Propuesta de implementación de Carrier-Grade NAT para Guifi.Net

Aaron Castro Sánchez
Grado de Ingeniería Informática
Tecnologías de la Información

Consultor: Victor Oncins Biosca
Profesor Responsable: Joan Manuel Marquès Puig

10/06/2020

A mi familia, por el apoyo que me han brindado todos estos años en la UOC y por el ánimo enorme que me facilitaron tanto en los buenos como en los malos momentos. Y por saber que llegaríamos hasta aquí.

A Victor Oncins por su ayuda y guía durante este proyecto.

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Propuesta de implementación de Carrier-Grade NAT para Guifi.Net*
Nombre del autor:	*Aaron Castro Sánchez*
Nombre del consultor/a:	*Victor Oncins Biosca*
Nombre del PRA:	*Joan Manuel Marqu*ès Puig
Fecha de entrega (mm/aaaa):	06/2020
Titulación:	*Grado Ingeniería Informática*
Área del Trabajo Final:	*Tecnologías de la Información*
Idioma del trabajo:	*Castellano*
Palabras clave	*Redes, NAT, Internet, IPv6, ISP, CGNAT*

Resumen del Trabajo:

El siguiente proyecto, como trabajo de fin de grado pretende unificar y asentar la mayor parte de los conocimientos adquiridos todos estos años cursando el Grado de Ingeniería Informática. Dentro del ámbito de las redes abiertas, consiste en la implementación de un sistema Carrier-Grade NAT como paso previo a una futura transición a IPv6 que podrá aplicarse en redes multioperador como Guifi.Net.

En este documento estudiamos los conceptos y teorías indispensables para diseñar y dimensionar de forma correcta el despliegue de los dispositivos necesarios para implementar esta tecnología, y que extiende enormemente la vida y uso de IPv4 dentro de los proveedores servicio.

Para la realización y validación de los conceptos expuestos, se ha construido un laboratorio que, en su escala, permite comprobar que las tecnologías descritas en este documento funcionan como se pretende y pueden ofrecer un servicio confiable, de forma que la adopción de CG-NAT solamente suponga una mejora en la calidad del servicio ofrecido por esos operadores.

Abstract:

This memory, as an end-of-degree project, aims to unify and settle most of the knowledge acquired all this years studying the Computer Engineering Degree. Within the scope of open networks, it consists on the implementation of a Carrier-Grade NAT solution, prior to a future transition to IPv6 that can be applied in multi-operator networks such

as Guifi.Net.

In this document we study the essential technologies and foundational concepts to correctly design and size the deployment of the necessary devices to implement this technology, which greatly extends the life and use of IPv4 within service providers.

In order to test and validate the exposed concepts, a virtual test-bed has been built from scratch, that allows verifying how the described technologies shown in this document work as intended and can offer a reliable service in such a way that the CGNAT adoption will only entail an improvement in the quality of service offered by these operators.

Índice

Tabla de ilustraciones

1. Introducción

Tal y como Internet fue diseñada, el direccionamiento IP es su base. Cada dirección IP identifica de manera unívoca a un nodo en la red. Inicialmente se diseñó pensando que 32 bits serían suficientes para su completo direccionamiento. Pronto se comprobó que no es así, el crecimiento sostenido de Internet desde su concepción provocó un agotamiento de estas direcciones [1] rápidamente.

De hecho, el protocolo que se había estandarizado en 1981 sufrió una explosión de uso en los años 90, y para 2011 la IANA[1] había asignado los últimos 5 bloques de direcciones IP. La ilustración 1 [2] muestra la línea temporal en la que se puede observar esta evolución más detalladamente.

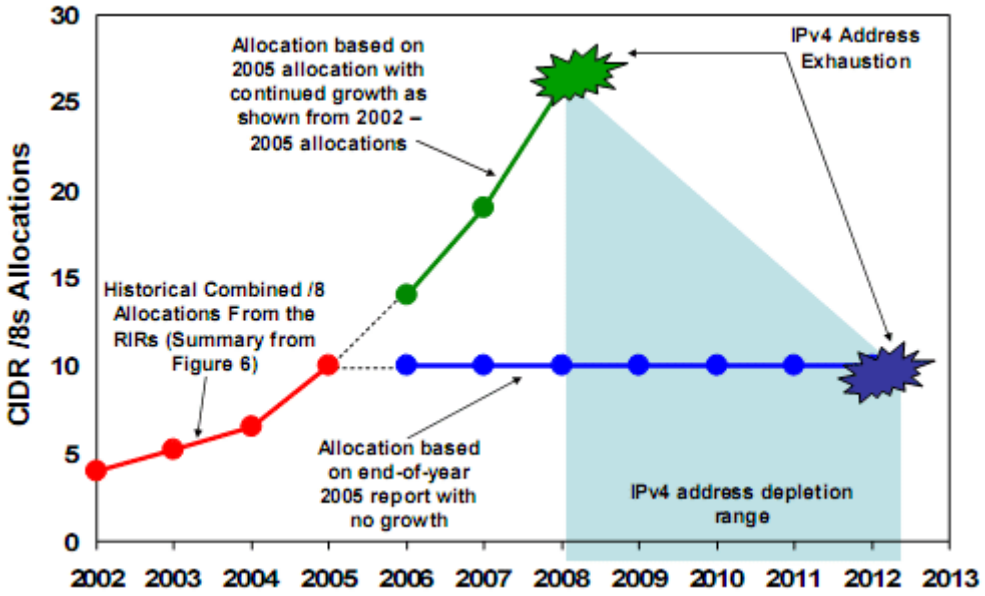


Ilustración 1: Agotamiento de direcciones IPv4

Para solucionar el problema del agotamiento de IPv4, la IETF[2] comenzó a trabajar en el diseño de IPv6, la evolución necesaria, que asignaba 128 bits para direccionar e identificar equipos, lo que significa $3{,}4 \times 10^{38}$ direcciones, en teoría un espacio inagotable a pesar de que al igual que sucede con su antecesor, existen rangos de estas direcciones que no son utilizables en la práctica al estar reservadas.

A pesar de tener IPv6 y ser un protocolo maduro [3] y ampliamente estudiado y probado, la transición de IPv4 a IPv6 todavía va a necesitar un largo tiempo para completarse. Son necesarias actualizaciones de *hardware*, aplicaciones, protocolos y servicios. Por lo tanto, es necesaria una coexistencia entre ambas versiones. Los ISP, han de diseñar, implementar, mantener y soportar redes en las que los dos protocolos coexistan. Se dispone ya de numerosos

1 Internet Assigned Numbers Authority. https://www.iana.org
2 Internet Engineering Task Force. https://www.ietf.org/

mecanismos de transición de uno a otro, pero siguen requiriendo de IPv4 para funcionar.

Es crucial por lo tanto mantener, conservar y racionar el escaso direccionamiento IPv4 que existe. Durante todos estos años se han generado mecanismos para conservar y retrasar el agotamiento de IPv4, como NAT (*Network Address Translation*). La traducción de direcciones de red, principalmente se basa en que varios nodos comparten un mismo direccionamiento IP de cara a otros nodos. A pesar de chocar frontalmente con el diseño inicial de la red, en la que cada nodo ha de ser identificado inequívocamente, NAT ha demostrado su utilidad y ha conseguido prolongar la vida de IPv4.

Mientras que NAT está pensado para organizaciones de cualquier tamaño, no está exento de limitaciones, y no soluciona el problema real, tan sólo lo retrasa. Es por esto que los ISP han pasado a utilizar sus propias implementaciones de NAT a gran escala, como Large-Scale NAT, también conocido como Carrier-Grade NAT [4][3].

1.1 Contexto y justificación

Como cualquier otro proveedor de servicios de conectividad a Internet, cada uno de los ISP que están adheridos a la Fundación Guifi.Net heredan los mismos problemas. Al utilizar la misma arquitectura de red basada en el protocolo IPv4, hereda todas y cada una de sus bondades y limitaciones. Además, la filosofía abierta de Guifi.Net, en cuanto a participación y ciertos criterios de libertad en cuanto a organización, genera ciertos inconvenientes, de forma que la asignación de segmentos y direcciones queda un poco al libre albedrío de cada uno de los proveedores integrantes, que proveen acceso desde sus propios nodos con su infraestructura privada. Esto supone falta de consenso y heterogeneidad en los diferentes diseños de infraestructura, así como en los fabricantes de dispositivos que se utilizan por parte de cada uno de ellos.

Tal como podemos observar el problema del direccionamiento es sistémico, y se multiplica al aumentar el número de nodos a gestionar. Por lo tanto, nos enfrentamos al desafío de implementar CGNAT sobre proveedores de servicio de tipo abierto cuyos mecanismos de organización y estructuración no son iguales a los de los proveedores privados.

1.2 Objetivos

A continuación, se enumeran los objetivos del presente trabajo y que intentarán dar solución al problema ya mencionado durante la introducción.

1.2.1 Objetivo principal

El objetivo principal de este proyecto es el de proporcionar una implementación de CGNAT válida para Guifi.Net y que ayude a extender la vida útil de IPv4 en

3 Asia Pacific Network Information Centre. https://www.apnic.net/

este proveedor durante el mayor tiempo posible hasta que se realice una transición completa a IPv6.

1.2.2 Objetivos secundarios

Como objetivos secundarios de este trabajo, y que serán fundamentales para la consecución del objetivo principal, se encuentran los siguientes:

- Evaluar las limitaciones de CGNAT.
- Proporcionar un diseño de CGNAT adecuado a Guifi.Net.
- Analizar si el equipamiento disponible actualmente en Guifi.Net proporciona capacidades de CGNAT.
- Valorar la transición a IPv6 en la medida de lo posible.

1.3 Alcance

Mientras el alcance de este proyecto se limita al análisis y validación de la red IP de Guifi.Net para una implementación de CGNAT, no significa que los datos y conclusiones que se extraigan del mismo no puedan ser exportadas a otros proyectos e incluso proveedores. Habrá que tener en cuenta en ese caso, que ciertas implicaciones e idiosincrasia propia de Guifi.Net que impregnan este documento pueden no ser válidas para otros casos de uso.

Es necesario recalcar que este trabajo y las pruebas de laboratorio que para él se realizan, contemplan una infraestructura de CGNAT que, si bien puede ser aplicable a las diferentes arquitecturas de red que pueden desplegar los proveedores de servicio, se han realizado teniendo en cuenta ciertas particularidades. Antes de extrapolar datos o aventurarse en implementaciones de esta tecnología, es necesario un análisis exhaustivo de los requerimientos del operador sobre el que decida implementarse. Son muchos los diferentes escenarios y casos de uso a los que podemos enfrentarnos, y no realizar una planificación adecuada, puede abocar al fracaso de un proyecto de semejante magnitud.

1.4 Planificación

El desglose de tareas que se llevarán a cabo durante la duración del presente proyecto se muestra a continuación:

	Tareas	Duración	Inicio	Final
1	Memoria TFG	105 días	02.03.20	15.06.20
2	Entrega 1	7 días	02.03.20	07.03.20
3	Introducción a Guifi.Net	1	02.03.20	02.03.20
4	Justificación del trabajo	2.5	03.03.20	05.03.20
5	Redacción de objetivos	2.5	04.03.20	06.03.20
6	Propuesta de planificación	1	07.03.20	07.03.20
9	Entrega 2	29 días	09.03.20	21.05.12
10	Estudio de CGNAT	8	09.03.20	16.03.20
11	Diseño	8	17.03.20	24.03.20
12	Topología	8	25.03.20	01.04.20
13	Transición a IPv6	5	02.04.20	06.04.20
16	Entrega 3	35 días	07.04.20	11.05.20
17	Provisión de laboratorio	8	07.04.20	14.04.20
18	Despliegue de la topología	8	15.04.20	22.04.20
19	Análisis de funcionamiento	10	24.04.20	02.05.20
20	Beta IPv6	9	03.05.20	11.05.20
23	Entrega 4	35 días	12.05.20	15.06.20
24	Compilar y redactar conclusiones	15	12.05.20	26.05.20
25	Diseño	7	27.05.20	02.06.20
26	Revisión y entrega	13	02.06.20	15.06.20

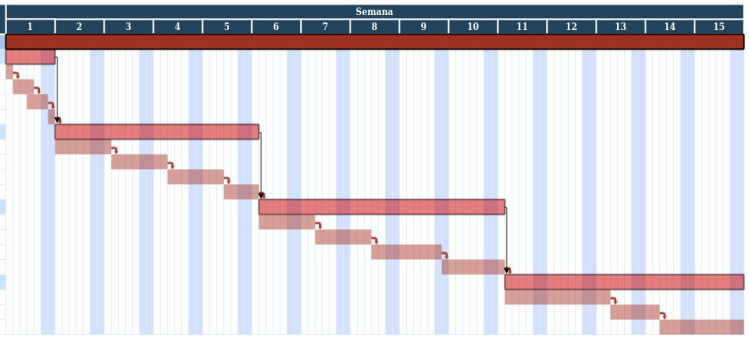


Ilustración 2: Diagrama de Gantt

Esta es una aproximación inicial de la planificación a grandes rasgos que se llevará a cabo. Durante la duración estimada del proyecto pueden producirse variaciones que se verán reflejadas en la versión final.

Cabe comentar que los fines de semana, tal y como se destaca en las columnas sombreadas, se consideran jornadas no laborables aunque sí se hayan invertido horas de trabajo durante los mismos. Además se alinean los hitos de este proyecto con las entregas realizadas en el aula de la asignatura.

1.4.1 Revisiones

Se realiza una revisión de la planificación durante la segunda fase del proyecto con el fin de ajustar los tiempos de la tercera fase. Esto fue necesario para preparar de manera correcta el laboratorio y las pruebas realizadas. La implementación de CGNAT probando diferentes tipos de arquitecturas de proveedor de servicios se ajustó a una sola de ellas. La implementación de CGNAT se realiza en una capa de abstracción que la convierte en independiente de la arquitectura de proveedor, lo cual simplificó en cierta medida el trabajo al eliminar por redundante parte de las configuraciones que se tomaron en cuenta al inicio del proyecto.

2. Metodología

2.1 Estudio de CGNAT

Para estudiar las limitaciones de CGNAT debemos primero estudiar brevemente el funcionamiento de NAT, dado que CGNAT no es más que NAT aplicado a gran escala.

2.1.1 Cómo funciona NAT

Se trata de un conjunto de mecanismos cuya base se centra en compartir direcciones IP entre varios dispositivos. El ejemplo más claro lo tenemos en las conexiones de banda ancha de suscriptores. En este caso el proveedor asigna una sola dirección IP a un suscriptor. Mediante un dispositivo con capacidades de NAT, se realiza la traducción del conjunto de direcciones privadas utilizadas en el domicilio del suscriptor, contra la única dirección IP que el proveedor le ha asignado a este abonado.

Existen múltiples estrategias de traducción de direcciones, nombradas de forma ligeramente diferente según la bibliografía que se consulte. Para este estudio, utilizaremos la nomenclatura [5] utilizada por uno de los principales fabricantes de dispositivos de red y acceso a Internet, Cisco Systems, Inc[4].

NAT	Estático	Traducción 1:1 manual. Se utiliza cuando han de comunicarse redes con direccionamiento incompatible. La traducción se realiza manualmente.
	Dinámico	Traducción 1:1 automática. Se utiliza cuando han de comunicarse redes con direccionamiento incompatible. La traducción se realiza automáticamente mediante un conjunto de direcciones asignado (*pool*).
PAT	Estático	Traducción de puertos. Un puerto específico en la dirección pública se traduce a un puerto específico diferente en la dirección privada.
	Sobrecargado	Traducción 1:N. Un conjunto de direcciones privadas se traduce en una o varias direcciones públicas. Es necesaria la traducción de múltiples tuplas dirección:puerto para realizarla, así como disponer de un *pool* en el segundo caso.

Tabla 1: Tipos de NAT

Para facilitar la comprensión de las diferencias entre cada uno de los tipos mencionados de NAT, presentamos a continuación algunas ilustraciones.

4 https://www.cisco.com

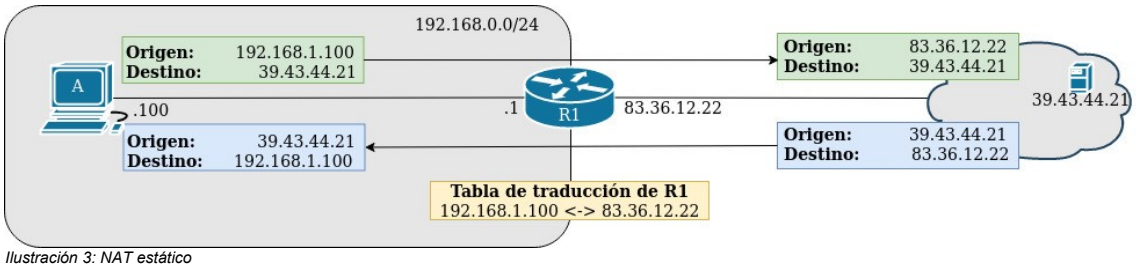


Ilustración 3: NAT estático

Como se aprecia en la ilustración 3, el NAT estático traduce la dirección privada del equipo A, a una única dirección pública enrutable en Internet y diferente de la original. La tabla de traducción de R1 ha de mantenerse para que el tráfico de respuesta originado pueda ser enrutado correctamente al equipo de la red interna correspondiente.

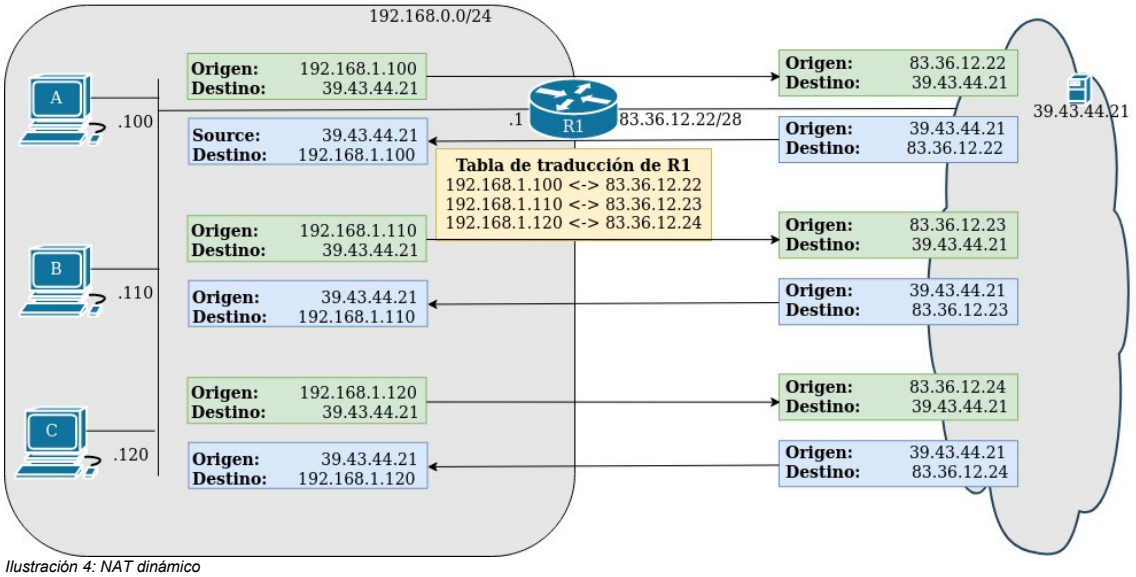


Ilustración 4: NAT dinámico

En el caso de la ilustración 4, cada equipo recibe una dirección diferente del *pool* público existente en R1, en este caso el rango va desde la dirección pública 83.36.12.17 hasta la 83.36.12.30. La tabla de traducciones se encarga de asignar direcciones de ese *pool* consecutivamente a cada uno de los flujos de tráfico originados en la red privada. Del mismo modo, es capaz de dirigir el tráfico devuelto al equipo que lo originó.

Es importante recalcar que las direcciones se asignan consecutivamente. Es decir, la primera sesión generada utilizará la primera dirección IP libre del *pool* y el puerto no reservado más bajo libre. La siguiente sesión, independientemente de si se generó por el mismo equipo de la red interna o no, utilizará el siguiente puerto no reservado de la misma IP del *pool*. Una vez se agoten los puertos, se pasará a utilizar la siguiente dirección IP libre, y así sucesivamente.

Esta estrategia es la utilizada en la mayoría de las implementaciones, y el algoritmo es algo más complejo, pero sería objetivo de un nuevo estudio en sí mismo y sale fuera del alcance de este documento.

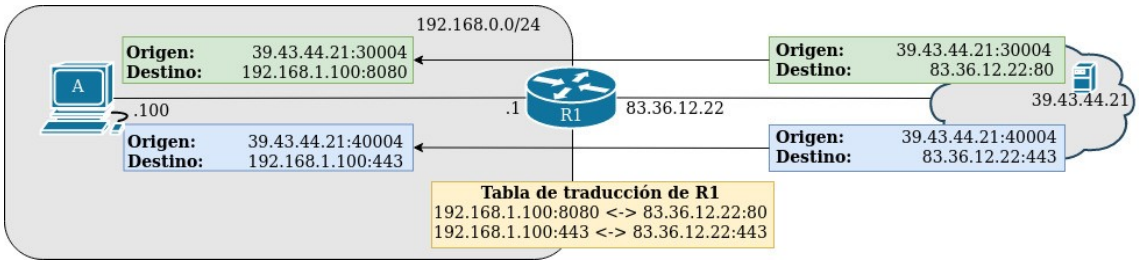


Ilustración 5: PAT estático

En el caso que nos ocupa, tal como se aprecia en la ilustración 5, se realiza una traducción de puertos de forma estática. De esta manera se permite el acceso a equipos de la red privada cuando las conexiones se originan desde equipos en Internet. Resulta de utilidad cuando ha de darse acceso ubicuo a servicios locales. Como se observa, la tabla de traducciones de R1 se encarga de redirigir el tráfico destinado a la dirección pública de R1 para un determinado servicio, al equipo de la red privada que lo mantiene.

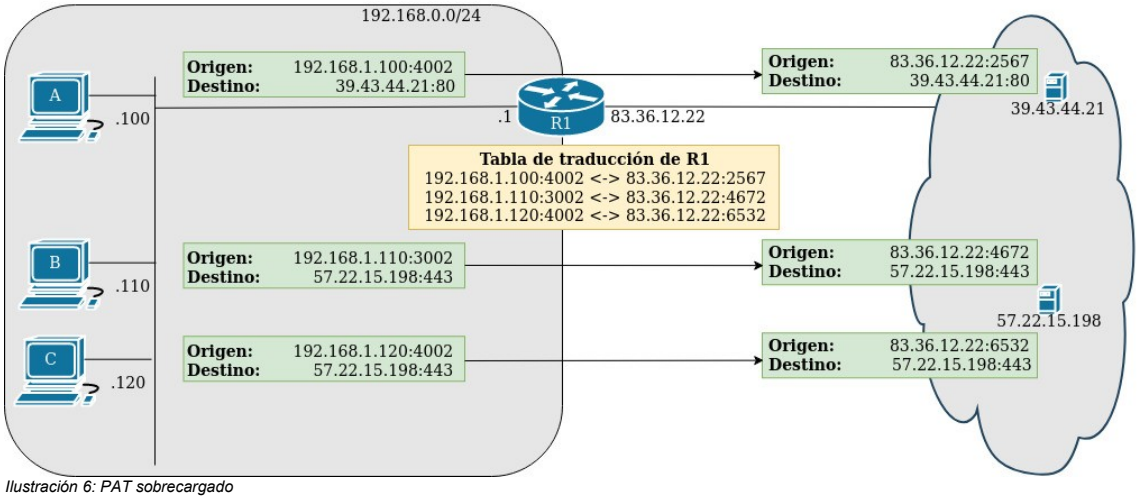


Ilustración 6: PAT sobrecargado

Este último caso es el que resulta más similar a la arquitectura de CGNAT. En él observamos como R1 realiza traducciones tanto de la dirección IP como del puerto asignado. De esta forma es capaz de mantener las sesiones correspondientes a cada equipo, independientemente del destino al que se dirijan. La tabla de traducciones ha de mantenerse suficiente tiempo en la memoria de R1 para evitar que haya interrupciones de tráfico por *timeout* en las aplicaciones de los clientes.

Podemos apreciar que varios equipos diferentes pueden acceder al mismo servicio en Internet incluso si los números de puerto originales se solapan entre varias máquinas. Recordemos que los números de puerto efímeros los escoge

el sistema operativo de forma aleatoria a la hora de iniciar la comunicación, pero éstos son más adelante sustituidos al ser procesados por el algoritmo de NAT que se ejecuta en el *router*.

Como ya se ha explicado, los nodos de Internet se identifican mediante una dirección IP. Estas direcciones se utilizan de forma similar al cómo funciona el servicio de correos. Las direcciones de los nodos de origen y destino se utilizan para que la red pueda enviar la información de manera óptima de un nodo a otro.

La información se encapsula en paquetes IP autocontenidos, lo que quiere decir que cualquier nodo de la red que lea ese paquete podrá saber su origen y destino. De hecho, es la información que utilizan los nodos intermedios para saber cuál es el siguiente nodo en el camino hacia el destino. Toda esa información se encuentra en la cabecera del paquete IP y se encuentra definida en el RFC791[5] [6][6].

Para el propósito de este estudio, no revisaremos todos y cada uno de los parámetros en una cabecera IP, tan sólo los que se verán afectados y modificados directamente por NAT y CGNAT, que para ésta son los campos *Source Address* y *Destination Address*.

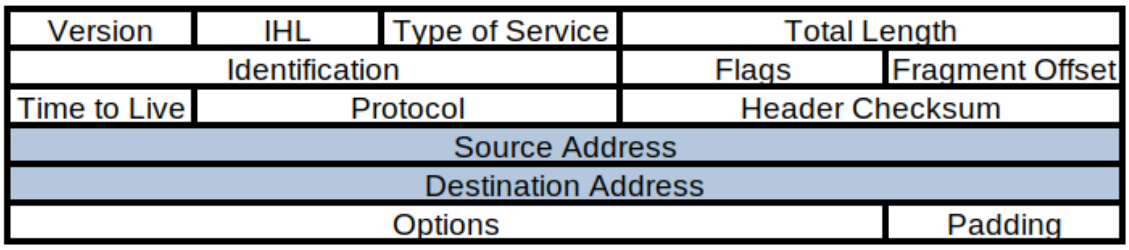


Ilustración 7: Cabecera IP

Los paquetes IP pueden llevar a su vez múltiples protocolos, entre ellos ICMP [7], UDP [8] y TCP [9]. Los dos últimos, pertenecientes a la capa de transporte, incorporan información de puertos de origen y destino. Ésta es fundamental a la hora de que las pilas de red de los nodos puedan decidir a qué capa superior del modelo OSI[7] corresponde el paquete que se está tratando en ese momento, pudiendo de esta forma transferir la información de forma correcta a la aplicación correspondiente.

La tupla dirección y puerto de origen junto a dirección y puerto de destino identifican cada conexión y sesión en particular que se denomina *socket*. Estas sesiones sirven para diferenciar las diferentes conversaciones que varios nodos de la red pueden estar teniendo simultáneamente. Por ejemplo, dos pestañas diferentes del navegador, o dos transferencias FTP [10] al mismo nodo.

5 Request for Comments. https://www.ietf.org/standards/rfcs/
6 Defense Advanced Research Projects Agency. https://www.darpa.mil/
7 Open Systems Interconnection

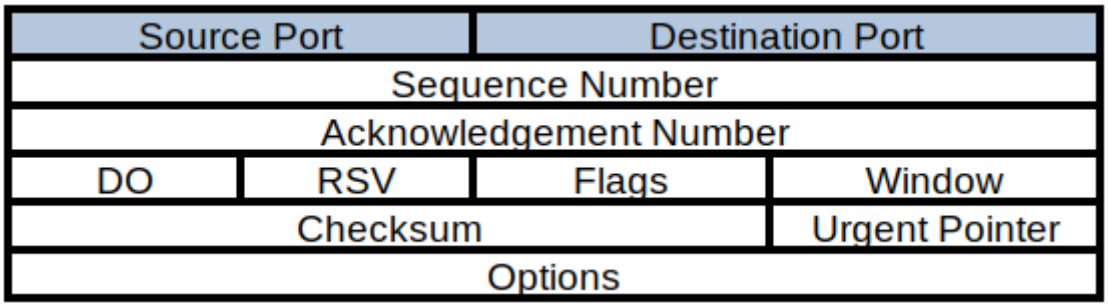


Ilustración 8: Cabecera TCP

La introducción de NAT provoca que hayan de crearse tablas de traducción para cada una de estas sesiones. Para compartir una única dirección IP, es necesario identificar de alguna manera cada una de las conexiones que los equipos detrás del dispositivo que hace NAT, si no la información no podría ser devuelta al nodo de origen correcto. La función de la tabla de traducción es ayudar a este dispositivo intermedio.

Tal como se ve en los ejemplos de las ilustraciones 3, 4, 5 y 6, las cabeceras IP de cada uno de los paquetes que han de traducirse, se ven modificadas. Para cada uno de esos paquetes ha de cambiar la dirección IP del nodo interno, y colocar la dirección pública. Por lo tanto, es necesario que modifique de la misma manera los números de puerto de cada conexión, manteniendo un registro que le permitirá identificar cada una de las sesiones establecidas. A esta estrategia de traducción se la denomina NAT44, que es la que se representa en la ilustración 6. Cada uno de los números en NAT44, indica la familia de direcciones a las que afecta. En este caso, indica que se realiza traducción de IPv4 a IPv4.

2.1.3 Qué implica CGNAT

CGNAT por su parte, adapta esta filosofía de traducir un conjunto de direcciones privadas mediante un dispositivo capaz hacia una sola dirección IP, y la lleva a la frontera del proveedor de servicios. En lugar de utilizar una dirección por abonado, el ISP se encarga de asignar una sola dirección a un conjunto de abonados. La idea es la misma, conservar direcciones públicas en tanto en cuanto se realiza la transición completa a IPv6.

Cuando los proveedores introducen CGNAT, lo que hacen es introducir una capa adicional a NAT44, comúnmente denominada NAT444 y representada en la ilustración 9 (en la que se realizan dos traducciones de IPv4 a IPv4), y que implica que los paquetes y la información contenida en la cabecera IP se modifica más de una vez utilizando el mismo mecanismo, pero con una nueva tabla de traducciones, esta vez en el lado del ISP. De este modo, en lugar de ser necesaria una dirección por cada subscriptor, se puede utilizar una sola para múltiples de ellos.

Por supuesto, esta nueva capa de direccionamiento utiliza un rango de direcciones también privado, y dado que las redes privadas del lado del

suscriptor y del lado del proveedor, no se pueden solapar, ha de introducirse un direccionamiento privado y a su vez diferente del que utilizan los suscriptores.

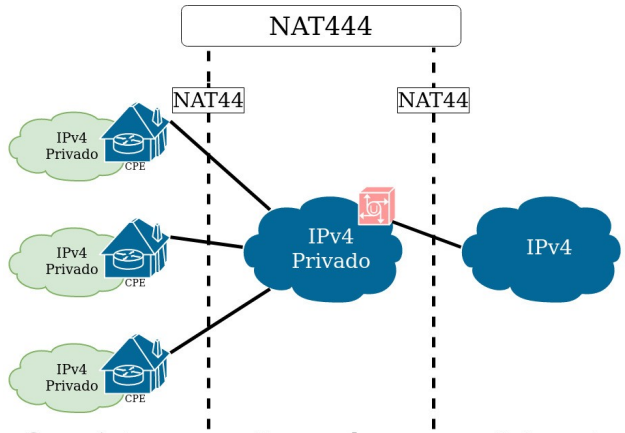


Ilustración 9: NAT 444

2.1.4 Limitaciones de CGNAT

Ahora que tenemos el funcionamiento de NAT y CGNAT definidos, pasamos a listar las limitaciones de estas estrategias.

En el ámbito técnico, existen numerosas limitaciones cruciales que afectan directamente a las conexiones a Internet de los suscriptores. Cada una de estas limitaciones tiene impacto tanto en el usuario final como en el proveedor de servicios, dependiendo de cuál de ella se trate.

Límite en el número de sesiones

Habitualmente los nodos cliente, suelen establecer varias conexiones con los nodos servidor para una misma aplicación. Los motivos de esta multiplicidad son, por ejemplo, proporcionar mayor rendimiento, agregar funcionalidad o establecer canales independientes de control. Así, por ejemplo, el protocolo FTP, en modo de transferencia activo, utiliza dos canales. Un canal de control que utiliza el puerto *well-known* TCP/21, y un canal para la transferencia de datos que utiliza el puerto *well-known* TCP/20. Los puertos *well-known* son aquellos definidos por la IANA que identifican unívocamente numerosos servicios. El RFC1340 [11] da cuenta de estas asignaciones.

El estudio del protocolo FTP queda fuera de este estudio, pero puede consultarse su definición completa en el RFC959 [10]. A partir del punto 5.2 de este RFC se encuentra la información relativa a la asignación de puertos para los modos de transferencia activo y pasivo.

Dado que TCP y UDP, como protocolos de capa 4, tienen asignados 16 bits para el número de puertos que pueden utilizar, quedan limitados a 65.536. A éstos es necesario quitar los *well-known* y los reservados, con lo que el número disminuye.

Los puertos *well-known* son aquellos definidos por la IANA en el RFC1340. Se trata de los 1024 puertos más bajos (0 al 1023) tanto en TCP como en UDP, y que se reservan a servicios de sistema. Al hacer traducción de direcciones, tal y como vimos en las ilustraciones de los diferentes tipos de traducciones, es necesario mapear y sustituir los números de puertos asignados a las conexiones.

El aumento del número de clientes a los que se hace NAT, disminuye en ese factor el número de puertos disponibles por cada conexión. Es decir, teniendo un sólo suscriptor (1 dirección IP pública), haciendo NAT44, implica que tendremos disponibles los 65.536 puertos para él. Si aumentamos el número de suscriptores que comparten la misma dirección, estaremos efectivamente limitando su número de conexiones.

Por lo tanto y para darle perspectiva a esta situación, hemos de quitar de las asignaciones 1024 puertos, lo que nos deja con tan solo 64.512 puertos utilizables. Ahora aumentemos el número de suscriptores. Con 10 suscriptores, dispondremos de 6.451 puertos para cada uno. Con 100 tendremos 645, y así sucesivamente.

Si todas las implementaciones de pila TCP/IP siguiesen las recomendaciones de asignación de puertos dadas por la IANA, habríamos de reducir esos puertos a muchísimos menos, ya que esas asignaciones comenzarían en el puerto 49.152 y llegando hasta el máximo (65.535), lo cual estrangula enormemente el margen de trabajo de los sistemas operativos de red. Afortunadamente esa recomendación no siempre se cumple y tal y como hemos dicho, cada implementación del protocolo TCP/IP lo realiza de forma diferente y no se aplica la limitación de puertos de forma estricta.

Por otro lado, el número de sesiones concurrentes que un equipo puede establecer de manera simultánea depende en gran medida de las aplicaciones que utilice. Un factor importante a la hora del dimensionado de CGNAT es el balanceo entre la eficiencia y capacidad de comunicación de las aplicaciones, y el número de sesiones que se van a permitir por cada suscriptor.

Para tener una aproximación a *grosso-modo* de estos números, se han recogido estadísticas de conexiones diarias de diferentes dispositivos conectados a Internet durante los períodos de uso de los mismos, obteniéndose los siguientes datos y que nos servirán de referencia para dimensionar correctamente la solución:

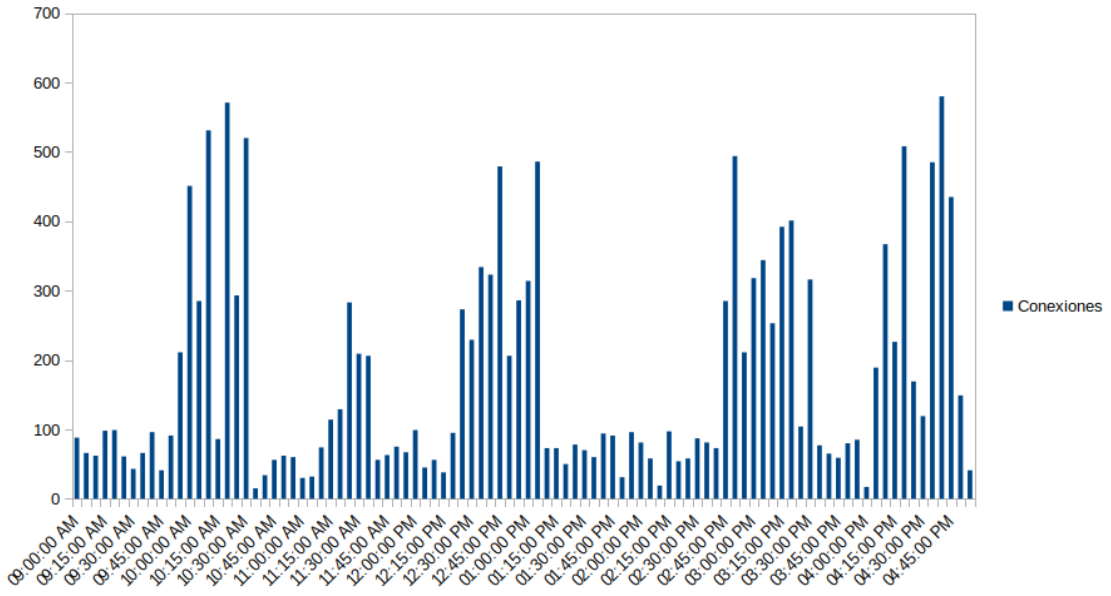


Ilustración 10: Media de conexiones a Internet por dispositivo. Tomadas cada 5 minutos.

La gráfica de la ilustración 10 se ha obtenido almacenando y analizando los patrones de tráfico en el hogar de un suscriptor estándar, utilizando 1 ordenador portátil, 2 teléfonos móviles y una *tablet*. Todo ello dentro del mismo entorno dónde se ha realizado este estudio y son los datos que se tomarán como referencia para dimensionar la solución de laboratorio más adelante.

Si los suscriptores consumen el número máximo de conexiones que les han sido asignadas, resultará en un fallo sostenido de las sucesivas sesiones que necesiten establecerse. El mejor caso de este escenario implicará un rendimiento de las aplicaciones muy pobre, mientras que el peor, provocará cortes continuos en las aplicaciones. Estos cortes serán consecutivos y recurrentes mientras las aplicaciones intentan por todos los medios mantenerse en comunicación, en lo que se liberan recursos en forma de puertos libres desde el lado del ISP.

Por otro lado, las sesiones no se pueden mantener indefinidamente en las tablas de traducción. Es necesario liberar esos recursos para que puedan ser reasignados. Esto hace imperativa una correcta planificación de los patrones de uso de los suscriptores, para evitar que colapsen los recursos del ISP, y éstos puedan mantener un nivel de servicio aceptable.

Limitaciones sobre los protocolos

Como vimos, NAT proporciona traducción de direcciones y mantenimiento de tablas de conexión para TCP y UDP. Pero el hecho es que las redes de datos utilizan muchos más protocolos que no se basan en TCP o UDP ya que no precisan o no implementan el concepto de puerto. Existen ciertos de estos protocolos que no toleran NAT o CGNAT, tales como IPsec [12], RTP [13], ICMP y ciertos mecanismos de transición a IPv6.

- IPsec.

Utilizado para la protección de las comunicaciones, se trata de un protocolo ampliamente extendido para el cifrado de canales de datos entre *hosts*. Básicamente introduce dos nuevas cabeceras que autentican y cifran los orígenes del tráfico. De esta forma ESP [14] encripta la capa de transporte, haciendo invisibles los puertos de la capa de transporte a los dispositivos que realizan NAT. Con AH [15] se autentica la cabecera IP, de forma que un cambio en alguno de sus campos, tal como hace CGNAT con las direcciones IP y puertos, provocan un fallo en esa autenticación.

- Mecanismos de transición a IPv6.

Siempre que el ISP no contemple un servicio nativo de IPv6 para sus suscriptores, mecanismos de transición como *6to4* [16] y *Teredo* [17] dejan de funcionar correctamente cuando se encuentran con interfaces que implementan CGNAT. Es necesario por parte de los proveedores la implementación nativa de IPv6 o al menos la implementación de *dual-stack*[8]. De esta forma se puede proporcionar a los suscriptores acceso directo a Internet mediante IPv6 y acceso con CGNAT utilizando IPv4.

Compartición de direcciones

Como CGNAT se fundamenta en compartir direcciones IP públicas entre varios suscriptores, se producen los siguientes hechos:

- Fallo en la seguridad de cierto tipo de comunicaciones que realizan seguimiento de las direcciones IPv4, tales como aplicaciones de banca *online*.

- Problema de reputación de dirección. Si se produce un abuso por parte de uno de los suscriptores, el resto de ellos que compartan su direccionamiento, sufrirán los posibles baneos sobre esa dirección.

- Publicación de servicios. Dado que se comparte la dirección IP, algunos servicios remotos dejan de funcionar, por ejemplo, el acceso a *webcams* o dispositivos de domótica.

Limitaciones en las redirecciones

Como Internet fue diseñado para ser bidireccional, se esperaba que las sesiones pudiesen establecerse de la misma forma en cualquier dirección, hacia o desde cualquier suscriptor. La introducción de tecnologías de traducción de direcciones rompe este paradigma, y con ello ciertos servicios dejan de funcionar en la forma que deberían hacerlo, como juego online (*p2p*), domótica, servicios de seguridad IP (CCTV) y sistemas de VoIP[9]. Recordemos que éstos requieren o bien que se mantengan intactas las direcciones de

8 *Dual-stack* indica que existe doble pila de protocolos, por lo tanto se pueden interpretar las dos familias, tanto IPv4 como IPv6.
9 Voice over IP. Servicios y protocolos de telefonía sobre IP.

origen y destino, o bien requieren servicios accesibles desde Internet que, como ya vimos requieren configuraciones especiales de *port-forwarding*.

Se han desarrollado a lo largo de todo este tiempo, diversos mecanismos para superar esta limitación, pero sólo aplican de forma satisfactoria a NAT44. La redirección de puertos (*port-forwarding*) puede realizarse desde los CPE de los abonados, pero es una tarea de relativa dificultad para el usuario medio. De esta forma, los mecanismos que mencionábamos permiten reducir esa sobrecarga al usuario, y realizan las redirecciones de forma automática sin intervención. Éstas incluyen UpNP [18], STUN [19], y demás.

Aun así, éstas se ven interrumpidas desde el momento en que los ISP introducen NAT444, ya que las redirecciones no son permitidas. El ISP ha de controlar todas y cada una de las conexiones, ya que han de ser compartidas por múltiples suscriptores. Por ejemplo, si un abonado solicitase al ISP redireccionar el puerto TCP/80 a su IP pública, para publicar un pequeño servidor *web*, esto resultaría en que ese puerto estaría redireccionado para todos los abonados que compartan la dirección IP a la que se hizo la redirección. Resulta obvio que el dispositivo que haya de realizar la traducción encontrará en su tabla múltiples entradas para el mismo puerto de destino, lo que provoca un fallo en la devolución de tráfico al origen.

Registro

Es necesario trazar los orígenes del tráfico que se produce en Internet por imperativo legal. Tal y como Jan Zorz menciona en su artículo "*CGN, IPv6 and figthing online crime*" [20], la implementación de CGNAT supone una dificultad añadida ya que la base del mismo es la compartición de las direcciones IP, lo cual dificulta el rastreo de los delincuentes. Además, ciertas medidas de protección que se implementan para paliar los abusos de forma inmediata penalizan a todos los suscriptores que utilizan su acceso a Internet en ese mismo momento de forma legítima.

Sin mecanismos de traducción de direcciones es extremadamente sencillo. Se identifica la dirección IP de origen y se solicita al ISP el registro de clientes que tuvieron esa dirección asignada en el momento del abuso. Tal como sabemos, las concesiones de direcciones se realizan mediante protocolos de asignación dinámica, con lo que únicamente necesita consultarse el registro correspondiente.

Dirección MAC	Dirección IP	Identificador de CPE	Inicio de concesión	Duración
0005.2329.dca6	185.202.16.123	vuyq6fl5i7	23/03/2020 18:37:22	6hr
0010.a702.ade1	185.202.19.25	hfbywgfpn2	23/03/2020 16:10:54	8hr
000a.1451.6168	185.202.17.200	lxnrqqackk	23/03/2020 21:04:46	3hr
005e.d4d3.2c3d	185.202.17.178	9ej2t2q6dw	23/03/2020 16:21:38	8hr

Table 2: Registro de conexiones

Desde que se introduce CGNAT, el registro y trazado es más complicado. Se hace necesario haber registrado la dirección IP pública utilizada en ese momento (hasta aquí igual que con NAT44), y además ser capaz de recuperar el mapeo instantáneo a la dirección interna (la del suscriptor). Si recordamos el funcionamiento de NAT44 y NAT444, ahora se hacen necesarios también los números de puerto utilizados, tanto interna como externamente.

Estos requisitos imponen sobrecargas en los ISP. Sobre todo, encontramos mayores necesidades de infraestructura y posibles problemas de latencia y rendimiento generados por el procesamiento de nuevas y más grandes tablas de traducción.

La tabla número 3 a continuación, muestra las diferencias que hay para cada tecnología de acuerdo a los requerimientos de registro necesarios.

Tipo de red	Requisitos de registro
Enrutada	Ninguno
CPE & NAT44 - Estático	Ninguno
CPE & NAT44 - Dinámico	Dirección IP del suscriptor y marca de tiempo
CGNAT	Con cada sesión • Marca de tiempo • Protocolo de transporte • Dirección IP privada • Puerto de origen privado • Dirección IP pública • Puerto de origen público

Table 3: Requisitos de registro

El almacenamiento de los datos indicados en la tabla es imprescindible para poder proporcionar un seguimiento óptimo de los direccionamientos utilizados por cada cliente. El problema que existe es que parece no haber aún definido un formato estándar de registro. La IETF tiene un borrador que contempla esta situación y ofrece una propuesta para solucionar este problema [21], pero parece no haberse implantando como estándar.

Los requisitos de registro expuestos anteriormente generan un tamaño de entrada de alrededor de 120 bytes en formato ASCII, 10 bytes para la marca de tiempo, 3 bytes para el protocolo, 15 bytes para cada dirección IP, y 5 bytes para cada puerto. Además, hay que agregar un campo para la identificación del suscriptor, ya que la asignación y recuperación de recursos es necesaria por parte del *router* que realiza CGNAT. Y, por último, el indicador del propio *router* si estos registros se van a almacenar externamente. Si cada suscriptor activo genera una media (C) de conexiones cada 5 minutos, el tamaño total (T) del registro durante un tiempo (t), viene determinado por la fórmula:

$$T(bytes/t) = S \cdot a \cdot C \cdot 120\,bytes \frac{t(min)}{5(min)}$$

Siendo *S* el número total de suscriptores del ISP y *a* el porcentaje de suscriptores activos en un período *t*.

Suponiendo un 30% de suscriptores activos al día sobre un total de 3000, y 400 conexiones por suscriptor, estamos almacenando 12.441$_1$600.000 bytes, o lo que es lo mismo 12.45 Gigabytes al día.

Respecto al consumo de ancho de banda que genera el registro de estas conexiones sobre servidores destinados específicamente a ello, esos 12.441$_1$600.000 bytes cada 5 minutos, suponen de media 4$_1$472.000 bytes/s, o lo que es lo mismo 35.776 Kbps de ancho de banda consumido.

Además de los expuestos a lo largo de este punto, existen también problemas adicionales generados por estas tecnologías, como son retrasos y latencia, que sobre todo impactan a aplicaciones sensibles a éstos, como juego *online*, servicios de banca y bolsa, y otros en tiempo real, tales como VoIP o s*treaming*. Es necesario tener en cuenta que se están añadiendo dispositivos intermedios que tienen que procesar el tráfico, lo cual supone parar los paquetes, leerlos, procesarlos, modificarlos adecuadamente y proceder a su envío. Obviamente ese tiempo de procesamiento, por mínimo que sea introducirá latencia en las comunicaciones extremo a extremo.

2.2 Diseño

El ámbito de este proyecto propone entre otras cosas un diseño válido para utilizar en Guifi.Net, de forma que se apliquen los conocimientos y estrategias de CGNAT que permitan prolongar la vida del servicio IPv4 proporcionado en lo que se produce la transición completa a IPv6. Como ya hemos mencionado, el agotamiento del espacio IPv4 (IANA-Allocation [23]) supone un problema a la hora de incrementar el número de abonados conectados a Internet. Recordemos la explosión de conectividad producida en los últimos años con las redes 4G y el IoT[10].

2.2.1 Consideraciones

La aproximación de CGNAT consistente en compartir el remanente escaso espacio de direcciones IPv4 entre varios abonados, incorpora una nueva capa de NAT a la frontera de red del ISP. CGNAT se emplaza entre las instalaciones de los abonados (CPE) y el núcleo de la red, que es el cual realmente incorpora los mecanismos que regulan la traducción de direcciones para los protocolos ICMP, UDP y TCP. Los RFC5508 [24], RFC4787 [25] y RFC5382 [26] definen los comportamientos que ha cumplir NAT para con sus respectivos protocolos. NAT444 (doble traducción dentro de la familia IPv4) es la estrategia que utiliza CGNAT y paliar el problema del agotamiento de direcciones.

10 Internet of Things. [22]

En este sentido, NAT444 incorpora 3 capas de direccionamiento IPv4:

- Una capa privada, utilizada en las instalaciones y equipos de los abonados.
- Una capa privada, en un espacio de direcciones diferente de la anterior que proporciona los enlaces entre el suscriptor e Internet.
- Una capa pública, con direcciones IPv4 enrutables fuera de la red del proveedor.

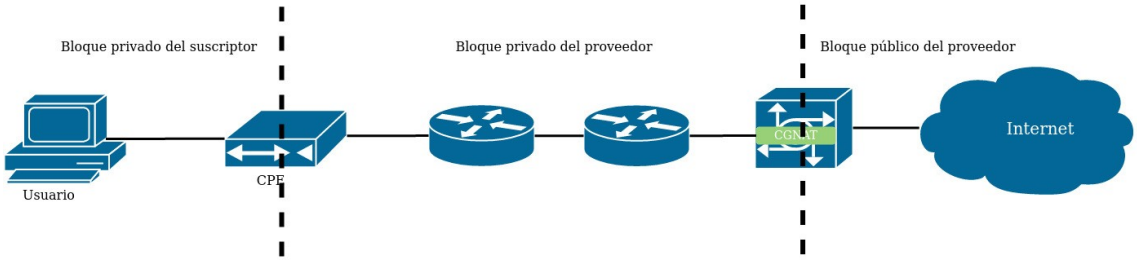


Ilustración 11: Arquitectura CGNAT

Para el ISP las ventajas son obvias:

- Proporciona más tiempo para la implementación de IPv6.
- No hay modificaciones en las redes detrás de los CPEs ni en sus servicios.
- Se pueden implementar ALGs.

Un ALG [27] es un software que gestiona protocolos de aplicación específicos y se encarga de interceptar y analizar el tráfico de esos protocolos convirtiendo, entre otros datos la información de capa de red (direccionamiento) que se encuentra encapsulado dentro de la capa de aplicación. De esta forma es capaz de solventar parte de la problemática que ya referimos anteriormente en el apartado 2.1.2. El protocolo FTP por ejemplo típicamente utiliza varias sesiones. Mediante la sesión de control, se envían los números de puerto que se utilizarán para las sesiones de transferencia de datos, incluyendo las direcciones IP del cliente y el servidor. Si existe un dispositivo intermediario realizando traducciones, las sesiones no pueden establecerse, rompiéndose la transmisión. Los ALG se encargan de enmendar este problema, ya que modifican la información de direccionamiento contenida dentro de los paquetes de control, adaptándola a las traducciones que el dispositivo traductor está realizando.

Como ya se ha explicado en varias ocasiones a lo largo de este documento, para que tanto NAT como CGNAT funcionen correctamente los direccionamientos IP sobre los que se realizan las traducciones, han de ser compatibles. Compatibilidad en este sentido significa que los espacios de direcciones no deben solaparse en redes contiguas (sólo separadas por un

router o un dispositivo de capa 3). Dado que la práctica totalidad del direccionamiento privado utilizado detrás de los CPEs se acoge al direccionamiento reflejado en el RFC1918 [28], las direcciones privadas utilizadas en esas redes están bastante bien delimitadas.

Si consultamos el RFC1918, veremos que se restringe a los prefijos 10/8, 172.16/12 y 192.168/16. Además, la clasificación de direcciones IPv4 realizada por la IANA, ha asignado ciertos segmentos dentro del espacio público que se destinan a ciertas tareas específicas de los ISP.

Segmentado el direccionamiento en el RFC6890 [29], consideraremos únicamente el bloque 100.64/10 como aplicable a este estudio. Este espacio, nombrado como *Shared Address Space*, es el que la IANA ha sugerido para utilización y despliegue de CGNAT tal como se indica en el RFC6598 [30]. Para no romper el estándar, y por supuesto proporcionar la máxima compatibilidad a Guifi.Net, es el que utilizaremos en los despliegues de laboratorio más adelante en este documento.

2.3 Topología

En el caso que nos ocupa, y tal y como mencionamos en el apartado anterior, se utilizarán 3 capas de direccionamiento. Esto acarrea una segmentación de la red en la que serán necesarios 2 rangos especiales, junto al rango público de direcciones que se utilizarán para proporcionar conectividad a los anteriores.

Consideraremos por lo tanto un rango privado del RFC1918 para las redes detrás de los CPEs. Se ha decidido aprovechar este dado que por un lado nos evita modificaciones del lado de los clientes, y por otro la ventaja de que el gran parque de *routers* y dispositivos de conectividad hogareño que se distribuyen actualmente, vienen preconfigurados con un direccionamiento de este espacio, en su gran mayoría 192.168/16, y dentro de éste 192.168.0/24 o 192.168.1/24.

Además, facilita también a las organizaciones que han migrado ese rango de direcciones a 10/8 y 172.16/12, bien sea debido al tamaño de las mismas, cantidad de equipos a conectar o necesidades de seguridad y segmentación, el no tener que realizar una nueva migración para adaptarse a un ISP que implemente CGNAT.

Por lo tanto y acogiéndonos a los RFC6890 y RFC6598, utilizaremos el rango 100.64/10 para la capa de red intermedia y sobre la que se realizará la agregación de suscriptores para sobrecargar el rango de direcciones público de Guifi.Net. Un cálculo rápido nos confirma que con un prefijo /10 nos proporciona $4_1194.302$ direcciones utilizables para suscriptores. Más concretamente 4 millones de CPEs, número que parece más que suficiente para un ISP pequeño tal como los adheridos a la Fundación Guifi.Net, incluso durante muchísimo tiempo. Si a esos 4 millones le añadimos el cálculo de equipos que cada uno de los suscriptores puede tener además dentro de su propia red privada, la conectividad se dispara de forma exponencial.

La siguiente imagen es una representación de la arquitectura final y topología que se utilizará para la validación y pruebas que llevaremos a cabo más adelante.

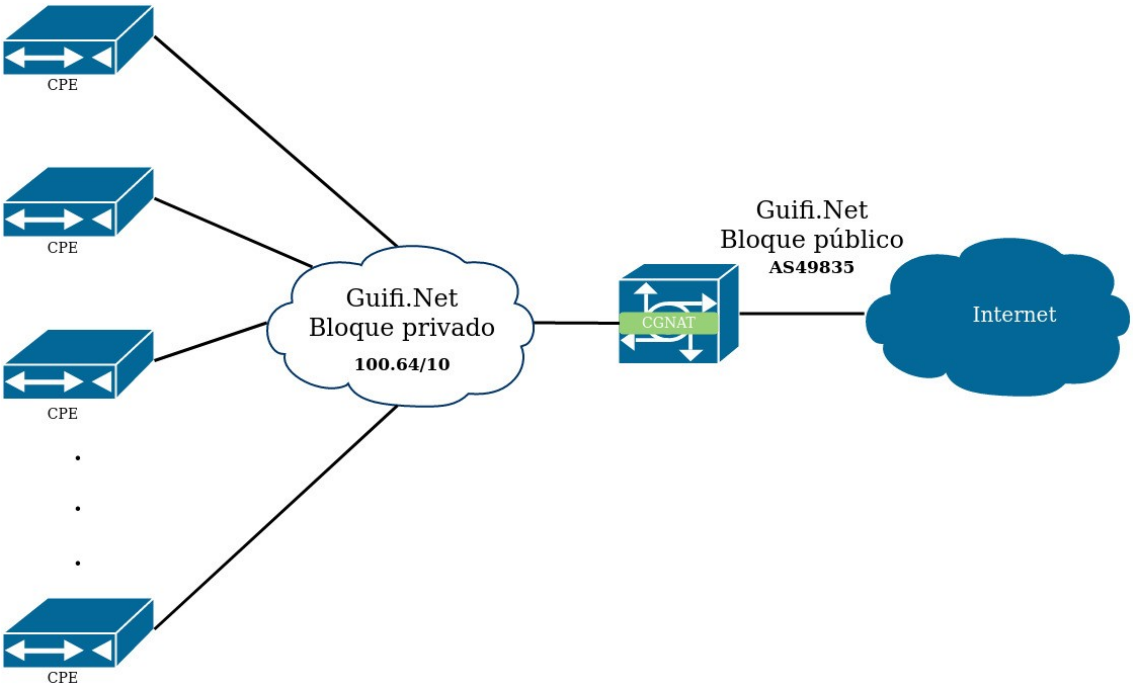


Ilustración 12: Arquitectura CGNAT para Guifi.Net

Como vemos, CGNAT a pesar de ser una solución temporal al agotamiento de direcciones, una prórroga necesaria a la implementación completa de IPv6, agrega muchísima flexibilidad y prolonga la vida de IPv4 de manera muy eficiente.

2.4 Transición a IPv6

Tal y como hemos visto, la utilización de CGNAT no se implementa para proporcionar una solución permanente. Como ya hemos estudiado, el agotamiento es un hecho y es necesaria la migración a IPv6, un protocolo diseñado con un espacio de direccionamiento virtualmente inagotable. Lo único que es necesario es el tiempo suficiente para migrar los direccionamientos, protocolos, aplicaciones y sistemas que no soportan todavía IPv6.

Sin duda alguna, existe un parque de sistemas embebidos, de control, IoT e industriales heredados gigantesco. En la mayoría de los casos esos sistemas dan servicio a infraestructuras que no permiten paradas de los mismos, con lo que han de aprovecharse reemplazos de *hardware* averiado o que ha cumplido su tiempo de vida para realizar la sustitución. Este hecho implica además la previsión, colaboración y actuación de numerosos entes privados, con lo que los tiempos esperados de transición se estiman muy largos.

2.4.1 Estrategias

Entre las diferentes posibilidades existentes, los ISP tienen la capacidad de elegir cuál de las siguientes estrategias es la más factible en cada caso particular. Mencionaremos para cada una de ellas algunas ventajas y desventajas.

Redes *dual-stack*

Se trata de redes en las que todos los nodos son capaces de procesar tanto tráfico IPv4 como IPv6. Las interfaces de red disponen de implementaciones de código que les permiten tratar ambas familias de direcciones, de manera que los dispositivos pueden recibir y entender contenido de la dos.

Se trata de la primera estrategia que se ideó y depende un espacio suficiente de direcciones IPv4.

Ventajas:

- Ambas familias de direcciones se procesan de forma independiente.
- Permite una transición parcial y progresiva.
- Como implementación nativa, no necesita de mecanismos de túnel en las redes de tránsito.
- Efectiva en coste y transición transparente. Tan pronto todos los servicios estuviesen ejecutándose sobre IPv6 se eliminaría IPv4 completamente.

Desventajas:

- Depende directamente de la disponibilidad de direcciones IPv4.
- Requiere soporte dual por parte de los fabricantes.
- Requiere recursos adicionales en el *hardware* que ha de soportarlo, en forma de memoria.
- Existen dispositivos que ejecutan IPv6 de forma nativa y no soportan IPv4.
- Los dispositivos de cierta edad no disponen de dobles pilas ni código con soporte IPv6.

6rd o *Rapid-Deploy* [31]

Se trata de introducir túneles IPv6 sobre redes IPv4 donde la infraestructura del ISP no soporte IPv6 todavía. De esta forma los clientes disponen de acceso

nativo IPv4 a la red mediante un segmento de direcciones asignado. En caso de que los clientes implementen IPv6, el ISP ha de crear túneles que encapsulen el tráfico IPv6 dentro de IPv4.

Ventajas:

- Extremadamente rápido de implementar por parte de los ISP.
- Los suscriptores tienen acceso a IPv6 de manera inmediata.
- No se dilata el despliegue de IPv6 globalmente.

Desventajas:

- No es una solución a largo plazo. Se están creando túneles para encapsular tráfico en los casos en los que la infraestructura del proveedor no soporta IPv6.
- Han de desplegarse dispositivos de terminación de *6rd*.
- Los CPE han de soportar esta tecnología y ser capaces de iniciar los túneles de encapsulación.
- Si el ISP utiliza NAT para IPv4, se heredan los mismos problemas en los diferentes protocolos que no lo soportan.

Carrier Grade NAT

Es el mecanismo objeto del presente estudio y puede considerarse como la versión a escala de ISP del NAT a nivel de suscriptor. Mientras que el NAT en la frontera del suscriptor solamente puede manejar un número restringido de traducciones, debido principalmente a las plataformas de hardware de los CPEs, el NAT implementado a nivel de ISP tiene capacidad para manejar millones de traducciones. Como ventaja añadida, CGNAT puede realizar también la traducción entre diferentes familias de direcciones IP, lo cual facilita mucho la tarea de implementación de IPv6 en el *core* de la red del ISP.

El caso de la Fundación Guifi.Net que nos ocupa tiene asignados 21 prefijos por el RIPE[11] para el AS49835[12]. Estos prefijos se asignan por el RIPE a cada uno de los diferentes operadores que conglomeran la Fundación Guifi.Net. Esos prefijos se distribuyen entre ellos tal y como se muestra en el siguiente enlace:

https://bgpview.io/asn/49835#prefixes-v4

Como vemos, este conjunto de operadores dispone de 20 prefijos IPv4 además de 1 prefijo IPv6 de longitud ::/32. Éstos ofrecen 5.888 direcciones IPv4 y $4{,}3 \times 10^9$ prefijos ::/64 para cada subred de suscriptor. Ambos espacios de

11 Réseaux IP Européens Network Coordination Centre. https://www.ripe.net/

12 Un AS (Sistema Autónomo) es una colección de prefijos IP administrados por uno o varios ISP en nombre de un ente administrativo único.

direcciones han de ser segmentados y repartidos entre los diferentes operadores de la Fundación Guifi.Net. Dado el espacio tan restringido en el primero y la enorme capacidad que brinda el segundo, trataremos de ofrecer una perspectiva de transición a IPv6 para cada uno de los operadores en la que se aprovechen al máximo las capacidades del espacio de direcciones IPv6 asignado.

Con esa premisa, el mecanismo más adecuado parece ser implementar una red *dual-stack* para las conexiones de los suscriptores. Utilizando *dual-stack* podremos garantizar el funcionamiento de dispositivos que no soporten IPv6 dentro de las premisas de los suscriptores. De hecho, los CPE desplegados más antiguos que no soporten *dual-stack* podrán seguir funcionando con IPv4 hasta que se realice la sustitución de los mismos. En el caso de que si soporten IPv6, el proveedor habrá de apoyarse en la delegación de prefijos [32] de forma que los CPEs y equipos de los suscriptores se auto configuren correctamente al iniciar sus pilas de protocolo IPv6.

La siguiente figura es una muestra de la solución de transición a IPv6 utilizando *dual-stack*.

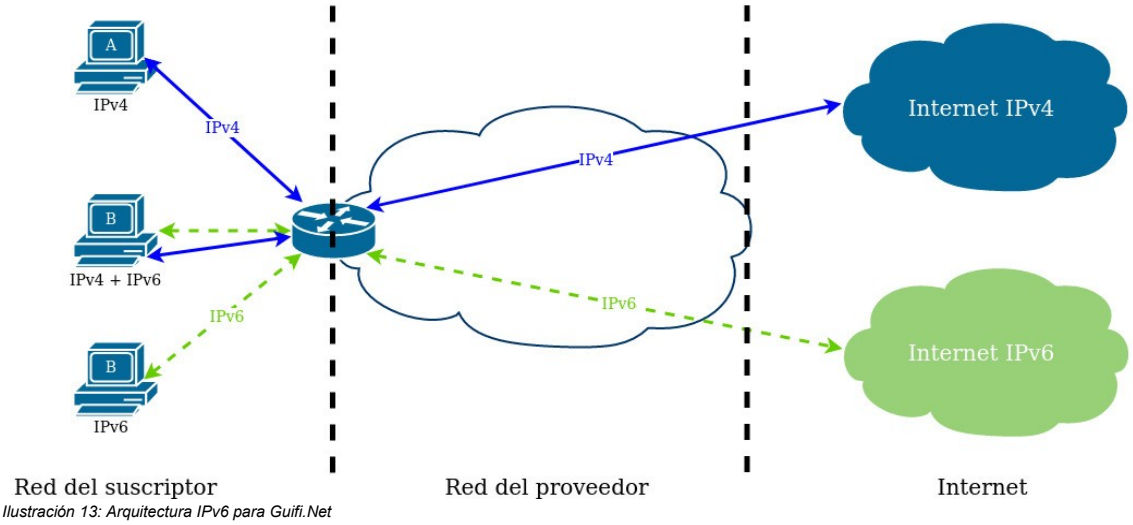


Ilustración 13: Arquitectura IPv6 para Guifi.Net

3. Validación y pruebas

Con el fin de demostrar la validez de CGNAT como mecanismo de conservación de direcciones, construiremos un entorno de laboratorio que simulará un subconjunto de un entorno real. Hemos de considerar que no podremos simular más que unos cuantos clientes, algunas secciones de lo que sería el *backbone* de alguno de los ISP que conforman Guifi.Net, y solamente podremos realizar pruebas a muy pequeña escala, pero que podrán sin duda extrapolarse de manera bastante fiable a un entorno real y de producción. Nuestro objetivo es comprobar si realmente CGNAT es la estrategia adecuada para extender el período de vida de IPv4 en tanto en cuanto los ISP que conglomeran Guifi.Net implementan IPv6 de manera definitiva.

Nuestro sistema tratará de comprobar la fiabilidad y resiliencia de una pasarela CGNAT, además de que trataremos de obtener la mayor cantidad de información disponible desde la propia pasarela. Es decir, recopilaremos información de registros de conexiones realizadas y podremos observar de primera mano si la información hasta ahora considerada en este estudio es realmente veraz.

Para la obtención de estos datos y la simulación parcial de un proveedor cualquiera, hemos de tener una idea real de las topologías de red típicas que los proveedores utilizan.

Podemos decir que el componente más importante en la arquitectura de un proveedor es el BNG[13]. Se trata de un dispositivo de red que interactúa con otros dispositivos de la red del proveedor para proporcionar acceso a los usuarios a los diferentes servicios y acceso a Internet.

Concretamente realiza las siguientes funciones:

- Conecta los diferentes CPE que recibirán acceso de banda ancha.
- Establece las sesiones de los usuarios utilizando protocolos como IPoE[14] o PPPoE[15].
- Interactúa con los servidores que realizan la autenticación de los usuarios y registran estos mismos accesos.
- Interactúa con los servidores que realizan la asignación de direcciones IP a los diferentes usuarios.

Por lo tanto, el BNG envía el tráfico del suscriptor al ISP, y la forma en la que éste se conecta en el ISP dependerá de la arquitectura de red en la que esté presente. Existen dos arquitecturas principales utilizadas por los proveedores:

13 Broadband Network Gateway.
14 Internet Protocol over Ethernet.
15 Point-to-Point Protocol over Ethernet.

- Proveedor de servicio de red.
- Proveedor de acceso de red.

Proveedor de servicio de red

En esta arquitectura, el proveedor ofrece acceso de banda ancha directamente al suscriptor. Es decir, el proveedor dispone de acceso de red al propio *backbone* de la red y control total sobre las conexiones y dispositivos intermedios implicados.

La ilustración 14 a continuación, muestra la topología de esta arquitectura:

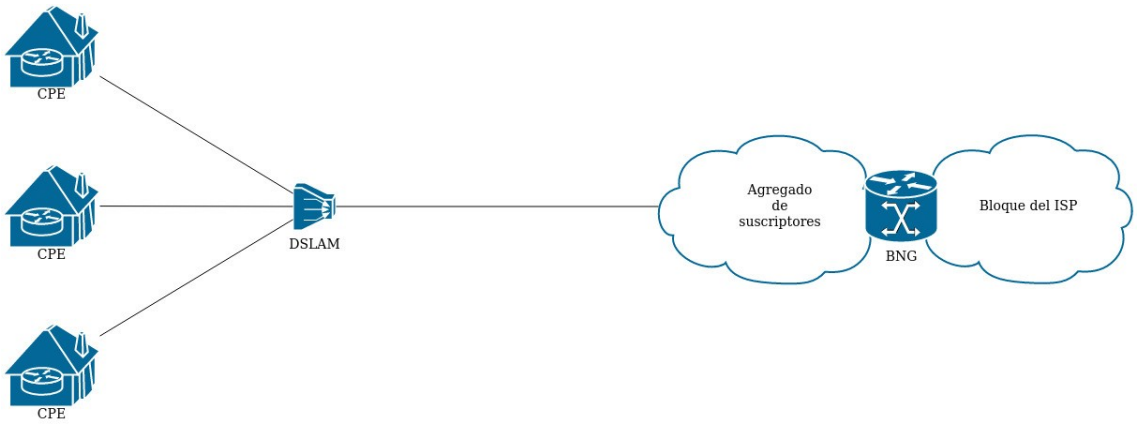


Ilustración 14: Arquitecura de proveedor de servicio

Proveedor de acceso de red

En esta arquitectura, sin embargo, el proveedor controla solamente los extremos de la red y provee del acceso a Internet, pero no es dueño ni controla en manera alguna el *backbone*. En este caso el proveedor ha de comunicarse con el ISP que si controla el servicio de red y el *backbone* de la misma. Para esta arquitectura son necesarios más componentes, agregando una capa adicional a la arquitectura de red subyacente, de forma que, mediante túneles L2TP [33] o L3VPNs [34] se puede proporcionar acceso de forma transparente a la red que el proveedor de servicio de red controla.

L2TP y L3VPN son ampliamente utilizados en los casos en los que es necesaria la abstracción sobre la infraestructura de red subyacente. Inicialmente ideados para generar esa independencia en redes DSL[16] [35] y más recientemente en arquitecturas tanto GPON[17] como FTTx[18] [36].

L2TP es el protocolo de túnel más utilizado en estos casos por su sencillez de despliegue y configuración. Aunque el estudio este protocolo se sale del propósito de este trabajo, es necesario mencionar que para implementarlo son

16 Digital Subscriber Line.
17 Gigabit Passive Optical Networks.
18 FTTx. Fiber to the Premises, Home, Building.

necesarios dos componentes adicionales que a continuación mencionamos y que tienen gran importancia a la hora de desplegar el entorno de laboratorio que estamos implementando:

- L2TP AC (Concentrador de acceso).
- L2TP NS (Servidor de red).

El primero recoge todas las conexiones de los CPEs y las envía al segundo, que se encarga de realizar la terminación lógica de las conexiones recogidas por el primero. Básicamente se trata de dos nodos que inician y establecen las conexiones PPPoE de los CPEs y las envían a la gestión del proveedor del servicio.

En la ilustración 15 a continuación vemos la topología de esta arquitectura:

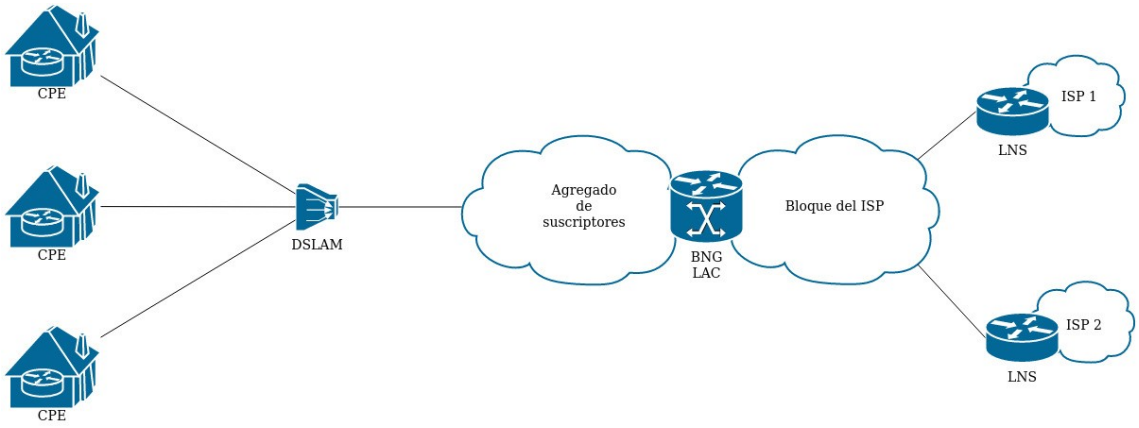


Ilustración 15: Arquitectura de proveedor de acceso

3.1 Provisión del entorno de laboratorio

Sobre un entorno virtual se han desplegado las dos arquitecturas descritas anteriormente. Para ello se han utilizado en total de 16 nodos puramente de red, simulando diferentes dispositivos del proveedor, junto a 6 nodos que realizarán las funciones tanto de clientes de los proveedores de servicio, como para gestión de la pasarela CGNAT que implementaremos.

La topología de referencia podemos verla en la ilustración 16 mostrada seguidamente:

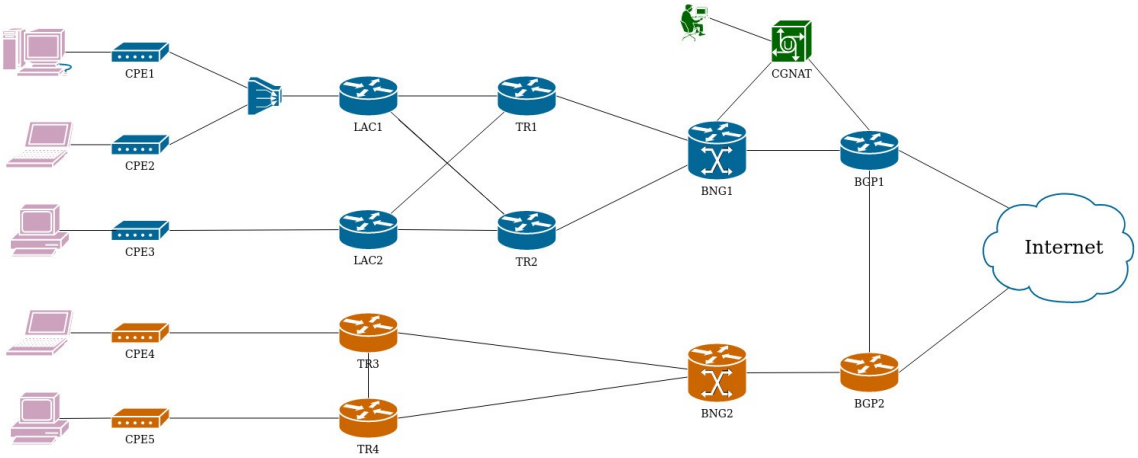


Ilustración 16: Topología de referencia

La tabla a continuación muestra un desglose de las tareas y funciones de cada uno de los nodos configurados en la topología de pruebas:

Equipo	Función	Proveedor
CPE1	Acceso de cliente	ISP A
CPE2	Acceso de cliente	ISP A
CPE3	Acceso de cliente	ISP A
CPE4	Acceso de cliente	ISP B
CPE5	Acceso de cliente	ISP B
LAC1	Concentrador de acceso	ISP A
LAC2	Concentrador de acceso	ISP A
TR1	Enrutador de tránsito	ISP A
TR2	Enrutador de tránsito	ISP A
TR3	Enrutador de tránsito	ISP B
TR4	Enrutador de tránsito	ISP B
BNG1	Pasarela de red	ISP A
BNG2	Pasarela de red	ISP B
BGP1	Enrutador de borde	ISP A
BGP2	Enrutador de borde	ISP B
CGNAT	Pasarela CGNAT	Guifi.Net

Table 4: Roles de los dispositivos

Como podemos comprobar, se ha simplificado la topología para ser una mera representación de un proveedor de servicio real. Para la simplificación se han consolidado ciertos roles de dispositivos, así como se ha reducido el número de ellos. Obviamente la escala a la que se realizan estas pruebas es muy reducida. Tómense de ejemplo el número de *routers* de tránsito que tenemos dista mucho de la realidad.

Respecto a la simplificación y unificación de roles, se han eliminado los LNS[19] específicos, que en nuestro caso se han consolidado en los propios BNG. Además, se han eliminado ciertas máquinas que también ejecutan servicios dedicados, como autenticación de usuarios y asignación de direcciones IP, y se han concentrado también en los BNG. Dado el reducido número de clientes que se presentan en este laboratorio, los BNG disponen de potencia y memoria suficientes para realizar todas las funciones que les estamos asignando.

Respecto a los proveedores representados, tanto ISP A como ISP B son nombres de proveedores ficticios que consideramos adheridos al paradigma que representa Guifi.Net.

Para que nuestro entorno sea lo más fiel a la realidad dentro de sus limitaciones, a cada uno de ellos le asignaremos un prefijo real del AS49835 perteneciente a Guifi.Net y que ésta ha delegado en cada uno de ellos. Concretamente configuraremos el prefijo IPv4 5.10.200.24/24 a ISP A y 91.97.120.200/22 a ISP B. Además, dispondremos del prefijo IPv6 2a00:1508::/32 para delegar en cada uno de ellos.

Consideraremos también que ISP A será del tipo Proveedor de acceso de red comentado anteriormente, mientras que ISP B será del tipo Proveedor de servicio de red. Esto implicará que ISP A habrá de establecer sus sesiones de abonados mediante asignación PPPoE a través de túneles L2TP configurados sobre la infraestructura subyacente. Por ejemplo, ISP A no tiene ningún control sobre los *routers* de tránsito sobre los que implementa su arquitectura. Por otro lado, ISP B si tiene acceso a los dispositivos de tránsito, lo que en cierta manera le facilita enormemente futuras migraciones o futuras mejoras.

3.2 Despliegue de la topología de pruebas

La siguiente tabla muestra un detalle de las configuraciones de *hardware* y *software* que se han utilizado para desplegar los equipos de laboratorio:

19 L2TP Network Server.

Equipo	Sistema Operativo	RAM
PC1	Ubuntu Linux x64	1GB
PC2	Windows 7	2GB
PC3	Ubuntu Linux x64	1GB
PC4	Ubuntu Linux x64	1GB
PC5	Windows 7	2GB
Mgmt1	Windows 7	2GB
CPE1	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	128MB
CPE2	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	128MB
CPE3	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	128MB
CPE4	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	128MB
CPE5	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	128MB
LAC1	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	128MB
LAC2	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	128MB
TR1	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	128MB
TR2	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	128MB
TR3	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	128MB
TR4	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	128MB
BNG1	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	512MB
BNG2	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	512MB
BGP1	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	256MB
BGP2	i86bi-linux-l3-adventerprisek9-15.2.4M1.bin	256MB
CGNAT	RouterOS 6.43.3	2GB

Table 5: Características de los dispositivos

Anexo a estos equipos que conforman en sí la topología de pruebas, el equipo de gestión de la pasarela CGNAT es el mismo que hospeda el laboratorio y que ejecuta las instancias de cada uno de los dispositivos que se ejecutan. En este caso se trata de un Debian[20] Linux x64 con 16GB de RAM y 8 CPUs ejecutando KVM[21] como hipervisor para la ejecución de los diferentes equipos.

Las siguientes ilustraciones nos muestran tanto las topologías físicas como lógicas del laboratorio desplegado. Éstas nos servirán como referencia a la hora de referenciar las interfaces de cada uno de los equipos y las direcciones IP durante el desarrollo de las pruebas de validación.

20 https://www.debian.org
21 Kernel Virtual Machine. Tecnología de virtualización para sistemas operativos Linux x86. https://www.linux-kvm.org

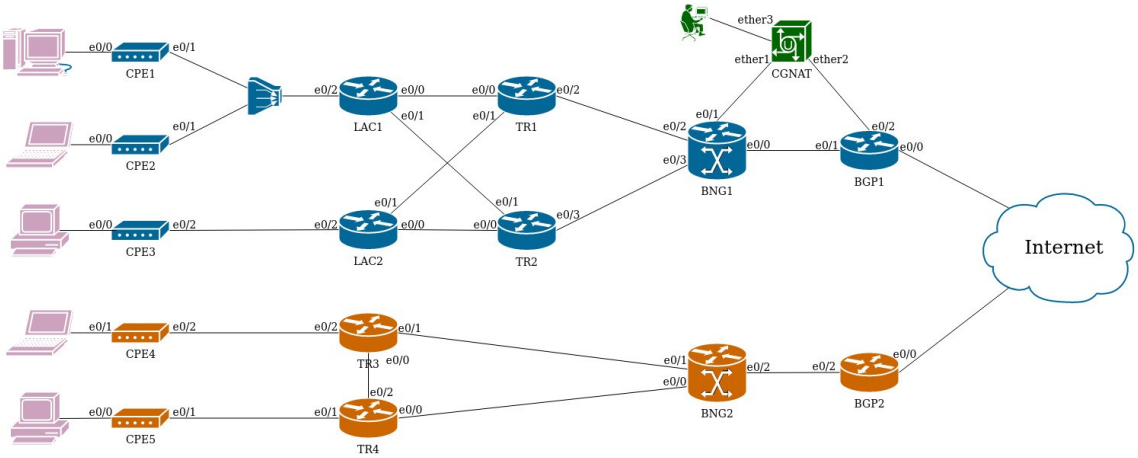


Ilustración 17: Topología física

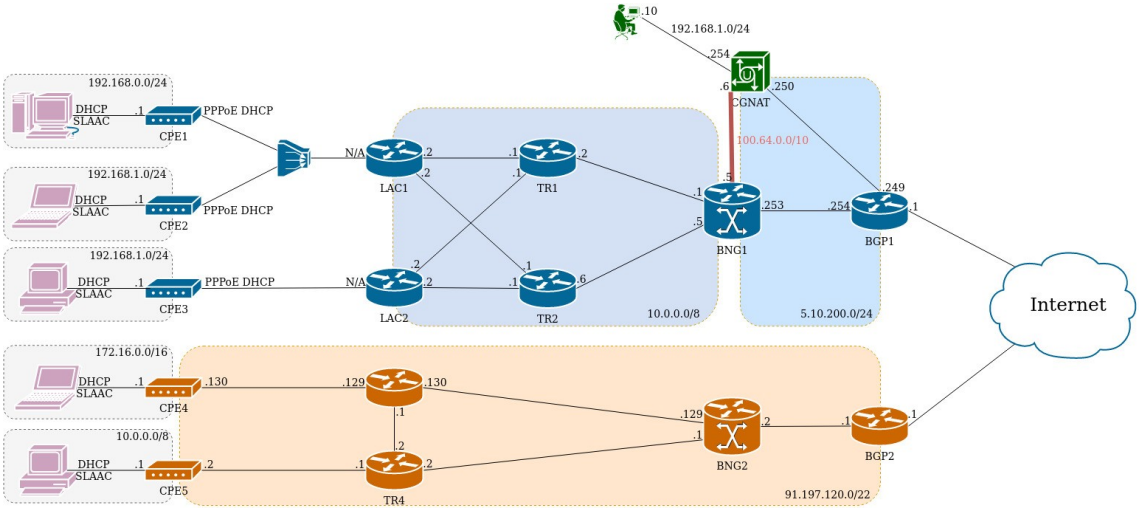


Ilustración 18: Topología lógica

En la ilustración 18 que precede, cabe destacar que el enlace de red entre los equipos BNG1 y CGNAT, será el que utilice el espacio de direcciones compartido que se indica en el RFC6598, 100.64.0.0/10. Desde este rango se asignarán las direcciones IP a los abonados de ISP A.

Recordemos que el objetivo de este estudio es comprobar la viabilidad de implementar CGNAT en Guifi.Net. Por lo tanto, debemos comprimir al máximo posible el espacio de direcciones asignado, mediante la estrategia NAT444. Para ello los CPE de los suscriptores quedarán sin modificar, y se implementará la segunda capa de NAT en el equipo CGNAT. Para ello, en lugar de asignar a las interfaces PPPoE de los CPE directamente direcciones del rango 5.10.200.0/24 tal y como se realiza en producción actualmente, pasaremos a asignar direcciones del rango compartido 100.64.0.0/10. De este modo, múltiples suscriptores utilizarán la misma dirección del rango 5.10.200.0/24 de forma pública, en lugar de una dirección por cada uno de ellos.

Hemos de observar un detalle de la topología de referencia mostrada en la ilustración 16 para el laboratorio, y es que no hay equipo CGNAT para el ISP ISP B (representado en naranja). Se ha omitido esta parte en las pruebas ya que no se trataría más que de duplicar el estudio que se va a realizar con ISP A. Con el fin de no extender en demasía el presente trabajo, se omite esta parte y se incluye solamente el estudio de CGNAT en ISP A por utilizar éste una arquitectura de Proveedor de acceso de red, que se considera un poco más desafiante al no disponerse de acceso a la red de capa 3 subyacente, tal y como pudimos ver anteriormente.

3.2.1 Consideraciones

Respecto a la configuración de la réplica virtual de ISP A, y teniendo en cuenta la simplificación a la que se la ha sometido para poder virtualizarla, hemos de considerar los siguientes aspectos:

- Los CPEs se han configurado tal y como lo estaría uno real, es decir, una interfaz interna, en un rango de direcciones privado del RFC1918 y sirviendo direcciones de manera automática a los dispositivos del suscriptor que se conecten a ellos. Además, están configurados de manera que la interfaz externa negocia automáticamente su dirección IP pública mediante PPPoE. De esta forma no es necesario realizar configuración específica en los mismos. Podríamos decir que son *plug-and-play*.

- Los dispositivos LAC1 y LAC2 son los que recogen las conexiones de los CPEs de los suscriptores y actúan como extremo de túnel L2TP que termina en el LNS del proveedor. Es decir, recogen el tráfico del suscriptor, lo encapsulan y envían al LNS correspondiente que des-encapsula ese tráfico para su tratamiento y reenvío. Como ya comentamos la función de LNS en nuestro entorno virtual se ha consolidado en los BNG.

- Los dispositivos TR#, son simplemente el *backbone* de la red del proveedor y proporcionan la capa de transporte necesaria para que los servicios del proveedor de acceso puedan ser entregados. En nuestro entorno simulan los nodos intermedios existentes en una topología real.

- Los dispositivos BNG, que unifican las funciones propias de BNG junto a las de LNS, servidor de autenticación y DHCP [37]. En lugar de desplegar más máquinas aumentando la complejidad innecesariamente, éstas funciones pasan a servirse en el mismo equipo. De esta forma, tenemos que el BNG por un lado termina el túnel L2TP iniciado en el LAC. Al mismo tiempo autentica las conexiones de los usuarios para establecer las sesiones PPPoE, y asigna direcciones IP a los CPEs dentro de los rangos asignados.

- Los dispositivos BGP, realizan la tarea de enrutadores de borde funcionando bajo el AS49835, que es el que tiene asignado la Fundación Guifi.Net. Estos *routers* se encargan de proporcionar el enrutamiento

real hacia y desde Internet para los operadores que están adheridos a Guifi.Net. Éstos en un entorno real, disponen de varias conexiones con otros proveedores y en puntos de acceso neutro, de forma que son capaces de anunciar sus *routers* BGP vecinos los prefijos que tienen asignados. Para este laboratorio y en este sentido se ha definido un *peering* BGP con un tercer *router* no representado en los diagramas, pero que será quien finalmente provea de acceso a Internet real, a todos los equipos virtualizados. Éste *peer* está definido como 20.0.0.4/32 en las configuraciones correspondientes de BGP1 y BGP2.

3.2.2 *Router* CGNAT

Este estudio se centra en la implementación de CGNAT en Guifi.Net, de manera que a tal propósito se ha implementado un *router* Mikrotik[22] ejecutando RouterOS[23] 6.43.3 y al que se le han asignado 2 CPUs y 2 GB de RAM. Además, se ha conectado este *router* a sus vecinos, BNG1 y BGP1, con interfaces Gigabit. Se han considerado los siguientes aspectos a la hora de asignar esos recursos:

- La cantidad de sesiones que ha de mantener en su memoria. CGNAT va a exigir una alta demanda de memoria, ya que, aunque cada cliente realice pocas conexiones simultáneas, el gran volumen de clientes que puede tener un proveedor real puede disparar ese número ampliamente. La concurrencia que este equipo ha de soportar en un entorno real puede llegar a ser muy elevada.

- De la misma manera, el ritmo de establecimiento de conexiones es un dato importante a tener en cuenta, y ya que establecer cada una de ellas y aplicar la traducción correspondiente consume potencia de procesamiento, es necesario dotar de la suficiente cantidad de CPUs al dispositivo.

- En cuanto a las conexiones de red, se ha optado por conexiones Gigabit con los dispositivos adyacentes BNG1 y BGP1 para que pueda manejarse de manera eficiente el volumen de datos que puedan transferir los suscriptores por estos enlaces. Dado un entorno real como aquel al que se enfrentan los ISP, es posible llegar a necesitar interfaces con transferencias de 10GigabitEthernet. Tan sólo hemos de pensar que las conexiones de banda ancha que se despliegan hoy en día ofrecen capacidades de transferencia de más de 100Mbps a cada suscriptor.

Teniendo en cuenta que CGNAT ha de realizar traducción de direcciones para poder conservar direcciones IP públicas de ISP A, es necesario modificar las configuraciones de los dispositivos de agregación de ISP A, de forma que si antes BNG1 asignaba direcciones del rango 5.10.200.0/24 a los suscriptores, y este rango era anunciado en Internet mediante el dispositivo BGP1, ahora serán necesarias algunas modificaciones que describimos a continuación.

22 https://mikrotik.com/
23 https://mikrotik.com/software

- **BNG1** pasará de asignar direcciones del rango 5.10.200.0/24 a los suscriptores, a asignar direcciones del rango 100.64.0.0/10.

- **BNG1** pasará a enrutar el tráfico de los suscriptores de la interfaz **e0/0** a la interfaz **e0/1**. Este paso es necesario porque 100.64.0.0/10 es un rango privado no enrutable en Internet.

- **CGNAT** recogerá las conexiones de los suscriptores en el rango 100.64.0.0/10 por la interfaz **ether1**, realizará las traducciones especificadas según su configuración, y dirigirá los paquetes traducidos, ahora sí dentro del rango público 5.10.200.0/24 al *router* BGP1 a través de la interfaz **ether2**.

- **BGP1** habrá de cambiar también su esquema de enrutamiento para devolver el tráfico de respuesta que viaja desde Internet hacia los suscriptores, cambiando la interfaz de entrada **e0/1** por la interfaz **e0/2**.

Con las modificaciones descritas, hemos pasado de un modelo en el que los suscriptores disponían de direcciones IP públicas únicas para cada uno, a un modelo en el que varios suscriptores comparten direcciones IP públicas entre varios de ellos.

Dado que disponemos un rango 100.64.0.0/10 que contiene $4_1194.302$ direcciones IP y el dispositivo CGNAT va a realizar traducciones sobre el rango 5.10.200.0/24, con 254 direcciones IP, estaríamos tratando con una ratio de compresión o ahorro de 1/16.513 teórico. Es decir, por cada dirección IP pública podríamos conectar a 16.513 suscriptores. Obviamente esto no se cumple a rajatabla, ya que los propios dispositivos de enrutamiento de ISP A, necesitan algunas de esas direcciones para poder comunicarse con sus vecinos, configurar puertas de enlace y demás direcciones IP de gestión necesarias para el funcionamiento correcto de la red.

Esta ratio teórica no es recomendable utilizarla en entornos de producción ya que existen factores que impactan al número de suscriptores que han de compartir la misma dirección IP pública, y que describimos anteriormente. Además, si se produce indisponibilidad de una dirección IP pública muy concurrida, se impacta a un gran número de suscriptores a la vez, por lo que es importante balancear correctamente la ratio de compartición real con la disponibilidad de servicio que se haya acordado con los suscriptores en sus contratos de conexión.

3.2.3 Configuración del *router* CGNAT

Teniendo en cuenta que esta máquina se introduce para combatir el agotamiento de direcciones IPv4, nos referimos al RFC6598 para realizar la configuración de direccionamiento. La idea básica consiste en utilizar el rango 100.64.0.0/10 dentro de la red de ISP A y aplicarle NAT hacia el *router* de borde utilizando un subconjunto del rango público del que el proveedor dispone.

Como ya se ha estudiado, a esta estrategia de segmentación se la denomina NAT444, y es una ampliación del NAT44 que se realiza en los CPEs de los suscriptores. Por lo tanto, nos enfrentamos a 3 rangos diferentes de direcciones IP, tal y como se refleja en la ilustración 9.

Una configuración NAT con sobrecarga de puertos (PAT sobrecargado) en RouterOS, tiene el siguiente aspecto:

```
/ip firewall nat
add action=src-nat chain=srcnat out-interface=ether2 src-address=192.168.0.0/24 to-address=5.10.200.100
```

Ilustración 19: Configuración de NAT N:1 RouterOS

Donde:

- **action**. Es la dirección en la que se realizará la traducción. En este caso de origen hacia destino.
- **chain**. Es el alias de las diferentes reglas del *firewall* que se ejecutan bajo un mismo conjunto.
- **src-address**. Es el rango o la dirección IP interna.
- **to-address**. Es el rango o la dirección IP pública.
- **out-if**. Es la interfaz que conecta al *router* de borde del proveedor.

Ésta sería una configuración válida si se tratase de un CPE y éste dispusiera de una dirección IP pública.

El caso que estamos manejando es diferente, ya que no disponemos de una sola dirección IP pública, sino varias dentro del rango 5.10.200.0/24, con lo que la configuración ha de ajustarse a este hecho. Por lo tanto, nuestra configuración debe de tener un aspecto más parecido a la siguiente:

```
/ip firewall nat
add action=src-nat chain=srcnat out-interface=ether2 src-address=192.168.0.0/24 to-addresses=5.10.200.10-5.10.200.250
```

Ilustración 20: Configuración de NAT N:N RouterOS

La configuración anterior sería válida si nuestro *router* tan sólo necesitase distribuir las conexiones de manera equitativa entre cierto número de direcciones IP públicas. Este caso se da en numerosas organizaciones que disponen de rangos de direcciones públicos y realizan traducción de las direcciones internas a su organización sobre ese *pool* de direcciones para dar salida a Internet. En muchos casos estas organizaciones, además de realizar sobrecargas pueden también hacer traducciones 1 a 1 del tipo NAT estático

cómo se ha explicado en la tabla 1, de forma que pueden publicar servicios internos en Internet.

El caso de nuestro *router* CGNAT es particular, ya que no ha de realizar traducciones para un suscriptor o una organización, sino para un proveedor de servicios, con las implicaciones que ya hemos estudiado.

Por otra parte, es obligatorio que los ISP registren las traducciones que se realicen. Como ya vimos, existe el requisito legal de registrar todas las traducciones y conexiones. En la red de un ISP esto es un problema, ya que el volumen de registros que se generan llega a ser extremadamente elevado. El RFC7422 [38] ofrece una solución para reducir la cantidad de registros, facilitando por tanto la traza de los mismos en casos de abusos o delitos informáticos.

CGNAT Determinista

Según el RFC7422, en lugar de registrar cada conexión, habrán de realizarse traducciones deterministas para las direcciones privadas de los suscriptores. Concretamente para las direcciones del rango 100.64.0.0/10 y sus traducciones al rango de direcciones públicas del proveedor, junto a los puertos de origen de estas conexiones, es decir las tuplas IP:puerto dentro del agregado del proveedor. Es decir, dispondremos de conjuntos prefijados de asignaciones, en lugar de dejar que sea el algoritmo del *router* quien las realice. De esta manera, podemos predecir qué asignaciones se realizarán, independientemente del algoritmo del *router*, facilitando y reduciendo el tamaño del registro de conexiones que se generará.

Así, por ejemplo, la asignación determinista podría tomar la siguiente forma:

IP interna	IP externa : rango de puertos
100.64.100.100	5.10.200.100 : 5001-6000
100.64.100.101	5.10.200.100 : 6001-7000
100.64.100.102	5.10.200.100 : 7001-8000
100.64.100.103	5.10.200.100 : 8001-9000
100.64.100.104	5.10.200.100 : 8001-9000

Table 6: Asignación determinista de ejemplo

Como vemos, estamos asignando traducciones a 5 direcciones IP compartidas, sobre 1 sola dirección IP pública. A cada dirección le estamos asignando 1.000 puertos comenzando en el 5.001.

Para nuestra implementación, si no basamos en el RFC6335 [39], en el que la IANA realiza recomendaciones sobre la asignación de puertos para el establecimiento de conexiones TCP y UDP, dispondríamos tan sólo de 15.361 puertos para realizar las asignaciones. Dado el volumen de suscriptores de un

ISP actual, junto al número de conexiones realizadas por cada dispositivo, cada vez mayor y con mayor número de aplicaciones conectadas, ese número parece claramente insuficiente.

De forma conservadora, asignaremos 200 direcciones del rango 5.10.200.0/24 para el pool de CGNAT, lo que nos proporciona $9_1625.400$ puertos en total. De forma igualmente conservadora asignaremos un límite de 2.000 puertos por cada dirección de suscriptor. De esta manera por un lado, estaremos proporcionando un número más que suficiente para el número de conexiones simultáneas que suele mantener un equipo conectado a Internet como vimos en el apartado 2.1.1. Y por otro lado, evitamos los posibles abusos que se puedan producir desde las direcciones de los suscriptores.

Nuestra tabla de asignaciones determinista tendrá el siguiente aspecto:

IP interna	IP externa : rango de puertos
100.64.10.10	5.10.200.11 : 2001-4000
100.64.10.11	5.10.200.11 : 4001-6000
...	...
100.64.31.98	5.10.200.245 : 44001-46000
100.64.31.99	5.10.200.245 : 46001-48000

Table 7: Asignación determinista para el laboratorio

Una representación gráfica más significativa del criterio de asignación de recursos asignado es la siguiente:

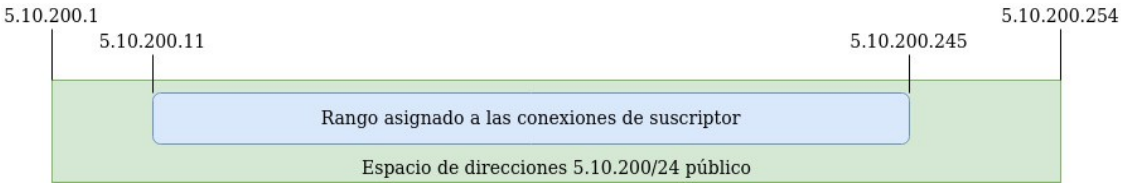


Ilustración 21: Direcciones asignadas

Observemos que se han excluido de las asignaciones para suscriptores 19 direcciones IP, 10 en el margen inferior y 9 en el superior. Consideramos éstas como reservas necesarias para la administración y gestión de la propia infraestructura del ISP, y que además le da margen para migraciones, cambios de configuración y adición de dispositivos en caso necesario por su parte.

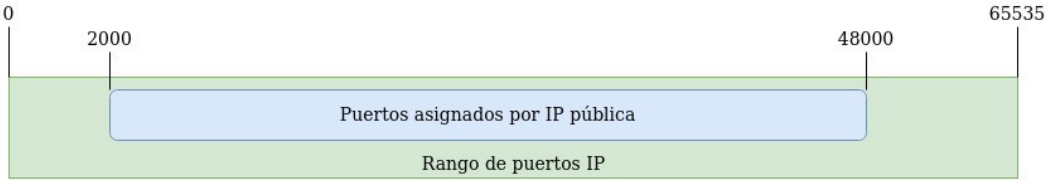


Ilustración 22: Puertos asignados

En el caso de la asignación de puertos, hemos omitido la recomendación de la IANA para aprovechar de manera más efectiva cada una de las direcciones IP que se asignarán acorde a la ilustración 21.

Estas asignaciones se traducen en que el espacio de direcciones público se comparte con una ratio de 23 a 1. Es decir, cada dirección IP pública puede ser utilizada por 23 suscriptores de manera simultánea. Como vemos la ratio real con el que tratamos dista mucho de la ratio teórica que calculamos anteriormente y que era de 16.513 a 1.

3.3 Análisis de funcionamiento

Utilizando el equipo MGMT1, que es la estación de trabajo que gestiona el *router* CGNAT, se podrán validar las configuraciones que se hagan sobre él. De esta manera podremos obtener estadísticas de uso de RAM, CPU y ancho de banda, así como contenidos completos de las tablas de traducción que se vayan generando a medida que los equipos de los suscriptores generan tráfico hacia Internet.

3.3.1 Configuración de NAT

El *router* CGNAT ha de encargarse del trabajo de traducir direcciones del rango privado 100.64.0.0/10 y mapearlas a direcciones del rango público 5.10.200.0/24 de acuerdo a las reglas deterministas que expusimos en el último punto del apartado anterior. Por cada grupo de 2000 puertos que se asignan a una dirección de abonado, son necesarias 2 líneas de configuración de NAT en el equipo, una para TCP y otra para UDP.

La configuración determinista para todo el rango que tenemos disponible va a generar más de 468 líneas de configuración encadenadas en el *router* CGNAT por cada dirección del rango público, con lo que sólo mostraremos a continuación un grupo extermadamente reducido de ellas, tan sólo para comprender el patrón de configuración que ha de seguirse.

La traducción determinista para las 2 primeras direcciones privadas, junto con sus rangos de puertos asignados para la primera dirección IP pública que utilizaremos, se convierte en las siguientes líneas de configuración:

```
/ip firewall nat
add action=jump chain=srcnat jump-target=ISPA src-address=100.64.10.50/31
add action=jump chain=ISPA jump-target=ISPA-0 src-address=100.64.10.50
add action=jump chain=ISPA jump-target=ISPA-1 src-address=100.64.10.51
add action=src-nat chain=ISPA-0 protocol=tcp src-address=100.64.10.50 to-addresses=5.10.200.11 to-ports=2001-4000
add action=src-nat chain=ISPA-0 protocol=udp src-address=100.64.10.50 to-addresses=5.10.200.11to-ports=2001-4000
add action=src-nat chain=ISPA-1 protocol=tcp src-address=100.64.10.51 to-addresses=5.10.200.11 to-ports=4001-6000
add action=src-nat chain=ISPA-1 protocol=udp src-address=100.64.10.51 to-addresses=5.10.200.11 to-ports=4001-6000
```

Ilustración 23: Configuración NAT determinista

Como vemos, cada dirección interna, es mapeada contra la primera dirección externa, utilizando el bloque de puertos que le corresponde de acuerdo a la asignación que realizamos anteriormente. Además, es necesaria una línea para UDP y otra para TCP. La evaluación sería como sigue suponiendo un origen de 100.64.10.50:

1. Para las direcciones en el rango 100.64.10.50/31, estas son 100.64.10.50 y 100.64.10.51, evaluar la cadena ISP A.

2. En este punto se evalúan las líneas 2 y 3 de la configuración y se encuentra una coincidencia en la línea 2, luego hay que evaluar la cadena ISPA-0.
3. Pasamos a evaluar las líneas 4 y 5 de la configuración. Según sea el protocolo de origen utilizado, se realizará la traducción con la instrucción que corresponda.

4. El puerto asignado de cada grupo (2.001-4.000) será elegido consecutivamente.

Esta configuración ha de implementarse en el *router* CGNAT para que utilice completamente el pool de direcciones públicas que le hemos asignado (5.10.200.11 a 5.10.200.245), lo cual viene a extender enormemente la configuración del mismo. Puede encontrarse una copia completa de la configuración en los Anexos de este documento.

De la misma manera en los Anexos se encuentra un *script* realizado en Python[24] para facilitar la generación de las configuraciones necesarias de acuerdo a las reglas que hemos expuesto en este punto.

El *script* incluye, además de la generación de las reglas expuestas para la traducción y mapeo de puertos TCP y UDP, una regla adicional que se encargará de traducir el resto de protocolos IP que no basan su funcionamiento en el concepto de puertos, tales como ICMP, GRE [40], ESP y demás. Puede encontrarse una lista completa de los números de protocolo IP en el siguiente enlace:

https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

24 https://www.python.org/

3.3.2 Traducciones

Una vez insertadas las reglas de traducción, podemos pasar a generar tráfico desde los equipos de los suscriptores. Se ha procedido a generar tráfico de manera aleatoria hacia Internet para comprobar que las traducciones funcionan de manera esperada y el acceso a Internet se produce de forma correcta.

Utilizando el equipo MGMT1 podemos comprobar como las gráficas de tráfico de las interfaces **ether1** y **ether2** del *router* CGNAT se disparan:

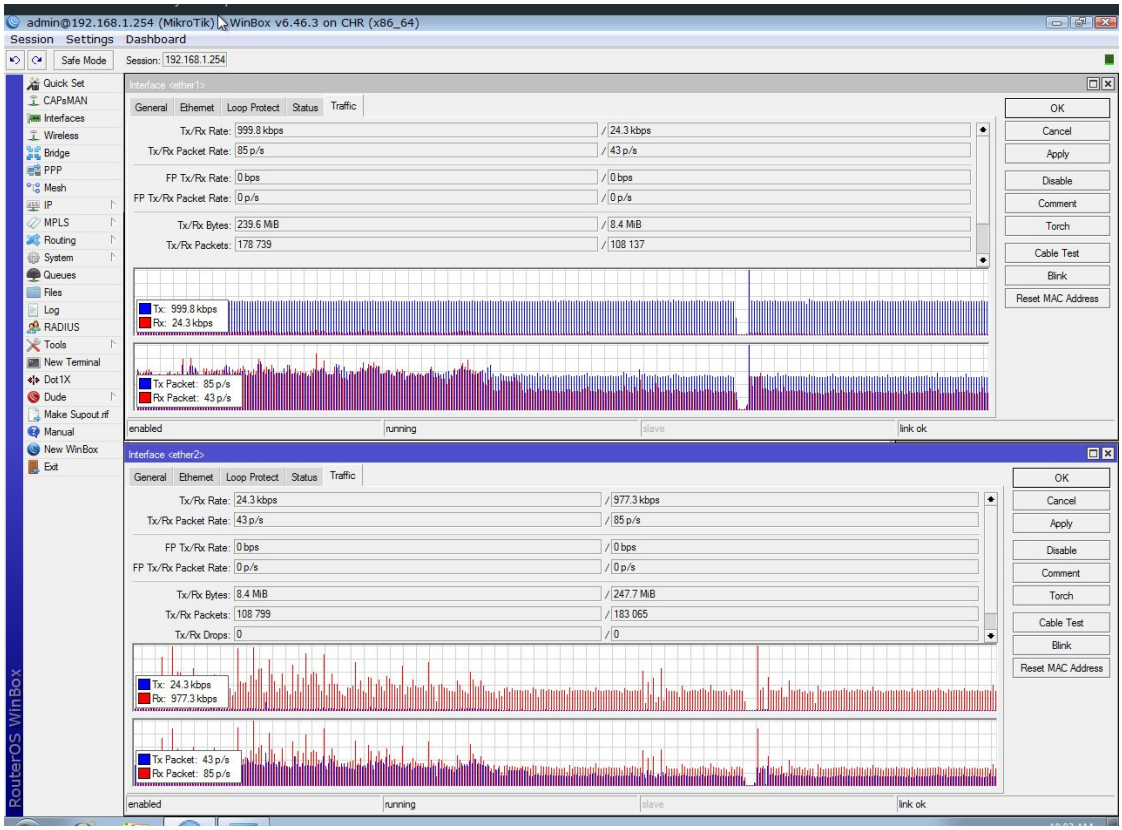


Ilustración 24: CGNAT: Tráfico de interfaces

También podemos comprobar que el conjunto de reglas que hemos configurado produce los resultados esperados, reflejándose también las gráficas en tiempo real de las traducciones realizadas por la máquina:

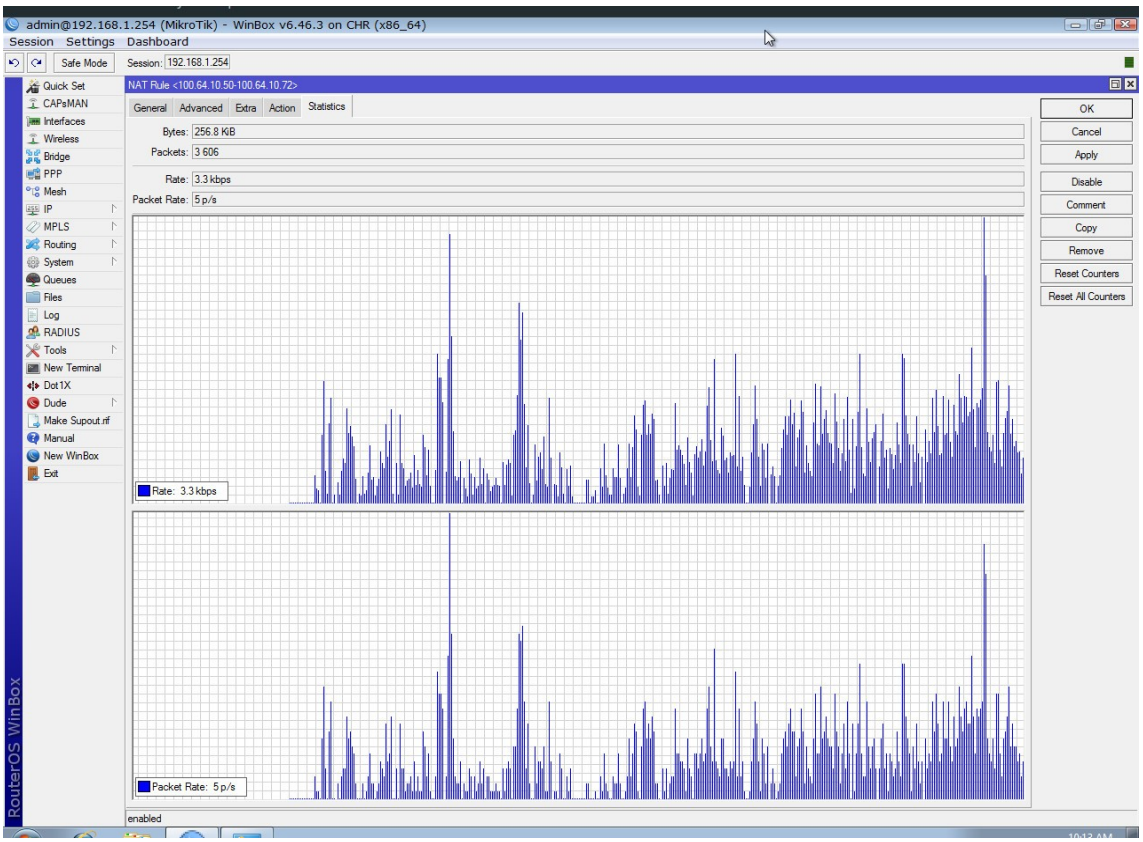
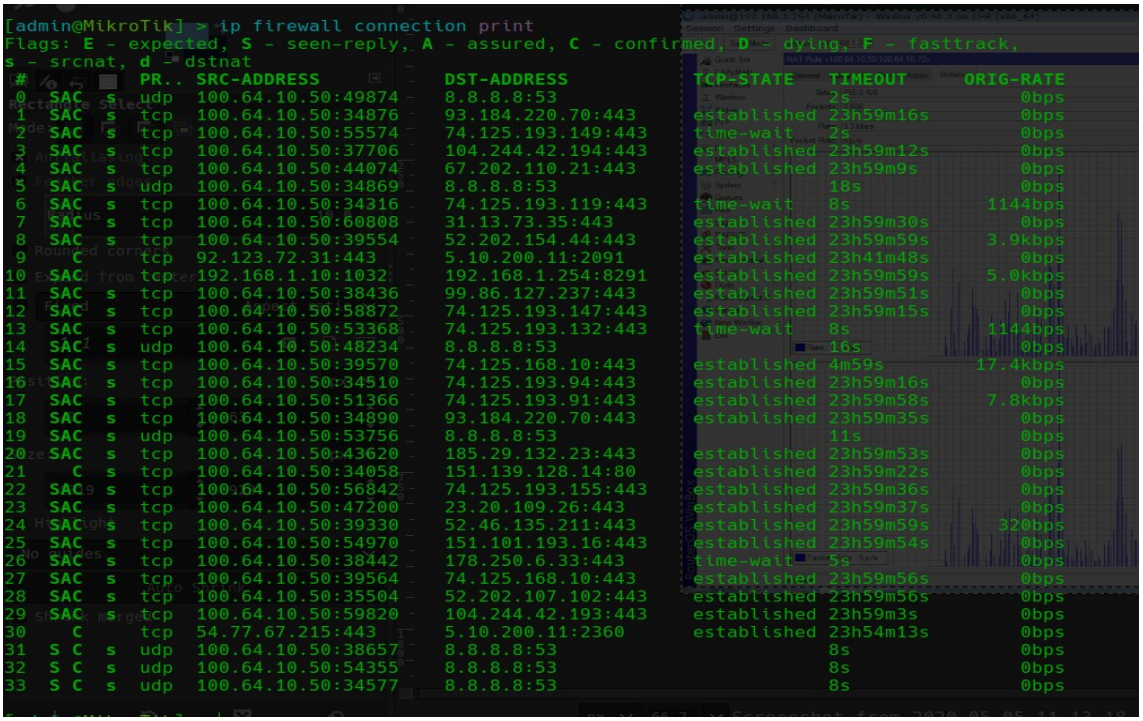


Ilustración 25: CGNAT: Tráfico de NAT

La tabla de traducciones de CGNAT refleja todas las conexiones de las máquinas cliente que han sido traducidas:



```
[admin@MikroTik] > ip firewall connection print
Flags: E - expected, S - seen-reply, A - assured, C - confirmed, D - dying, F - fasttrack,
s - srcnat, d - dstnat
#    PR.. SRC-ADDRESS          DST-ADDRESS          TCP-STATE    TIMEOUT     ORIG-RATE
0    SAC  s udp  100.64.10.50:49874  8.8.8.8:53                  2s              0bps
1    SAC  s tcp  100.64.10.50:34876  93.184.220.70:443  established 23h59m16s       0bps
2    SAC  s tcp  100.64.10.50:55574  74.125.193.149:443 time-wait   2s              0bps
3    SAC  s tcp  100.64.10.50:37706  104.244.42.194:443  established 23h59m12s       0bps
4    SAC  s tcp  100.64.10.50:44074  67.202.110.21:443   established 23h59m9s        0bps
5    SAC  s udp  100.64.10.50:34869  8.8.8.8:53                  18s             0bps
6    SAC  s tcp  100.64.10.50:34316  74.125.193.119:443 time-wait   8s          1144bps
7    SAC  s tcp  100.64.10.50:60808  31.13.73.35:443     established 23h59m30s       0bps
8    SAC  s tcp  100.64.10.50:39554  52.202.154.44:443   established 23h59m59s     3.9kbps
9      C  s tcp  92.123.72.31:443    5.10.200.11:2091    established 23h41m48s       0bps
10   SAC  s tcp  192.168.1.10:1032   192.168.1.254:8291  established 23h59m59s     5.0kbps
11   SAC  s tcp  100.64.10.50:38436  99.86.127.237:443   established 23h59m51s       0bps
12   SAC  s tcp  100.64.10.50:58872  74.125.193.147:443  established 23h59m15s       0bps
13   SAC  s tcp  100.64.10.50:53368  74.125.193.132:443  time-wait   8s          1144bps
14   SAC  s udp  100.64.10.50:48234  8.8.8.8:53                  16s             0bps
15   SAC  s tcp  100.64.10.50:39570  74.125.168.10:443   established 4m59s       17.4kbps
16   SAC  s tcp  100.64.10.50:34510  74.125.193.94:443   established 23h59m16s       0bps
17   SAC  s tcp  100.64.10.50:51366  74.125.193.91:443   established 23h59m58s     7.8kbps
18   SAC  s tcp  100.64.10.50:34890  93.184.220.70:443   established 23h59m35s       0bps
19   SAC  s udp  100.64.10.50:53756  8.8.8.8:53                  11s             0bps
20   SAC  s tcp  100.64.10.50:43620  185.29.132.23:443   established 23h59m53s       0bps
21     C  s tcp  100.64.10.50:34058  151.139.128.14:80   established 23h59m22s       0bps
22   SAC  s tcp  100.64.10.50:56842  74.125.193.155:443  established 23h59m36s       0bps
23   SAC  s tcp  100.64.10.50:47200  23.20.109.26:443    established 23h59m37s       0bps
24   SAC  s tcp  100.64.10.50:39330  52.46.135.211:443   established 23h59m59s     320bps
25   SAC  s tcp  100.64.10.50:54970  151.101.193.16:443  established 23h59m54s       0bps
26   SAC  s tcp  100.64.10.50:38442  178.250.6.33:443    time-wait   5s              0bps
27   SAC  s tcp  100.64.10.50:39564  74.125.168.10:443   established 23h59m56s       0bps
28   SAC  s tcp  100.64.10.50:35504  52.202.107.102:443  established 23h59m56s       0bps
29   SAC  s tcp  100.64.10.50:59820  104.244.42.193:443  established 23h59m3s        0bps
30     C    tcp  54.77.67.215:443   5.10.200.11:2360    established 23h54m13s       0bps
31   S C  s udp  100.64.10.50:38657  8.8.8.8:53                  8s              0bps
32   S C  s udp  100.64.10.50:54355  8.8.8.8:53                  8s              0bps
33   S C  s udp  100.64.10.50:34577  8.8.8.8:53                  8s              0bps
```

Ilustración 26: CGNAT: Tabla de conexiones

En la ilustración 26 anterior solamente se refleja un resumen de la información que realmente almacena el *router* sobre las conexiones. La información que nos proporciona es:

- **Flags**. Que permiten el seguimiento de las conexiones.
- **TYPE**. En este caso **s**, que indica que es NAT originado en la parte privada de la red.
- **PROTO**. El tipo de protocolo de transporte que se ha traducido.
- **SRC-ADDRESS**. En este caso, siempre serán direcciones del rango 100.64.0.0/10 que estén asignadas a CPEs de suscriptores.
- **DST-ADDRESS**. Las direcciones IP originales de los equipos en Internet a los que se está accediendo.
- **TCP-STATE**. Sólo se registra en caso de este tipo de conexión de capa 4. En UDP no tiene sentido y no aplica.
- **TIMEOUT**. El tiempo que esa asignación va a mantenerse en la memoria del *router*.
- **ORIG-RATE**. El ancho de banda en bits por segundo y sus múltiplos que se están utilizando para esa entada.

En la ilustración 27 justo debajo, observamos información mucho más detallada:

Ilustración 27: CGNAT: Tabla de traducciones

Con la información de la ilustración 27 podemos observar que las traducciones se están realizando acorde a las reglas que implementamos.

Como podemos apreciar, los campos **reply-dst-address**, que indican a qué tupla dirección:puerto ha de destinarse el tráfico de regreso, siempre incluyen la dirección IP pública compartida 5.10.200.11 y un número de puerto entre 2.001 y 4.000 que, en nuestra tabla determinista, corresponden a la IP del bloque agregado 100.64.10.50 y que es la dirección del CPE que ha originado ese tráfico. Por supuesto, el CPE al que corresponde esta dirección IP en el momento en el que se registra, puede saberse desde la tabla de asignaciones que se produce en el LNS. Recordemos que, en nuestra topología el LNS se incluye en el BNG correspondiente a este ISP.

Un detalle que no debemos pasar por alto es que estamos reduciendo mucho la cantidad de conexiones que puede establecer cada suscriptor. Es decir, si normalmente cada uno de ellos dispone de una IP pública, puede en teoría mantener 48.127 conexiones diferentes tanto en TCP como en UDP, ya que dispone de todo el rango de puertos en su dirección IP. Ahora bien, hemos recortado a 4.000 el número de conexiones combinadas TCP y UDP que va a poder establecer. Es necesario recortar el tiempo que las conexiones establecidas se mantienen en la memoria del *router* que hace la traducción, de otra forma, los clientes se quedarían sin puertos disponibles dentro de su *pool* rápidamente.

Como ya observamos en la ilustración 10, una sesión estándar de acceso a Internet de un equipo mantiene del orden de entre 60 y 100 conexiones, con picos de 450 a 600 conexiones. Como se puede deducir, mantener un *timeout* de sesión a 24 horas, tal y como viene configurado por defecto el sistema

operativo RouterOS, no es sostenible. Se hace por tanto necesario recortar ese valor por defecto:

```
/ip firewall connection tracking
set tcp-established-timeout=10m
```

Ilustración 28: Reducción del timeout por defecto RouterOS

La cuenta atrás del *timeout* se reinicia siempre que haya actividad o tráfico de red por esa conexión (tupla dirección:puerto), por lo tanto, no existe peligro de que la sesión se cierre cuando todavía se está utilizando. El *timeout* consecuentemente es de ayuda para liberar conexiones que ya no se están utilizando, liberando memoria del router por un lado, y facilitando a los suscriptores establecer nuevas conexiones.

Cabe destacar que las tablas de conexiones y traducciones que se incluyen en este documento están recortadas con doble propósito. Primero, reducir el tamaño de las ilustraciones, que con miles de traducciones sería imposible de incluir aquí. Segundo, incluir solamente la información mínima relevante para el propósito de este estudio.

3.3.3 Registro de conexiones

Ya hemos visto anteriormente que es necesario registrar las conexiones que se originan en nuestro ISP. Debido a posibles requerimientos judiciales o investigaciones por abusos o delitos informáticos, hay que guardar registros de conexión de cada uno de los suscriptores. La traducción determinista que hemos implementado facilita mucho esta labor, ya que nos proporciona de antemano la información que necesitamos para hacer seguimiento a la dirección original del suscriptor que estuvo involucrado en las conexiones maliciosas.

Una tabla de *log* de un servidor *web* Apache®[25] típico contiene la siguiente información, de acuerdo al *Common Log Format*[26]:

- IP del cliente remoto que se conecta.
- ID del usuario. Según lo transmite la autenticación HTTP[27].
- Marca de tiempo de la conexión.
- Petición. En formato HTML[28].
- Código de estado.
- Tamaño del objeto solicitado.

25 https://httpd.apache.org/
26 Formato de registro de conexiones adoptado por Apache®. https://httpd.apache.org/docs/1.3/logs.html
27 Hypertext Transfer Protocol.
28 Hypertext Markup Language.

Que proporcionan el siguiente aspecto:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
```

Ilustración 29: Entrada de registro de un servidor Apache®

No cabe en el ámbito de este trabajo el estudio pormenorizado de los formatos de registro de los diferentes servidores y servicios, y utilizaremos el que utilizaría un servidor Apache® por su sencillez de formato y para poder explicar cómo nos ayuda la traducción determinista a reducir el tamaño de los registros que, como ISP se han de almacenar.

Luego en el caso de una solicitud judicial, el servidor ofendido proporcionaría un registro de accesos como el siguiente:

```
5.10.200.11:2105 - frank [27/Apr/2020:13:55:36 -0700] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.11:2106 - steve [27/Apr/2020:13:55:36 -0750] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.11:2107 - root [27/Apr/2020:13:55:36 -0800] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.11:6010 - admin [27/Apr/2020:13:55:36 -0850] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.11:6011 - operator [27/Apr/2020:13:55:36 -0900] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.11:6012 - root [27/Apr/2020:13:55:36 -0950] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.12:40003 - admin [27/Apr/2020:13:55:37 -0000] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.12:40004 - operator [27/Apr/2020:13:55:37 -0050] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.12:40005 - steve [27/Apr/2020:13:55:37 -0100] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.13:20102 - admin [27/Apr/2020:13:55:37 -0150] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.13:20103 - root [27/Apr/2020:13:55:37 -0200] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.13:20104 - root [27/Apr/2020:13:55:37 -0250] "GET /.htaccess HTTP/1.0" 200 312
5.10.200.11:2108 - admin [27/Apr/2020:13:55:37 -0300] "GET /.htaccess HTTP/1.0" 200 312
```

Ilustración 30: Registro de accesos a un servidor Apache®

Como se puede observar, si evaluamos la tabla de traducciones que programamos anteriormente, y hacemos un escrutinio del registro que se nos presenta, podemos observar que en el momento indicado obtenemos las siguientes direcciones de suscriptores:

IP de suscriptor	IP : Puerto registrados por el servidor
100.64.10.50	5.10.200.11 : 2105
	5.10.200.11 : 2106
	5.10.200.11 : 2107
100.64.10.52	5.10.200.11 : 6010
	5.10.200.11 : 6011
	5.10.200.11 : 6012
100.64.10.92	5.10.200.12 : 40003
	5.10.200.12 : 40004
	5.10.200.12 : 40005

100.64.10.105	5.10.200.13 : 20102
	5.10.200.13 : 20103
	5.10.200.13 : 20104
100.64.10.50	5.10.200.11 : 2108

Table 8: Mapeo del registro de traducciones

De esta manera el registro que ha de guardar el ISP es muy similar al que guardaría si no se implementase CGNAT, evitando tener que almacenar registros que incluyan las marcas de tiempo, las tuplas dirección:puerto para la dirección en el bloque 100.64.0.0/10, su traducción a la tupla dirección:puerto para la dirección pública y el protocolo de transporte utilizado.

3.4 Beta IPv6

Para la transición a IPv6, Guifi.Net ha de asignar prefijos IPv6 a cada uno de los proveedores que dan servicio de banda ancha y están adheridos a la Fundación. Como vimos anteriormente, el AS49835 administrado por Guifi.Net dispone del prefijo 2a00:1508::/32 y dado que el espacio de direcciones IPv6 es virtualmente inagotable, la recomendación [41] general ofrecida por el RIPE es que, en el eventual caso de ser necesarios más bloques de direcciones, se pueden solicitar. Además, el bloque mínimo que asigna el RIPE es de ::/32 con lo que recibir asignaciones de bloques ::/32 adicionales no es problema alguno.

Con esa premisa, utilizaremos el prefijo entero que Guifi.Net tiene asignado 2a00:1508::/32 y lo utilizaremos al completo para ISP A, que es el ISP que hemos estado utilizando durante todas nuestras pruebas. Como ya hemos comentado en el caso de que esta prueba beta sea satisfactoria, podría extenderse a los demás ISPs adheridos a Guifi.Net utilizando los nuevos prefijos que se asignasen.

Siguiendo con las recomendaciones del RIPE sobre las asignaciones de direcciones IPv6, implementaremos un *pool* de direcciones con el prefijo ::/48 para cada uno de los suscriptores. Independientemente de que se trate de organizaciones o usuarios finales, dado el tamaño del prefijo de ISP A, es irrelevante ajustar mucho el direccionamiento tal y como se venía haciendo con IPv4.

Para facilitar la implementación y despliegue de IPv6, y evitar inconvenientes de posibles dispositivos obsoletos conectados a través de los CPEs de los suscriptores, se optará por utilizar una estrategia ***dual-stack*** que ya introdujimos en el apartado 2.4.1.

3.4.1 Red del suscriptor

Tal y como se estudió en el apartado 2.4.1, la implementación de *dual-stack*, aporta la ventaja de mantener la conexión IPv4 para dispositivos obsoletos que

aún no soportan IPv6. Tal y como se observa en la ilustración 13, nos enfrentamos a 4 posibilidades:

- CPE sin soporte IPv6. El CPE seguirá recibiendo una dirección IPv4 del LNS de ISP A.
    - Equipos de suscriptor sin soporte IPv6. El CPE sigue siendo IPv4 nativo, con lo cual la comunicación tendrá lugar sin problemas.
    - Equipos de suscriptor con soporte de IPv6. En este caso los equipos toleran *dual-stack*, con lo que utilizarán la pila IPv4 para la comunicación.
- CPE con soporte de IPv6. El CPE soporta *dual-stack* y recibirá del LNS una dirección IPv4 y otra IPv6.
    - Equipos del suscriptor sin soporte IPv6. El CPE ha recibido una asignación IPv4 y realizará NAT44 de manera tradicional.
    - Equipos con soporte IPv6. El CPE habrá recibido un prefijo IPv6 delegado, con lo que el equipo del suscriptor podrá utilizar SLAAC [42] para autoconfigurar su direccionamiento IPv6. Aunque los equipos implementen *dual-stack*, las conexiones se establecerán de forma principal utilizando IPv6, e IPv4 quedará restringido a servicios en Internet que todavía no soporten IPv6.

En todos los casos la comunicación desde los equipos de los suscriptores ha de realizarse sin problemas y de forma transparente. En tanto en cuanto los ISP actualicen el parque de CPEs obsoletos que pueda haber desplegados, podrá seguir utilizándose IPv4 en la red.

3.4.2 Red del proveedor

En cuanto al *backbone* de la red, dado el caso con el que trabajamos, ISP A, un proveedor de acceso de banda ancha, tan sólo es necesario que configure su LNS y servidores DHCPs para que asignen rangos de direcciones IPv6 a los CPEs que inician las sesiones PPPoE. En nuestro entorno de laboratorio esas funciones están consolidadas en el BNG, con lo que los cambios de configuración se realizarán en este equipo.

El tráfico tanto IPv6 como IPv4 de los CPEs de ISP A recordemos que era establecido mediante sesiones PPPoE encapsuladas en túneles L2TP. Como estos túneles agregan una capa adicional de transporte independiente de la red subyacente, no hace tan importante a que ésta última funcione sobre IPv6 nativo o IPv4. Es algo que se deja al período de transición del proveedor de servicio de red.

Por lo tanto, se trata de habilitar el protocolo IPv6, configurar el *pool* de direcciones IPv6 que se repartirán a los suscriptores, y autorizar la delegación

de prefijos asignados a los CPEs, para que estos a su vez permitan la autoconfiguración de sus clientes dentro del rango recibido.

Las instrucciones necesarias para habilitar IPv6, junto a la delegación de prefijos, para el caso del BNG de ISP A con el que estamos trabajando serían:

```
ipv6 unicast-routing
ipv6 cef
ipv6 dhcp pool ISPAPool
 prefix-delegation pool ISPAPool
!
interface Loopback0
  ipv6 address 2A00:1508::1/128
!
interface Virtual-Template1
 peer default ip address pool ISPAPool
 peer default ipv6 pool ISPAPool
 ipv6 unnumbered Loopback0
 ipv6 enable
 ipv6 mtu 1492
 no ipv6 nd ra suppress
 ipv6 dhcp server ISPAPool
!
ipv6 local pool ISPAPool 2A00:1508:FF01::/48 64
```

Ilustración 31: Configuración IPv6 del BNG de ISP A

Y por supuesto habilitar los CPE a que transmitan la información del prefijo que les ha sido delgado a sus clientes:

```
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 ipv6 address ISPA_Prefix ::1:0:0:1/64
 ipv6 enable
!
 ipv6 address autoconfig default
 ipv6 enable
 ipv6 dhcp client pd ISPA_Prefix
```

Ilustración 32: Configuración IPv6 de un CPE de ISP A

3.4.3 Validación

Se han implementado los cambios anunciados en el apartado anterior y podemos comprobar que surten el efecto deseado, proporcionando conectividad *dual-stack* tanto a los CPE de los suscriptores a través de PPPoE. Además, los CPE delegan los prefijos recibidos por el LNS a los equipos cliente.

Comprobamos que la asignación de direcciones IPv6 a los CPE mediante PPPoE funciona como se espera. Podemos comprobar que el equipo ha asignado dos direcciones del *pool* que le fue configurado.

```
BNG1#sh ipv6 local pool
Pool                    Prefix                                  Free  In use
ISPAPool                2A00:1508:FF01::/48                     65534      2
BNG1#
```

Ilustración 33: Pool IPv6 de ISP A

A continuación, verificamos que el CPE ha recibido su dirección IPv6. Hemos de matizar que todos los equipos que implementan IPv6 deben disponer de manera obligatoria de una dirección IPv6 del *scope link-local* (definidos en el RFC4007 [43]). Es por eso que en la siguiente captura observamos direcciones asignadas del rango FE80::/10. Las direcciones que nos interesan son las del prefijo que ISP A tiene asignado, concretamente aquellas pertenecientes a 2A00:1508::/32.

```
CPE1#sh ipv6 int bri | e unass
Ethernet0/0             [up/up]
    FE80::A8BB:CCFF:FE00:500
    2A00:1508:FF01:1:1::1
Ethernet0/1             [up/up]
Ethernet0/2             [administratively down/down]
Ethernet0/3             [administratively down/down]
Serial1/0               [administratively down/down]
Serial1/1               [administratively down/down]
Serial1/2               [administratively down/down]
Serial1/3               [administratively down/down]
Dialer1                 [up/up]
    FE80::A8BB:CCFF:FE00:500
    2A00:1508:FF01:0:A8BB:CCFF:FE00:500
NVI0                    [up/up]
Virtual-Access1         [up/up]
Virtual-Access2         [up/up]
    FE80::A8BB:CCFF:FE00:500

CPE1#
```

Ilustración 34: Direccionamiento IPv6 de CPE1

Comprobamos también que la delegación de prefijos ha sido recibida correctamente ya que en la ilustración 34, la interfaz local **ethernet0/0** dispone de una dirección *unicast* dentro del rango correcto.

```
CPE1#sh ipv6 general-pre
IPv6 Prefix ISPA_Prefix, acquired via DHCP PD
  2A00:1508:FF01:1::/64 Valid lifetime 2591969, preferred lifetime 604769
   Ethernet0/0 (Address command)
CPE1#
```

Ilustración 35: Delegación de prefijo IPv6 de CPE1

Por último, podemos comprobar como un equipo cliente del CPE1 ha recibido su dirección IP con ámbito ::/64 tal y como se ordenaba desde el *pool* configurado en el LNS.

```
osboxes@osboxes:~$ ip -6 a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1000
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 2a00:1508:ff01:1:3056:5bee:737e:ec35/64 scope global temporary dynamic
       valid_lft 604612sec preferred_lft 86072sec
    inet6 2a00:1508:ff01:1:7e6a:e2eb:8f0f:965e/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 2591812sec preferred_lft 604612sec
    inet6 fe80::566f:741c:543a:cef1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
osboxes@osboxes:~$
```

Ilustración 36: Dirección IPv6 de PC1

4. Conclusiones

El objetivo principal de este estudio era considerar la viabilidad de implementación de Carrier-Grade NAT en un entorno multioperador como el que se presenta bajo el marco de la Fundación Guifi.Net. Se ha detallado en qué consiste CGNAT, y se han mostrado sus ventajas e inconvenientes ya que es imprescindible conocer de forma adecuada la tecnología subyacente para aportar las soluciones más adecuadas a cada proyecto.

Para el desarrollo de este trabajo, se han utilizado como base los dos tipos principales de arquitecturas de red utilizadas por los ISP actualmente. Y sobre estas bases se ha establecido el modelo de despliegue adecuado a las mismas. Como se ha podido observar, la implementación de CGNAT es relativamente transparente a los suscriptores y al servicio que reciben. Salvo las limitaciones inherentes al propio protocolo IP y sus implementaciones, la viabilidad de CGNAT en este sentido es perfectamente posible, en tanto en cuanto no se utilice como una medida permanente. Queda claro pues, que esta solución es un mero sistema para proporcionar margen de tiempo a la implementación de IPv6.

Para tener una imagen de la plena funcionalidad de esta implementación, se ha utilizado un entorno de trabajo emulado y sobre este se han reproducido, dentro de las limitaciones inherentes a las tecnologías de virtualización y a la escala reproducible en este entorno, las configuraciones necesarias para probar su funcionamiento.

Como base se partió de una topología de red multioperador en la que se configuraron redes de proveedor de acceso y proveedor de servicio, utilizando las tecnologías más extendidas como son PPPoE e IPoE para proporcionar conectividad *end-to-end* a los dispositivos del entorno virtual que simulaban los equipos de los suscriptores finales tras sus respectivos CPEs.

A partir de esta base, se introdujo un equipo Mikrotik virtual también, que realizaría las funciones de CGNAT sobre la red. Se ha elegido Mikrotik por ser un equipo muy frecuente en el core de Guifi.Net, además de tener una gran cantidad de funcionalidad disponible sin coste de licenciamento adicional como sucede con otros fabricantes. Las configuraciones relativas a CGNAT determinista resultaron muy sencillas de realizar tras implementar un *script* el Python para automatizar la configuración.

Para la introducción de este *router* CGNAT, y dado que la traducción de direcciones es una tecnología de capa 3, se hizo necesario modificar en cierta manera el esquema de enrutamiento en la capa agregada del ISP. Los cambios introducidos se realizaron de forma satisfactoria, con el nuevo esquema de tráfico funcionando, todas las conexiones de red generadas por los suscriptores estaban obligadas a pasar indefectiblemente por el *router* CGNAT, que por su parte se encargaba de realizar primero la asignación de recursos, en forma de tabla de traducciones, y posteriormente reenviando aguas arriba los paquetes con las traducciones ya aplicadas en las cabeceras IP de cada uno de ellos.

Además, mediante la configuración determinista que se aplicó, se pudo constatar que la reducción del tamaño de registro es enorme, facilitando por tanto el almacenamiento y abaratando el consumo de espacio en disco de forma considerable. De cara a posibles auditorías, menos tamaño en disco significa mayor velocidad en el análisis, búsqueda y tratamiento de los datos almacenados.

Esa misma configuración determinista permitió una ratio de compresión muy elevada, tanto que pudimos consolidar 23 suscriptores por cada dirección IP del prefijo disponible por el ISP. Lo cual a efectos de conservación del espacio de direcciones es un logro bastante considerable.

Por lo tanto podemos considerar que la implementación de CGNAT para conservar direccionamiento es totalmente alcanzable, tanto en cuanto a los equipos a utilizar, como por la ratio de ahorro de direcciones que alcanzamos. Estos datos arrojan bastante luz sobre cómo esta tecnología puede ayudar a retrasar el despliegue de IPv6 hasta que éste último pueda realizarse de forma segura y sin detrimento de servicio.

Todas las labores llevadas a cabo durante el desarrollo de este trabajo, han supuesto además la consecución de los objetivos secundarios planteados, como eran:

- Análisis de las limitaciones de CGNAT. Se estudiaron todas y cada una de ellas así como sus posibles implicaciones en entornos reales de proveedor.

- Proporcionar un diseño adecuado. El despliegue y configuración del entorno de laboratorio, proporcionó un esquemático de los pasos necesarios a dar cuando una solución de este tipo se despliegue en producción.

- Análisis del equipamiento disponible. Aunque el laboratorio implementa equipos virtuales de otros fabricantes, para el equipo alrededor del cual gira este estudio, el router CGNAT se implementó utilizando Mikrotik con una versión de *software* cuya publicación se realizó hace menos de un año, lo cual garantiza que el fabricante haya publicado suficientes parches y pueda considerarse estable.

- Valorar una posible transición a IPv6. En este objetivo pudimos constatar que la transición a IPv6 es perfectamente posible en tanto en cuanto los CPEs de los suscriptores lo soporten. Pero también que la heterogeneidad no supondrá ningún problema, ya que las dos versiones del protocolo van a poder convivir tanto tiempo como sea necesario.

5. Anexos

5.1 Configuración de los equipos

Se incluyen en este anexo las configuraciones de todos los dispositivos de red utilizados en el laboratorio. Por motivos de espacio y concisión y para evitar información innecesaria en este documento, se han omitido aquellas secciones de las configuraciones irrelevantes al propósito de este estudio.

5.1.1 CPE1

```
!
version 15.2
!
hostname CPE1
!
ip dhcp excluded-address 192.168.0.1 192.168.0.10
!
ip dhcp pool Customer_Pool
 network 192.168.0.0 255.255.255.0
 default-router 192.168.0.1
 dns-server 8.8.8.8
!
ip cef
ipv6 unicast-routing
ipv6 cef
!
bba-group pppoe global
!
interface Ethernet0/0
 ip address 192.168.0.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 ipv6 address ISP_A_Prefix ::1:0:0:1/64
 ipv6 enable
!
interface Ethernet0/1
 no ip address
 pppoe enable group global
 pppoe-client dial-pool-number 1
!
interface Dialer1
 ip address negotiated
 ip nat outside
 ip virtual-reassembly in
 encapsulation ppp
 dialer pool 1
 ipv6 address autoconfig default
 ipv6 enable
```

```
 ipv6 dhcp client pd ISP_A_Prefix
 ppp pap sent-username pppoe@uoc.lab password 0 uoc
!
ip nat inside source list local_nat interface Dialer1
overload
ip route 0.0.0.0 0.0.0.0 Dialer1
!
ip access-list extended local_nat
 permit ip 192.168.0.0 0.0.0.255 any
!
end
```

5.1.2 CPE2

```
!
version 15.2
!
hostname CPE2
!
ip dhcp excluded-address 192.168.1.1 192.168.1.10
!
ip dhcp pool Customer
 network 192.168.1.0 255.255.255.0
 default-router 192.168.1.1
 dns-server 8.8.8.8
!
ip cef
ipv6 unicast-routing
ipv6 cef
!
bba-group pppoe global
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 ipv6 address ISP_APrefix ::1:0:0:1/64
 ipv6 enable
!
interface Ethernet0/1
 no ip address
 pppoe enable group global
 pppoe-client dial-pool-number 1
!
interface Dialer1
 ip address negotiated
 ip nat outside
 ip virtual-reassembly in
 encapsulation ppp
 dialer pool 1
 ipv6 address autoconfig default
```

```
 ipv6 enable
 ipv6 dhcp client pd ISP_APrefix
 ppp pap sent-username pppoe@uoc.lab password 0 uoc
!
ip nat inside source list local_nat interface Dialer1
overload
ip route 0.0.0.0 0.0.0.0 Dialer1
!
ip access-list extended local_nat
 permit ip 192.168.1.0 0.0.0.255 any
!
end
```

5.1.3 CPE3

```
!
version 15.2
!
hostname CPE3
!
ip dhcp excluded-address 192.168.1.1 192.168.1.10
!
ip dhcp pool Customer
 network 192.168.1.0 255.255.255.0
 default-router 192.168.1.1
 dns-server 8.8.8.8
!
ip cef
ipv6 unicast-routing
ipv6 cef
!
bba-group pppoe global
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 ipv6 address ISP_A_Prefix ::1:0:0:1/64
 ipv6 enable
!
interface Ethernet0/2
 no ip address
 pppoe enable group global
 pppoe-client dial-pool-number 1
!
interface Dialer1
 ip address negotiated
 ip nat outside
 ip virtual-reassembly in
 encapsulation ppp
 dialer pool 1
```

```
 ipv6 address autoconfig default
 ipv6 enable
 ipv6 dhcp client pd ISP_A_Prefix
 ppp pap sent-username pppoe@uoc.lab password 0 uoc
!
ip nat inside source list local_nat interface Dialer1
overload
ip route 0.0.0.0 0.0.0.0 Dialer1
!
ip access-list extended local_nat
 permit ip 192.168.1.0 0.0.0.255 any
!
end
```

5.1.4 CPE4

```
!
version 15.2
!
hostname CPE4
!
ip dhcp excluded-address 172.16.0.0 172.16.0.10
!
ip dhcp pool Customer
 network 172.16.0.0 255.255.0.0
 default-router 172.16.0.1
 dns-server 8.8.8.8
!
ip cef
no ipv6 cef
!
interface Ethernet0/1
 ip address 172.16.0.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
!
interface Ethernet0/2
 ip address 91.197.122.130 255.255.255.128
 ip nat outside
 ip virtual-reassembly in
!
router eigrp 1
 network 91.197.0.0 0.0.255.255
!
ip nat inside source list local_nat interface Ethernet0/2
overload
ip route 0.0.0.0 0.0.0.0 91.197.124.1
!
ip access-list extended local_nat
 permit ip 172.16.0.0 0.0.0.255 any
!
```

```
end
```

5.1.5 CPE5

```
!
version 15.2
!
hostname CPE5
!
ip dhcp excluded-address 10.0.0.1 10.0.0.10
!
ip dhcp pool Customer
 network 10.0.0.0 255.0.0.0
 default-router 10.0.0.1
 dns-server 8.8.8.8
!
ip cef
no ipv6 cef
!
interface Ethernet0/0
 ip address 10.0.0.1 255.0.0.0
 ip nat inside
 ip virtual-reassembly in
!
interface Ethernet0/1
 ip address 91.197.123.2 255.255.255.128
 ip nat outside
 ip virtual-reassembly in
!
router eigrp 1
 network 91.197.0.0 0.0.255.255
!
ip nat inside source list local_nat interface Ethernet0/1
overload
ip route 0.0.0.0 0.0.0.0 91.197.128.1
!
ip access-list extended local_nat
 permit ip 10.0.0.0 0.0.0.255 any
!
end
```

5.1.6 LAC1

```
!
version 15.2
!
hostname LAC1
!
ip cef
no ipv6 cef
!
```

```
vpdn enable
!
vpdn-group 1
 request-dialin
  protocol l2tp
  domain uoc.lab
 initiate-to ip 10.0.254.1 priority 1
 initiate-to ip 10.0.254.5 priority 2
 local name LAC1
 l2tp tunnel password 0 uoc
!
bba-group pppoe global
 virtual-template 1
!
interface Ethernet0/0
 ip address 10.0.200.2 255.255.255.252
!
interface Ethernet0/1
 ip address 10.0.50.2 255.255.255.252
!
interface Ethernet0/2
 no ip address
 pppoe enable group global
!
interface Virtual-Template1
 ip unnumbered Ethernet0/2
 ppp authentication pap
!
router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
!
end
```

5.1.7 LAC2

```
!
version 15.2
!
hostname LAC2
!
ip cef
no ipv6 cef
!
vpdn enable
!
vpdn-group 1
 request-dialin
  protocol l2tp
  domain uoc.lab
 initiate-to ip 10.0.254.1 priority 1
 initiate-to ip 10.0.254.5 priority 2
```

```
 local name LAC2
 l2tp tunnel password 0 uoc
!
bba-group pppoe global
 virtual-template 1
!
interface Ethernet0/0
 ip address 10.0.150.2 255.255.255.252
!
interface Ethernet0/1
 ip address 10.0.100.2 255.255.255.252
!
interface Ethernet0/2
 no ip address
 pppoe enable group global
!
interface Virtual-Template1
 ip unnumbered Ethernet0/2
 ppp authentication pap
!
router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
!
end
```

5.1.8 TR1

```
!
version 15.2
!
hostname TR1
!
ip cef
no ipv6 cef
!
interface Ethernet0/0
 ip address 10.0.200.1 255.255.255.252
!
interface Ethernet0/1
 ip address 10.0.100.1 255.255.255.252
!
interface Ethernet0/2
 ip address 10.0.254.2 255.255.255.252
!
router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
!
end
```

5.1.9 TR2

```
!
version 15.2
!
hostname TR2
!
ip cef
no ipv6 cef
!
interface Ethernet0/0
 ip address 10.0.150.1 255.255.255.252
!
interface Ethernet0/1
 ip address 10.0.50.1 255.255.255.252
!
interface Ethernet0/3
 ip address 10.0.254.6 255.255.255.252
!
router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
!
end
```

5.1.10 TR3

```
!
version 15.2
!
hostname TR3
!
ip cef
no ipv6 cef
!
interface Ethernet0/0
 ip address 91.197.121.129 255.255.255.128
!
interface Ethernet0/1
 ip address 91.197.120.130 255.255.255.128
!
interface Ethernet0/2
 ip address 91.197.122.1 255.255.255.128
!
router eigrp 1
 network 91.197.0.0 0.0.255.255
!
end
```

5.1.10 TR4

```
!
```

```
version 15.2
!
hostname TR4
!
ip cef
no ipv6 cef
!
interface Ethernet0/0
 ip address 91.197.121.2 255.255.255.128
!
interface Ethernet0/1
 ip address 91.197.123.1 255.255.255.128
!
interface Ethernet0/2
 ip address 91.197.122.129 255.255.255.128
!
router eigrp 1
 network 91.197.0.0 0.0.255.255
!
end
```

5.1.11 BNG1

```
!
version 15.2
!
hostname BNG1
!
aaa new-model
!
aaa authentication login default local
aaa authentication login console none
aaa authentication ppp default local
aaa authorization network default local
!
aaa session-id common
ip cef
ipv6 unicast-routing
ipv6 cef
ipv6 dhcp pool ISP_APool
 prefix-delegation pool ISP_APool
!
vpdn enable
!
vpdn-group 1
 accept-dialin
  protocol l2tp
  virtual-template 1
 terminate-from hostname LAC1
 source-ip 10.0.254.1
 l2tp tunnel password 0 uoc
```

```
!
vpdn-group 2
 accept-dialin
  protocol l2tp
  virtual-template 2
 terminate-from hostname LAC1
 source-ip 10.0.254.5
 l2tp tunnel password 0 uoc
!
vpdn-group 3
 accept-dialin
  protocol l2tp
  virtual-template 3
 terminate-from hostname LAC2
 source-ip 10.0.254.1
 l2tp tunnel password 0 uoc
!
vpdn-group 4
 accept-dialin
  protocol l2tp
  virtual-template 4
 terminate-from hostname LAC2
 source-ip 10.0.254.5
 l2tp tunnel password 0 uoc
!
username pppoe@uoc.lab password 0 uoc
!
interface Loopback0
 ip address 100.64.0.1 255.255.255.255
 ipv6 address 2A00:1508::1/128
!
interface Ethernet0/0
 ip address 5.10.200.253 255.255.255.252
 ipv6 enable
!
interface Ethernet0/1
 ip address 100.64.0.5 255.255.255.252
!
interface Ethernet0/2
 ip address 10.0.254.1 255.255.255.252
!
interface Ethernet0/3
 ip address 10.0.254.5 255.255.255.252
!
interface Virtual-Template1
 ip unnumbered Loopback0
 ip mtu 1492
 peer default ip address pool ISP_APool
 peer default ipv6 pool ISP_APool
 ipv6 unnumbered Loopback0
 ipv6 enable
```

```
 ipv6 mtu 1492
 no ipv6 nd ra suppress
 ipv6 dhcp server ISP_APool
 ppp authentication pap
!
router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
!
ip local pool ISP_APool 100.64.10.50 100.64.255.250
ip forward-protocol nd
!
ip route 0.0.0.0 0.0.0.0 100.64.0.6
!
ipv6 local pool ISP_APool 2A00:1508:FF01::/48 64
!
end
```

5.1.12 BNG2

```
!
version 15.2
!
hostname BNG2
!
ip cef
no ipv6 cef
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface Ethernet0/0
 ip address 91.197.121.1 255.255.255.252
!
interface Ethernet0/1
 ip address 91.197.120.129 255.255.255.252
!
interface Ethernet0/2
 ip address 91.197.120.2 255.255.255.252
!
router eigrp 1
 network 91.197.0.0 0.0.255.255
 redistribute static
!
ip route 0.0.0.0 0.0.0.0 91.197.120.1
!
end
```

5.1.13 BGP1

```
!
version 15.2
```

```
!
hostname BGP1
!
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 ip address 20.0.0.1 255.255.255.0
 ipv6 enable
!
interface Ethernet0/1
 ip address 5.10.200.254 255.255.255.252
 ipv6 enable
!
interface Ethernet0/2
 ip address 5.10.200.249 255.255.255.252
!
router bgp 49835
 bgp log-neighbor-changes
 neighbor 20.0.0.2 remote-as 49835
 neighbor 20.0.0.4 remote-as 49835
 !
 address-family ipv4
  network 5.10.200.0 mask 255.255.255.0
  neighbor 20.0.0.2 activate
  neighbor 20.0.0.2 next-hop-self
  neighbor 20.0.0.4 activate
  neighbor 20.0.0.4 next-hop-self
  auto-summary
 exit-address-family
!
ip route 5.10.200.0 255.255.255.0 5.10.200.250
ip route 100.64.0.0 255.255.0.0 5.10.200.250
!
end
```

5.1.14 BGP2

```
!
version 15.2
!
hostname BGP2
!
ip cef
no ipv6 cef
!
interface Ethernet0/0
 ip address 20.0.0.2 255.255.255.0
!
interface Ethernet0/1
```

```
 no ip address
 shutdown
!
interface Ethernet0/2
 ip address 91.197.120.1 255.255.255.252
!
router bgp 49835
 bgp log-neighbor-changes
 neighbor 20.0.0.1 remote-as 49835
 neighbor 20.0.0.4 remote-as 49835
 !
 address-family ipv4
  network 91.197.120.0 mask 255.255.252.0
  neighbor 20.0.0.1 activate
  neighbor 20.0.0.1 next-hop-self
  neighbor 20.0.0.4 activate
  neighbor 20.0.0.4 next-hop-self
  auto-summary
 exit-address-family
!
ip route 91.197.120.0 255.255.252.0 91.197.120.2
!
end
```

5.1.15 CGNAT

En el caso de este dispositivo solamente se incluyen las reglas de traducción determinista para la dirección 100.64.10.50. Incluir reglas para todo el rango generaría una configuración demasiado extensa para incluirla en este estudio.

El Anexo 5.2 de este documento incluye el *script* en Python que se encarga de la generación de las reglas de traducción pertinentes.

```
/system logging action
set 3 bsd-syslog=yes remote=192.168.1.10
/ip firewall connection tracking
set tcp-established-timeout=10m
/ip address
add address=5.10.200.250/30 interface=ether2
network=5.10.200.248
add address=100.64.0.6/30 interface=ether1
network=100.64.0.4
add address=192.168.1.254/24 interface=ether3
network=192.168.1.0
/ip dhcp-client
add disabled=no interface=ether1
/ip firewall nat
add action=jump chain=srcnat jump-target=ISPA log=yes log-
prefix="CGNAT INFO" src-address=\
    100.64.10.50
```

```
add action=jump chain=ISPA jump-target=ISPA-0 src-
address=100.64.10.50
add action=src-nat chain=ISPA-0 protocol=tcp src-
address=100.64.10.50 to-addresses=5.10.200.11 \
    to-ports=2001-4000
add action=src-nat chain=ISPA-0 protocol=udp src-
address=100.64.10.50 to-addresses=5.10.200.11 \
    to-ports=2001-4000
add action=src-nat chain=ISPA-0 src-address=100.64.10.50
to-addresses=5.10.200.11
/ip route
add distance=1 gateway=5.10.200.249
add distance=1 dst-address=100.64.0.0/16 gateway=100.64.0.5
/system logging
add action=remote topics=info
```

5.2 Script Python para la configuración CGNAT determinista

```
import argparse, sys, ipaddress

def getOptions(args=sys.argv[1:]):
 parser = argparse.ArgumentParser(description="Crea las
cadenas de configuración CGNAT para RouterOS")
 parser.add_argument("-n", "--numero", type=int,
help="Número de direcciones a traducir.", required=True)
 parser.add_argument("-o", "--origen", type=str, help="La
primera dirección IP privada.", required=True)
 parser.add_argument("-d", "--destino", type=str, help="La
dirección IP pública.", required=True)
 parser.add_argument("-p", "--puerto", type=int, help="El
primer puerto asignado.", required=True)
 parser.add_argument("-c", "--cantidad", type=int,
help="Número de puertos asignados.", required=True)
 options = parser.parse_args(args)
 return options

def main():
 opciones = getOptions(sys.argv[1:])
 puerto_maximo = 48000

 try:
  if ipaddress.IPv4Address(opciones.origen):
   protocolos = ['tcp', 'udp', 'ip', 'icmp']
  if opciones.numero * opciones.puerto - 1 > puerto_maximo:
   print("Se va a sobrepasar el número de puertos asignados
para la IP pública.")
  else:
   print("/ip firewall nat add chain=srcnat action=jump
jump-target= ISPA src-address={}-
{}".format(ipaddress.IPv4Address(opciones.origen),
```

```
ipaddress.IPv4Address(opciones.origen) + opciones.numero -
1))
   for i in range(opciones.numero):
    print("/ip firewall nat add chain= ISPA action=jump
jump-target=ISPA-{} src-address={}".format(i,
ipaddress.IPv4Address(opciones.origen) + i))
puerto_i=(i + 1) * opciones.puerto - i
    for protocolo in protocolos:
     print("/ip firewall nat add chain=ISPA-{} action=src-
nat protocol={} src-address={} to-address={} to-ports={}-
{}".format(i, protocolo,
ipaddress.IPv4Address(opciones.origen),
ipaddress.IPv4Address(opciones.destino), puerto_i, puerto_i
+ opciones.cantidad-1))
    print("/ip firewall nat add chain=ISPA-{} action=src-
nat src-address={} to-address={}".format(i,
ipaddress.IPv4Address(opciones.origen) + i,
ipaddress.IPv4Address(opciones.destino)))

 except ipaddress.AddressValueError as error:
  print(error)

if __name__ == "__main__":
 main()
```

6. Bibliografía

1. **RIPE NCC**. **2019**. IPv4 Run-Out. [artículo en línea]. Noviembre de 2019. https://www.ripe.net/manage-ips-and-asns/ipv4/ipv4-run-out.

2. **S. Joshi, B. Dawadi**. **2008**. IPv6-only Network for Today's Internet: Prospects and Problems. [artículo en línea]. Enero de 2008. https://www.researchgate.net/figure/Pv4-address-exhaustion-Timeline-8_fig1_315769551.

3. **ESnet**. . ESnet IPv6 History. [artículo en línea]. . http://es.net/engineering-services/ipv6-network/esnet-ipv6-history/.

4. **APNIC**. . About Carrier Grade NAT (CGN). [artículo en línea]. . https://www.apnic.net/community/ipv6-program/about-cgn/.

5. **Cisco Systems, Inc**. **2019**. IP addressing: NAT Configuration Guide. [artículo en línea]. Diciembre de 2019. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_nat/configuration/xe-16/nat-xe-16-book/iadnat-addr-consv.html#GUID-6882F409-51AC-43C6-B718-1AB3DC781FBB.

6. **DARPA**. **1981**. Internet Protocol. [artículo en línea]. Septiembre de 1981. https://tools.ietf.org/html/rfc791.

7. **J. Postel**. **1981**. Internet Control Message Protocol. [artículo en línea]. Septiembre de 1981. https://tools.ietf.org/html/rfc792.

8. **J. Postel**. **1980**. User Datagram Protocol. [artículo en línea]. Agosto de 1980. https://tools.ietf.org/html/rfc768.

9. **DARPA**. **1981**. Tranmission Control Protocol. [artículo en línea]. Septiembre de 1981. https://tools.ietf.org/html/rfc793.

10. **J. Postel**. **1985**. File Transfer Protocol. [artículo en línea]. Octubre de 1985. https://tools.ietf.org/html/rfc959.

11. **J. Reynolds, J. Postel**. **1992**. Assigned Numbers. [artículo en línea]. Julio de 1992. https://tools.ietf.org/html/rfc1340.

12. **S. Frankel, S. Krishnan**. **2011**. IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap . [artículo en línea]. Febrero de 2011. https://tools.ietf.org/html/rfc6071.

13. **H. Schulzrinne, S. Casner, R. Frederik, V. Jacobson**. **2003**. A Transport Protocol for Real-Time Applications. [artículo en línea]. Julio de 2003. https://tools.ietf.org/html/rfc3550.

14. **R. Atkinson**. **1995**. IP Encapsulating Security Payload (ESP). [artículo en línea]. Agosto de 1995. https://tools.ietf.org/html/rfc1827.

15. **R. Atkinson**. **1995**. IP Authentication Header. [artículo en línea]. Agosto de 1995. https://tools.ietf.org/html/rfc1826.

16. **B. Carpenter, K. Moore**. **2001**. Connection of IPv6 Domains via IPv4 Clouds. [artículo en línea]. Febrero de 2001. https://tools.ietf.org/html/rfc3056.

17. **C. Huitema**. **2006**. Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). [artículo en línea]. Febrero de 2006. https://tools.ietf.org/html/rfc4380.

18. **M. Boucadair, R. Penno, D. Wing**. **2013**. Universal Plug and Play (UPnP) Internet Gateway Device - Port Control Protocol Interworking Function (IGD-PCP IWF). [artículo en línea]. Julio de 2013. https://tools.ietf.org/html/rfc6970.

19. **J. Rosenberg, R. Mahy, P. Mathews, D. Wing**. **2008**. Session Traversal Utilities for NAT (STUN). [artículo en línea]. Octubre de 2008. https://tools.ietf.org/html/rfc5389.

20. **J. Zorz**. **2018**. CGN, IPv6 and fighting online crime. [artículo en línea]. Marzo de 2018. https://www.internetsociety.org/blog/2018/03/cgn-ipv6-fighting-online-crime/.

21. **S. Sivakumar, R. Penno**. **2013**. IPFIX Information Elements for logging NAT Events. [artículo en línea]. Enero de 2013. https://tools.ietf.org/html/draft-sivakumar-behave-nat-logging-06.

22. **M. Rose**. **2020**. Internet of Things. [artículo en línea]. Febrero de 2020. https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT.

23. **IANA**. **2019**. IANA IPv4 Address Space Registry. [artículo en línea]. Diciembre de 2019. https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml.

24. **P. Srisuresh, B. Ford, S. Sivakumar, S. Guha**. **2009**. NAT Behavioral Requirements for ICMP. [artículo en línea]. Abril de 2009. https://tools.ietf.org/html/rfc5508.

25. **F. Audet, C. Jenning**. **2007**. Network Address Translation (NAT) Behavioral Requirements for Unicast UDP. [artículo en línea]. Enero de 2007. https://tools.ietf.org/html/rfc4787.

26. **S. Guha, K. Biswas, B. Ford, S. Sivakumar, P. Srisuresh**. **2008**. NAT Behavioral Requirements for TCP. [artículo en línea]. Octubre de 2008. https://tools.ietf.org/html/rfc5382.

27. Application-level gateways. [artículo en línea]. . https://www.3cx.es/voip-sip/alg/.

28. **Y. Rekhter, B. Moskowitz, D. Karrenberg, J. G. de Groot, E. Lear**. **1996**. Address Allocation for Private Internets. [artículo en línea]. Febrero de 1996. https://tools.ietf.org/html/rfc1918.

29. **M. Cotton, L. Vegoda, R. Bonica, B. Haberman**. **2013**. Special-Purpose IP Address Registries. [artículo en línea]. Abril de 2013. https://tools.ietf.org/html/rfc6890.

30. **J. Weil, V. Kuarsingh, C. Donley, C. Liljentolpe, M. Azinger**. **2012**. IANA-Reserved IPv4 Prefix for Shared Address Space. [artículo en línea]. Abril de 2012. https://tools.ietf.org/html/rfc6598.

31. **R. Despres**. **2010**. IPv6 Rapid Deployment on IPv4 Infrastructures (6rd). [artículo en línea]. Enero de 2010. https://tools.ietf.org/html/rfc5569.

32. **F. Templin**. **2017**. IPv6 Prefix Delegation Models. [artículo en línea]. Diciembre de 2017. https://tools.ietf.org/id/draft-templin-v6ops-pdhost-17.html.

33. **W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, B. Palter**. **1999**. Layer Two Tunneling Protocol "L2TP". [artículo en línea]. Agosto de 1999. https://tools.ietf.org/html/rfc2661.

34. **Y. El Mghazli, T. Nadeau, M. Boudacair, K. Chan, A. Gonguet**. . Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management. [artículo en línea]. Octubre de 2005. https://tools.ietf.org/html/rfc4176.

35. **DSL Forum**. **2003**. Technical Report DSL Forum TR-059. [artículo en línea]. Septiembre de 2003. https://www.broadband-forum.org/technical/download/TR-059.pdf.

36. **Broadband Forum**. **2017**. Using GPON Access in the context of TR-101. [artículo en línea]. Noviembre de 2017. https://www.broadband-forum.org/technical/download/TR-156_Issue-4.pdf.

37. **R. Droms**. **1997**. Dynamic Host Configuration Protocol. [artículo en línea]. Marzo de 1997. https://tools.ietf.org/html/rfc2131.

38. **C. Donley, C. Grundemann, V. Sarawat, K. Sundaresan, O. Vautrin**. **2014**. Deterministic Address Mapping to Reduce Logging in Carrier-Grade NAT Deployments. [artículo en línea]. Diciembre de 2014. https://tools.ietf.org/html/rfc7422.

39. **M. Cotton, L. Eggert, J. Touch, M. Westerlund, S. Cheshire**. **2011**. Internet Assigned Numbers Authority (IANA) Procedures for the Management

of the Service Name and Transport Protocol Port Number Registry. [artículo en línea]. Agosto de 2011. https://tools.ietf.org/html/rfc6335.

40. **D. Farinacci, T. Li, S. Hanks, D. Meyer, P. Traina**. **2000**. Generic Routing Encapsulation (GRE). [artículo en línea]. Marzo de 2000. https://tools.ietf.org/html/rfc2784.

41. **RIPE NCC**. **2017**. Best Current Operational Practice for Operators: IPv6 prefix assignment for end-users – persistent vs non-persistent, and what size to choose. [artículo en línea]. Octubre de 2017. https://www.ripe.net/publications/docs/ripe-690.

42. **S. Thomson, T. Narten, T. Jinmei**. **2007**. IPv6 Stateless Address Autoconfiguration. [artículo en línea]. Septiembre de 2007. https://tools.ietf.org/html/rfc4862.

43. **S. Deering, B. Haberman, T.Jinmei, E. Nordmark, B.Zill**. **2005**. IPv6 Scoped Address Architecture. [artículo en línea]. Marzo de 2005. https://tools.ietf.org/html/rfc4007.