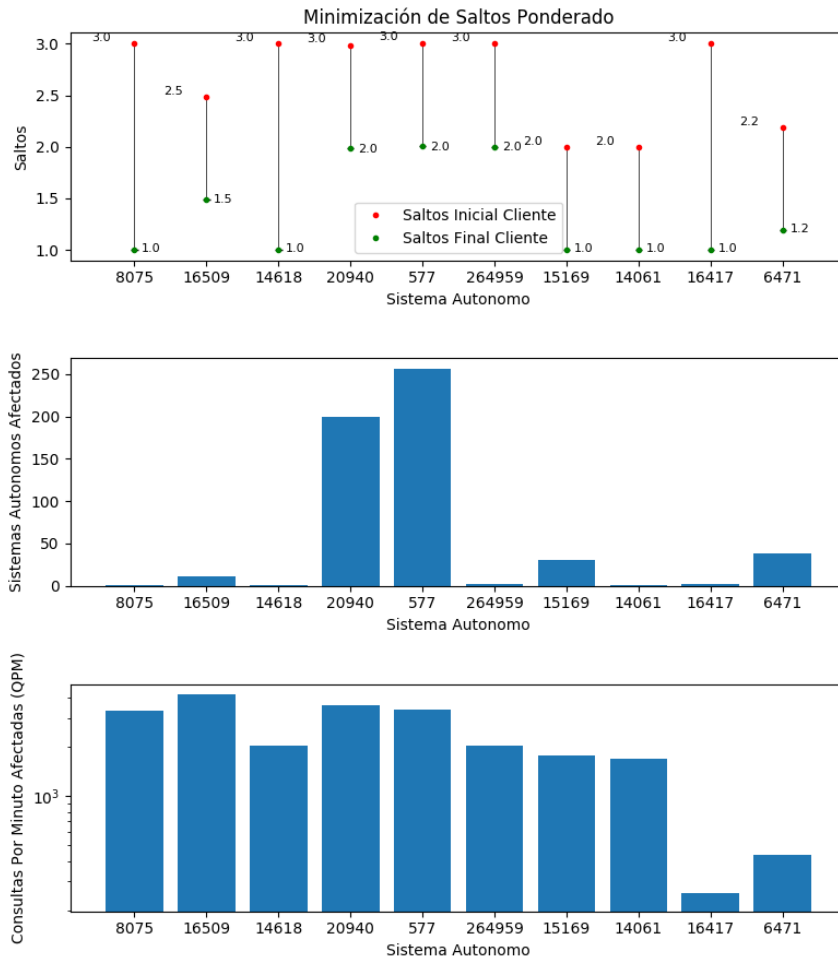


Figura 5.5: Gráfico que muestra la variación de saltos media, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la cantidad de saltos requeridos por los *Sistemas Autónomos* para acceder a la nube *anycast* analizada y el tráfico generado por estos.

5.3.2.3. Minimizar Saltos Ponderados Limitando Saltos

Los resultados expuestos en la figura 5.6 nos muestra los diez lugares que minimizan la cantidad de saltos requeridos por los *Sistema Autónomo* para acceder a la nube *anycast*, ponderando la diferencia de saltos y el tráfico generado por cada uno de los *Sistemas Autónomos* afectados al posicionar el nuevo nodo *anycast* en el *Sistema Autónomo* indicado, limitando la distancia máxima de las rutas generadas por este nuevo nodo en dos saltos. Las figuras que contienen los límites aplicados desde uno a cuatro saltos se pueden observar en el *Apéndice 2: Minimizar Saltos Ponderados Limitando Saltos*.

Podemos observar que la variación de estos resultados y los expuestos en el punto 5.3.2.2 es mínima, donde la única diferencia encontrada corresponde a la cantidad de *Sistemas Autónomos* afectados.

Además de esto, el tráfico capturado por estos nodos se mantiene prácticamente constante con o sin limitaciones de saltos, permitiéndonos verificar que las sugerencias presentadas se encuentran muy cercanas a las fuentes de tráfico.

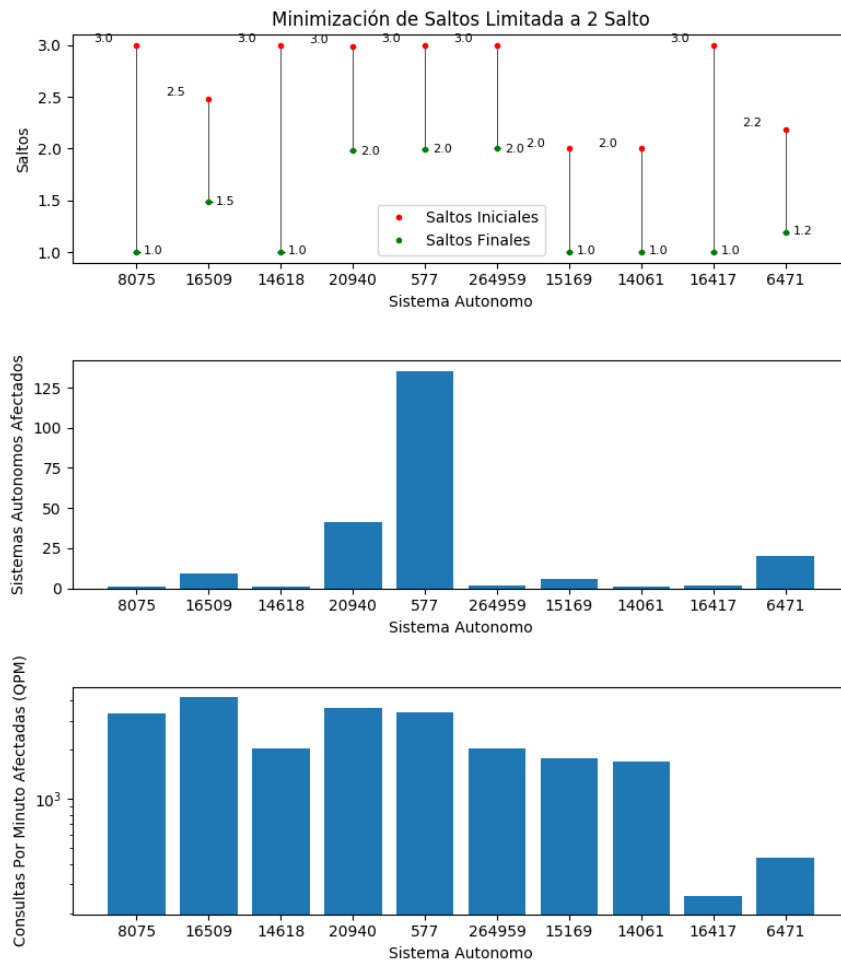


Figura 5.6: Gráfico que muestra la variación de saltos media, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la cantidad de saltos requeridos por los *Sistemas Autónomos* para acceder a la nube *anycast* analizada y el tráfico generado por estos, limitando la propagación de esta ruta en dos saltos.

Sistema Autónomo	Localización
8075	Estados Unidos (Microsoft)
16509	Estados Unidos (Amazon)
14618	Estados Unidos (Amazon)
20940	Estados Unidos (Akami)
577	Canada

Sistema Autónomo	Localización
264959	Brasil
15169	Estados Unidos (Google)
14061	Estados Unidos (DO ²)
16417	Estados Unidos (Cisco)
6471	Chile (Entel)

Tabla 5.6: Tabla con la localización de los *Sistemas Autónomos* mostrados en la figura 5.6

5.3.3. Verificación

Para realizar la verificación de los resultados entregados, realizamos la medición de las áreas de servicio de la nube *PEERING* utilizando los nodos *amsterdam*, *seattle* y *ufmg*, ubicados en Ámsterdam (Países Bajos), Seattle (Estados Unidos) y Belo Horizonte (Brasil).

Las áreas de servicio se midieron en dos configuraciones diferentes, las cuales se muestran a continuación.

- Anuncio de la red $184.164.228.0/24$, desde los nodos *seattle* y *ufmg*.
- Anuncio de la red $184.164.228.0/24$, desde los nodos *amsterdam*, *seattle* y *ufmg*.

El número de prefijos de red $/24$ y *Sistemas Autónomos* resultante de estas configuraciones se pueden apreciar en las tablas 5.7 y 5.8.

Nodo	Número de Prefijos de red $/24$	Número de Sistemas Autónomos
seattle	121,989	5574
ufmg	1,524,652	34320

Tabla 5.7: Tabla que muestra la cantidad de prefijos $/24$ y *Sistemas Autónomos* únicos que componen el área de servicio de la nube *PEERING*, utilizando la configuración *anycast* con los nodos *seattle* y *ufmg*.

Nodo	Número de Prefijos de red $/24$	Número de Sistemas Autónomos
amsterdam	1,394,662	27160
seattle	48,164	3278
ufmg	36,481	1960

Tabla 5.8: Tabla que muestra la cantidad de prefijos $/24$ y *Sistemas Autónomos* únicos que componen el área de servicio de la nube *PEERING*, utilizando la configuración *anycast* con los nodos *amsterdam*, *seattle* y *ufmg*.

En estas tablas, logramos apreciar que el cambio en la asignación que ocurre es muy alto, redirigiendo casi todo el tráfico hacia el nodo *amsterdam* cuando este nodo se une a la nube *anycast*.

Esto contrastó en gran manera con los cambios predichos al ejecutar las predicciones, las cuales mostraron un cambio en las rutas de solo 7677 *Sistemas Autónomos*.

²Digital Ocean

Esta diferencia en la asignación se puede explicar por la alta conectividad del nodo *amsterdam* en comparación a los otros, dado que este se encuentra conectado a grandes redes que enrutan una gran cantidad de tráfico a diferentes partes del mundo.

El nivel de conectividad del nodo *amsterdam* se puede visualizar en la figura 5.7 y los nodos *seattle* y *ufmg* en el Apéndice 3: Ruteo Entre Sistemas Autónomo. En estas figuras, se muestran las conexiones que poseen los diferentes *Sistemas Autónomos* que son utilizados como punto de conexión a internet, mostrando una densidad mucho mayor de conexiones en el punto de entrada de *amsterdam* en comparación a los demás.

Dada la gran diferencia entre los efectos de la predicción realizada y los resultados observados, **no** logramos realizar la verificación del método propuesto de manera exitosa.

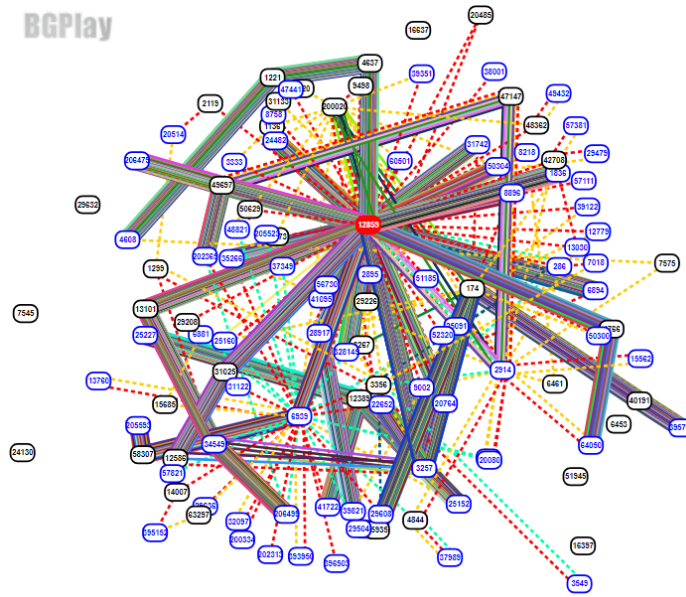


Figura 5.7: Rutas disponibles desde el *Sistema Autónomo 10417*, punto de entrada de la nube *anycast* de prueba *PEERING*, nodo *ufmg*. Visualización generada a través de BGPlay.

5.4. Conclusión

En este capítulo, propusimos y ejecutamos una metodología que nos permite determinar diferentes posiciones para realizar la instalación de nueva infraestructura en una nube *anycast*, buscando reducir la latencia y número de saltos necesarios para conectarse desde los diferentes *Sistemas Autónomos* que componen internet hasta una nube *anycast*.

Para observar los resultados generados por esta metodología, aplicamos esta a la nube *anycast a.nic.cl* de *NIC Chile*, específicamente a los nodos *arica*, *merced*, *saopaulo* y *tucapel*.

Dado que no se utilizaron los datos de la nube *anycast* completa, todos los datos y predicciones debieron ser filtrados según las áreas de servicio y rutas utilizadas extraídas desde

el capítulo 2. Esto significó que los resultados solo poseían una parte de la internet abierta, correspondiente a la vista por estos nodos que se encuentran concentrados en América Latina.

Cuando buscamos minimizar la latencia global de los *Sistemas Autónomos* hacia la nube *anycast*, y no consideramos el tráfico recibido por la nube, esta forma de optimización tiende a posicionar el nuevo nodo en los extremos de la red, lugares que poseen la latencia más alta, o lugares que poseen una alta concentración de sistemas autónomos, afectando a un bajo nivel de tráfico.

Por otro lado, si optimizamos por latencia global considerando el tráfico, logramos observar que el posicionamiento del nuevo nodo *anycast* se realiza cerca de las fuentes de tráfico que poseen una alta latencia, sin afectar a una gran cantidad de *Sistemas Autónomos*, aun cuando limitemos la propagación de las nuevas rutas anunciadas.

Al optimizar los saltos promedios sin considerar el tráfico, logramos observar que este método busca lugares que poseen la mayor conectividad posible, lo cual se refleja en la cantidad de *Sistemas Autónomos* afectados, en conjunto con el tráfico de estos lugares.

Si utilizamos al tráfico como un ponderador, este método nos entregó posiciones generalmente cercanas a las grandes empresas, las cuales poseen una gran cantidad de clientes que generan tráfico hacia la nube *anycast* analizada, sin ser afectado por limitaciones de la propagación de las nuevas rutas generadas.

La gran variedad de resultados presentados pueden ser utilizados de distintas maneras por los operadores según las diferentes métricas que deseen optimizar. Por ejemplo, si desearan minimizar la latencia de los clientes según el tráfico, posicionaríamos un nuevo nodo muy cercano a la fuente de mayor tráfico que posea una latencia considerable. Si quisiéramos distribuir la carga de la nube en un nuevo nodo, posicionaremos este en un lugar que posea una alta conectividad con la internet abierta, vigilando que este no genere una sobrecaptura en otros clientes.

Si bien no se logró realizar la verificación de este método de manera exitosa, la metodología de verificación utilizada nos mostró que posicionar un nuevo nodo en un lugar altamente conectado, no siempre provee una reducción de latencia a los clientes, y puede redirigir una cantidad más alta de tráfico a un lugar más lejano. Además de esto, realizar cambios en los anuncios de las rutas utilizadas es un paso muy importante, ya sea en un experimento o en producción, para obtener los mejores resultados.

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

El trabajo realizado en esta tesis buscó definir y ejecutar diferentes metodologías que permitan realizar el análisis geográfico y topológico de una nube *anycast DNS*, determinando las áreas de servicio de cada nodo perteneciente a esta nube, y las posiciones óptimas según diferentes parámetros para posicionar un nuevo servidor.

Para esto, separamos el trabajo en cuatro diferentes etapas *Captura de Datos*, *Análisis Geográfico*, *Análisis Topológico* y *Posicionamiento de Nueva Infraestructura*, cada etapa con su propia metodología que nos permita obtener la información necesaria.

La principal fuente de datos de este trabajo corresponde a la nube *anycast a.nic.cl* de *NIC Chile*, una de tres nubes *anycast* que *NIC Chile* administra directamente. Esta nube se encarga de responder las consultas *DNS* de todos los dominios *.cl*, asegurándonos que los datos obtenidos corresponden a una nube altamente utilizada. Los análisis se realizaron sobre los nodos *arica*, *merced*, *saopaulo* y *tucapel* de la nube *anycast*, representando a los nodos que se encuentran en América del Sur.

En la etapa de *Captura de Datos*, se definió y ejecutó una metodología que nos permitió encontrar las áreas de servicio de cada uno de los nodos analizados, en términos de los prefijos de red /24 sobre los cuales cada uno de los nodos recibe información, además de las rutas utilizadas para comunicarse con estas.

Esta metodología definida corresponde a una extensión de la metodología de medición *Verfploeter* [8], que ahora nos permite medir las rutas utilizadas para conectarse a las diferentes redes, en conjunto con una ejecución unitaria que nos permita realizar las mediciones en ambientes seguros de manera más sencilla.

Las áreas de servicio capturadas se verificaron a través de la plataforma de medición *RIPE Atlas*, donde logramos identificar al nodo *anycast* asignado satisfactoriamente en 106 puntos de medición diferentes, con tan solo un error atribuible al desfase entre el tiempo de medición

del área de servicio y la medición desde la plataforma.

Para el *Análisis Geográfico*, se definió y ejecutó una metodología que nos permitió geolocalizar las diferentes redes que componen las áreas de servicio de los nodos analizados, mostrando redes que pueden no encontrarse en capturas de datos realizadas directamente desde los nodos. Esto nos permitió observar anomalías, tal como la gran comunicación del nodo *merced* ubicado en Chile con países tal como Italia y Turquía, aun cuando existen servidores desplegados en Europa que se encuentran más cercanos geográficamente.

La captura de las rutas utilizadas para realizar estas conexiones, nos permitieron determinar los *Sistemas Autónomos* involucrados en estas asignaciones. Esta información puede ser utilizada por los operadores para establecer diferentes medidas de mitigación tales como prevenir el enrutamiento a través de este sistema, o realizar variaciones en los largos de los anuncios a través del protocolo *BGP*.

Utilizando esta información, se realizó el cálculo de las distancias geográficas desde todos los nodos en la nube *anycast* hacia las redes en el área de servicio de los nodos analizados en la etapa *Captura de Datos*, donde se encontró que una gran parte de las áreas de servicio no se realiza hacia el nodo más cercano geográficamente, aun cuando se utilizan capturas de datos para ponderar el tráfico de cada red, asignando una mayor importancia a las redes que generan una mayor cantidad de tráfico a la nube *anycast*.

En el *Análisis Topológico*, se definió y ejecutó una metodología que nos permitió determinar la distribución de saltos y latencias de cada nodo de la nube *anycast* analizada, en conjunto con un método que nos permite observar si la asignación entre las redes y los nodos corresponde a la óptima en términos de latencia.

Los resultados obtenidos nos permitieron observar la distribución de latencia que posee cada uno de los nodos, agrupando los *Sistemas Autónomos* pertenecientes a cada área de servicio en diferentes rangos de latencia, comparando estos con el efecto de aplicar el tráfico recibido desde las diferentes redes como un ponderador.

El análisis realizado sobre la asignación de las diferentes redes, según su latencia hacia los diferentes nodos que pertenecen a la nube *anycast DNS* analizada, nos mostró que la asignación topológica no corresponde a la óptima en términos de latencia en la mayoría de los casos, encontrando que, al utilizar 687 puntos de medición distribuidos, solo el 35 % de estos es asignado al nodo con el cual existe una latencia mínima.

Cabe destacar que esta asignación de redes no es solo importante para servidores *anycast DNS* que poseen una baja tasa de transferencia, sino también para servicios *anycast CDN*, donde un aumento en la latencia puede llevar a una disminución de la velocidad de transferencia en más de un 40 % [3].

Finalmente, en la etapa de *Posicionamiento de Nueva Infraestructura*, definimos una metodología que nos permite estimar diferentes posiciones en la internet abierta para realizar la instalación de nueva infraestructura, expandiendo un despliegue *anycast* actualmente desplegado, o generando sugerencias sobre el posicionamiento del primer nodo.

Los resultados obtenidos nos permitieron observar los efectos producidos al optimizar los saltos o latencia de la nube analizada, además de los cambios producidos al considerar el tráfico generado por los diferentes *Sistemas Autónomos* como un ponderador en la optimización.

La verificación de este método se realizó en la nube *anycast* de prueba *PEERING*, donde se encontró que realizar predicciones certeras sobre los cambios en la red al ingresar un nuevo nodo *anycast* es más difícil de lo esperado, encontrando grandes diferencias entre los valores teóricos y los observados. Esto generó que no se pudiera validar el método propuesto, sin embargo, las predicciones resultantes pueden utilizarse como una base para seleccionar una posición en la red para instalar un nuevo nodo *anycast*.

Considerando lo anterior, se lograron cumplir los objetivos planteados por esta tesis, donde el único punto que no se logró verificar corresponde al posicionamiento óptimo de un nuevo nodo *anycast* según los diferentes parámetros definidos.

En las etapas de *Análisis Geográfico* y *Análisis Topológico*, logramos observar que las selecciones de servidor realizadas por los diferentes *Sistemas Autónomos* no privilegian las distancias geográficas ni topológicas en términos de latencia, con lo cual logramos concluir que la tesis expuesta “*En una nube anycast DNS, el servidor más cercano al cliente en términos de su distancia geográfica y latencia es el que siempre responde las consultas DNS*” es **falsa**.

6.2. Trabajo Futuro

La metodología de *Captura de Datos* expuesta puede aplicarse para realizar optimizaciones sobre los anuncios *BGP* actualmente realizados por cada uno de los nodos *anycast*. Al realizar un cambio en el largo del anuncio, es posible realizar la captura del área de servicio antes y después del cambio, y así obtener directamente los efectos producidos en el enrutamiento de la nube. Al aplicar esto a un prefijo no utilizado, es posible realizar un ajuste previo a los anuncios desde los diferentes nodos *anycast*, hasta llegar a una configuración óptima. De esta manera, es posible ajustar una red utilizada con el menor impacto a los usuarios, obteniendo resultados optimizados de manera previa. Esto se puede complementar con el conocimiento sobre las rutas utilizadas, permitiendo identificar *Sistemas Autónomos* que se encuentran anunciando las rutas en caminos no deseados.

Como trabajo futuro, se debería probar la metodología de estimación considerando las tablas de rutas de los nodos *anycast* analizados, en las cuales es posible observar los largos de ruta de cada *Sistema Autónomo*, con lo cual sería posible aumentar la precisión de las predicciones. En caso de no ser esto posible, se recomienda ajustar el largo de los anuncios antes de realizar las mediciones en una nueva nube, dado que esto puede afectar en gran manera las áreas de servicio resultante.

Por otro lado, un nuevo parámetro a considerar en los análisis de rutas y estimaciones de posicionamiento corresponde a la pérdida de paquetes en las conexiones entre *Sistemas Autónomos*, los cuales afectan en gran manera a los tiempos de respuesta observados por los

usuarios.

Las mediciones de área de servicio realizadas pueden ser utilizadas en procesos de enumeración de nubes *anycast*, donde un *Sistema Autónomo* encontrado en el área de servicio de múltiples nodos *anycast* pueden indicar la presencia de una nube *anycast* en esta red, permitiendo acelerar los métodos actualmente utilizados para realizar esta detección [5].

Por último, las áreas de servicio calculadas pueden utilizarse como un método de mitigación de ataques de reflexión *DNS*, dado que podemos verificar si la fuente del tráfico puede comunicarse o no con el nodo que recibe la respuesta.

Capítulo 7

Bibliografía

- [1] R. Liu L. Zhang A. Fei, G. Pei. Measurements on delay and hop-count of the internet. In *Proceedings of Globecom '98*, Globecom '98, 1998.
- [2] N. Ahmed and K. Sarac. An experimental study on inter-domain routing dynamics using ip-level path traces. In *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, pages 510–517, Oct 2015.
- [3] Zakaria Al-Qudah, Seungjoon Lee, Michael Rabinovich, Oliver Spatscheck, and Jacobus Van der Merwe. Anycast-aware transport for content delivery networks. In *Proceedings of the 18th International Conference on World Wide Web, WWW '09*, pages 301–310, New York, NY, USA, 2009. ACM.
- [4] Vaibhav Bajpai, Steffie Jacob Eravuchira, and Jürgen Schönwälder. Lessons learned from using the ripe atlas platform for measurement research. *SIGCOMM Comput. Commun. Rev.*, 45(3):35–42, July 2015.
- [5] D. Cicalese, D. Joumblatt, D. Rossi, M. Buob, J. Augé, and T. Friedman. A fistful of pings: Accurate and lightweight anycast enumeration and geolocation. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2776–2784, April 2015.
- [6] Danilo Cicalese, Jordan Augé, Diana Joumblatt, Timur Friedman, and Dario Rossi. Characterizing ipv4 anycast adoption and deployment. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT '15*, pages 16:1–16:13, New York, NY, USA, 2015. ACM.
- [7] Ricardo de Oliveira Schmidt, John S. Heidemann, and Jan Harm Kuipers. Anycast latency: How many sites are enough? In *PAM*, 2017.
- [8] Wouter B. de Vries, Ricardo de O. Schmidt, Wes Hardaker, John Heidemann, Pieter-Tjerk de Boer, and Aiko Pras. Broad and load-aware anycast mapping with verfloeter. In *Proceedings of the 2017 Internet Measurement Conference, IMC '17*, pages 477–488, New York, NY, USA, 2017. ACM.

- [9] X. Fan, J. Heidemann, and R. Govindan. Evaluating anycast in the domain name system. In *2013 Proceedings IEEE INFOCOM*, pages 1681–1689, April 2013.
- [10] Xun Fan and John Heidemann. Selecting representative ip addresses for internet topology studies. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, IMC '10*, pages 411–423, New York, NY, USA, 2010. ACM.
- [11] M. Gharaibeh, A. Shah, B. Huffaker, H. Zhang, R. Ensafi, and C. Papadopoulos. A Look at Router Geolocation in Public and Commercial Databases. In *Internet Measurement Conference (IMC)*, Nov 2017.
- [12] Md Ahsan Habib and Marc Abrams. Analysis of sources of latency in downloading web pages. In *PROCEEDINGS OF WEBNET 2000*, 2000.
- [13] James Hiebert, Peter Boothe, Randy Bush, and Lucy Lynch. Determining the cause and frequency of routing instability with anycast. In Kenjiro Cho and Philippe Jacquet, editors, *Technologies for Advanced Heterogeneous Networks II*, pages 172–185, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [14] P. Mockapetris, ISI. *DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. RFC 1035*. Internet Engineering Task Force, Enero 1987.
- [15] NIC Chile Research Labs. niclabs/anytrace: An anycast path and area of service discovery tool. [En línea]. Disponible en: <https://github.com/niclabs/AnyTrace/> [Accedido: 04/01/2019].
- [16] Bu-Sung Lee, Yu Shyang Tan, Y. Sekiya, A. Narishige, and S. Date. Availability and effectiveness of root dns servers: A long term study. In *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, pages 862–865, April 2010.
- [17] Ziqian Liu, Bradley Huffaker, Marina Fomenkov, Nevil Brownlee, and Kc Claffy. Two days in the life of the dns anycast root servers. In *Proceedings of the 8th International Conference on Passive and Active Network Measurement, PAM'07*, pages 125–134, Berlin, Heidelberg, 2007. Springer-Verlag.
- [18] MaxMind. Maxmind geolite2. [En línea]. Disponible en: <http://dev.maxmind.com/geoip/geoip2/geolite2/> [Accedido: 04/01/2019].
- [19] S. Sarat, V. Pappas, and A. Terzis. On the use of anycast in dns. In *Proceedings of 15th International Conference on Computer Communications and Networks*, pages 71–78, Oct 2006.
- [20] L. Wei and J. Heidemann. Does anycast hang up on you? In *2017 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–9, June 2017.
- [21] S. Hares, Ed. Y. Rekhter, Ed., T. Li, Ed. *A Border Gateway Protocol 4 (BGP-4). RFC 4271*. Internet Engineering Task Force, Enero 2006.
- [22] B. Zhang, J. Bi, Y. Wang, Y. Zhang, and J. Wu. Refining ip-to-as mappings for as-level

traceroute. In *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7, July 2013.

Capítulo 8

Apéndice

8.1. Apéndice 1: Resultados Minimizar Latencia Ponderada Limitando Saltos

8.1.1. Minimizar Latencia Ponderada Limitando a 1 Salto

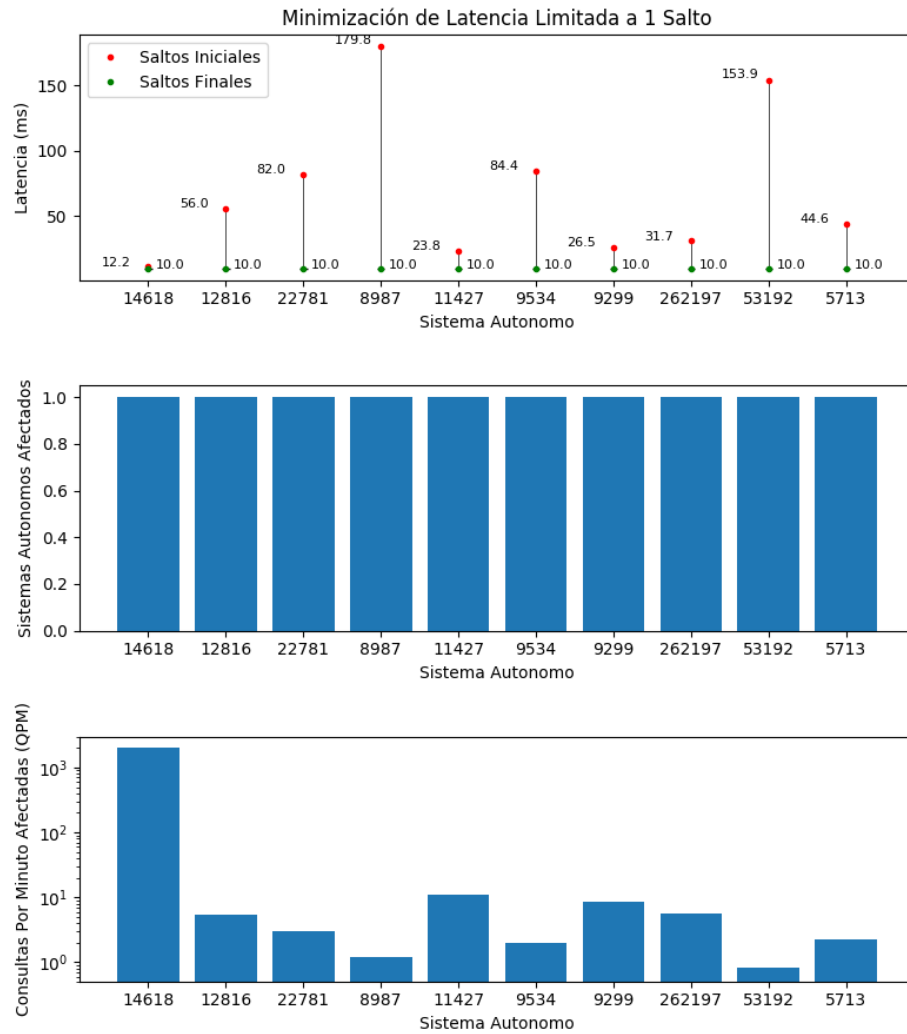


Figura 8.1: Gráfico que muestra la variación de latencia, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la ponderación entre el tráfico afectado y la variación de latencia, limitando el máximo de propagación del área de servicio de este nuevo nodo en un salto.

8.1.2. Minimizar Latencia Ponderada Limitando a 2 Saltos

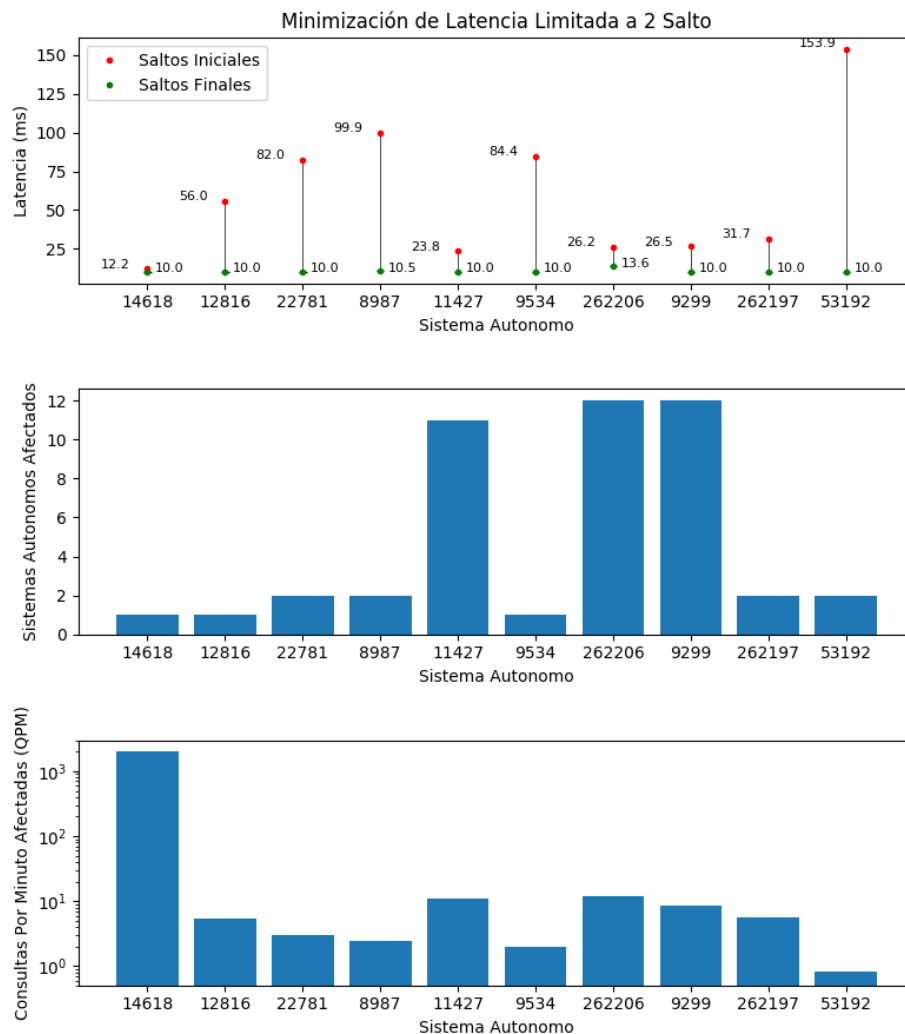


Figura 8.2: Gráfico que muestra la variación de latencia, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la ponderación entre el tráfico afectado y la variación de latencia, limitando el máximo de propagación del área de servicio de este nuevo nodo en dos saltos.

8.1.3. Minimizar Latencia Ponderada Limitando a 3 Saltos

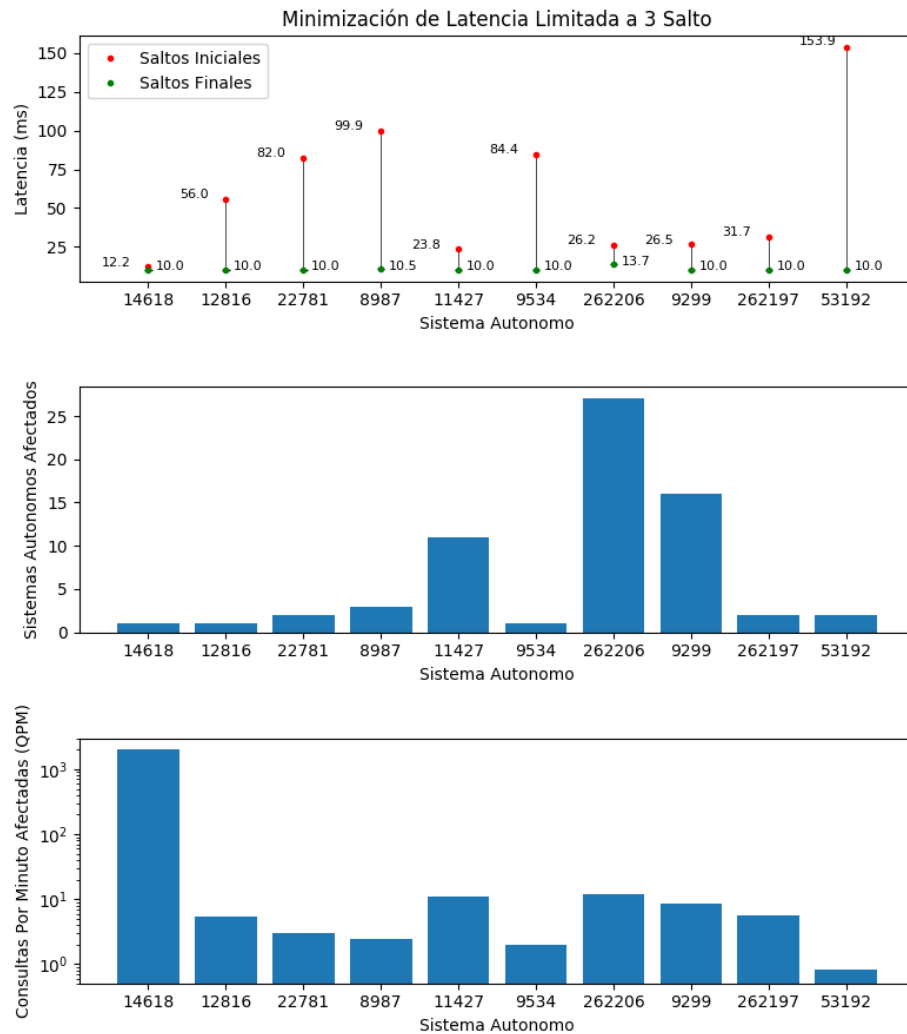


Figura 8.3: Gráfico que muestra la variación de latencia, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la ponderación entre el tráfico afectado y la variación de latencia, limitando el máximo de propagación del área de servicio de este nuevo nodo en tres saltos.

8.1.4. Minimizar Latencia Ponderada Limitando a 4 Saltos

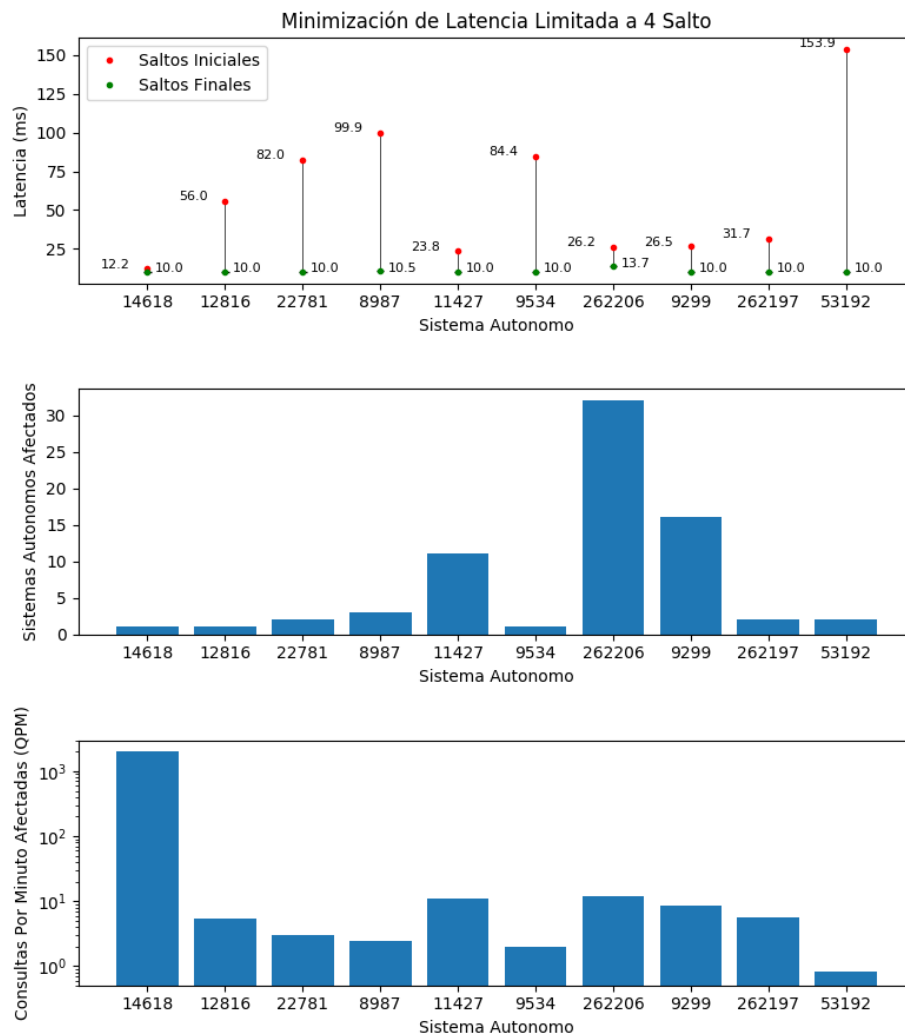


Figura 8.4: Gráfico que muestra la variación de latencia, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la ponderación entre el tráfico afectado y la variación de latencia, limitando el máximo de propagación del área de servicio de este nuevo nodo en cuatro saltos.

8.2. Apéndice 2: Minimizar Saltos Ponderados Limitando Saltos

8.2.1. Minimizar Saltos Ponderados Limitando a 1 Saltos

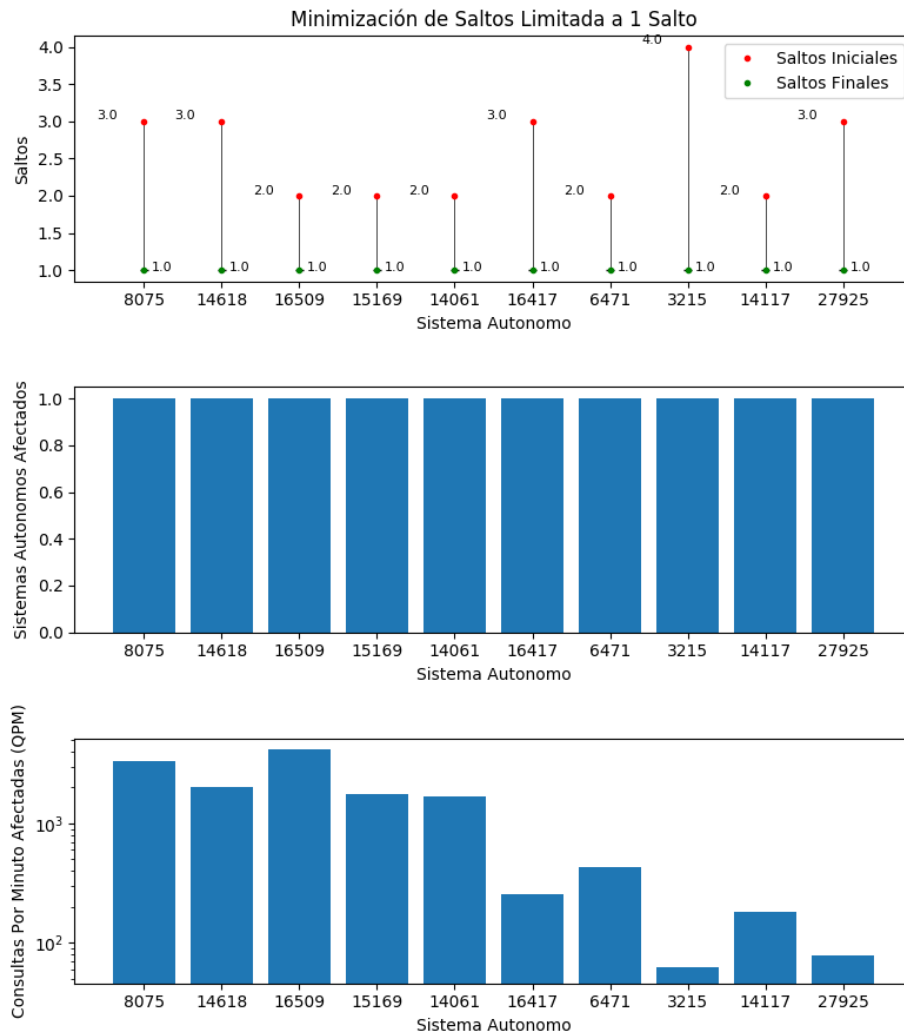


Figura 8.5: Gráfico que muestra la variación de saltos media, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la cantidad de saltos requeridos por los *Sistemas Autónomos* para acceder a la nube *anycast* analizada y el tráfico generado por estos, limitando la propagación de esta ruta en un saltos.

8.2.2. Minimizar Saltos Ponderados Limitando a 2 Saltos

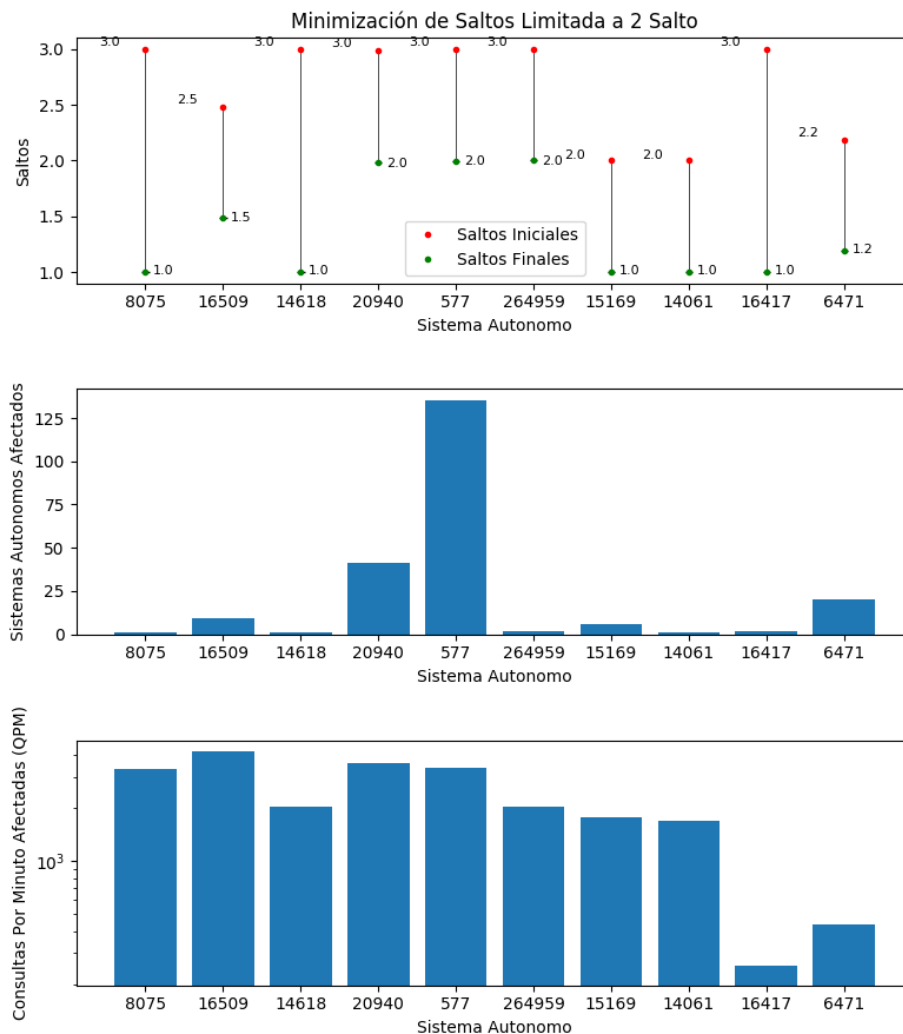


Figura 8.6: Gráfico que muestra la variación de saltos media, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la cantidad de saltos requeridos por los *Sistemas Autónomos* para acceder a la nube *anycast* analizada y el tráfico generado por estos, limitando la propagación de esta ruta en dos saltos.

8.2.3. Minimizar Saltos Ponderados Limitando a 3 Saltos

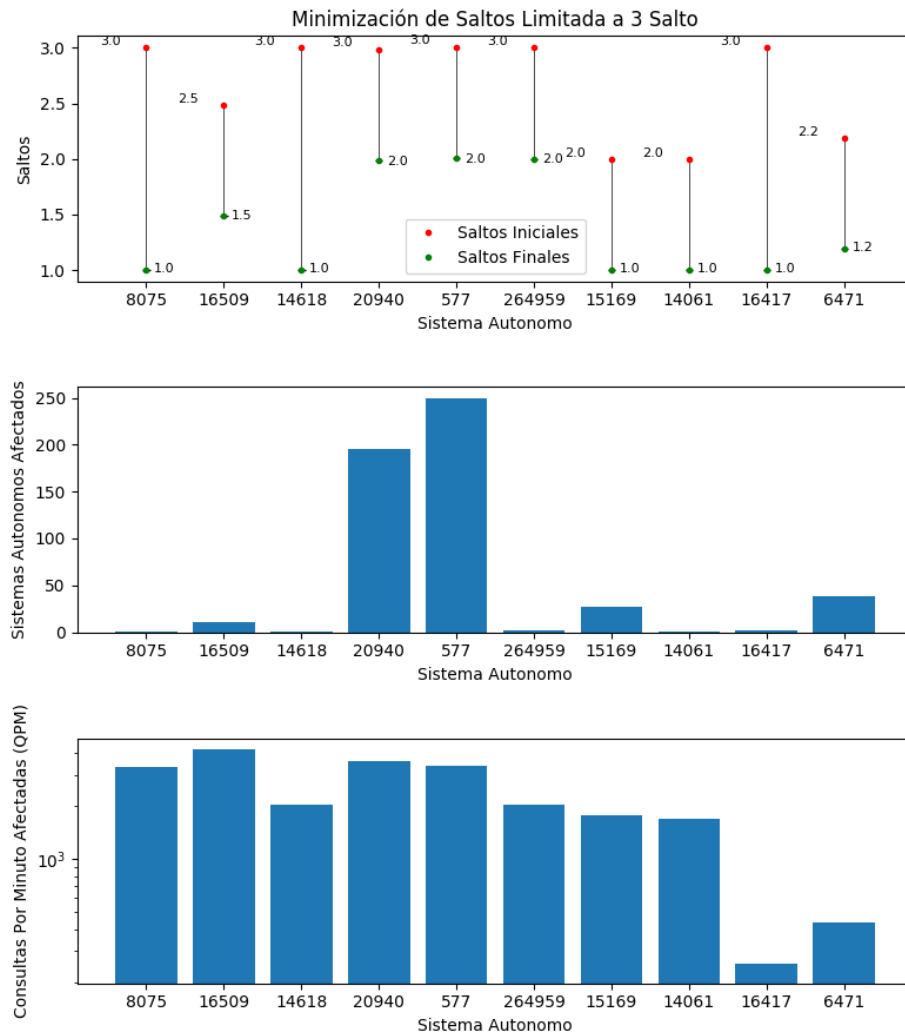


Figura 8.7: Gráfico que muestra la variación de saltos media, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la cantidad de saltos requeridos por los *Sistemas Autónomos* para acceder a la nube *anycast* analizada y el tráfico generado por estos, limitando la propagación de esta ruta en tres saltos.

8.2.4. Minimizar Saltos Ponderados Limitando a 4 Saltos

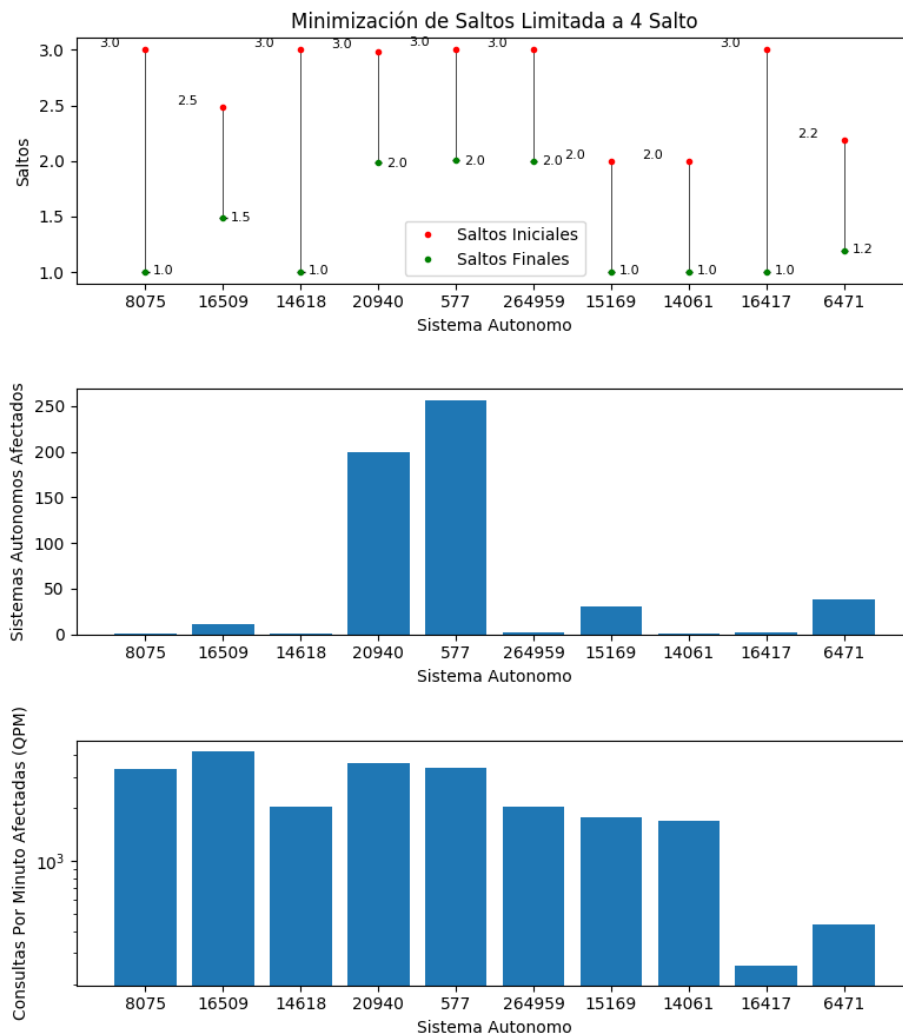


Figura 8.8: Gráfico que muestra la variación de saltos media, número de *Sistemas Autónomos* afectados y consultas por minuto afectadas (escala logarítmica) por posicionar un nuevo servidor en el *Sistema Autónomo* indicado, optimizando por la cantidad de saltos requeridos por los *Sistemas Autónomos* para acceder a la nube *anycast* analizada y el tráfico generado por estos, limitando la propagación de esta ruta en cuatro saltos.

8.3. Apéndice 3: Ruteo Entre Sistemas Autónomo

8.3.1. Peering Amsterdam

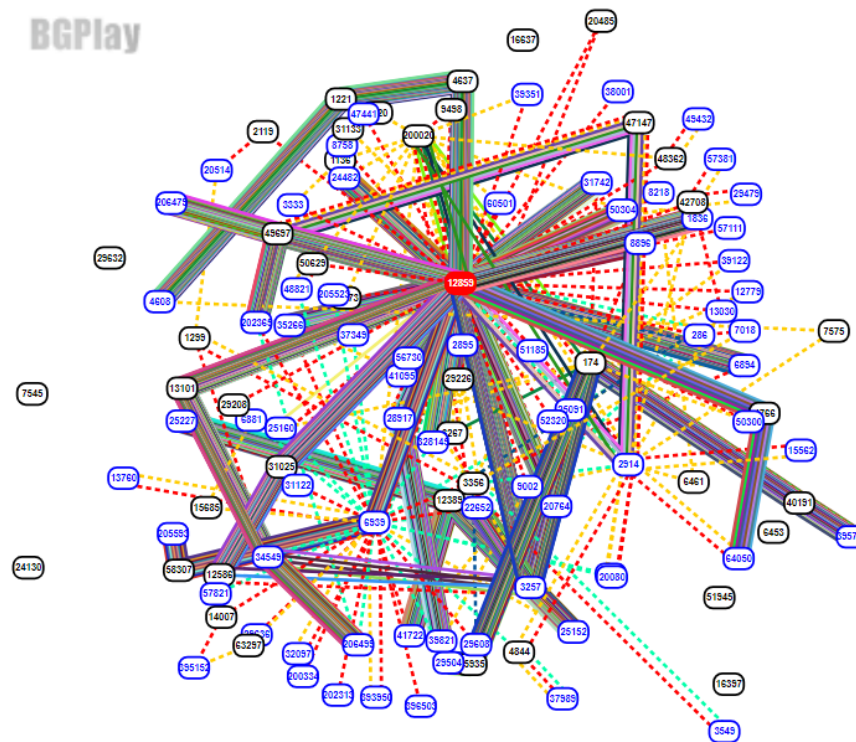


Figura 8.9: Rutas disponibles desde el *Sistema Autónomo 12859*, punto de entrada de la nube *anycast* de prueba *PEERING*. Visualización generada a través de BGPlay.

8.3.2. Peering UFMG

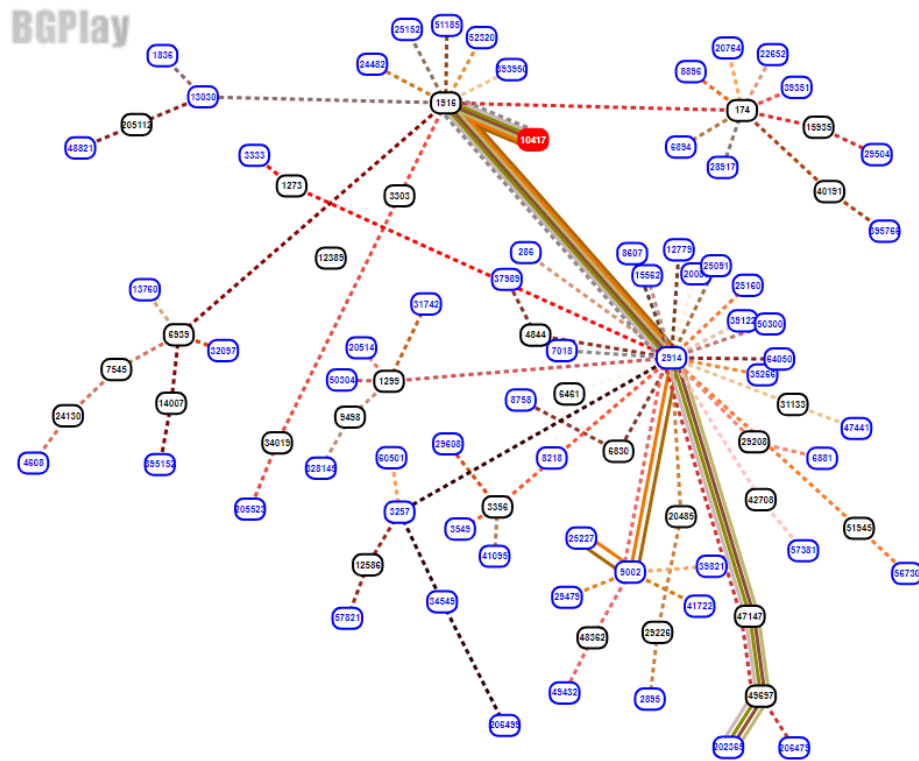


Figura 8.10: Rutas disponibles desde el *Sistema Autónomo 10417*, punto de entrada de la nube *anycast* de prueba *PEERING*, nodo *ufmg*. Visualización generada a través de BGPlay.

8.3.3. Peering Seattle

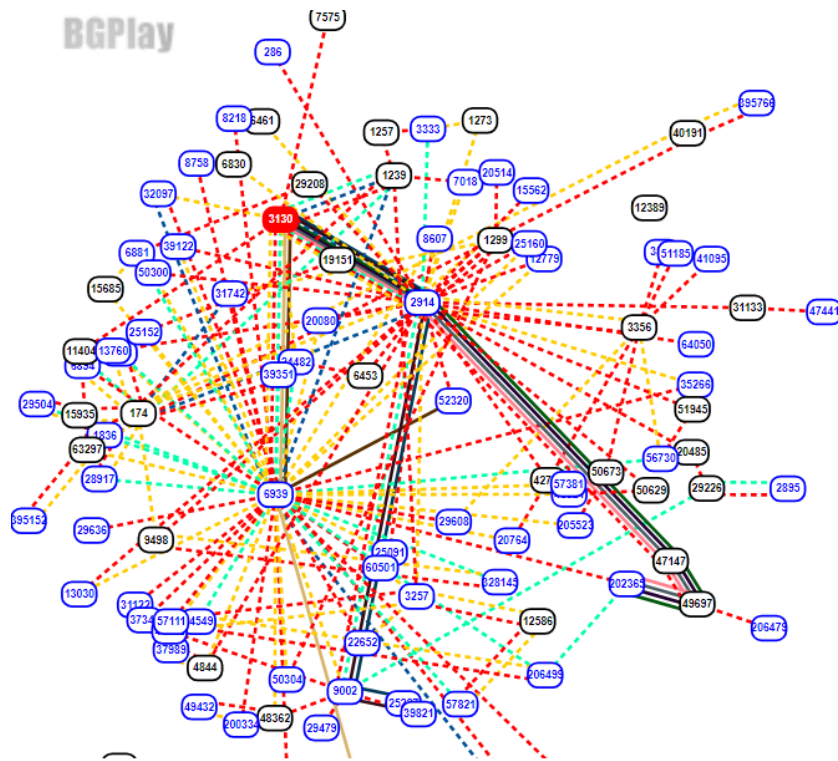


Figura 8.11: Rutas disponibles desde el *Sistema Autónomo 3130*, punto de entrada de la nube *anycast* de prueba *PEERING*, nodo *seattle*. Visualización generada a través de BGPlay.