academy xperts

DNS Caché Transparente DNS Estático DHCP RFC2132 DHCP-Server DHCP-Client DHCP BOOTP Opciones DHCP Pools DHCP-Leases DHCP-Alerts DHCP-Authoritative DHCP-Relay Firewall Filter input-output-forward Connection Track Connection State Network Intrusion port scan-DoS attack-DDoS attack- ping flood Bogon IPs Firewall NAT srcnat-dstnat netmap-same Firewall Mangle mark-connection mark-packet HTB Estructura Limitación Dual Prioridad Queues PCQ Simple Queues Queue Tree Burst

RouterOS

Control de Tráfico, Firewall y QoS con MikroTik RouterOS

por Mauro Escalante

DNS, DHCP, Firewall Filter, Firewall NAT, Firewall Mangle, HTB, Queues, Burst, Web Proxy, Balanceo de Carga

RouterOS v6.33.5

Control de Tráfico Firewall y QoS con MikroTik RouterOS

v6.33.5.01

Libro de Estudio

ABC Xperts ®
Network Xperts ®
Academy Xperts ®

Tabla de Contenido

Introducción 4
Resumen 5
Audiencia 5
Convenciones usadas en este libro 5
Comentarios y preguntas 5
Un poco de Historia (Costa Rica) 7
Cubriendo un País con MikroTik. 7
Detalle de cambios en las tres últimas versiones de RouterOS 8
0 – Red de Trabajo 10
Laboratorio 0-1 10
Capítulo 1: DNS 11
Qué es DNS 11
MikroTik DNS Caché 11
Monitoreo del Cache 12
Borrando el DNS Cache (flushing) 13
Entradas de DNS Estático 13
Preguntas de repaso del Capítulo 1 13
Laboratorio 1-1: DNS Transparente 14
Laboratorio 1-2: DNS Estático 14
Capítulo 2: DHCP 15
Escenario de comunicación DHCP 16
Identificación del Cliente DHCP 17
Configuración Rápida (Quick Setup) 18
IPv6 18
DHCP-Server 18
Configuración del almacenamiento de reservas (leases) 20
Laboratorio 2-1: DHCP Server 20
DHCP-Networks 20
DHCP-Options 21
Extensiones de fabricante BOOTP y Opciones DHCP 22
Pool de direcciones IP 26
DHCP-Leases 26
DHCP-Alerts 27
DHCP-Client 28
IPv6 28
Estatus 29
Comandos Específicos 29
DHCP Autoritativo 29
DHCP-Relay 30
Laboratorio 2-2: DHCP Relay 30
Capítulo 3: Firewall Filter 31
Estructura de Filtros de Firewall 31
Diagrama de Estructura de Filtros de Firewall 31
Connection Tracking 31
Diagrama de ubicación del Connection Tracking 32
Condition: Connection State 32
Capítulo 4: Firewall Filter - Chain Input 36
Chain Input 36
Laboratorio 4-1: Filter Input – Reglas Básicas 36
Tipos de Intrusión de Red (Network Intrusion) 36
Port Scan 36
Ataques DoS 37
Ataques DDoS 38
ICMP 38
Estructura del segmento ICMP 39
Ping Flood 40
Laboratorio 4-2: Filter Input – Network Intrusion 41
Servicios de RouterOS 42
Capítulo 5: Firewall Filter - Chain Forward 43
Chain Forward 43
Laboratorio 5-1: Filter Input – Network Intrusion 43

Capítulo 6: Bogon IPs 44
Bogon IPs 44
Capítulo 7: Firewall NAT 45
Estructura del Firewall NAT 48
dst-nat 49
redirect 49
Laboratorio 7-1: Ejercicio de dstnat 50
Laboratorio 7-2: Ejercicio de dstnat 50
Inconvenientes del Source NAT 51
NAT Helpers (Service Ports) 51
NAT `action=netmap` 51
NAT `action=same` 51
Capítulo 8: Firewall Mangle 53
Change MSS 56
Marcado de Paquetes 57
Estructura del Mangle 57
Marcado de Conexiones 58
Marcado de Paquetes 58
Laboratorio 8-1: Ejercicio de mangle #1 59
Laboratorio 8-2: Ejercicio de mangle #2 59
Capítulo 9: HTB 60
Colores de las colas en Winbox 60
Estructura HTB 60
Limitación Dual 61
Prioridad 61
Ejemplo 1 – caso típico 62
Ejemplo 2 – caso típico con max-limit 62
Ejemplo 3 – limit-at de la cola más interna 63
Ejemplo 4 – limit-at de la cola hoja 63
Capítulo 10: Queues 64
Principios de limitación de velocidad 64
Simple Queues (Colas Simples) 65
Identificadores de Flujo 65
Propiedades HTB 66
Tipos de Colas (Queue Types) 67
PFIFO, BFIFO y MQ PFIFO 68
RED 69
SFQ 69
PCQ 70
Nueva implementación de PCQ (a partir de la v5.0RC5) 73
Interface Queue 73
Capítulo 11: Burst 74
Capítulo 12: Web proxy 75
Ejemplo de configuración de Proxy Transparente 76
Firewall basado en Proxy 76
Habilitación de RAM o Caché basado en almacenamiento 77
Access List 80
Direct Access 81
Administración del Caché 82
Connections 82
Métodos HTTP 83
Capítulo 13: TTL 84
Mangle action=change-ttl 85
Capítulo 14: Balanceo de Carga 86
Balanceo de Carga Usando PCC (Per Connection Classifier) 86
Balanceo de Carga Usando ECMP (Equal Cost Multi-Path) 87
Balanceo de Carga usando Nésimo Paquete (Nth Packet) 88

Introducción

MikroTik es una empresa que nace en Latvia (Letonia) en 1995 con el claro objetivo de proveer un sistema operativo de red altamente robusto y eficiente al cual llamó RouterOS en 1997. La evolución del mismo llevó a la creación y lanzamiento al mercado en el 2002 de un hardware que aprovechara al máximo sus grandes capacidades de multiprocesamiento simétrico y multi-núcleo, este hardware es el RouterBOARD.

A lo largo de los años a partir del nacimiento del Internet, los administradores de red hemos visto desfilar varios fabricantes por nuestros racks, siendo Cisco el referente, sin embargo siempre había representado un costo más o menos importante a la hora de implementar una solución de red ruteada en especial si se trataba de un ISP/WISP.

No es sino hasta hace una década aproximadamente en que MikroTik se empieza a hacer conocer en Latinoamérica y varios emprendedores, y por sobre entusiastas, se vuelcan a la implementación de soluciones basadas en RouterOS y RouterBOARD. Claro ejemplo de ello son nuestros grandes amigos de Index México (Ezequiel García) y REICO Costa Rica (Miguel Solís) quienes tomaron la iniciativa de confiar en los productos ofrecidos por MikroTik. Es muy interesante y gratificante conversar con ellos y escuchar los relatos sobre los primeros pasos del fabricante letón en tierras americanas.

Estoy convencido de que MikroTik llegó no solo para quedarse sino para formar una parte muy importante en la historia del networking y de las telecomunicaciones. De hecho, cientos de miles (quizá millones a esta fecha - Junio 2015) obtienen su internet de banda ancha a un bajo costo a través de una red ruteada gracias a que los proveedores de Internet, pequeños y medianos, pueden estructurar e implementar redes sumamente complejas y completas usando los RouterBOARD.

Las soluciones en RouterOS y RouterBOARD no se han quedado estancadas en las empresas de Telecom pequeñas, sino que han ido escalando en credibilidad en las empresa medianas y grandes en Latinoamérica, rompiendo paradigmas de fabricantes y costos de implementación.

Este libro nace como un aporte a la comunidad tecnológica de habla hispana y latinoamericana que ha decidido incursionar en MikroTik y desea obtener un conocimiento formal. De igual manera queremos que esta guía constituya una fuente importante de aprendizaje para quienes empiezan a realizar sus primeras configuraciones en RouterOS.

Mauro Escalante

CEO Academy Xperts
CEO Network Xperts

Resumen

Hemos tenido un especial cuidado en ampliar la información de aquellos puntos que no se profundizan en los cursos de certificación, pero que resultan claves para el correcto entendimiento de la materia.

La información aquí presentada se complementa con nuestros recursos en *www.abcxperts.com* y *www.youtube.com/abcxperts*

Este libro no pretende reemplazar la interacción face-to-face con un instructor ya que su experiencia y conocimiento es invaluable y únicamente explotable a través del contacto interpersonal de un curso de certificación. Sin embargo, todo el material de apoyo junto con los videos tutoriales, webinars, tips, etc., representan un importante aporte para aquellos colegas que optan por leer un libro y estudiar a su propio ritmo.

Esta es la primera revisión dedicada a la versión 6.33.5 Las posteriores revisiones al material y a los nuevos releases de RouterOS serán agregadas a esta edición y estarán a disponibilidad de las personas que compren la suscripción.

Tenemos una tarea inmensa por delante, pero estamos muy claros en nuestro objetivo de hacer de este libro la mejor guía de autoestudio MikroTik.

Audiencia

Las personas que leen este libro deben estar familiarizados con:

- Operaciones de red en Capa 2
- Conjunto de protocolos IP, incluyendo TCP. UDP e ICMP

Este libro está dirigido a:

- Ingenieros y Técnicos en Redes, Telecomunicaciones y afines, que desea implementar y dar soporte a:
  - Redes Corporativas
  - Clientes WISP e ISP
- Ingenieros de Redes involucrados en actividades de pre-venta y post-venta en soporte e instalación de redes corporativa y PYMES
- Ingenieros de Redes, Administradores de Red, Técnicos en Soporte de Redes, y Técnicos de Soporte a Usuario (Help Desk)

Convenciones usadas en este libro

En este libro se utilizarán las siguientes convenciones tipográficas:

Itálicas

Indica comandos, direcciones de correo, claves, mensajes de error, nombres de archivos, énfasis, y el primer uso de términos técnicos

`Courier new`

Indica direcciones IP y ejemplos de línea de comando

`Courier new en itálica`

Indica texto que puede ser reemplazado

`Courier new en negrita`

Indica datos de entrada del usuario

Este icono significa un consejo, sugerencia, o una nota general.

Este icono indica una advertencia o precaución.

Comentarios y preguntas

Puede enviar sus comentarios y preguntas sobre este libro por correo tradicional a la siguiente dirección:

Network Xperts S.A.
Av. Juan T. Marengo y J. Orrantia
Edificio Professional Center, Piso 5, Ofic. 507
Guayaquil, ECUADOR
+593-4-600-8590
+593-9-9535-2132

A través del sitio web y por medio de su usuario y contraseña, tendrá acceso a las actualizaciones, ejemplos, e información adicional:

http://cursos.abcxperts.com

Puede enviarnos sus comentarios o preguntas técnicas sobre este libro enviándonos un email a:

libro@abcxperts.com

Para más información sobre libros, conferencias, centros de recursos, y la red educativa de Academy Xperts, visite nuestros Websites y canal de YouTube

http://www.abcxperts.com

http://www.academyxperts.com

http://www.youtube.com/abcxperts

Un poco de Historia (Costa Rica)

Cubriendo un País con MikroTik.

En el año 1998, estando en una empresa de servicios públicos en Costa Rica, el Ing. Miguel Solís en conjunto con el Ing. Paulino Solano, comenzaron a utilizar MikroTik con gran éxito en las telecomunicaciones de esta empresa. Se lograron 2 Mbps en una distancia de 8 Km, una velocidad record para aquellos tiempos en que la velocidad rondaba los 256 Kbps.

En esta empresa de Servicios Públicos, se logró la interconexión de 52 sucursales mediante tecnología inalámbrica, todas bajo la misma marca MikroTik y su sistema operativo RouterOS.

Dado el éxito alcanzado en este proyecto, ambos ingenieros en conjunto con uno más llamado Olman González, decidieron formar una empresa que se dedicara a solventar los problemas de telecomunicaciones en donde el cobre no fuera factible o se necesitara más velocidad. Esta empresa fue nombrada Redes Inalámbricas de Costa Rica S.A (REICO).

Es así como a la fecha (Julio 2015), REICO, con solo Miguel Solís como propietario, tiene el liderato en telecomunicaciones inalámbricas en el país Centroamericano Costa Rica. REICO posee más de 3,800 Km de red troncal inalámbrica y más de 80,000 Km de red de acceso. Posee más de 100 radio bases instaladas estratégicamente para alcanzar una cobertura de más del 80% del territorio y a más del 90% de la población.

La empresa se dedica 100% a proveer transporte de datos corporativos y sirve a sectores financieros, agroindustriales, turísticos, comerciales, etc.

Su plataforma tiene una particularidad única en el mundo, con sus más de 1,000 clientes corporativos y empresariales y sus más de 1,500 equipos de acceso, CPE, transporte, Core secundario y Core primario: EL 100% SON MARCA MIKROTIK.

REICO es un ejemplo del gran potencial que tiene MikroTik y RouterOS ya que esta empresa compite en el mercado con grandes de las telecomunicaciones y aun así mantiene una posición privilegiada, siendo el cuarto operador en Costa Rica en importancia en Transporte de Datos Corporativos, por debajo de ICE, Tigo y de RACSA pero por encima de Claro, Telefónica, Cables & Wireless, etc. Esto según el último informe de Estadísticas del Sector de Telecomunicaciones de Costa Rica 2014.

Texto desarrollado por el Ing. Miguel Solís, a quien agradezco por su aporte histórico sobre los inicios de MikroTik en Latinoamérica.

Detalle de cambios en las tres últimas versiones de RouterOS

Para una revisión del histórico de cambios en la versión 6.x le recomendamos visitar el siguiente link:

http://abcxperts.com/index.php/bitacora-de-cambios

Número de Versión		6.33.5	6.33.3	6.33.2
Fecha Emisión		Wed, 13/Jan/2016	Thu, 03/Dec/2015	Fri, 27/Nov/2015
Hora de Emisión		16:08	16:08	9:55
# Días transcurridos desde versión anterior		41	6	10
arp		Muestra entradas ARP incompletas		
bridge				Se corrigió un problema de power-cycle-ping en los puertos bridge (que afectaba a todo el bridge)
btest		Se corrige una caida potencial después que se libera el btest Se mejora la precisión de la tasa UDP Tx		
crypto		Se corrigió una falla en el kernel en la encriptación Talitos HW		
dhcpv4	**server**			Se corrigió un problema de caida cuando se restauraba el lease con una cola (queue) a un server que ya no existe
dhcpv4	**client**			Soporte para la asignación de dirección /32
dhcpv6	**client**	Se corrige problemas de asignación de dirección DNS Se configura los parámetros correctos cuando se utiliza rapid-commit		
email		No resetea la dirección del server después de cambiar la configuración	Se hace que el campo de clave (password) sea sensitivo en la consola	
ethernet			Se corrigió un problema de autonegociación 10/100 Mbps en el ether1 del RB922UAGS (introducido en v6.33.2)	Se corrigió un problema en el que se resetaba el enlace cuando se hacía un cambio de valor en power-cycle-ping
fastpath		Se corrigió una posibe falla del kernel en sistemas multi núcleo		
fetch		Se agregó un time-out de conexión de 30 segundos		
hotspot		Se agregó el archivo perdido favicon.ico en las páginas htmls del hotspot		
kernel		Mejora general en la planificación de los procesos del core		
led		Se agregó el led WLAN al RB951Ui		
LTE		Se mejoró el soporte para el Sierra Wireless 320U Se mejoró la velocidad de la conexión por primera vez a una red LTE en los SXT LTE		
log		Se hace log de los eventos up/down únicamente cuando el enlace actual cambia su estado		
net		Se aplica la configuración esclavo (slave) únicamente si se ha cambiado la configuración del master		

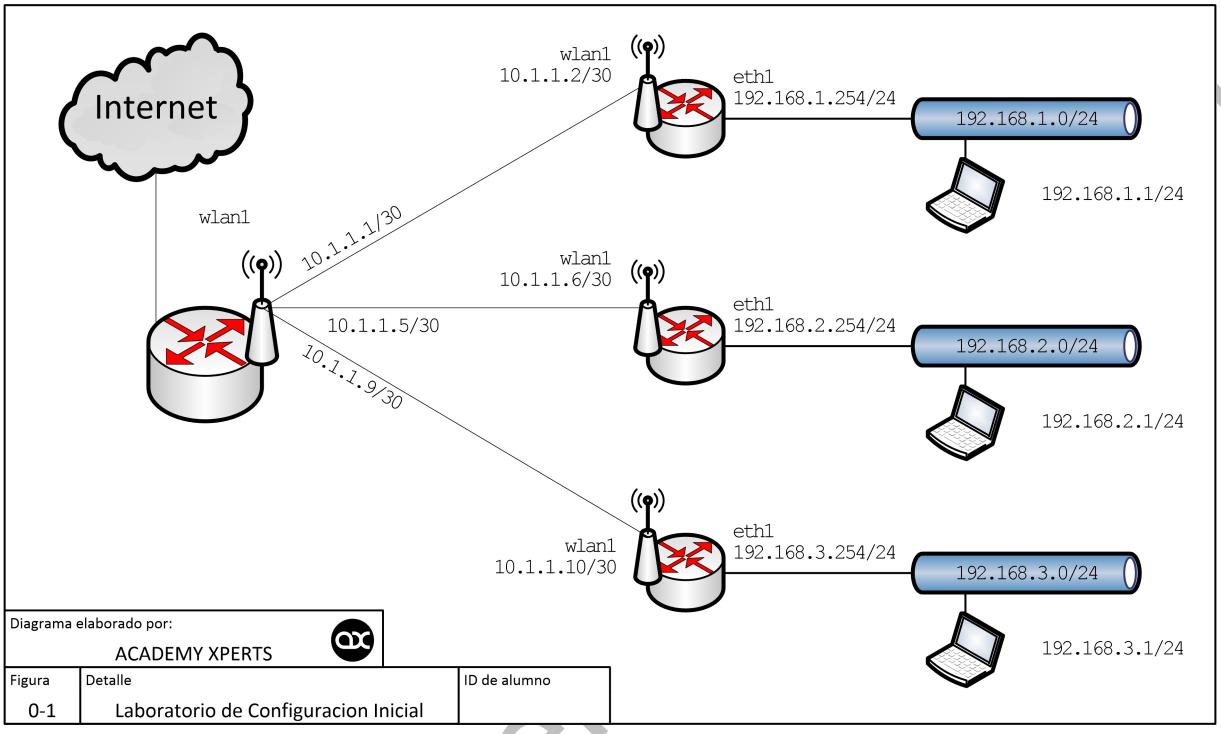
Número de Versión	6.33.5	6.33.3	6.33.2
	No se muestra el L2MTU en VLAN cuando se hace un export compacto		
netwatch	Se hace que el netwatch trabaje de una forma más precisa con el ping time-out		
ppp	Se hace que trabajen las condiciones -PPP active print radius- y -!radius!-		Se corrigió la adición de una regla de filtrado que se generaba dinámicamente en algunas configuraciones de /ip firewall filter
pppoe			Se mejoró la compatibilidad de MTU discovery con otros fabricantes Se hizo el MTU discovery más robusto Se corrigió nuevamente la conformidad con el RFC4638 (MTU más grande que 1488)
romon	No acepta mulitcast id Se corrigió una caida en RoMON si fast-path estaba activa		
smb	Se muestra el nombre de la interface correcta en los logs de debug del SMB		
ssh	Se corrigió session clean-up	Se evita el clean-up de la doble sesión	Se corrigió el intercambio de clave cuando se sigue el primer paquete kex
sshd	Se resolvió un problema de incompatibilidad de la clave compartida		
tile	Se corrigió una falla del kernel en la encriptación HW		
vrrp			Se corrigió arp=reply-only No advierte sobre la incompatibilidad de versión si el VRID no coincide Se permite que el VRRP trabaje detrás de reglas de Firewall y NAT Se corrigió el script on-backup
webfig	No se muestran los valores cero (zero) en las reglas de traslación CRS ingress/egress VLAN		
winbox	Se agregó + & - al IGMP proxy MFC Se agregó el menú LCD para el RB3011 Se permite especificar el umbral del monitor de tráfico (traffic-monitor threshold) en unidades k & M y se especifican aquellos que son en bits Se muestra los contadores por interface fast-path		

0 – Red de Trabajo

Laboratorio de configuración inicial

Laboratorio 0-1

En este laboratorio el estudiante realizará la configuración en su router para poder conectarse a la red de trabajo. Se debe aplicar los conocimientos necesarios para habilitar reglas de NAT, Rutas, configuración Wireless, DNS, etc., con el objetivo final de que la laptop del alumno obtenga acceso a internet a través de su router.



Los pasos a seguir se detallan en el Manual de Laboratorio.

Capítulo 1: DNS

Domain Name System

Qué es DNS

DNS es un sistema jerárquico distribuido de nombres para computadoras, servicios o cualquier recurso que esté conectado a Internet o a una red privada. Asocia información variada con los nombres de dominio asignados a cada una de las entidades que participan. Traduce los nombres de dominio (que pueden ser fácilmente memorizados) a direcciones IP.

El DNS es un componente esencial para la mayoría de los servicios de Internet ya que constituye el servicio de directorio primario de Internet.

El DNS distribuye la responsabilidad de asignar los nombres de dominio y mapear estos nombres a direcciones IP designando Servidores de Nombre Autoritativos para cada dominio. Los Servidores de Nombres Autoritativos son asignados para ser responsables de sus dominios soportados, y pueden delegar autoridad sobre sub-dominios a otros Servidores de Nombres. Este mecanismo proporciona un servicio distribuido y tolerante a falla, y fue diseñado para evitar la necesidad de una Base de Datos Central.

El DNS también especifica la funcionalidad técnica del servicio de base de datos. Define el protocolo DNS, una especificación detallada de las estructuras de datos y los intercambios de comunicación de datos usados en DNS, como parte de la Suite de Protocolos de Internet.

DNS ha estado en uso desde los 1980s. Previo a esta fecha existieron otros servicios de directorio que precedieron al DNS, pero no eran escalables y estaban basados originalmente en archivos de texto.

Internet mantiene dos espacios de nombre (namespaces) principales: la jerarquía de nombre de dominio, y los espacios de dirección IP (IP address spaces). El DNS mantiene la jerarquía de nombre de dominio y provee la traducción de servicios entre el DNS y los espacios de dirección.

Los servidores de nombre de Internet y un protocolo de comunicación son los que implementan el DNS. Un servidor de nombre DNS es un servidor que almacena los registros DNS de un nombre de dominio; un servidor de nombre DNS responde con respuestas a búsquedas contra su base de datos.

Los tipos más comunes de registros almacenados en la base de datos DNS son:

- **SOA** – Autoridad de zona DNS (DNS zone authority)
- **A** y **AAAA** – Direcciones IP
- **MX** – Intercambiadores de mail SMTP (SMTP mail exchangers)
- **NS** – Servidores de Nombre (Name Servers)
- **PTR** – Apuntadores para reverse DNS lookips
- **CNAME** – Alias de nombres de dominio

MikroTik DNS Caché

El Caché DNS se utiliza para minimizar los requerimientos DNS hacia un servidor DNS externo, así como también para disminuir el tiempo de resolución DNS. Es un Caché DNS simple con ítems locales.

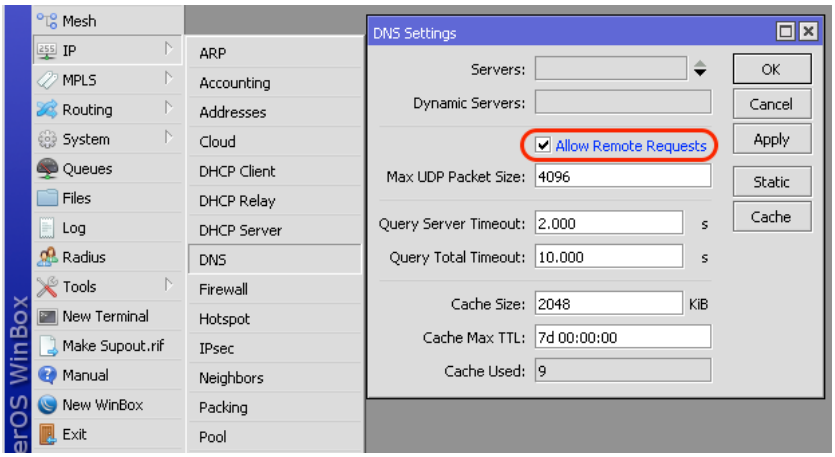
Un router MikroTik con la característica de DNS habilitada puede ser configurado como un DNS server para cualquier cliente DNS-compatible. Más aún, el router MikroTik puede especificarse como un Servidor DNS Primario en las configuraciones de DHCP Server. Cuando está habilitada la opción de requerimientos remotos (`allow-remote-requests`), el router MikroTik responde a los requerimientos DNS TCP y UDP en el puerto 53.

DNS utiliza principalmente el protocolo UDP en el puerto 53 para atender las solicitudes. Las búsquedas DNS se componen de una sola solicitud UDP desde el cliente, seguido de una sola respuesta UDP del servidor. El protocolo TCP se usa cuando el tamaño de los datos de respuesta es superior a 512 bytes, o para tareas tales como las transferencias de zona. Algunas implementaciones del resolver utilizan TCP para todas las consultas.

La facilidad DNS se usa para proveer la resolución de nombres de dominio para el mismo router, así como también para los clientes conectados a él.

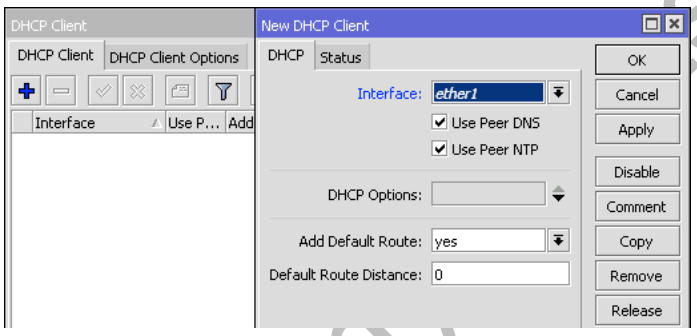
Parámetros

- **allow-remote-requests** *(yes | no; default: no):* Especifica si se va a permitir o no los requerimientos remotos de la red. Cuando se activa esta opción (yes) el router MikroTik se convierte en un DNS Caché
- **cache-max-ttl** *(time; default: 1w):* Especifica el time-to-live para los registros en caché. Esto significa que los registros del caché expirarán de manera incondicional después de que se haya alcanzado el tiempo especificado en cache-max-ttl. Si un servidor DNS especifica un TTL más pequeño que el establecido en este parámetro, el RouterOS respetará dicho TTL.
- **cache-size** *(integer: 512..10240; default: 2048KiB):* Especifica el tamaño del DNS en KiB. Nótese que el RouterOS trabaja bien hasta con un caché de tamaño 10240 KiB, más allá de ese valor se puede empezar a notar lentitud en los tiempos de respuesta de la resolución DNS.
- **cache-used** *(read-only: integer):* Muestra el tamaño del caché usado actualmente en KiB
- **servers** *(IPv4/IPv6 address list; default: 0.0.0.0):* Aquí se especifican los servidores DNS. Cuando se lo hace por CLI se deben separar los servers por una coma.



Notas importantes

- Si en `/ip dhcp-client` se configura la propiedad `use-peer-dns=yes`, entonces el DNS Primario (`primary-dns`) en `/ip dns` cambiará a la dirección DNS dada por el DHCP Server.
- Si se utiliza `allow-remote-requests`, debe asegurarse de limitar el acceso a su server sobre los protocolos TCP y UDP



Ejemplo de configuración DNS

Para configurar `159.148.60.2` como servidor DNS Primario, y permitir que el Router sea usado como un servidor DNS, se debe realizar lo siguiente:

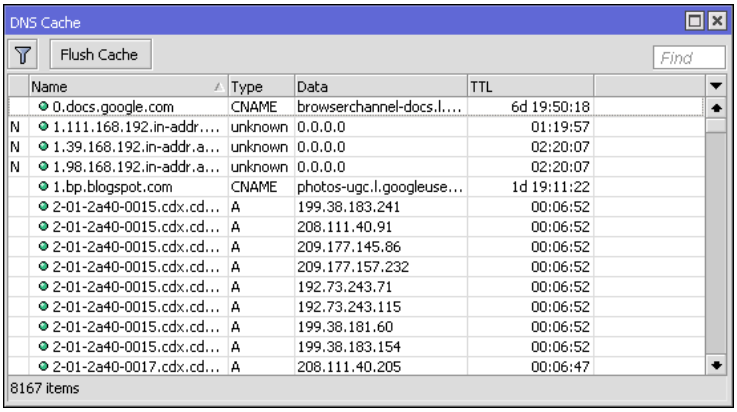
```
/ip dns set servers=159.148.60.2 allow-remote-requests=yes
/ip dns print
                  servers: 8.8.8.8
  allow-remote-requests: yes
             cache-size: 2048KiB
          cache-max-ttl: 1w
             cache-used: 7KiB
```

Monitoreo del Cache

`/ip dns cache all`

Este menú provee una lista de todos los registros de direcciones (DNS tipo “A”) almacenados en el server

- **`address`** *(read-only: IP address):* Dirección IP del host
- **`name`** *(read-only: name):* Nombre DNS del host
- **`ttl`** *(read-only: time):* Tiempo de vida del registro
- **`type`** *(read-only: time):* Tipo de registro DNS



Borrando el DNS Cache (flushing)

```
/ip dns cache flush
```

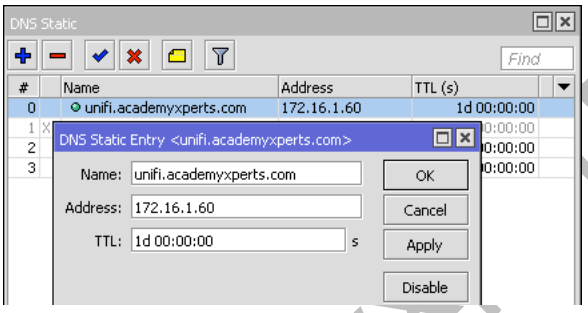
- **`flush`**: limpia el caché DNS interno

Entradas de DNS Estático

El MikroTik RouterOS tiene una característica de funcionamiento como DNS Server embebido en el DNS Cache. Esto permite enlazar los nombres de dominio particulares con las respectivas direcciones IP y avisar de estos links (enlaces) a los clientes DNS que usan el router como su servidor DNS. Esta característica también se puede usar para proveer información “falsa” a los clientes de la red. Por ejemplo, cuando se resuelva cualquier requerimiento DNS para un cierto conjunto de dominios (o para todo el internet) se re direccione a su propia página

El server es capaz de resolver requerimientos DNS basados en expresiones regulares básicas POSIX. por lo tanto, múltiples requerimientos pueden coincidir con la misma entrada. En caso de que una entrada no esté acorde con los estándares de nombre DNS, es considerada como una expresión regular y se la marca con la etiqueta “R”. La lista es ordena y chequeada de arriba hacia abajo. Las expresiones regulares se chequean primero, y luego los registros planos.

- **`address`** *(IP address):* Dirección IP para resolver el nombre de dominio
- **`name`** *(text):* Nombre DNS a ser resuelto para una dirección específica. Puede ser una expresión regular.
- **`ttl`** *(time):* Tiempo de vida del registro DNS



Notas importantes

- No es posible realizar la búsqueda reversa de DNS (reverse DNS lookup) de las entradas de expresión regular. Se puede sin embargo agregar un registro plano adicional con la misma dirección IP y especificar algún nombre para dicho registro.
- Recuerde que el significado de un punto (.) en una expresión regular es cualquier carácter, por lo que la expresión debería ser “escapada” apropiadamente. Por ejemplo, si se necesita emparejar (match) cualquier cosa dentro del dominio `example.com`, pero no todos los dominios que terminan con `example.com`, como `www.another-example.com`, entonces debe configurar `name=".*\\.example\\.com"`
- La concordancia de la expresión regular es significativamente más lenta que la de las entradas planas, por lo que se aconseja reducir al mínimo el número de reglas de expresiones regulares y optimizar las expresiones existentes.

Por ejemplo, si se desea agregar una entrada DNS estática para que www.example.com sea resuelta a la dirección IP 10.0.0.1, se debe configurar de la siguiente forma:

```
/ip dns static add name www.example.com address=10.0.0.1
/ip dns static print
Flags: D - dynamic, X - disabled, R - regexp
 #     NAME              ADDRESS                 TTL
 0     www.example.com   10.0.0.1                1d
```

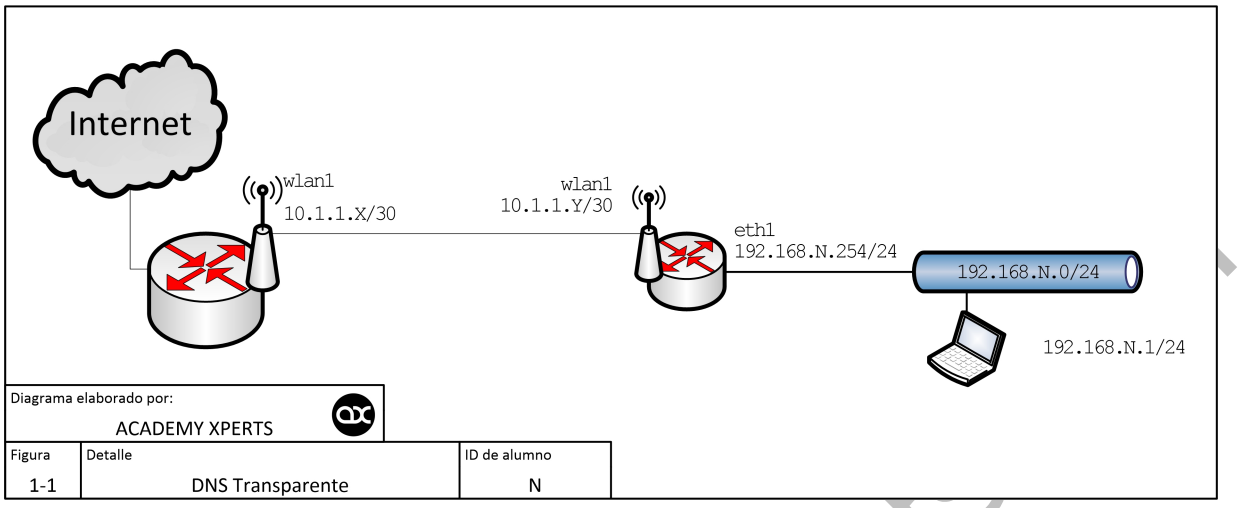
Preguntas de repaso del Capítulo 1

1. En qué protocolo y puertos trabaja DNS
2. Qué beneficios tiene que un router MikroTik se configure como un DNS Caché
3. Cuál es la función del parámetro `allow-remote-requests`
4. Cuál es el tamaño máximo de caché recommendable para un óptimo funcionamiento del Caché DNS en un router MikroTik?
5. Qué sucede cuando se excede el tamaño máximo recomendable para cache-size

Laboratorio 1-1: DNS Transparente

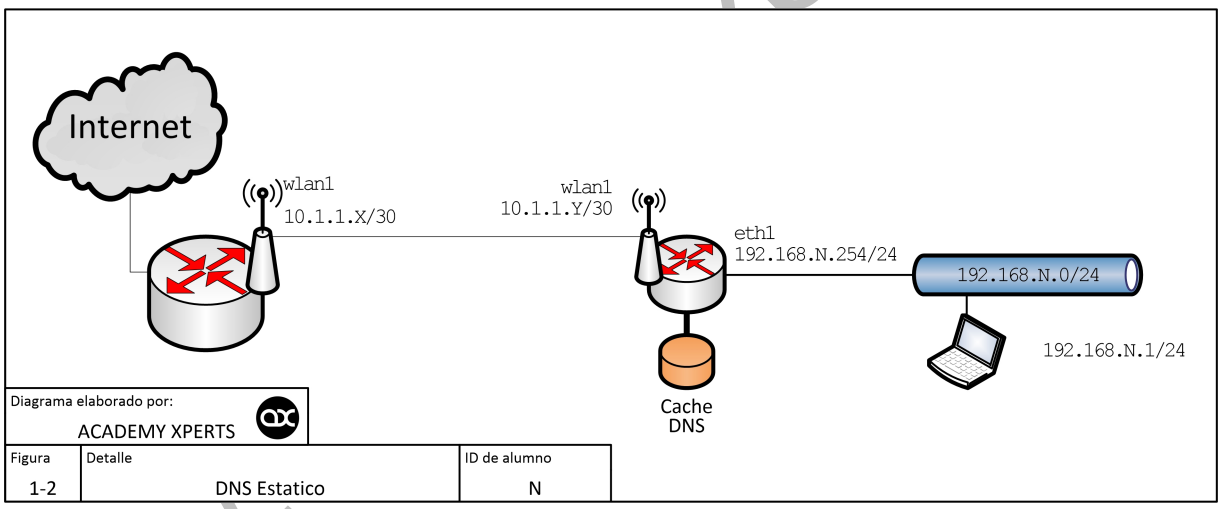
En este laboratorio el estudiante activará el DNS caché. Capturará el tráfico DNS y lo direccionará a su propio router.

Se debe comprobar el funcionamiento del DNS transparente. Los pasos a seguir se detallan en el Manual de Laboratorio.



Laboratorio 1-2: DNS Estático

En este laboratorio el estudiante deberá comprobar el funcionamiento de las entradas estáticas DNS. Los pasos a seguir se detallan en el Manual de Laboratorio.



Capítulo 2: DHCP

Dynamic Host Configuration Protocol

El Protocolo de Configuración Dinámica de Host (Dynamic Host Configuration Protocol) se usa para la fácil distribución de direcciones IP en una red. La implementación de MikroTik RouterOS incluye las partes Server y Cliente, y es compatible con el RFC2131.

El router soporta un servidor individual para cada interface tipo Ethernet. El servidor DHCP de MikroTik RouterOS soporta las funciones básicas de proveer a cada cliente que solicita los siguientes valores:

- Reserva (lease) de dirección IP/máscara (mask)
- Default gateway
- Nombre de dominio
- Servidor(es) DNS
- Servidor(es) WINS, para clientes Windows

Para que el servidor DHCP trabaje, se debe configurar también un Pool de direcciones IP y las redes DHCP (DHCP networks). Se debe tener cuidado de no incluir la dirección IP del propio servidor DHCP.

- DHCP es inseguro y debería ser usado únicamente en redes confiables

También se puede repartir las direcciones IP a los clientes DHCP usando el servidor RADIUS. Los parámetros soportados por un servidor RADIUS son los siguientes:

Access-Request:

- NAS-Identifier – Identidad del Router
- NAS-IP-Address – Dirección IP del mismo router
- NAS-Port – ID de sesión única
- NAS-Port-Type – Ethernet
- Calling-Station-Id – Identificador del Cliente (`active-client-id`)
- Framed-IP-Address – Dirección IP del Cliente (`active-address`)
- Called-Station-Id – Nombre del servidor DHCP
- User-Name – Dirección MAC del Cliente (`active-mac-address`)
- Password - ""

Access-Accept:

- Framed-IP-Address – Dirección IP que será asignada al cliente
- Framed-Pool – Pool de direcciones IP desde la cual se asignará la dirección IP al Cliente
- Rate-Limit – Limitación de la tasa de datos (datarate) para clientes DHCP. El formato es:
  - `rx-rate[/tx-rate] [rx-burst-rate[/tx-burst-rate] [rx-burst-threshold[/tx-burst-threshold] [rx-burst-time[/tx-burst-time][priority] [rx-rate-min[/tx-rate-min]]]]`.
  - Todas las tasas deben ser números con '`k`' (`1,000s`) o '`M`' (`1,000,000s`).
  - Si no se especifica el `tx-rate`, `rx-rate` es como `tx-rate`.
  - Lo mismo para `tx-burst-rate` y `tx-burst-threshold` y `tx-burst-time`. Si ambos `rx-burst-threshold` y `tx-burst-threshold` no se especifican (pero se especifica `burst-rate`), `rx-rate` y `tx-rate` se usan como burst thresholds.
  - Si ambos `rx-burst-time` y `tx-burst-time` no se especifican, se usa `1s` como default.
  - La prioridad toma valores entre `1..8`, donde 1 es la más alta prioridad y 8 la más baja.
  - Si `rx-rate-min` y `tx-rate-min` no se especifican, se usan los valores de `rx-rate` y `tx-rate` values.
  - Los valores de `rx-rate-min` y `tx-rate-min` no pueden exceder los valores `rx-rate` y `tx-rate`.
- Ascend-Data-Rate – Limitación de la tasa de datos `tx/rx` si se proveen múltiples atributos
  - Primero limita la tasa de datos `tx`
  - Segundo la tasa de datos `rx`
  - Si se usa junto con `Ascend-Xmit-Rate`, se especifica una tasa `rx`.
  - `0` si es ilimitado
- Ascend-Xmit-Rate – Limitación de la tasa de datos `tx`. Puede ser usada para especificar únicamente el límite `tx` en lugar de enviar dos atributos secuenciales `Ascend-Data-Rate` (en ese caso `Ascend-Data-Rate` especificará la tasa de recepción). `0` si es ilimitado.
- Session-Timeout – Tiempo reserva (lease) máximo (`lease-time`)

Nota importante

- El servidor DHCP requiere una interface real para recibir los paquetes Ethernet crudos. Si la interface es una interface `bridge`, entonces el `bridge` debe tener una interface real anexada como un puerto a ese `bridge` en el cual recibirá los paquetes crudos. El servidor DHCP no puede funcionar correctamente en una interface tonta (dummy) o bridge vacío (empty bridge)

Escenario de comunicación DHCP

- DHCP Server siempre escucha en UDP 67
- DHCP Cliente escucha en UDP 68
- La negociación inicial involucra la comunicación entre direcciones broadcast.
- En algunas fases el “sender” usa dirección origen 0.0.0.0 y/o direcciones destino 255.255.255.255
- Se debe tener cuidado de esto cuando se construye un firewall

Proceso DHCP

- DHCP Discovery

  `src-mac=<client>, dst-mac=<broadcast>, protocol=udp, src-ip=0.0.0.0:68, dst-ip=255.255.255.255:67`

- DHCP Offer

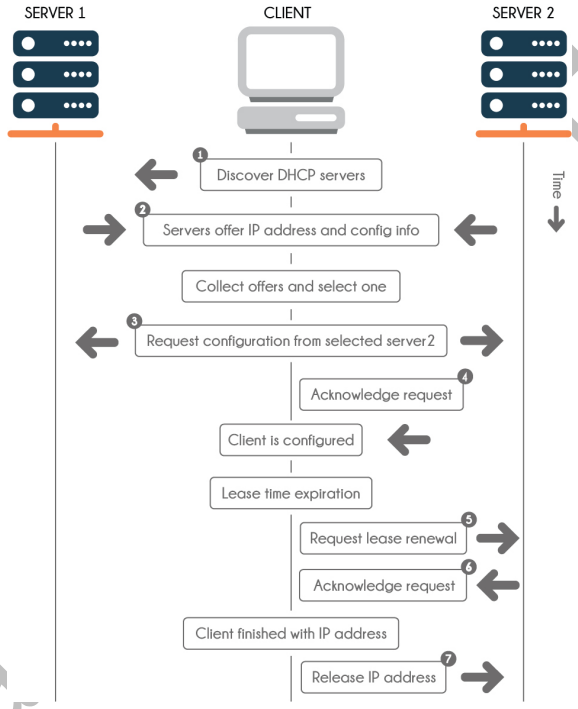
  `src-mac=<DHCP-server>, dst-mac=<broadcast>, protocol=udp, src-ip=<DHCP-server>:67, dst-ip=255.255.255.255:67`

- DHCP Request

  `src-mac=<client>, dst-mac=<broadcast>, protocol=udp, src-ip=0.0.0.0:68, dst-ip=255.255.255.255:67`

- DHCP Acknowledgment

  `src-mac=<DHCP-server>, dst-mac=<broadcast>, protocol=udp, src-ip=<DHCP-server>:67, dst-ip=255.255.255.255:67`



1 El cliente descubre un servidor DHCP mediante la difusión de un mensaje de descubrimiento a la dirección de broadcast limitada (`255.255.255.255`) por la máscara de subred local. Si un router está presente y configurado para comportarse como un agente de reenvío de BOOTP, la solicitud se pasa a otros servidores DHCP en diferentes subredes. La emisión del cliente incluye su identificador único, que es la dirección MAC del cliente. Los servidores DHCP que reciben el mensaje de descubrir pueden determinar la red del cliente al ver la siguiente información:

   - **Por cuál de las interfaces de red del servidor llegó la solicitud?** Esto le dice al servidor que el cliente está en la red a la que dicha interfaz está conectado, o que el cliente está utilizando un agente de reenvío de BOOTP conectado a esa red.
   - **Incluye la solicitud la dirección IP de un agente de retransmisión (relay) BOOTP**? Cuando un requerimiento pasa a través de un agente relay, el agente relay inserta su dirección en la cabecera del requerimiento. Cuando el server detecta la dirección de un agente relay, conoce que la porción de la dirección indica la dirección de red del cliente ya que el agente relay debe estar conectado a la red del cliente.
   - **Está subneteada la red del cliente?** El servidor consulta la tabla de netmasks, enfocándose en la dirección del agente relay o la dirección de la interface de red que recibe el requerimiento. Una vez que el servidor conoce la mascara de subred utilizada, puede determinar qué porción de la dirección de red es la parte del host, y entonces seleccionar la dirección IP apropiada para el cliente.

2 Después de determinar la red del cliente, los servidores DHCP, seleccionan una dirección IP adecuada y verifican que la dirección ya no esté en uso. Los servidores DHCP entonces responden al cliente emitiendo un mensaje de oferta que incluye la dirección IP seleccionada e información acerca de los servicios que se pueden configurar para

el cliente. Cada servidor reserva temporalmente la dirección IP ofrecida hasta que se pueda determinar si el cliente la va a usar.

3 El cliente selecciona la mejor oferta (en función del número y tipo de los servicios ofrecidos) y difunde una solicitud, especificando la dirección IP del servidor que hizo la mejor oferta. El broadcast asegura de que todos los servidores DHCP que responden saben que el cliente ha elegido un servidor, y esos servidores sin elegir pueden cancelar las reservas de las direcciones IP que habían ofrecido.

4 El servidor seleccionado asigna la dirección IP para el cliente, el almacenamiento de la información en el área de almacenamiento de datos DHCP, y envía un acuse de recibo al cliente. El mensaje de reconocimiento contiene los parámetros de configuración de red para el cliente. El cliente comprueba la dirección IP para asegurarse de que ningún otro sistema lo está utilizando, y continúa el arranque para unirse la red.

5 El cliente controla el tiempo de concesión, y cuando ha transcurrido un período de tiempo determinado, el cliente envía un nuevo mensaje al servidor elegido para aumentar su tiempo de concesión.

6 El servidor DHCP que recibe la solicitud extiende el tiempo de concesión si todavía se adhiere a la política de arrendamiento de local establecido por el administrador. Si el servidor no responde dentro de los 20 segundos, el cliente emite un pedido, por lo que uno de los otros servidores DHCP puede extender el contrato de arrendamiento.

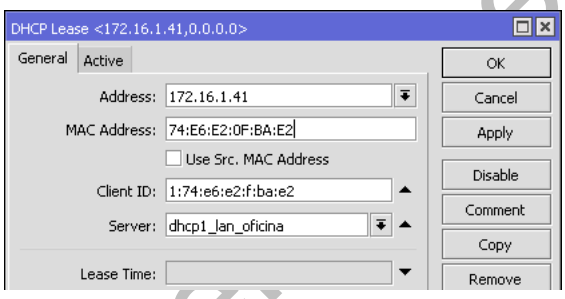
Cuando el cliente ya no necesita la dirección IP, envía un mensaje de notificación al servidor que se está liberando la dirección IP. Esto puede suceder durante un cierre ordenado y también se puede hacer manualmente.

Identificación del Cliente DHCP

El Servidor DHCP puede rastrear la asociación con un cliente particular basado en su identificación. La identificación puede ser ejecutada en 2 formas

- Basado en la opción “client-id” (RFC2132)
- Basado en la MAC address, si la opción “client-id” no está disponible

La opción “hostname” permite a los clientes RouterOS enviar identificación adicional al server



RouterOS y DHCP Cliente

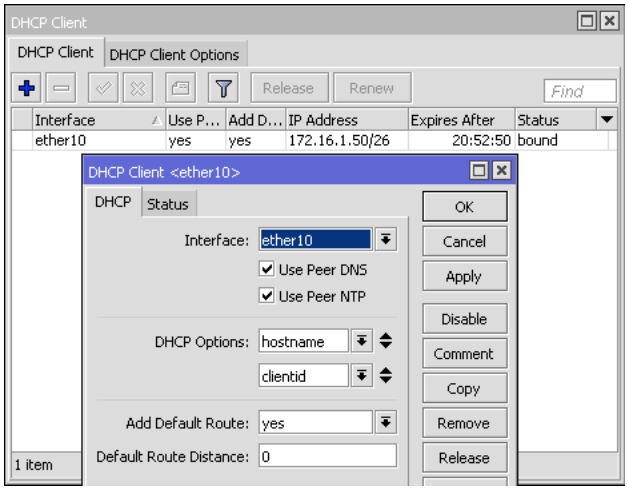
Solo puede haber un cliente DHCP activo por interface. El cliente aceptará:

- 1 dirección
- 1 netmask
- 1 default gateway
- 2 direcciones DNS server
- 2 direcciones NTP server

La IP recibida con el netmask será añadido a la interfaz correspondiente

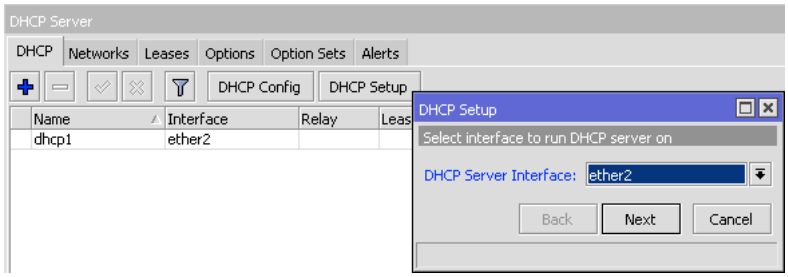
El default gateway será añadido a la tabla de ruteo como una entrada dinámica

Nota Importante: Si ya existe una ruta por default instalada antes de que el cliente DHCP obtenga una, la ruta obtenida por el cliente DHCP se mostrará como inválida



Configuración Rápida (Quick Setup)

RouterOS posee un comando que permite configurar fácilmente un servidor DHCP. Supongamos que se quiere configurar un servidor DHCP en la interface `ether1` para asignar direcciones desde la `192.168.0.2` hasta la `192.168.0.254`, las mismas que pertenecen a la red `192.168.0.0/24`. El gateway y server DNS es `192.168.0.1`



Desde el menú `/ip dhcp-server` se debe ejecutar el comando `setup` y seguir las instrucciones

```
/ip dhcp-server> setup
Select interface to run DHCP server on

dhcp server interface: ether1
Select network for DHCP addresses

dhcp address space: 192.168.0.0/24
Select gateway for given network

gateway for dhcp network: 192.168.0.1
Select pool of ip addresses given out by DHCP server

addresses to give out: 192.168.0.2-192.168.0.254
Select DNS servers

dns servers: 192.168.0.1
Select lease time

lease time: 3d
```

El asistente de configuración habrá creado la siguiente configuración basado en las respuestas de arriba:

```
/ip dhcp-server> print
Flags: X - disabled, I - invalid
  #   NAME            INTERFACE RELAY           ADDRESS-POOL LEASE-TIME ADD-ARP
  0   dhcp1           ether1    0.0.0.0         dhcp_pool1   3d         no

/ip dhcp-server network print
  # ADDRESS            GATEWAY         DNS-SERVER      WINS-SERVER     DOMAIN
  0 192.168.0.0/24       192.168.0.1       192.168.0.1

/ip pool print
  # NAME              RANGES
  0 dhcp_pool1        192.168.0.2-192.168.0.254
```

IPv6

- A partir de la v5.8 el RouterOS soporta la delegación de prefijo IPv6 acorde a RFC3315 y RFC3633
- A partir de la v5.9, la configuración del server DHCPv6 se movió al menú `/ipv6 sub-menu`

DHCP-Server

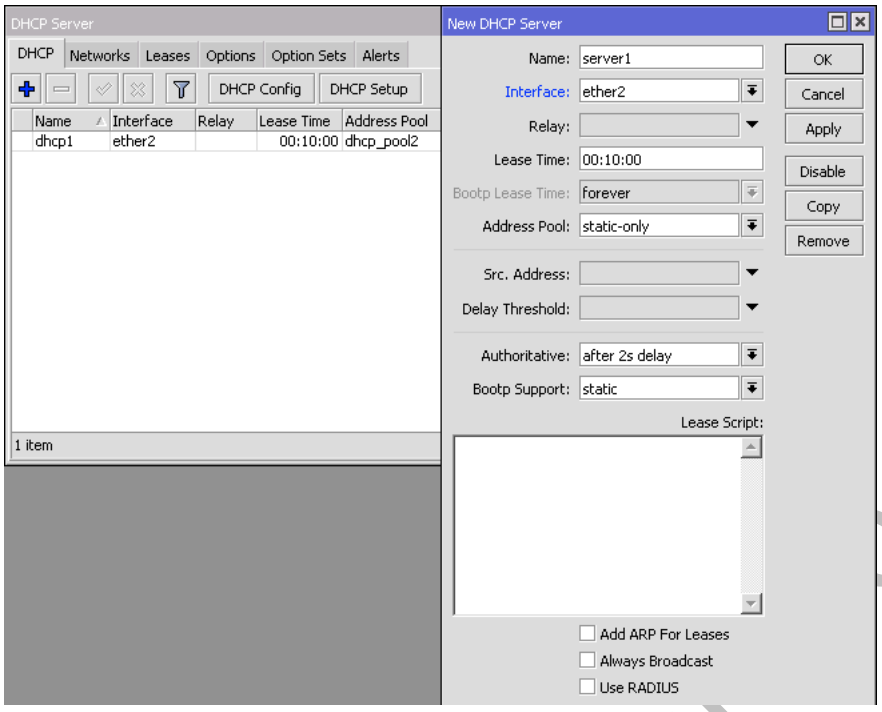
Solo puede haber un DHCP Server por combinación interface/relay

Para crear un DHCP Server se debe tener

- Dirección IP en la interface DHCP Server deseada
- Pool de direcciones (address pool) para los clientes
- Información sobre la red DHCP

Los 3 requerimientos deben corresponder

“Lease on Disk” debería ser usado para reducir el número de escrituras en el drive (útil con flash drives)



Parámetros

- **`add-arp`** *(yes | no; Default: no)* – Define si se agrega una entrada dinámica ARP. Si se configura `add-arp=no`, se debería habilitar en esa interface ya sea el modo ARP, o las entradas estáticas ARP deberían ser definidas administrativamente en el sub menú `/ip arp`
- **`address-pool`** *(string | static-only; Default: static-only)* – Pool de direcciones IP desde la cual tomará las direcciones IP para los clientes.
    - Si se configura `address-pool=static-only`, entonces únicamente se permitirá a los clientes que tienen una reserva (lease) estática (agregada en el submenú lease).
- **`always-broadcast`** *(yes | no; Default: no)* – Siempre envía respuestas (replies) como broadcasts
- **`authoritative`** *(after-10sec-delay | after-2sec-delay | yes | no; Default: after-2sec-delay)* – Esta opción cambia la forma como el server responde a los requerimientos DHCP
    - **`yes`** – Responde a los clientes que solicitan una dirección que no está disponible en este servidor. El servidor DHCP enviará un reconocimiento negativo (negative acknowledgment DHCPNAK)
    - **`no`** – El servidor DHCP ignora los requerimientos de los clientes de direcciones que no están disponibles en este servidor
    - **`after-10sec-delay`** – Los requerimientos con "`segundos<10`" serán procesados como `authoritative=no`. Los requerimientos con "`segundos>=10`" será procesados como `authoritative=yes`.
    - **`after-2sec-delay`** – Los requerimientos con "`segundos<2`" serán procesados como `authoritative=no`. Los requerimientos con "`segundos>=2`" será procesados como `authoritative=yes`.
    - Si todos los requerimientos con "`secs < x`" deben ser ignorados, entonces se debería usar la configuración `delay-threshold=x`.
- **`bootp-support`** *(none | static | dynamic; Default: static)* – Soporte para clientes BOOTP
    - **`none`** – No responde a requerimientos BOOTP
    - **`static`** – Ofrece únicamente reservas (leases) estáticas para clientes BOOTP
    - **`dynamic`** – Ofrece reservas (leases) estáticas y dinámicas para clientes BOOTP
- **`delay-threshold`** *(time | none; Default: none)* – Si el campo de segundos (sec) en el paquete DHCP es más pequeño que el `delay-threshold`, entonces este paquete es ignorado. Si se configura `delay-threshold=none`, entonces no hay threshold, es decir que se procesarán todos los paquetes.
- **`interface`** *(string; Default:)* – Especifica la interface en la cual el server estará corriendo
- **`lease-script`** *(string; Default:)* – Script que será ejecutado después que la reserva (lease) es asignada o deasignada. Las siguientes son variables "globales" internas que pueden ser usadas en el script:
    - **`leaseBound`** - Configurado como "1" si está confinado, si no configurado como "0"
    - **`leaseServerName`** – Nombre del server DHCP
    - **`leaseActMAC`** – Dirección MAC activa
    - **`leaseActIP`** – Dirección IP activa

- **`lease-time`** *(time; Default: 72h)* – Especifica el tiempo que un cliente puede utilizar la dirección asignada. El cliente intentará renovar esta dirección después de que haya transcurrido la mitad de este tiempo (lease-time) y requerirá una nueva dirección después de que el tiempo límite expira.
- **`name`** *(string; Default:)* – Nombre de referencia
- **`relay`** *(IP; Default: 0.0.0.0)* – La dirección IP del relay desde el que este server DHCP debería procesar los requerimientos:
  - **`0.0.0.0`** – El server DHCP se utilizará únicamente para las solicitudes directas de los clientes (no DHCP realmente permitido)
  - **`255.255.255.255`** – El servidor DHCP debería ser usado para cualquier requerimiento entrante de un DHCP relay excepto para aquellos que son procesados por otro servidor DHCP que existe en `/ip dhcp-server`
- **`src-address`** *(IP; Default: 0.0.0.0)* – La dirección con la cual el cliente DHCP debe enviar las solicitudes a fin de renovar una reserva (lease) de dirección IP. Si existe únicamente una dirección estática en la interface del servidor DHCP y el `src-address=0.0.0.0`, entonces se usará la dirección estática. Si existen múltiples direcciones en la interface, una dirección en la misma subred que el rango de direcciones dada se debe utilizar.
- **`use-radius`** *(yes | no; Default: no)* - Especifica si se utiliza un servidor RADIUS para las asignaciones dinámicas (dynamic leases)

Comando específico del menú

- **`setup`** *()* – Inicia el asistente de configuración del server DHCP, el cual lo guía a través de los pasos necesarios para fácilmente crear toda la configuración necesaria.

Configuración del almacenamiento de reservas (leases)

`/ip dhcp-server config`

Este submenú permite configurar cuan frecuente las reservas DHCP serán almacenadas en disco. Si estas reservas deberían ser grabadas en disco en cada cambio de lease, entonces debería ocurrir una gran cantidad de escrituras en disco, lo cual es muy malo para la Compact Flash (especialmente si los tiempos de reserva son muy cortos). Para minimizar las escrituras a disco, todos los cambios se almacenan en disco cada `store-leases-disk` segundos. Adicionalmente las reservas siempre se graban en disco cuando se realiza el shutdown y reboot de forma adecuada.

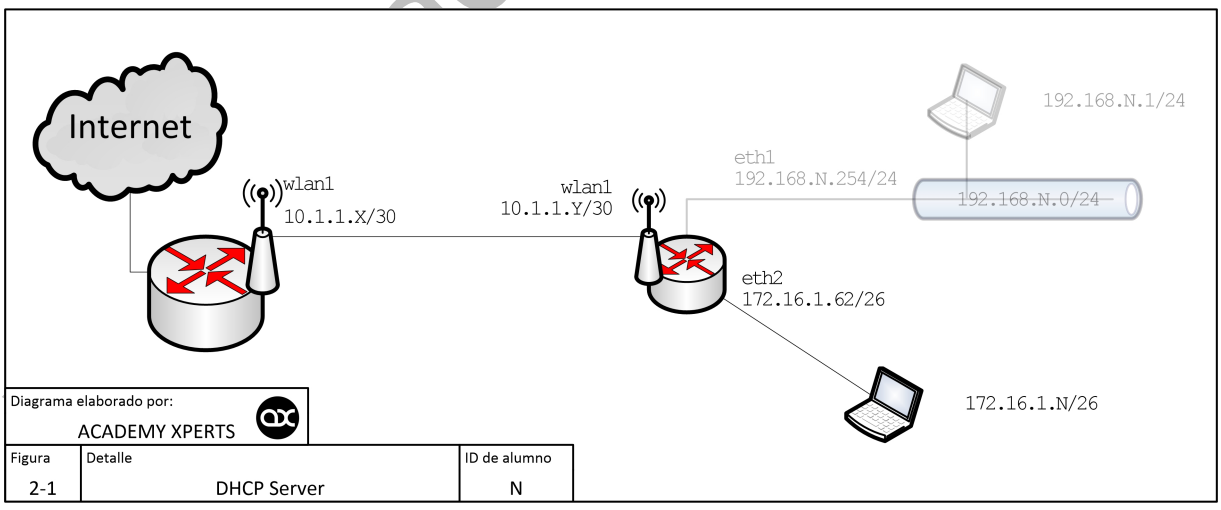
Nota Importante

- Los cambios manuales a las reservas (leases), adición/remoción de leases estáticos, la remoción de las reservas (leases) dinámicas, ocasionarán que los cambios empujen a que las reservas sean almacenadas

Laboratorio 2-1: DHCP Server

En este laboratorio el estudiante deberá:

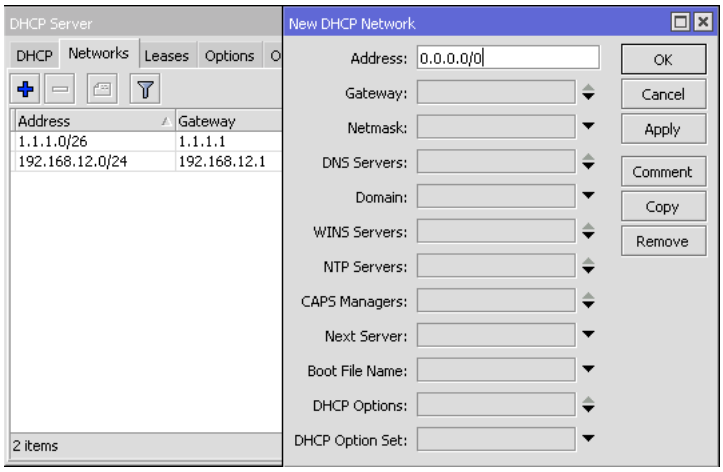
- Configurar manualmente el DHCP Server.
- Asignar automáticamente la dirección IP a la laptop del cliente
- Confirmar que la laptop sigue teniendo salida a Internet



DHCP-Networks

En el menú DHCP Networks se puede configurar opciones específicas DHCP para una red particular. Algunas de las opciones están integradas en el RouterOS, otras pueden ser asignadas en forma “cruda”

- El DHCP Server está habilitado para enviar cualquier opción
- El DHCP Client puede recibir únicamente las opciones implementadas



Parámetros

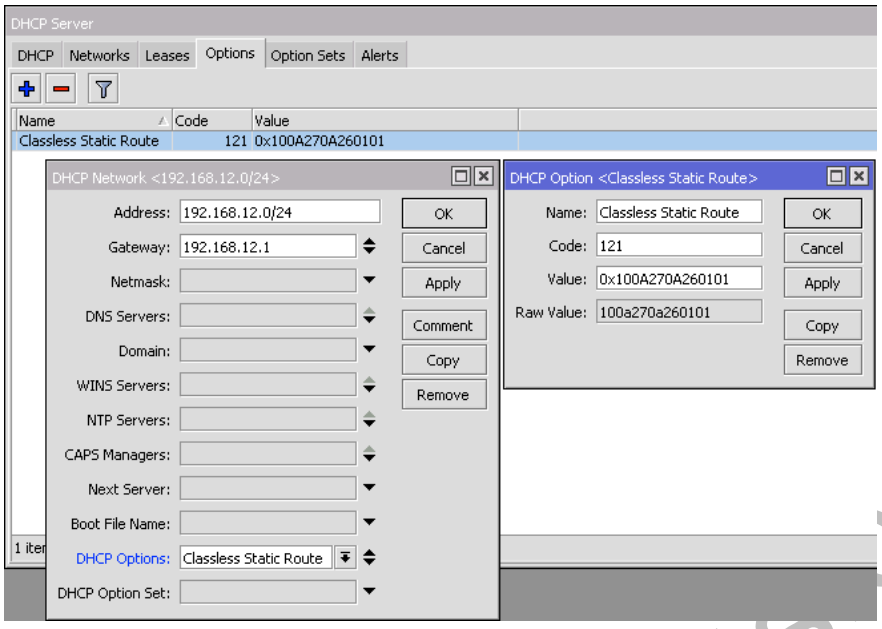
- **`address`** *(IP/netmask; Default:)* – Especifica la red del servidor(es) DHCP de donde se tomarán las direcciones que se reservarán.
- **`boot-file-name`** *(string; Default:)* – Nombre del archivo de boot
- **`caps-manager`** *(string; Default:)* – Lista separada por comas de las direcciones IP de uno o más administradores de sistema CAPsMan.
- **`dhcp-option`** *(string; Default:)* – Añade opciones adicionales de DHCP.
- **`dns-server`** *(string; Default:)* – El cliente DHCP usará estos valores como los servidores DNS por default. Se pueden especificar dos servidores DNS separados por coma, para que sean usados por el cliente DHCP como servidores DNS primario y secundario.
- **`domain`** *(string; Default)* – El cliente DHCP usará este valor como la configuración del "Dominio DNS" para el adaptador de red.
- **`gateway`** *(IP; Default: 0.0.0.0)* – Especifica el gateway por default que será usado por el cliente DHCP.
- **`netmask`** *(integer: 0..32; Default: 0)* – Especifica la máscara de red actual de será usada por el cliente DHCP. Si `netmask=0` se utilizará el netmask de la dirección de red
- **`next-server`** *(IP; Default)* – Especifica la dirección IP del próximo servidor que se usará en `bootstrap`
- **`ntp-server`** *(IP; Default:)* – El cliente DHCP usará estos valores como los servidores NTP por default. Se pueden especificar hasta dos servidores NTP separados por coma, los mismos que serán usados por el cliente DHCP como servidores NTP primario y secundario.
- **`wins-server`** *(IP; Default:)* – El cliente DHCP Windows usará estos parámetros como los servidores WINS por default. Se pueden especificar dos servidores WINS separados por coma, para que sean utilizados por el cliente DHCP como servidores WINS primario y secundario.

DHCP-Options

`/ip dhcp-server option`

Con la ayuda de las listas de opciones DHCP, es posible definir opciones personalizadas adicionales del Servidor DHCP.

- En el menú DHCP Networks se puede configurar opciones específicas DHCP para una red particular
- Algunas de las opciones están integradas en el RouterOS, otras pueden ser asignadas en forma "cruda"
- DHCP Server está habilitado para enviar cualquier opción



Extensiones de fabricante BOOTP y Opciones DHCP

http://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml

El protocolo Bootstrap (BOOTP, según el RFC951) describe un protocolo bootstrap IP/UDP que permite a los clientes que no poseen disco, descubrir su propia dirección IP, la dirección de un server host, y el nombre de un archivo que será cargado en memoria para ser ejecutado.

El protocolo DHCP (Dynamic Host Configuration Protocol, según el RFC2131) provee un marco de referencia para la configuración automática de las direcciones IP en los hosts. El documento "DHCP Options and BOOTP Vendor Information Extensions" conforme al RFC2132, describe las opciones de DHCP, algunas de las cuales pueden también usarse con BOOTP. Adicionalmente las opciones DHCP se describen también en otros RFCs

Tag	Name	Data Length	Meaning	Reference
0	Pad	0	None	[RFC2132]
1	Subnet Mask	4	Subnet Mask Value	[RFC2132]
2	Time Offset	4	Time Offset in Seconds from UTC (note: deprecated by 100 and 101)	[RFC2132]
3	Router	N	N/4 Router addresses	[RFC2132]
4	Time Server	N	N/4 Timeserver addresses	[RFC2132]
5	Name Server	N	N/4 IEN-116 Server addresses	[RFC2132]
6	Domain Server	N	N/4 DNS Server addresses	[RFC2132]
7	Log Server	N	N/4 Logging Server addresses	[RFC2132]
8	Quotes Server	N	N/4 Quotes Server addresses	[RFC2132]
9	LPR Server	N	N/4 Printer Server addresses	[RFC2132]
10	Impress Server	N	N/4 Impress Server addresses	[RFC2132]
11	RLP Server	N	N/4 RLP Server addresses	[RFC2132]
12	Hostname	N	Hostname string	[RFC2132]
13	Boot File Size	2	Size of boot file in 512 byte chunks	[RFC2132]
14	Merit Dump File	N	Client to dump and name the file to dump it to	[RFC2132]
15	Domain Name	N	The DNS domain name of the client	[RFC2132]
16	Swap Server	N	Swap Server address	[RFC2132]
17	Root Path	N	Path name for root disk	[RFC2132]
18	Extension File	N	Path name for more BOOTP info	[RFC2132]
19	Forward On/Off	1	Enable/Disable IP Forwarding	[RFC2132]
20	SrcRte On/Off	1	Enable/Disable Source Routing	[RFC2132]
21	Policy Filter	N	Routing Policy Filters	[RFC2132]
22	Max DG Assembly	2	Max Datagram Reassembly Size	[RFC2132]
23	Default IP TTL	1	Default IP Time to Live	[RFC2132]
24	MTU Timeout	4	Path MTU Aging Timeout	[RFC2132]
25	MTU Plateau	N	Path MTU Plateau Table	[RFC2132]
26	MTU Interface	2	Interface MTU Size	[RFC2132]
27	MTU Subnet	1	All Subnets are Local	[RFC2132]
28	Broadcast Address	4	Broadcast Address	[RFC2132]
29	Mask Discovery	1	Perform Mask Discovery	[RFC2132]
30	Mask Supplier	1	Provide Mask to Others	[RFC2132]
31	Router Discovery	1	Perform Router Discovery	[RFC2132]
32	Router Request	4	Router Solicitation Address	[RFC2132]
33	Static Route	N	Static Routing Table	[RFC2132]
34	Trailers	1	Trailer Encapsulation	[RFC2132]
35	ARP Timeout	4	ARP Cache Timeout	[RFC2132]

Tag	Name	Data Length	Meaning	Reference
36	Ethernet	1	Ethernet Encapsulation	[RFC2132]
37	Default TCP TTL	1	Default TCP Time to Live	[RFC2132]
38	Keepalive Time	4	TCP Keepalive Interval	[RFC2132]
39	Keepalive Data	1	TCP Keepalive Garbage	[RFC2132]
40	NIS Domain	N	NIS Domain Name	[RFC2132]
41	NIS Servers	N	NIS Server Addresses	[RFC2132]
42	NTP Servers	N	NTP Server Addresses	[RFC2132]
43	Vendor Specific	N	Vendor Specific Information	[RFC2132]
44	NETBIOS Name Srv	N	NETBIOS Name Servers	[RFC2132]
45	NETBIOS Dist Srv	N	NETBIOS Datagram Distribution	[RFC2132]
46	NETBIOS Node Type	1	NETBIOS Node Type	[RFC2132]
47	NETBIOS Scope	N	NETBIOS Scope	[RFC2132]
48	X Window Font	N	X Window Font Server	[RFC2132]
49	X Window Manager	N	X Window Display Manager	[RFC2132]
50	Address Request	4	Requested IP Address	[RFC2132]
51	Address Time	4	IP Address Lease Time	[RFC2132]
52	Overload	1	Overload "sname" or "file"	[RFC2132]
53	DHCP Msg Type	1	DHCP Message Type	[RFC2132]
54	DHCP Server Id	4	DHCP Server Identification	[RFC2132]
55	Parameter List	N	Parameter Request List	[RFC2132]
56	DHCP Message	N	DHCP Error Message	[RFC2132]
57	DHCP Max Msg Size	2	DHCP Maximum Message Size	[RFC2132]
58	Renewal Time	4	DHCP Renewal (T1) Time	[RFC2132]
59	Rebinding Time	4	DHCP Rebinding (T2) Time	[RFC2132]
60	Class Id	N	Class Identifier	[RFC2132]
61	Client Id	N	Client Identifier	[RFC2132]
62	NetWare/IP Domain	N	NetWare/IP Domain Name	[RFC2242]
63	NetWare/IP Option	N	NetWare/IP sub Options	[RFC2242]
64	NIS-Domain-Name	N	NIS+ v3 Client Domain Name	[RFC2132]
65	NIS-Server-Addr	N	NIS+ v3 Server Addresses	[RFC2132]
66	Server-Name	N	TFTP Server Name	[RFC2132]
67	Bootfile-Name	N	Boot File Name	[RFC2132]
68	Home-Agent-Addrs	N	Home Agent Addresses	[RFC2132]
69	SMTP-Server	N	Simple Mail Server Addresses	[RFC2132]
70	POP3-Server	N	Post Office Server Addresses	[RFC2132]
71	NNTP-Server	N	Network News Server Addresses	[RFC2132]
72	WWW-Server	N	WWW Server Addresses	[RFC2132]
73	Finger-Server	N	Finger Server Addresses	[RFC2132]
74	IRC-Server	N	Chat Server Addresses	[RFC2132]
75	StreetTalk-Server	N	StreetTalk Server Addresses	[RFC2132]
76	STDA-Server	N	ST Directory Assist. Addresses	[RFC2132]
77	User-Class	N	User Class Information	[RFC3004]
78	Directory Agent	N	directory agent information	[RFC2610]
79	Service Scope	N	service location agent scope	[RFC2610]
80	Rapid Commit	0	Rapid Commit	[RFC4039]
81	Client FQDN	N	Fully Qualified Domain Name	[RFC4702]
82	Relay Agent Information	N	Relay Agent Information	[RFC3046]
83	iSNS	N	Internet Storage Name Service	[RFC4174]
84	REMOVED/Unassigned			[RFC3679]
85	NDS Servers	N	Novell Directory Services	[RFC2241]
86	NDS Tree Name	N	Novell Directory Services	[RFC2241]
87	NDS Context	N	Novell Directory Services	[RFC2241]
88	BCMCS Controller Domain Name list			[RFC4280]
89	BCMCS Controller IPv4 address option			[RFC4280]
90	Authentication	N	Authentication	[RFC3118]
91	client-last-transaction-time option			[RFC4388]
92	associated-ip option			[RFC4388]
93	Client System	N	Client System Architecture	[RFC4578]
94	Client NDI	N	Client Network Device Interface	[RFC4578]
95	LDAP	N	Lightweight Directory Access Protocol	[RFC3679]
96	REMOVED/Unassigned			[RFC3679]
97	UUID/GUID	N	UUID/GUID-based Client Identifier	[RFC4578]
98	User-Auth	N	Open Group's User Authentication	[RFC2485]
99	GEOCONF_CIVIC			[RFC4776]
100	PCode	N	IEEE 1003.1 TZ String	[RFC4833]
101	TCode	N	Reference to the TZ Database	[RFC4833]
102-107	REMOVED/Unassigned			[RFC3679]
108	REMOVED/Unassigned			[RFC3679]
109	Unassigned			[RFC3679]
110	REMOVED/Unassigned			[RFC3679]
111	Unassigned			[RFC3679]
112	Netinfo Address	N	NetInfo Parent Server Address	[RFC3679]
113	Netinfo Tag	N	NetInfo Parent Server Tag	[RFC3679]
114	URL	N	URL	[RFC3679]
115	REMOVED/Unassigned			[RFC3679]
116	Auto-Config	N	DHCP Auto-Configuration	[RFC2563]

Tag	Name	Data Length	Meaning	Reference
117	Name Service Search	N	Name Service Search	[RFC2937]
118	Subnet Selection Option	4	Subnet Selection Option	[RFC3011]
119	Domain Search	N	DNS domain search list	[RFC3397]
120	SIP Servers DHCP Option	N	SIP Servers DHCP Option	[RFC3361]
121	Classless Static Route Option	N	Classless Static Route Option	[RFC3442]
122	CCC	N	CableLabs Client Configuration	[RFC3495]
123	GeoConf Option	16	GeoConf Option	[RFC6225]
124	V-I Vendor Class		Vendor-Identifying Vendor Class	[RFC3925]
125	V-I Vendor-Specific Information		Vendor-Identifying Vendor-Specific Information	[RFC3925]
126	Removed/Unassigned			[RFC3679]
127	Removed/Unassigned			[RFC3679]
128	PXE - undefined (vendor specific)			[RFC4578]
128	Etherboot signature. 6 bytes: E4:45:74:68:00:00			
128	DOCSIS "full security" server IP address			
128	TFTP Server IP address (for IP Phone software load)			
129	PXE - undefined (vendor specific)			[RFC4578]
129	Kernel options. Variable length string			
129	Call Server IP address			
130	PXE - undefined (vendor specific)			[RFC4578]
130	Ethernet interface. Variable length string.			
130	Discrimination string (to identify vendor)			
131	PXE - undefined (vendor specific)			[RFC4578]
131	Remote statistics server IP address			
132	PXE - undefined (vendor specific)			[RFC4578]
132	IEEE 802.1Q VLAN ID			
133	PXE - undefined (vendor specific)			[RFC4578]
133	IEEE 802.1D/p Layer 2 Priority			
134	PXE - undefined (vendor specific)			[RFC4578]
134	Diffserv Code Point (DSCP) for VoIP signalling and media streams			
135	PXE - undefined (vendor specific)			[RFC4578]
135	HTTP Proxy for phone-specific applications			
136	OPTION_PANA_AGENT			[RFC5192]
137	OPTION_V4_LOST			[RFC5223]
138	OPTION_CAPWAP_AC_V4	N	CAPWAP Access Controller addresses	[RFC5417]
139	OPTION-IPv4_Address-MoS	N	a series of suboptions	[RFC5678]
140	OPTION-IPv4_FQDN-MoS	N	a series of suboptions	[RFC5678]
141	SIP UA Configuration Service Domains	N	List of domain names to search for SIP User Agent Configuration	[RFC6011]
142	OPTION-IPv4_Address-ANDSF	N	ANDSF IPv4 Address Option for DHCPv4	[RFC6153]
143	Unassigned			
144	GeoLoc	16	Geospatial Location with Uncertainty	[RFC6225]
145	FORCERENEW_NONCE_CAPABLE	1	Forcerenew Nonce Capable	[RFC6704]
146	RDNSS Selection	N	Information for selecting RDNSS	[RFC6731]
147-149	Unassigned			[RFC3942]
150	TFTP server address			[RFC5859]
150	Etherboot			
150	GRUB configuration path name			
151	status-code	N+1	Status code and optional N byte text message describing status.	[RFC6926]
152	base-time	4	Absolute time (seconds since Jan 1, 1970) message was sent.	[RFC6926]
153	start-time-of-state	4	Number of seconds in the past when client entered current state.	[RFC6926]
154	query-start-time	4	Absolute time (seconds since Jan 1, 1970) for beginning of query.	[RFC6926]
155	query-end-time	4	Absolute time (seconds since Jan 1, 1970) for end of query.	[RFC6926]
156	dhcp-state	1	State of IP address.	[RFC6926]
157	data-source	1	Indicates information came from local or remote server.	[RFC6926]
158	OPTION_V4_PCP_SERVER	Variable; the minimum length is 5.	Includes one or multiple lists of PCP server IP addresses; each list is treated as a separate PCP server.	[RFC7291]
159	OPTION_V4_PORTPARAMS	4	This option is used to configure a set of ports bound to a shared IPv4 address.	[RFC7618]
160	DHCP Captive-Portal	N	DHCP Captive-Portal	[RFC-wkumari-dhc-capport-16]

Tag	Name	Data Length	Meaning	Reference
161-174	Unassigned			[RFC3942]
175	Etherboot (Tentatively Assigned - 2005-06-23)			
176	IP Telephone (Tentatively Assigned - 2005-06-23)			
177	Etherboot (Tentatively Assigned - 2005-06-23)			
177	PacketCable and CableHome (replaced by 122)			
178-207	Unassigned			[RFC3942]
208	PXELINUX Magic	4	magic string = F1:00:74:7E	[RFC5071][Deprecated]
209	Configuration File	N	Configuration file	[RFC5071]
210	Path Prefix	N	Path Prefix Option	[RFC5071]
211	Reboot Time	4	Reboot Time	[RFC5071]
212	OPTION_6RD	18 + N	OPTION_6RD with N/4 6rd BR addresses	[RFC5969]
213	OPTION_V4_ACCESS_DOMAIN	N	Access Network Domain Name	[RFC5986]
214-219	Unassigned			
220	Subnet Allocation Option	N	Subnet Allocation Option	[RFC6656]
221	Virtual Subnet Selection (VSS) Option			[RFC6607]
222-223	Unassigned			[RFC3942]
224-254	Reserved (Private Use)			
255	End	0	None	[RFC2132]

De acuerdo al protocolo DHCP, se regresa al cliente DHCP únicamente si solicita este parámetro, especificando el respectivo código en el atributo de petición de la lista de parámetros (código 55). Si el código no está incluido en el atributo de lista de parámetros, el servidor DHCP no lo enviará al cliente DHCP.

Propiedades

- **`code`** *(integer:1..254; Default:)* – Código de opción DHCP. Todos los códigos están disponibles en
  - http://www.iana.org/assignments/bootp-dhcp-parameters
- **`name`** *(string; Default)* – Nombres descriptivo de la opción
- **`value`** *(string; Default)* – Valor del Parámetro.
  - A partir de la v6.8 los tipos de datos disponibles para las opciones son:
    - `0xXXXX` – string hexadecimal (trabaja también en v5)
    - `'XXXXX'` – string (trabaja también en v5 pero sin '' alrededor del texto)
    - `$(XXXXX)` - variable (actualmente no hay variables para el server)
    - `'10.10.10.10'` – Dirección IP
    - `s'10.10.10.10'` – Dirección IP convertida a string
    - `'10'` – Número decimal
    - `s'10'` – Número decimal convertido a string
  - Ahora es posible combinar los tipos de datos en uno, por ejemplo: `"0x01'vards'$(HOSTNAME)"`
  - Por ejemplo, si el HOSTNAME es 'kvm', entonces el valor crudo será `0x0176617264736b766d`
- **`raw-value`** *(HEX string)* – Campo de solo lectura que muestra el valor crudo de opción DHCP (el formato en realidad lo envió)

Ejemplo: Ruta Estática sin Clase (Classless Route)

Una ruta sin clase agrega una ruta especificada en la tabla de ruta de los clientes. En este ejemplo agregará

```
dst-address=160.0.0.0/24 gateway=10.1.101.1
dst-address=0.0.0.0/0 gateway=10.1.101.1
```

De acuerdo al RFC 3442:

- La primera parte es el netmask (`"18" = netmask /24`)
- La segunda parte es la parte significante de la red destino (`"A00000" = 160.0.0`)
- La tercera parte es la dirección IP del gateway (`"0A016501" = 10.1.101.1`)
- Entonces hay partes de la ruta por default, netmask destino (`0x00 = 0.0.0.0/0`) seguido por la ruta por default (`0x0A016501 = 10.1.101.1`)

```
/ip dhcp-server option
add code=121 name=classless value=0x18A000000A016501000A016501
/ip dhcp-server network
set 0 dhcp-option=classless
```

Resultado:

```
/ip route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip,
```

```
b - bgp, o - ospf, m - mme, B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 ADS  0.0.0.0/0                          10.1.101.1         0
 1 ADS  160.0.0.0/24                       10.1.101.1         0
```

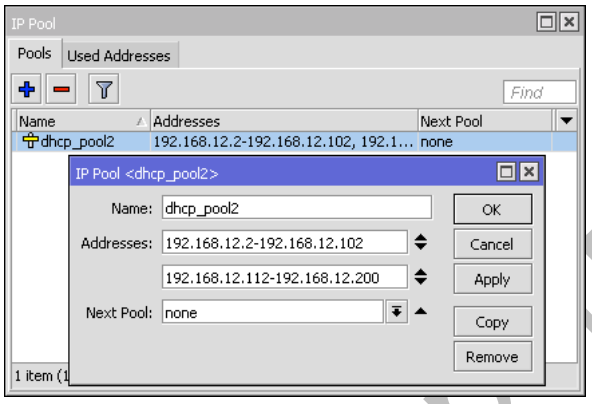
Configuración AutoProxy

```
/ip dhcp-server option
add code=252 name=auto-proxy-config value="'http://autoconfig.something.lv/wpad.dat'"
```

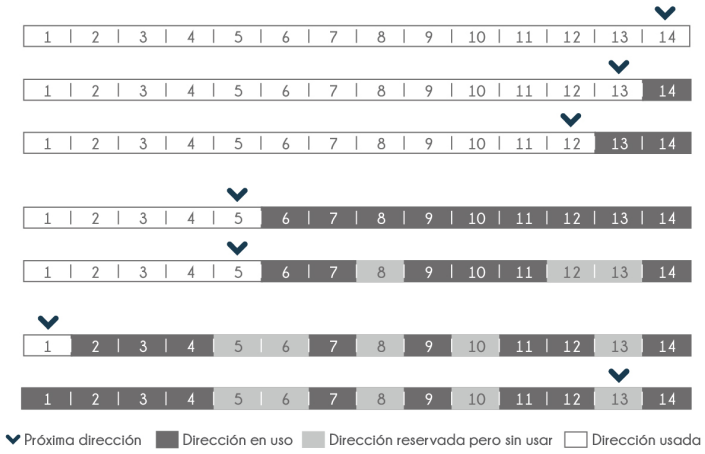
Pool de direcciones IP

Los IP Address Pool son usados para definir el rango de direcciones IP para distribución dinámica (DHCP, PPP, HotSpot)

- Los Address Pool deben excluir las direcciones ya ocupadas (direcciones estáticas o de servers)
- Es posible asignar más de un rango a un Pool
- Es posible encadenar varios Pools usando la opción “Next Pool”



Pool de Direcciones en Acción



DHCP-Leases

`/ip dhcp-server lease`

Esta opción se utiliza para monitorear y administrar las reservas (leases) del servidor. Las reservas emitidas se muestran aquí como entradas dinámicas. También se puede agregar leases estáticos para generar una dirección IP específica para un cliente en particular (identificado por dirección MAC).

Generalmente, la concesión DHCP se asigna de la siguiente manera:

- Una reserva sin usar está en estado de espera
- Si un cliente solicita una dirección IP, el servidor elige una dirección
- Si el cliente recibe una dirección estática asignada, el lease se convierte en ofrecido, y luego es ligado con el respectivo tiempo de concesión (lease)
- Si el cliente recibe una dirección dinámica (tomada desde un pool de direcciones IP), el router envía un paquete `ping` y espera 0.5 segundos. Durante este tiempo, el lease es marcado como prueba.
- En el caso en el que la dirección no responda, el lease se convierte en ofrecido, y luego es ligado con el respectivo tiempo de reserva (lease time)
- En otro caso, el lease se convierte en ocupado por el período de reserva (hay un comando para volver a probar todas las direcciones ocupadas), y la solicitud del cliente permanece sin responder (el cliente intentará nuevamente dentro de poco).

Un cliente puede liberar la dirección asignada. El lease dinámico es removido, y la dirección ubicada es retornada al pool de direcciones. Pero la reserva estática se vuelve ocupada hasta que el cliente vuelve a adquirir la dirección.

Nota importante

- Las direcciones IP asignadas estáticamente no son sondeadas

Propiedades

- **address** *(IP; Default)* – Especifica la dirección IP (o el pool IP) para la asignación estática. Si se configura `address=0.0.0.0`, se utilizará el pool del servidor.
- **address-list** *(string; Default)* – Lista de direcciones a la cual la dirección será agregada si el lease es asociado
- **always-broadcast** *(yes | no; Default)* – Envía todas las respuestas como broadcasts
- **block-access** *(yes | no; Default: no)* – Bloquea el acceso para este cliente
- **client-id** *(string; Default)* – Si se especifica este valor, debe coincidir la opción de "identificador del cliente" DHCP con el requerimiento.
- **lease-time** *(time; Default: 0s)* – Especifica el tiempo que el cliente puede usar la dirección. Si se configura `lease-time=0` entonces la reserva nunca expira.
- **mac-address** *(MAC; Default: 00:00:00:00:00:00)* – Si especifica este valor, debe coincidir con la dirección MAC del cliente.
- **src-mac-address** *(MAC; Default)* – Dirección MAC origen
- **use-src-mac** *(MAC; Default)* – Usa esta dirección MAC origen en lugar de `src-mac-address`

Propiedades de solo lectura

- **active-address** *(IP)* – La dirección IP actual para este lease
- **active-client-id** *(string)* – El client-id actual del cliente
- **active-mac-address** *(MAC)* – La dirección MAC actual del cliente
- **active-server** *(list)* – Servidor DHCP actual, que sirve este cliente
- **agent-circuit-id** *(string)* – ID de Circuito del agente DHCP Relay
- **agent-remote-id** *(string)* – ID Remoto, configurado por el agente DHCP Relay
- **blocked** *(flag)* – Especifica si es el lease está bloqueado
- **expires-after** *(time)* – Tiempo hasta que la reserva (lease) expira
- **host-name** *(text)* – Muestra la opción `host-name` desde el último requerimiento DHCP recibido
- **radius** *(yes | no)* – Muestra si el lease dinámico es autenticado o no por RADIUS
- **rate-limit** – Configura el límite de tasa para un lease activo. El formato es:
  - `rx-rate[/tx-rate] [rx-burst-rate[/tx-burst-rate] [rx-burst-threshold[/tx-burst-threshold] [rx-burst-time[/tx-burst-time]]]]`.
  - Todas las tasas deben ser números con 'k' (`1,000s`) o 'M' (`1,000,000s`).
  - Si no se especifica el `tx-rate`, `rx-rate` es como `tx-rate`.
  - Lo mismo para `tx-burst-rate` y `tx-burst-threshold` y `tx-burst-time`. Si ambos `rx-burst-threshold` y `tx-burst-threshold` no se especifican (pero se especifica `burst-rate`), `rx-rate` y `tx-rate` se usan como burst thresholds.
  - Si ambos `rx-burst-time` y `tx-burst-time` no se especifican, se usa `1s` como default.
- **server** *(string)* – Nombre del servidor que sirve a este cliente
- **status** *(waiting | testing | authorizing | busy | offered | bound)* – Estatus de la reserva:
  - **waiting** – Asignación estática no usada
  - **testing** - En prueba si es que esta dirección está siendo usada o no (únicamente para reservas dinámicas) haciéndole un ping con un timeout de 0.5s
  - **authorizing** – Espera por la respuesta del servidor RADIUS
  - **busy** – Esta dirección está asignada estáticamente a un cliente, o que ya existe en la red, por lo que no puede ser reservada
  - **offered** – El servidor ha ofrecido este lease a un cliente, pero no ha recibido confirmación del cliente.
  - **bound** – El servidor ha recibido la confirmación del cliente que acepta la dirección ofrecida, está usándola en este momento y liberará la dirección no más tarde que el tiempo estipulado en el parámetro `lease-time`

Comando específicos

- **check-status** (id) – Verifica el estatus de un lease dinámico ocupado, y lo libera en caso de que no responda
- **make-static** (id) – Convierte un lease dinámico en uno estático

DHCP-Alerts

`/ip dhcp-server alert`

Para encontrar servidores DHCP falsos tan pronto como aparecen en la red, se de usar la herramienta DHCP Alert. Esta herramienta monitoreará la interface Ethernet de todas las respuestas DHCP y verificará si esta respuesta proviene de un servidor DHCP válido. Si se detecta una respuesta de un servidor DHCP desconocido, se dispara una alerta:

Cuando el sistema alerta sobre un servidor DHCP falso, se puede ejecutar un script personalizado.

Como las respuestas DHCP pueden ser unicast, el "detector de DHCP falso" no puede recibir ninguna oferta de otros clientes DHCP. Para lidiar con esto, el detector de DHCP falso actúa también como un cliente DHCP, y envía las solicitudes de discover-dhcp una vez cada minuto.

Propiedades

- **`alert-timeout`** *(none | time; Default: none)* – Tiempo después del cual se olvidará la alerta. Si después de ese tiempo se detecta el mismo servidor falso, se generará una nueva alerta. Si se configura `alert-timeout=none`, ésta nunca expirará.
- **`interface`** *(string; Default:)* – Especifica la interface en la cual se ejecutará el buscador de servidor DHCP falso.
- **`on-alert`** *(string; Default:)* – Script para ejecutar, cuando se detecta un servidor DHCP desconocido.
- **`valid-server`** *(string; Default:)* – Lista de direcciones MAC de servidores DHCP válidos.

Propiedades de sólo lectura

- **`unknown-server`** *(string)* – Lista de direcciones MAC de servidores DHCP desconocidos que se han detectado. El servidor se remueve de esta lista después que se cumple el `alert-timeout`.

Comandos específicos

- **`reset-alert`** *(id)* – Limpia todas las alertas en una interface

DHCP-Client

El cliente DHCP de MikroTik RouterOS puede ser habilitado en cualquier interface tipo Ethernet.

El cliente aceptará

- Una dirección
- Netmask
- Default gateway
- Dos direcciones de servidor DNS

La dirección IP que se reciba será agregada a la interface con la respectiva netmask. El default gateway será agregado a la tabla de ruteo como una entrada dinámica. Si el cliente DHCP se deshabilita o no se renueva la dirección IP, entonces la ruta por default será removida.

Si ya existe una ruta por default instalada anterior a que el cliente DHCP obtenga una (ruta por default), la ruta obtenida por el cliente DHCP deberá ser mostrada como inválida.

El cliente DHCP RouterOS pide las siguientes opciones:

- option 1 - SUBNET_MASK
- option 3 - GATEWAY_LIST
- option 6 - TAG_DNS_LIST
- option 33 - STATIC_ROUTE
- option 42 - NTP_LIST
- option 121 - CLASSLESS_ROUTE

Opción

El cliente DHCP tiene la posibilidad de configurar las opciones que son enviadas al servidor DHCP. Por ejemplo, el host-name y la dirección MAC. La sintaxis es la misma que para las opciones de servidor DHCP.

Nota importante

- Esta característica está disponible desde la v6.0

Actualmente hay dos variables que pueden ser usadas en opciones:

- HOSTNAME
- CLIENT_MAC – Dirección MAC de la interface del cliente

IPv6

A partir de la v5.8 el cliente DHCP puede recibir prefijos delegados del servidor DHCPv6. Actualmente el prefijo recibido se agrega al pool IPv6, el cual puede ser usado más tarde por ejemplo en la configuración del server PPPoE.

A partir de la v5.9, la configuración del cliente DHCPv6 fue movida a `/ipv6 sub-menu`

Nota importante

- Si la interface usada por el cliente DHCP es parte de la configuración VRF, entonces la ruta por default y otras rutas recibidas del servidor DHCP serán agregadas a la tabla de ruteo VRF

Propiedades

`/ip dhcp-client`

- **`add-default-route`** *(yes | no | special-classless; Default: yes)* – Instalar la ruta predeterminada en la tabla de enrutamiento recibida del servidor DHCP. Por default el cliente RouterOS cumple con el RFC e ignora la opción 3 si se recibe la opción 121 (classless). Para forzar a que el cliente no ignore la opción 3 se configura `special-classless`. Este parámetro está disponible a partir de la v6rc12+

    - **yes** – Agrega la ruta sin clase (classless) si se recibe. Si no se la recibe entonces se agrega la ruta por default (viejo comportamiento)
    - **special-classless** – Agrega tanto la ruta sin clase (classless) si se recibe y la ruta por default (estilo MS)
- **client-id** *(string; Default:)* – Corresponde a la configuración sugerida por el administrador de red o ISP. Si no se especifica, entonces se enviará la dirección MAC del cliente
- **comment** *(string; Default:)* – Una descripción corta del cliente
- **default-route-distance** *(integer:0..255; Default:)* – Distancia de la ruta por default. Se aplica si `add-default-route=yes`.
- **disabled** *(yes | no; Default: yes)*
- **host-name** *(string; Default:)* – Nombre del host del cliente que se envía al servidor DHCP. Si no se especifica, se usará la identidad de sistema del cliente.
- **interface** *(string; Default:)* – Interface en la cual se ejecutará el cliente DHCP.
- **use-peer-dns** *(yes | no; Default: yes)* – Especifica si se acepta la configuración DNS anunciada por el servidor DHCP. Sobre escribirá las configuraciones puestas en `/ip dns`
- **use-peer-ntp** *(yes | no; Default: yes)* – Especifica si se acepta la configuración NTP anunciada por el servidor DHCP. Sobre escribirá las configuraciones puestas en `/system ntp client`

Estatus

El comando `/ip dhcp-client print detail` mostrará el estatus actual del cliente DHCP y las propiedades de solo lectura listadas en la tabla que se muestra a continuación:

- **address** *(IP/Netmask)* – La dirección IP y el netmask, el cual es asignado desde el servidor al cliente DHCP
- **dhcp-server** *(IP)* – La dirección IP del servidor DHCP.
- **expires-after** *(time)* – Tiempo cuando expira el lease (especificado por el servidor DHCP).
- **gateway** *(IP)* – Dirección IP del gateway el cual es asignado por el servidor DHCP
- **invalid** *(yes | no)* – Muestra si la configuración es inválida.
- **netmask** *(IP)*
- **primary-dns** *(IP)* – Dirección IP del servidor DNS primario, asignada por el servidor DHCP
- **primary-ntp** *(IP)* – Dirección IP del servidor NTP primario, asignada por el servidor DHCP
- **secondary-dns** *(IP)* – Dirección IP del servidor DNS secundario, asignada por el servidor DHCP
- **secondary-ntp** *(IP)* – Dirección IP del servidor NTP secundario, asignada por el servidor DHCP
- **status** *(bound | error | rebinding... | requesting... | searching... | stopped)* – Muestra es estatus del cliente DHCP

Comandos Específicos

- **release** (numbers) – Libera el enlace actual y reinicia el cliente DHCP
- **renew** (numbers) – Renueva el lease actual. Si la operación de renovación no se completa, el cliente trata de reinicializar el lease, es decir que inicia el proceso de petición de lease (rebind) como si no hubiese recibido aún una dirección IP

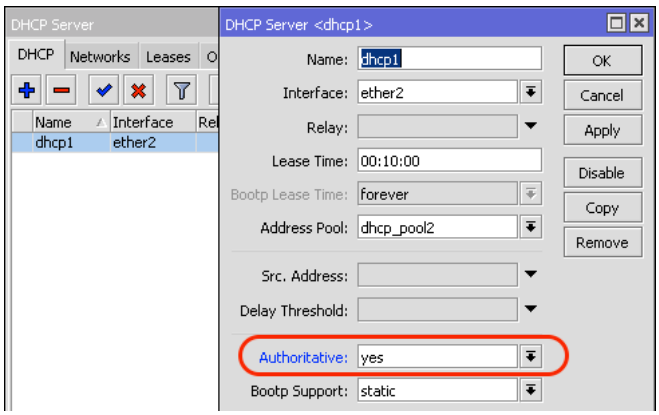
DHCP Autoritativo

authoritative=yes – Permite al DHCP Server responder a los broadcasts de cliente no conocidos y pide al cliente renovar la reserva (lease).

El cliente envía un broadcast solo si el unicast al server falla cuando se renueva el lease

Athoritative permite:

- Prevenir las operaciones de Rogue DHCP Servers
- Una adaptación más rápida de la red a cambios de configuración DHCP



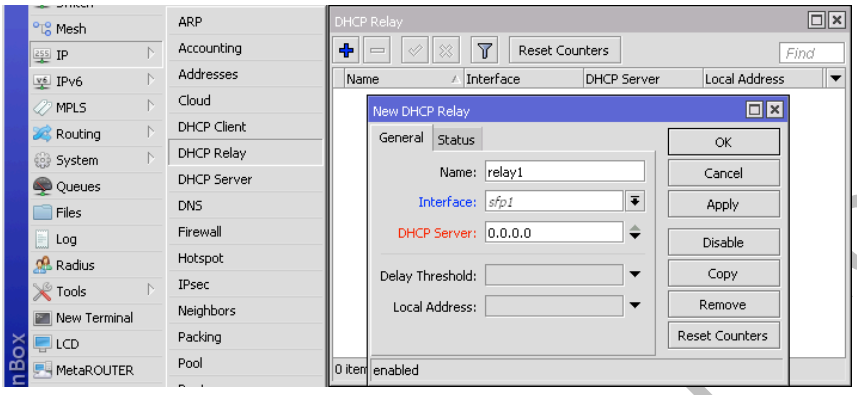
DHCP-Relay

`/ip dhcp-relay`

DHCP Relay es un proxy que es capaz de recibir una solicitud DHCP y reenviarla a un servidor DHCP real

DHCP Relay no elige el servidor DHCP en la lista `dhcp-server`, tan solo envía el requerimiento entrante a todos los servidores listados.

- DHCP Relay es un proxy que permite recibir un DHCP Discovery y un DHCP Request y reenviarlos al DHCP Server
- Solo puede haber un DHCP Relay entre el DHCP Server y el Cliente DHCP
- La comunicación DHCP con Relay no requiere de dirección IP en el Relay, sin embargo, la opción “local address” del Relay debe ser la misma que la configurada en la opción “relay address” del Server.



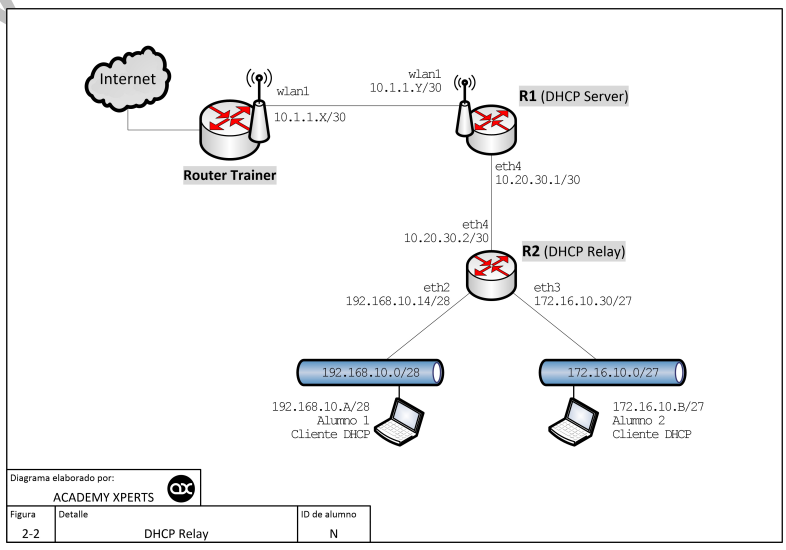
Propiedades

- **`add-relay-info`** *(yes | no; Default: no)* – Agrega la información del agente DHCP Relay si está habilitado de acuerdo al RFC 3046. El `Agent Circuit Circuit ID Sub-option` contiene la dirección MAC de una interface, el `Agent Remote ID Sub-option` contiene la dirección MAC del cliente desde el cual se recibió la solicitud.
- **`delay-threshold`** *(time | none; Default: none)* – Si el campo “`segundos`” en el paquete DHCP es más pequeño que `delay-threshold`, entonces este paquete es ignorado
- **`dhcp-server`** *(string; Default:)* – Lista de las direcciones IP de los servidores DHCP hacia donde deben ser enviados los requerimientos DHCP
- **`interface`** (string; Default:) – Nombre de la interface donde el DHCP Relay trabajará
- **`local-address`** *(IP; Default: 0.0.0.0)* – Dirección IP única de este DHCP Relay que se necesita para que el Server DHCP distinga los relays. Si se configura como `0.0.0.0`, la dirección IP se elegirá automáticamente.
- **`relay-info-remote-id`** *(string; Default:)* - El Relay usará este string en lugar de la dirección MAC del cliente cuando se construye la Opción 82 para ser enviada al DHCP Server. La opción 82 consiste de paquetes de interface que fueron recibidos de la dirección mac del cliente o del parámetro `relay-info-remote-id`
- **`name`** *(string; Default:)* – Nombre descriptivo del relay

Laboratorio 2-2: DHCP Relay

En este laboratorio el estudiante deberá:

- Configurar manualmente el DHCP Server.
- Asignar automáticamente la dirección IP a la laptop del cliente
- Confirmar que la laptop sigue teniendo salida a Internet



Capítulo 3: Firewall Filter

Estructura de Filtros de Firewall

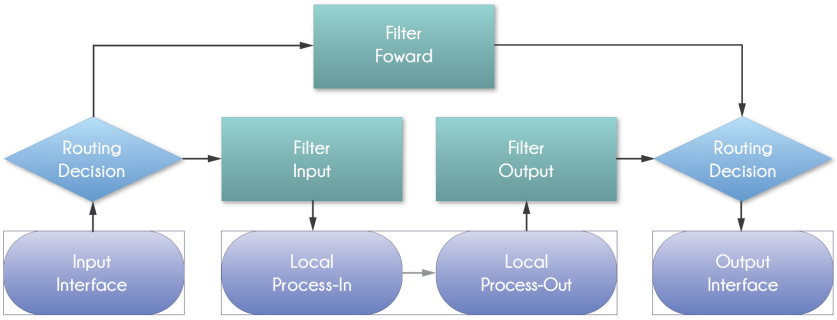
Las reglas de filtros de firewall están organizadas en `chains`. Existen “default chains” y “user-defined chains”

Los chain por default (default chains) son 3:

- **`input`**: procesa los paquetes enviados al router
- **`output`**: procesa los paquetes enviados por el router
- **`forward`**: procesa los paquetes enviados a través del router

Cada chain definido por el usuario (user-defined chain) debe estar subordinada a al menos una de los “chain” de default.

Diagrama de Estructura de Filtros de Firewall



Connection Tracking

Conntrack System: es el corazón del firewall. Obtiene y maneja la información de TODAS las conexiones activas

- Si se deshabilita el conntrack system se perderá la funcionabilidad del NAT y también la mayoría de los filtros y de las conexiones de mangle
- Cada entrada en la tabla conntrack representa un intercambio bidireccional de datos
- Conntrack hace uso de gran cantidad de recursos de CPU. Solo debe ser deshabilitado si no se usa el firewall

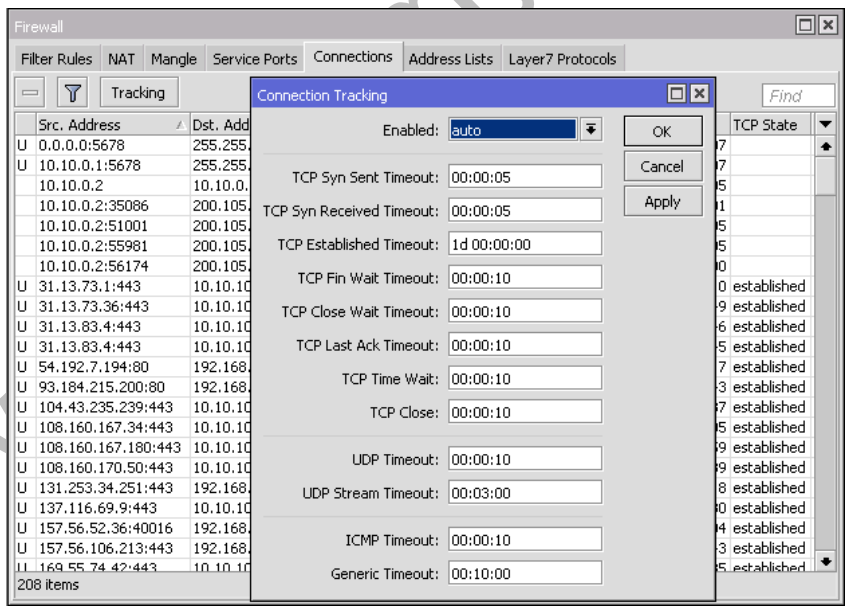
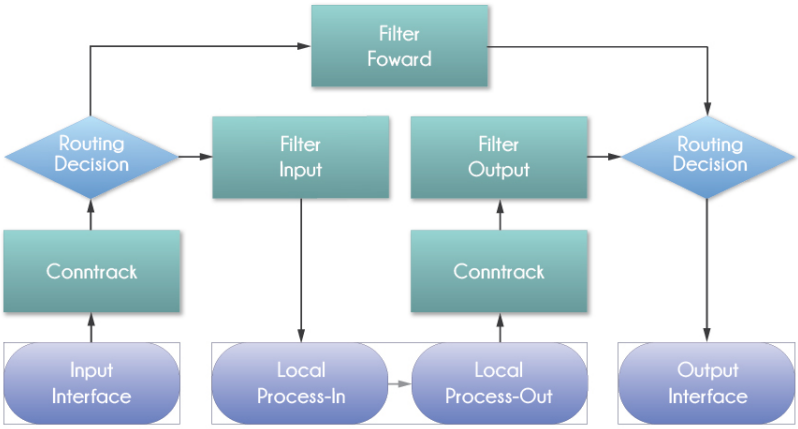


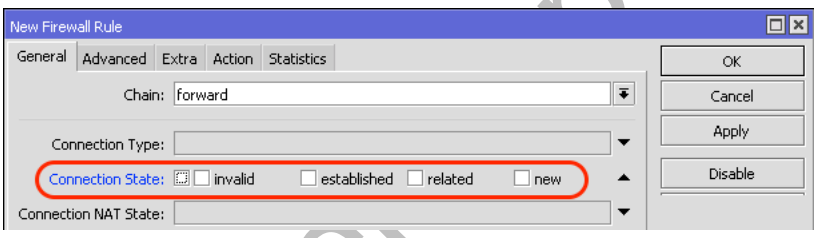
Diagrama de ubicación del Connection Tracking



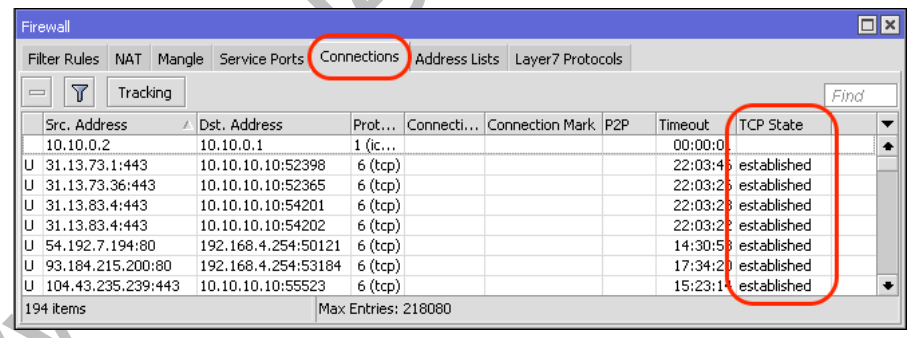
Condition: Connection State

Connection State es un estatus asignado a cada paquete por el sistema Conntrack

- **`new`** – El paquete está abriendo una nueva conexión
- **`established`** – El paquete perteneces a una conexión ya conocida
- **`invalid`** – El paquete no pertenece a ninguna de las conexiones conocidas
- **`related`** – El paquete está también abriendo una nueva conexión, pero está en algún tipo de relación a una conexión ya conocida

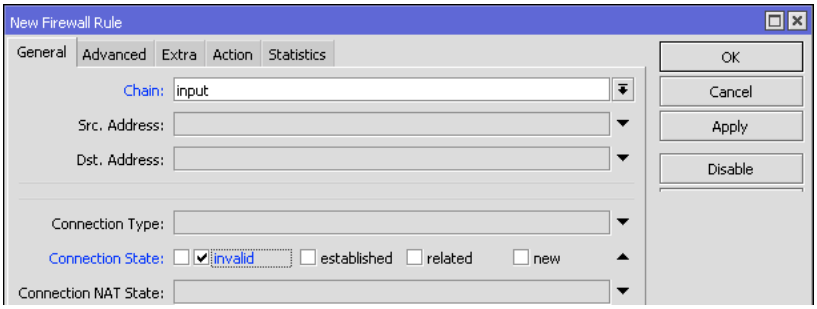


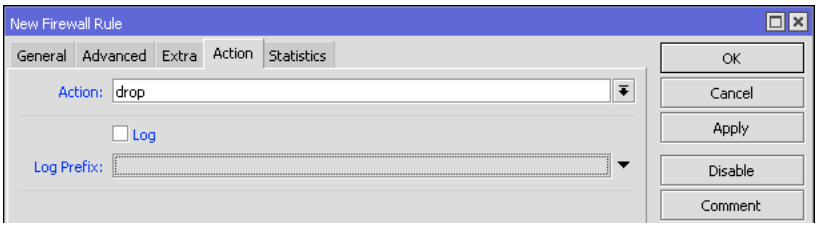
Nota importante: Connection state ≠ TCP state



Ejemplo de Regla Básica

```
/ip firewall filter add chain=input connection-state=invalid action=drop comment="Descartar paquetes invalidos"
```





Propiedades

- **`action`** *(action name; Default: accept)* – Acción que se va a tomar si el paquete coincide con la regla:
  - **`accept`** – Acepta el paquete. El paquete no se pasa a la siguiente regla de firewall
  - **`add-dst-to-address-list`** – Se agrega una dirección destino a un address list especificado en el parámetro `address-list`
  - **`add-src-to-address-list`** – Se agrega una dirección origen a un address list especificado en el parámetro `address-list`
  - **`drop`** – Se rechaza (dropea) el paquete de forma silenciosa
  - **`jump`** – Se salta al chain definido por el usuario. Este valor se especifica en el parámetro `jump-target`
  - **`log`** – Se agrega un mensaje al log del sistema (system log) que contiene la siguiente información: `in-interface, out-interface, src-mac, protocol, src-ip:port->dst-ip:port` y longitud del paquete (`length of the packet`). Después de que el paquete coincide con esta regla, se pasa a la siguiente regla en la lista, similar a la acción `passthrough`
  - **`passthrough`** – Ignora esta regla y el paquete pasa a la siguiente regla. Esta acción es útil para trabajar con estadísticas.
  - **`reject`** – Se rechaza el paquete y se envía un mensaje `ICMP reject`
  - **`return`** – Pasa el control de nuevo al `chain` donde tuvo lugar el `jump`
  - **`tarpit`** – Captura y mantiene las conexiones TCP (responde con un `SYN/ACK` al paquete `TCP SYN` entrante)
- **`address-list`** *(string; Default:)* - Nombre de la lista de direcciones (`address list`) que se utilizará. Se puede aplicar si `action=add-dst-to-address-list` o `action=add-src-to-address-list`
- **`address-list-timeout`** *(time; Default: 00:00:00)* – Especifica el intervalo de tiempo después del cual la dirección será removida del address list especificado en el parámetro `address-list`. Se usa en conjunto con las acciones `add-dst-to-address-list` o `add-src-to-address-list`. Cuando se especifica un valor de `00:00:00` significa que la dirección IP se dejará por siempre en el address list
- **`chain`** *(name; Default)* – Especifica a que `chain` se agregará la regla. Si la entrada no coincide con el nombre de un `chain` ya definido, se creará un nuevo `chain`.
- **`comment`** *(string; Default)* – Comentario descriptivo de esta regla.
- **`connection-bytes`** *(integer-integer; Default)* - Coincide con los paquetes solamente si una determinada cantidad de bytes ha sido transferido a través de la conexión particular. 0 – significa infinito. Por ejemplo, `connection-bytes=2000000-0` significa que la regla hace coincidencia si más de 2MB han sido transferidos a través de esa conexión relevante.
- **`connection-limit`** *(integer, netmask; Default:)* - Restringir el límite de conexiones por dirección o bloque de direcciones hasta e incluyendo el valor dado
- **`connection-mark`** *(no-mark | string; Default)* – Coincide con los paquetes marcados vía mangle con una marca de conexión particular (`connection mark`). Si se configura como `no-mark`, la regla coincidirá con cualquier conexión que no tenga marca.
- **`connection-rate`** *(Integer 0..4294967295; Default:)* – Permite capturar el tráfico basado en la velocidad actual de la conexión.
- **`connection-state`** *(established | invalid | new | related; Default:)* – Interpreta el análisis de datos del `connection tracking` de un paquete en particular:
  - **`established`** – Un paquete que pertenece a una conexión existente
  - **`invalid`** – Un paquete que no pudo ser identificado por algún motivo
  - **`new`** – El paquete ha iniciado una nueva conexión, o de otra manera asociado con una conexión de la que no se ha visto paquetes en ambas direcciones.
  - **`related`** – Un paquete que se relaciona con, pero que no es parte de una conexión existente, tales como errores ICMP o un paquete que inicia la conexión de datos
- **`connection-type`** *(ftp | h323 | irc | pptp | quake3 | sip | tftp; Default:)* - Coincide con los paquetes de conexiones relacionadas basado en la información de sus ayudantes de seguimiento de conexión (connection tracking helpers). Un ayudante (helper) de conexión relevante debe estar habilitado en `/ip firewall service-port`
- **`content`** *(string; Default:)* – Coincide con los paquetes que contienen un texto específico
- **`dscp`** *(integer: 0..63; Default:)* – Coincide con el campo de cabecera (header field) `DSCP IP`.
- **`dst-address`** *(IP/netmask | IP range; Default:)* - Coincide con los paquetes cuyo destino es igual a la IP especificada o cae dentro del rango IP especificado.

- **dst-address-list** *(name; Default:)* - Coincide con la dirección de destino de un paquete contra la lista de direcciones definido por el usuario
- **dst-address-type** *(unicast | local | broadcast | multicast; Default:)* – Coincide con el tipo de dirección destino:
  - **unicast** – Dirección IP usada para transmisión punto a punto
  - **local** – Si el `dst-address` está signado a una de las interfaces del router
  - **broadcast** – El paquete es enviado a todos los dispositivos en una subred
  - **multicast** – El paquete es enviado a un grupo definido de dispositivos
- **dst-limit** *(integer[/time],integer, dst-address | dst-port | src-address[/time]; Default:)* – Coincide con los paquetes hasta que una tasa (rate) dada es excedida. La tasa (rate) se define como paquetes por intervalo de. A diferencia del parámetro `limit`, cada flujo tiene su propio límite. El flujo se define por el parámetro `mode` (modo). Los parámetros se escriben en el siguiente formato: `count[/time],burst,mode[/expire]`.
  - **count** – Conteo de paquetes por intervalo de tiempo por flujo para que coincida
  - **time** – Especifica el intervalo de tiempo en el cual el conteo de paquetes por flujo no puede ser excedido (opcional, Si no se especifica nada se utilizará `1s`)
  - **burst** – Número inicial de paquete por flujo para coincidir: este número se recarga en uno cada vez/cuenta, hasta este número
  - **mode** – Este parámetro especifica qué campos únicos definen el flujo (`src-address, dst-address, src-and-dst-address, dst-address-and-port, addresses-and-dst-port`)
  - **expire** – Especifica el intervalo después del cual el flujo sin paquetes será permitido eliminar (opcional)
- **dst-port** *(integer[-integer]: 0..65535; Default:)* – Lista de números de puerto destino o rangos de número de puerto
- **fragment** (yeslno; Default:) – Coincide con los paquetes fragmentados. El primer paquete fragmentado (inicial) no cuenta. Si el `connection tracking` está habilitado entonces no habrá fragmentos ya que el sistema ensambla automáticamente cada paquete
- **hotspot** *(auth | from-client | http | local-dst | to-client; Default:)*
- **icmp-options** *(integer:integer; Default:)* – Coincide con los campos ICMP type:code
- **in-bridge-port** *(name; Default:)* – Interface real por la que el paquete ha ingresado al router, si la interface entrante es `bridge`. Trabaja únicamente si en la configuración de `bridge` se habilita `use-ip-firewall`.
- **in-interface** *(name; Default:)* – Interface por la que el paquete ha ingresado al router
- **ingress-priority** *(integer: 0..63; Default:)* – Coincide con la prioridad de ingreso del paquete. Prioridad se puede derivar de VLAN, WMM o MPLS EXP bit.
- **ipsec-policy** *(in | out, ipsec | none; Default:)* – Coincide con la política usada por `IPSec`. El valor es escrito en el siguiente formato: `direction, policy`. Direction se utiliza para seleccionar si coincide con la política usada para `decapsulation` o la política que será usada para `encapsulation`.
  - **in** – Válida en los chains `PREROUTING`, `INPUT` y `FORWARD`
  - **out** – Válida en los chains `POSTROUTING`, `OUTPUT` y `FORWARD`
  - **ipsec** – Coincide si el paquete está sujeto a procesamiento `IPsec`
  - **none** – Coincide con el paquete de transporte `IPsec`
  - Por ejemplo, si el router recibe `IPsec` con el paquete `GRE` encapsulado, entonces la regla `ipsec-policy=in,ipsec` coincidirá el paquete `GRE`, sino la regla `ipsec-policy=in,none` coincidirá el paquete `ESP`.
- **ipv4-options** *(any | loose-source-routing | no-record-route | no-router-alert | no-source-routing | no-timestamp | none | record-route | router-alert | strict-source-routing | timestamp; Default:)* – Coincide con las opciones de la cabecera IPv4.
  - **any** – Coincide con el paquete con al menos una de las opciones `IPv4`
  - **loose-source-routing** – Coinciden con los paquetes con opción `loose source routing`. Esta opción se utiliza para rutear el datagrama internet en base a la información suministrada por la fuente
  - **no-record-route** – Coincide con los paquetes con opción `no record route`. Esta opción se utiliza para rutear el datagrama internet en base a la información suministrada por la fuente
  - **no-router-alert** – Coincide con los paquetes con opción `no router alter`
  - **no-source-routing** - Coincide con los paquetes con opción `no source routing`
  - **no-timestamp** – Coincide con los paquetes con opción `no timestamp`
  - **record-route** – Coincide con los paquetes con opción `record route`
  - **router-alert** – Coincide con los paquetes con opción `router alter`
  - **strict-source-routing** – Coincide con los paquetes con opción `strict source routing`
  - **timestamp** - Coincide con los paquetes con `timestamp`
- **jump-target** *(name; Default:)* – Nombre del chain destino al cual debe saltar. Se aplcia solo si `action=jump`
- **layer7-protocol** *(name; Default:)* – Nombre del filtro `Layer7` definido en menú de protocolo `layer7`.
- **limit** *(integer,time,integer; Default:)* – Coincide con los paquetes a una velocidad limitada. Regla que usa este matcher coincidirá hasta que se alcance este límite. Los parámetros se escriben en formato siguiente: `count[/time],burst`.
  - **count** – Conteo de paquetes por intervalo de tiempo para que coincida

    - **time** – Especifica el intervalo de tiempo en el cual el conteo de paquetes no puede ser excedido (opcional, si no especifica nada entonces se usará `1s`)
    - **burst** – Número inicial de paquetes para coincidir: este número se recarga en uno cada vez/cuenta, hasta este número
- **log-prefix** *(string; Default:)* - Añade texto especificado al principio de cada mensaje de registro (`log`). Se aplica si `action=log`
- **nth** *(integer, integer; Default:)* – Coincide cada n-ésimo (nth) paquete.
- **out-bridge-port** *(name; Default:)* - Interface real por la que el paquete abandona al router, si la interface de salida es `bridge`. Trabaja únicamente si en la configuración de `bridge` se habilita `use-ip-firewall`.
- **out-interface** *(; Default:)* – Interface por la que el paquete abandona el router
- **p2p** (all-p2p | bit-torrent | blubster | direct-connect | edonkey | fasttrack | gnutella | soulseek | warez | winmx; Default:) - Coincide con los paquetes de parios protocolos peer-to-peer (P2P). No trabaja en paquetes p2p encriptados.
- **packet-mark** *(no-mark | string; Default:)* - Coincide con los paquetes marcados vía mangle con una marca de paquete particular (`packet mark`). Si se configura como `no-mark`, la regla coincidirá con cualquier paquete que no tenga marca.
- **packet-size** *(integer[-integer]:0..65535; Default:)* – Coincide con los paquetes de un tamaño específico o un rango de tamaño en bytes.
- **per-connection-classifier** *(ValuesToHash:Denominator/Remainder; Default:)* - Permite dividir el tráfico en streams iguales con la capacidad de mantener los paquetes con un conjunto específico de opciones en una stream particular.
- **port** *(integer[-integer]: 0..65535; Default:)* – Coincide si cualquier puerto (origen o destino) coincide con la lista específica de puertos o rangos de puertos. Se aplica solo si el protocolo es `TCP` o `UDP`
- **protocol** *(name or protocol ID; Default: tcp)* – Coincide con un protocolo IP específico por nombre de protocolo o por número de protocolo
- **psd** *(integer, time, integer, integer; Default:)* – Intenta detectar los escaneos TCP y UDP. Los parámetros están en el siguiente formato `WeightThreshold`, `DelayThreshold`, `LowPortWeight`, `HighPortWeight`
    - **WeightThreshold** – Peso total de los últimos paquetes `TCP/UDP` con diferentes puertos de destino procedentes del mismo host para ser tratado como una secuencia de escaneo de puertos
    - **DelayThreshold** – Retardo de los paquetes con diferentes puertos destino que proceden del mismo host paa ser tratados como una posible subsecuencia de escaneo de puertos
    - **LowPortWeight** – Peso de los paquetes con puerto de destino privilegiado (<=1024)
    - **HighPortWeight** – Peso de los paquetes con puerto de destino no-privilegiado
- **random** *(integer: 1..99; Default:)* – Coincide con los paquetes al azar con una probabilidad dada.
- **reject-with** *(; Default:)* – Especifica el mensaje de error que será devuelto siel paquete es rechazado. Se aplica si `action=reject`
- **routing-mark** *(string; Default:)* – Coincide con los paquetes marcados por mangle con una marca de ruteo (`routing mark`) particular
- **src-address** *(Ip/Netmaks, Ip range; Default:)* – Coincide con los paquetes cuya fuente es igual a la IP especificada o cae dentro de un rango IP específico.
- **src-address-list** *(name; Default:)* – Coincide con la dirección origen de un paquete contra el `address list` definido por el usuario
- **src-address-type** *(unicast | local | broadcast | multicast; Default:)* – Coincide con el tipo de dirección origen:
    - **unicast** – Dirección IP usada para transmisión punto a punto
    - **local** – Si la dirección es asignada a una de las interfaces del router
    - **broadcast** – El paquete es enviado a todos los dispositivos en una subred
    - **multicast** – El paquete es enviado a un grupo definido de dispositivos
- **src-port** *(integer[-integer]: 0..65535; Default:)* – Lista de puertos origen y rangos de puertos origen. Aplicable solo si el protocolo es `TCP` o `UDP`.
- **src-mac-address** *(MAC address; Default:)* – Coincide con la dirección MAC del paquete
- **tcp-flags** *(ack | cwr | ece | fin | psh | rst | syn | urg; Default:)* – Coincide con las banderas TCP específicas
    - **ack** – Reconocimiento de la data
    - **cwr** – Venta de congestión reducida
    - **ece** – Bandera `ECN-echo` (notificación de congestión explícita)
    - **fin** – Conexión cerrar (close)
    - **psh** – Función push (empujar)
    - **rst** – Conexión rechazar (drop)
    - **syn** – Nueva conexión (new)
    - **urg** – Data urgente
- **tcp-mss** *(integer: 0..65535; Default:)* – Coincide con el valor `TCP MSS` de un paquete IP
- **time** *(time-time,sat | fri | thu | wed | tue | mon | sun; Default:)* – Permite crear un filtro basado en el tiempo y fecha de arribo de un paquete, o para paquetes generados localmente, tiempo y fecha de partida
- **ttl** *(integer: 0..255; Default:)* – Coincide con el valor `TTL` de los paquetes

Capítulo 4: Firewall Filter - Chain Input

En este capítulo trataremos sobre la protección del router permitiendo únicamente los servicios necesarios de fuentes confiables con carga satisfactoria.

Chain Input

Conexiones Establecidas, Relacionadas e Inválidas

Estas reglas aseguran que solo las conexiones válidas vayan al router y hará un "drop" a las inválidas

```
/ip firewall filter
add chain=input connection-state=established, related action=accept comment="acepta los paquetes de conexiones establecidas y relacionadas" disabled=no
add chain=input connection-state=invalid action=drop comment="descarta los paquetes inválidos" disabled=no
```

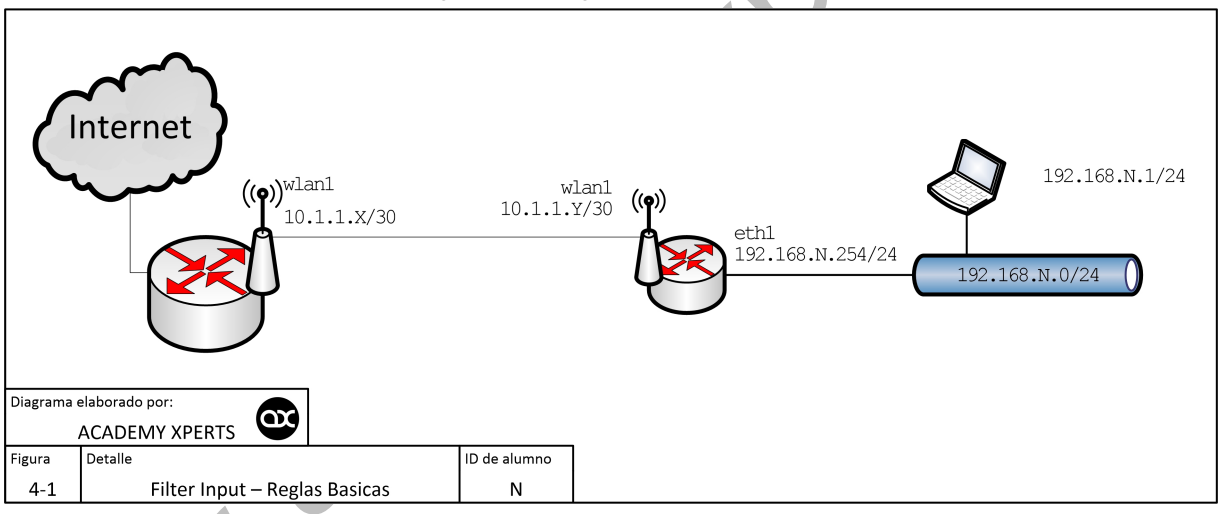
Address List

Esta regla permite el acceso total al router para ciertas direcciones IP

```
/ip firewall filter
add chain=input src-address-list=safe action=accept comment="permite el acceso al router desde la red" disabled=no
```

Laboratorio 4-1: Filter Input – Reglas Básicas

En este laboratorio el estudiante deberá configurar las 4 reglas básicas de Firewall INPUT



Tipos de Intrusión de Red (Network Intrusion)

Network Intrusion es un riesgo de seguridad muy serio que podría resultar no solo en una denegación temporal sino también en un rechazo total del servicio de red

Podemos hablar de 4 tipos principales de Network Intrusion

- Port scan
- DoS attack
- DDoS attack
- Ping flood

Port Scan

Port Scan es una “indagación” secuencial de puertos TCP (UDP). PSD (Port Scan Detection) es posible para el protocolo TCP y UDP

Los Nombres de Servicios y los Números de Puerto se utilizan para distinguir entre diferentes servicios que se ejecutan a través de protocolos de transporte como TCP, UDP, DCCP, y SCTP.

Los nombres de servicios son asignados en un esquema de “primero que llega, primero que se sirve” según lo documentado en el RFC6335.

Los números de puerto se asignan de varias maneras, basado en tres rangos:

1. Puertos de Sistema: 0 a 1023
2. Puertos de Usuario: 1024 a 49151
3. Puertos dinámicos y/o privados: 49152 a 65535

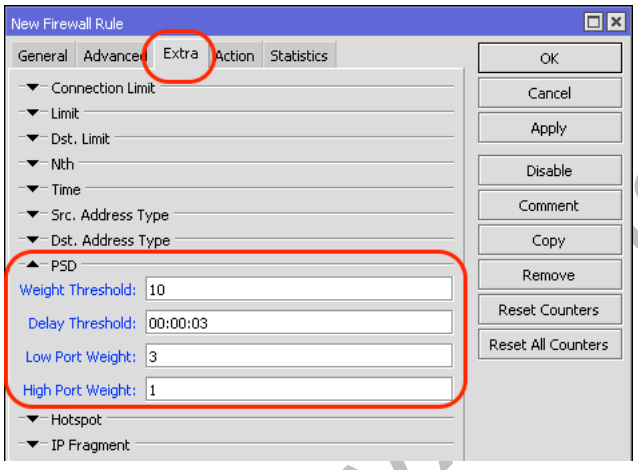
Los diferentes usos de estos rangos se describe en el RFC6335. Los Puertos de Sistema son asignados por el proceso IETF. Los Puertos de Usuario son asignados por el IANA usando el proceso de "IETF Review", el proceso "IESG Aproval" o el proceso “Expert Review", según lo especificado en el RFC6335. Los Puertos Dinámicos no están asignados.

Los procedimientos de registro de nombres de servicios y números de puerto se describen en el RFC6335.

Los puertos de sistema como de usuario, ya asignados, no deben ser usados sin un registro previo del IANA.

En los parámetros de RouterOS se refiere a Puertos Bajos (low ports) y Puertos Altos (high ports):

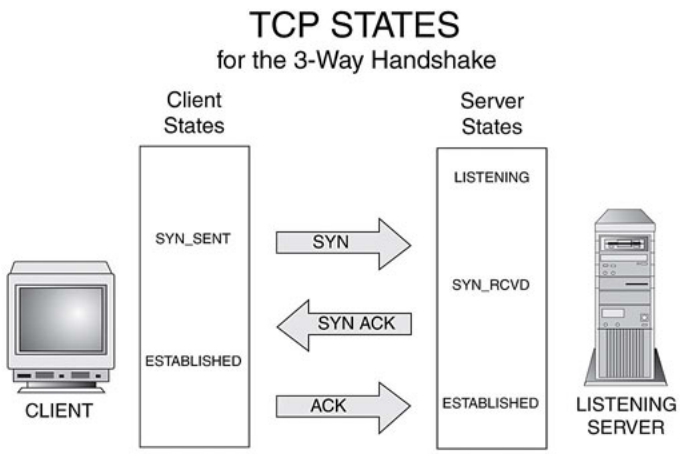
- Puertos bajos: 0 a 1023
- Puertos altos: 1024 a 65535



Ataques DoS

El principal objetivo de los ataques DoS es el consumo de recursos, como el tiempo de CPU o el ancho de banda, por lo que los servicios estándares obtendrán un Denial of Services (DoS).

Usualmente el router es inundado con paquetes TCP/SYN (requerimiento de conexión). Ocasionando que el server responda con un paquete TCP/SYN-ACK, y esperando por un paquete TCP/ACK

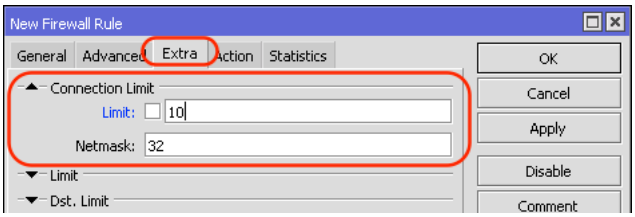


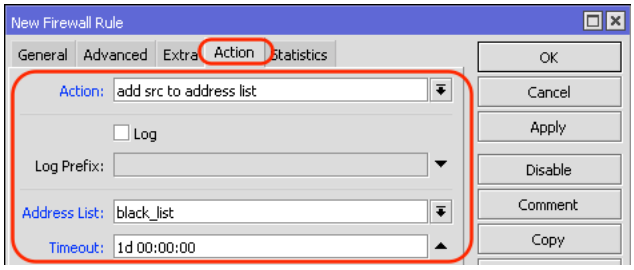
La mayoría de atacantes DoS son clientes infectados de virus

Protección contra ataques DoS

- Todas las IPs con más de 10 conexiones al router deberían ser consideradas como atacantes DoS
- Con cada conexión TCP descartada (dropped) se permitirá al atacante crear una nueva conexión
- Se debe implementar la protección DoS en 2 pasos
  - Detección: creando una lista de atacantes DoS en base a la conexión límite
  - Supresión: aplicando restricciones a los atacantes DoS detectados

Detección de ataques DoS

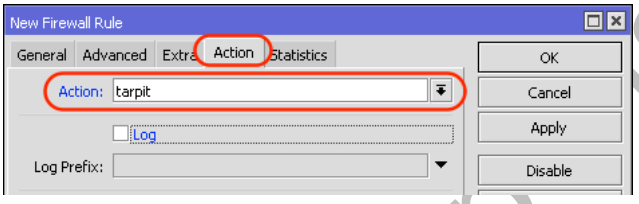




Supresión de ataques DoS

Para detener al atacante de crear nuevas conexiones se usa la opción `action=tarpit`

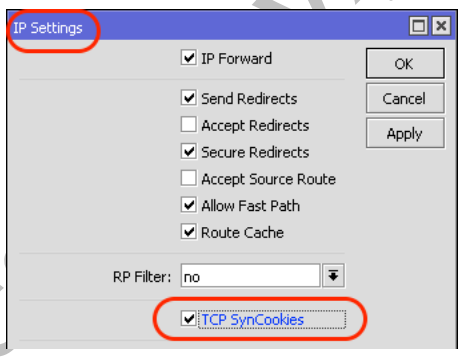
Se debe ubicar esta regla antes de la regla de detección o sino la entrada del address-list la reescribirá todo el tiempo



Ataques DDoS

Un ataque Distributed Denial of Service es muy similar al ataque DoS, y ocurre desde múltiples sistemas comprometidos

La única cosa que podría ayudar es la opción “TCPSyn Cookie” en IP Settings



ICMP

- Internet Control Message Protocol (inglés)
- Protocolo de Mensajes de Control de Internet (español)

https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol

El protocolo ICMP es uno de los principales protocolos de la Internet Protocol Suite, como se define en el RFC 792. Es utilizado por los dispositivos de red, como routers, para enviar mensajes de error que indican, por ejemplo, que un servicio solicitado no está disponible o que un host o router no pudieron ser contactados. ICMP también puede ser utilizado para retransmitir mensajes de consulta. ICMP tiene asignado el número 1 como número de protocolo.

ICMP se diferencia de los protocolos de transporte tales como TCP y UDP en que no se utiliza normalmente para el intercambio de datos entre los sistemas, ni tampoco se emplea regularmente por las aplicaciones de red de usuario final (con la excepción de algunas herramientas de diagnóstico como ping y traceroute).

Los mensajes ICMP se suelen utilizar con fines de diagnóstico o de control, o se generan en respuesta a errores en las operaciones de IP. Los errores ICMP se dirigen a la dirección IP de origen del paquete de origen.

Por ejemplo, todos los dispositivos (como un enrutador intermedio) que reenvían un datagrama IP, primero disminuyen el tiempo de vida (TTL) en la cabecera IP en uno. Si el TTL resultante es 0, el paquete se descarta y se envía un mensaje `ICMP Time To Live exceeded in transit` a la dirección de origen del datagrama.

Aunque los mensajes ICMP están contenidos dentro del paquete IP estándar, los mensajes ICMP se procesan generalmente como un caso especial, que se distingue de la transformación IP normal, en lugar de ser procesados como un sub-protocolo normal de IP. En muchos casos, es necesario inspeccionar el contenido del mensaje ICMP y entregar el mensaje de error

correspondiente a la aplicación que generó el paquete IP original, el que envió el paquete que provocó el envío del mensaje de ICMP.

Muchas utilidades de red comúnmente utilizados se basan en mensajes ICMP. El comando `traceroute` se puede implementar mediante la transmisión de datagramas con campos de cabecera IP TTL configurados, y buscando los mensajes `Time to live exceeded in transit` y `Destination unreachable` generados en respuesta. La utilidad relacionada `ping` se implementa utilizando los mensajes ICMP `Echo request` y `Echo reply`.

Estructura del segmento ICMP

Header (cabecera)

La cabecera ICMP comienza después de la cabecera IPv4 y se identifica con el número de protocolo IP '1'. Todos los paquetes ICMP tienen una cabecera de 8 bytes y la sección de datos de tamaño variable. Los primeros 4 bytes de la cabecera tienen formato fijo, mientras que los últimos 4 bytes dependen del type/code de ese paquete ICMP.

Formato de la cabecera (header) ICMP

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Type								Code								Checksum															
4	32	Rest of Header																															

Checksum

Comprobación de errores de datos. Se calcula a partir de la cabecera ICMP y de los datos, con el valor 0 sustituido para este campo. El Internet Checksum se utiliza según se especifica en el RFC 1071.

Rest of Header (resto de la cabecera)

Campo de cuatro bytes, el contenido varía en función del tipo (type) y código (code) ICMP.

Data

Los mensajes de error ICMP contienen una sección de datos que incluye toda la cabecera IPv4, además de los primeros ocho bytes de datos del paquete IPv4 que provocó el mensaje de error. El paquete ICMP se encapsula entonces en un nuevo paquete IPv4.

El tamaño variable de la sección de datos de paquetes ICMP ha sido explotado. En el conocido "Ping de la muerte" (ping of death), paquetes de ping grandes o fragmentados son utilizados para los ataques de denegación de servicio (DoS). ICMP también puede ser usado para crear canales encubiertos (covert channels) para la comunicación, así como también el exploit LOKI.

Mensajes de Control

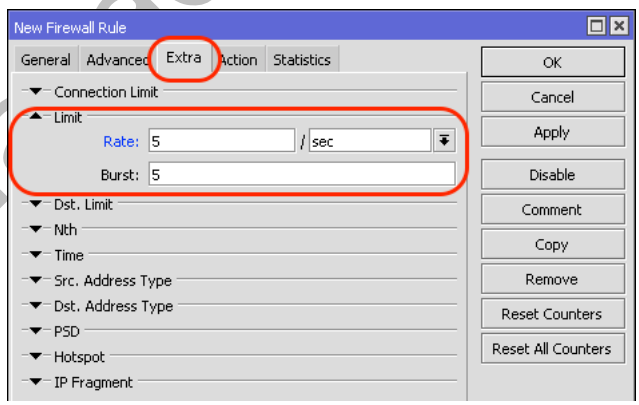
Type	Code	Status	Description
0 – Echo Reply	0		Echo reply (used to ping)
1 and 2		unassigned	Reserved
3 – Destination Unreachable	0		Destination network unreachable
	1		Destination host unreachable
	2		Destination protocol unreachable
	3		Destination port unreachable
	4		Fragmentation required, and DF flag set
	5		Source route failed
	6		Destination network unknown
	7		Destination host unknown
	8		Source host isolated
	9		Network administratively prohibited
	10		Host administratively prohibited
	11		Network unreachable for TOS
	12		Host unreachable for TOS
	13		Communication administratively prohibited
	14		Host Precedence Violation
	15		Precedence cutoff in effect
4 – Source Quench	0	deprecated	Source quench (congestion control)
5 – Redirect Message	0		Redirect Datagram for the Network
	1		Redirect Datagram for the Host
	2		Redirect Datagram for the TOS & network
	3		Redirect Datagram for the TOS & host
6		deprecated	Alternate Host Address
7		unassigned	Reserved
8 – Echo Request	0		Echo request (used to ping)
9 – Router Advertisement	0		Router Advertisement
10 – Router Solicitation	0		Router discovery/selection/solicitation

Type	Code	Status	Description
11 – Time Exceeded	0		TTL expired in transit
	1		Fragment reassembly time exceeded
12 – Parameter Problem: Bad IP header	0		Pointer indicates the error
	1		Missing a required option
	2		Bad length
13 – Timestamp	0		Timestamp
14 – Timestamp Reply	0		Timestamp reply
15 – Information Request	0	deprecated	Information Request
16 – Information Reply	0	deprecated	Information Reply
17 – Address Mask Request	0	deprecated	Address Mask Request
18 – Address Mask Reply	0	deprecated	Address Mask Reply
19		reserved	Reserved for security
20 through 29		reserved	Reserved for robustness experiment
30 – Traceroute	0	deprecated	Information Request
31		deprecated	Datagram Conversion Error
32		deprecated	Mobile Host Redirect
33		deprecated	Where-Are-You (originally meant for IPv6)
34		deprecated	Here-I-Am (originally meant for IPv6)
35		deprecated	Mobile Registration Request
36		deprecated	Mobile Registration Reply
37		deprecated	Domain Name Request
38		deprecated	Domain Name Reply
39		deprecated	SKIP Algorithm Discovery Protocol, Simple Key-Management for Internet Protocol
40			Photuris, Security failures
41		experimental	ICMP for experimental mobility protocols such as Seamoby [RFC4065]
42 through 252		unassigned	Reserved
253		experimental	RFC3692-style Experiment 1 (RFC 4727)
254		experimental	RFC3692-style Experiment 2 (RFC 4727)
255		reserved	Reserved

Ping Flood

Ping flood usualmente consiste de volúmenes de mensajes ICMP aleatorios.

Con la condición “limit” es posible limitar la regla para que coincida con un límite dado. Esta condición se usa frecuentemente con la acción “log”.

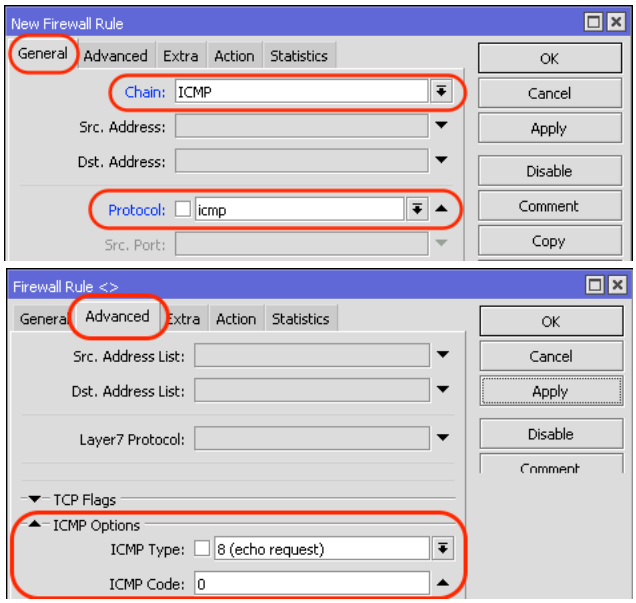


Tipos de mensaje ICMP

El router típico usa solamente 5 tipos de mensajes ICMP (type:code)

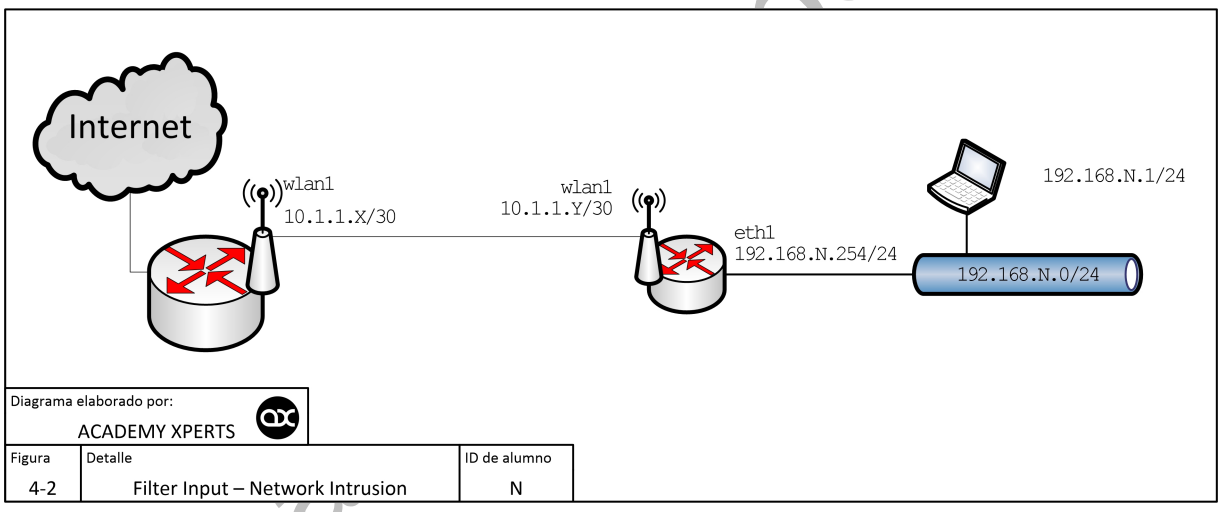
- Para **PING**: mensajes 0:0 y 8:0
- Para **TRACEROUTE**: mensajes 11:0 y 3:3
- Para Path **MTU** discovery: mensaje 3:4
- Otros tipos de mensajes ICMP deberían ser bloqueados

Ejemplo de regla de mensaje ICMP



Laboratorio 4-2: Filter Input – Network Intrusion

Configurar las reglas fundamentales para proteger al router de las Intrusiones de Red más conocidas



Servicios de RouterOS

Nr.	Port	Protocol	Comments
1	20	tcp	FTP
2	21	tcp	FTP
3	22	tcp	SSH,SFTP
4	23	tcp	Telnet
5	53	tcp	DNS
6	80	tcp	HTTP
7	179	tcp	BGP
8	443	tcp	SHTTP (Hotspot)
9	646	tcp	LDP (MPLS)
10	1080	tcp	SoCKS (Hotspot)
11	1723	tcp	PPTP
12	1968	tcp	MME
13	2000	tcp	Bandwidth server
14	2210	tcp	Dude server
15	2211	tcp	Dude server
16	2828	tcp	uPnP
17	3128	tcp	WEB Proxy
18	8291	tcp	Winbox
19	8728	tcp	API
20	-------	/1	ICMP
21	-------	/2	IGMP (Multicast)
22	-------	/4	IPIP

Nr.	Port	Protocol	Comments
23	53	udp	DNS
24	123	udp	NTP
25	161	udp	SNMP
26	500	udp	IPSec
27	520	udp	RIP
28	521	udp	RIP
29	646	udp	LDP (MPLS)
30	1698	udp	RSVP (MPLS)
31	1699	udp	RSVP (MPLS)
32	1701	udp	L2TP
33	1812	udp	User-manager
34	1813	udp	User-manager
35	1900	udp	uPnP
36	1966	udp	MME
37	5678	udp	Neighbour Discovery
38	-------	/46	RSVP (MPLS)
39	-------	/47	PPTP, EoIP
40	-------	/50	IPSec
41	-------	/51	IPSec
42	-------	/89	OSPF
43	-------	/103	PIM (Multicast)
44	-------	/112	VRRP

Chain Services

Los siguientes son servicios que permiten acceder al router. La mayoría están deshabilitados por default

Usualmente los siguientes servicios deberían estar siempre disponibles

- MAC Telnet
- Bandwidth Test Server
- MTU Discovery

```
/ip firewall filter
add chain=input action=jump jump-target=services comment="jump al chain services" disabled=no

/ip firewall filter
add chain=services src-address-list=127.0.0.1 dst-address=127.0.0.1 action=accept comment="accept localhost"
disabled=no
add chain=services protocol=udp dst-port=20561 action=accept comment="allow MACwinbox " disabled=no
add chain=services protocol=tcp dst-port=2000 action=accept comment="Bandwidth server" disabled=no
add chain=services protocol=udp dst-port=5678 action=accept comment=" MT Discovery Protocol" disabled=no
add chain=services protocol=tcp dst-port=161 action=accept comment="allow SNMP" disabled=yes
add chain=services protocol=tcp dst-port=179 action=accept comment="Allow BGP" disabled=yes
add chain=services protocol=udp dst-port=5000-5100 action=accept comment="allow BGP" disabled=yes
add chain=services protocol=udp dst-port=123 action=accept comment="Allow NTP" disabled=yes
add chain=services protocol=tcp dst-port=1723 action=accept comment="Allow PPTP" disabled=yes
add chain=services protocol=gre action=accept comment="allow PPTP and EoIP" disabled=yes
add chain=services protocol=tcp dst-port=53 action=accept comment="allow DNS request" disabled=yes
add chain=services protocol=udp dst-port=53 action=accept comment="Allow DNS request" disabled=yes
add chain=services protocol=udp dst-port=1900 action=accept comment="UPnP" disabled=yes
add chain=services protocol=tcp dst-port=2828 action=accept comment="UPnP" disabled=yes
add chain=services protocol=udp dst-port=67-68 action=accept comment="allow DHCP" disabled=yes
add chain=services protocol=tcp dst-port=8080 action=accept comment="allow Web Proxy" disabled=yes
add chain=services protocol=ipencap action=accept comment="allow IPIP" disabled=yes
add chain=services protocol=tcp dst-port=443 action=accept comment="allow https for Hotspot" disabled=yes
add chain=services protocol=tcp dst-port=1080 action=accept comment="allow Socks for Hotspot" disabled=yes
add chain=services protocol=udp dst-port=500 action=accept comment="allow IPsec connections" disabled=yes
add chain=services protocol=ipsec-esp action=accept comment="allow IPSec" disabled=yes
add chain=services protocol=ipsec-ah action=accept comment="allow IPSec" disabled=yes
add chain=services protocol=udp dst-port=520-521 action=accept comment="allow RIP" disabled=yes
add chain=services protocol=ospf action=accept comment="allow OSPF" disabled=yes
add chain=services action=return comment="" disabled=no
```

Chain Input & Broadcast

Esta regla permite el tráfico broadcast al router. Esto es necesario algunas veces por servicios como NTP

```
/ip firewall filter
add chain=input dst-address-type=broadcast action=accept comment="permitir trafico Broadcast" disabled=no
```

Capítulo 5: Firewall Filter - Chain Forward

Protección a los clientes de ICMP-Flooding/Virus y Protección del internet de los clientes

Chain Forward

Conexiones Establecidas, Relacionadas e Inválidas

Estas reglas aseguran que solo se acepten las conexiones válidas QUE PASAN A TRAVES del router y hará un "drop" a las inválidas

```
/ip firewall filter
add chain=forward connection-state=established,related action=accept comment="acepta los paquetes de conexiones establecidas y relacionadas" disabled=no
add chain=forward connection-state=invalid action=drop comment="descarta los paquetes inválidos" disabled=no
```

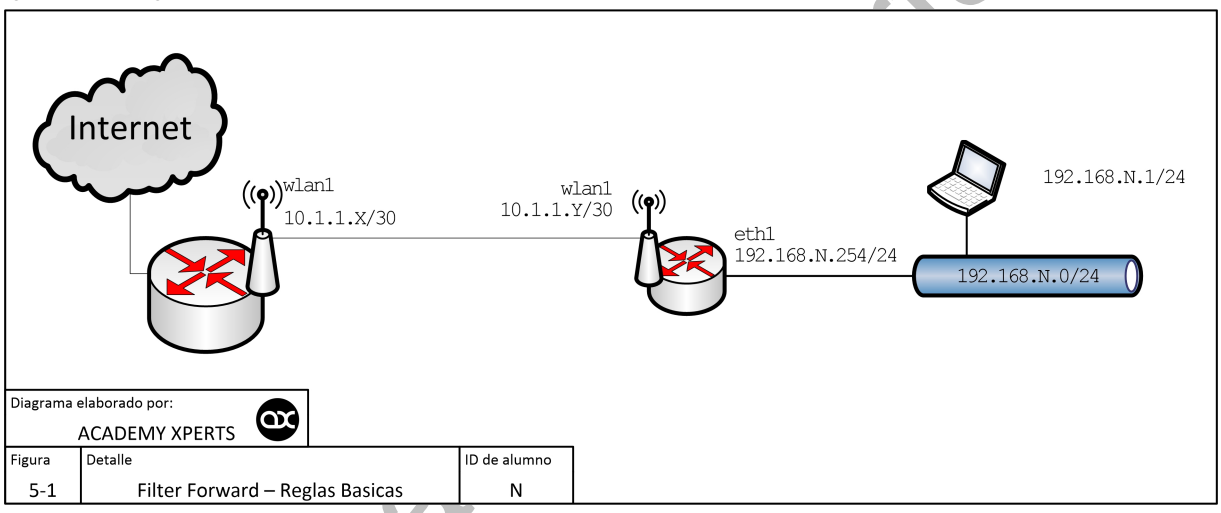
Control ICMP

Se crea un JUMP hacia el conjunto de reglas ICMP desarrolladas en capítulo anterior

```
/ip firewall filter
add chain=forward protocol=icmp action=jump jump-target=ICMP comment="Salto al chain ICMP" disabled=no
```

Laboratorio 5-1: Filter Input – Network Intrusion

Configurar las 4 reglas básicas de Firewall FORWARD



Control de Virus (ejemplo)

Filtro para descartar todos los paquetes no-deseados que parezcan venir de equipos infectados. En lugar de añadir esta regla en el chain forward, vamos a crear un nuevo chain que haga un salto

Un conjunto de reglas más extenso se puede encontrar el Wiki.

```
/ip firewall filter
add chain=forward action=jump jump-target=virus comment="saltar al chain virus"

/ip firewall filter
add chain=virus protocol=tcp dst-port=135-139 action=drop comment="Drop Blaster Worm"
add chain=virus protocol=udp dst-port=135-139 action=drop comment="Drop Messenger Worm"
add chain=virus protocol=tcp dst-port=445 action=drop comment="Drop Blaster Worm"
add chain=virus protocol=udp dst-port=445 action=drop comment="Drop Blaster Worm"
```

Chain Forward – HTTP/SNMP (otros ejemplos)

Bloquear todo el trafico excepto el que específicamente se permite pasar.

Permitir HTTP y SMTP, algunos TCP y UDP, e ICMP (ping)

```
/ip firewall filter
add chain=forward action=accept protocol=tcp dst-port=80 comment="permitir HTTP"
add chain=forward action=accept protocol=tcp dst-port=25 comment="permitir SMTP"
add chain=forward protocol=tcp comment="permitir TCP"
add chain=forward protocol=icmp comment="permitir ping"
add chain=forward protocol=udp comment="permitir udp"
add chain=forward action=drop comment="drop todo lo demas"
```

Capítulo 6: Bogon IPs

IPs reservadas. IPs públicamente ruteables

Hay aproximadamente 4,3 billones de direcciones IPv4

Existen varios rangos de direcciones IP restringidas en la red pública. Hay varios rangos de IP reservados (que no se usan hasta el momento) para propósitos específicos. Hay muchos rangos de IP sin usar

http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml

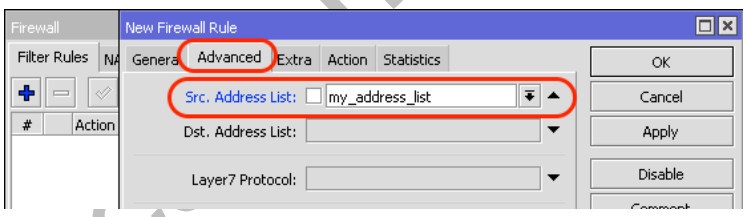
Bogon IPs

- `0.0.0.0/8` reserved for self-identification [RFC1122] , section 3.2.1.3
- `10.0.0.0/8` (10.x.x.x) reserved for Private-Use Networks [RFC1918]
- `100.64.0.0/10` reserved for Shared Address Space [RFC6598]
- `127.0.0.0/8` reserved for Loopback [RFC1122], section 3.2.1.3
- `169.254.0.0/16` reserved for Link Local [RFC3927]
- `172.16.0.0/12` (172.16.x.x - 172.31.x.x) reserved for Private-Use Networks [RFC1918]
- `192.0.2.0/24` reserved for TEST-NET-1 [RFC5737].
- `192.88.99.0/24` reserved for 6to4 Relay Anycast [RFC3068]
- `192.88.99.2/32` reserved for 6a44 Relay Anycast [RFC6751
- `192.168.0.0/16` (192.168.x.x) reserved for Private-Use Networks [RFC1918]
- `192.0.0.0/24` reserved for IANA IPv4 Special Purpose Address Registry [RFC5736]
- `198.18.0.0/15` reserved for Network Interconnect Device Benchmark Testing [RFC2544]
- `198.51.100.0/24` reserved for TEST-NET-2 [RFC5737]
- `203.0.113.0/24` reserved for TEST-NET-3 [RFC5737]
- `Multicast` (formerly "Class D") [RFC5771]
- Unicast-Prefix-Based IPv4 Multicast Addresses [RFC6034]
- Administratively Scoped IP Multicast [RFC2365]
- Reserved for future use (formerly "Class E") [RFC1112]
- `255.255.255.255` is reserved for "limited broadcast" destination address [RFC919] and [RFC922]

Opciones Address List

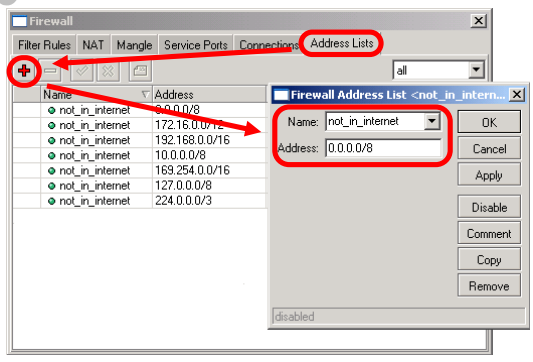
En lugar de crear una regla de filtrado para cada dirección de red IP, usted puede crear una única regla de lista de direcciones IP. Use las opciones “Src./Dst. Address List

Cree un Address List en el menú `/ip firewall address-list`



Laboratorio Address List (opcional)

Haga un Address List de Bogon IPs más comunes



Address List – Laboratorio (opcional)

- Permitir que solo los paquetes de direcciones válidas de internet ingresen a la red
- Permitir que solo los paquetes de direcciones de clientes conocidos ingresen a la red
- Permitir que solo los paquetes de direcciones de clientes válidos salgan de la red
- Permitir que solo los paquetes a direcciones válidas de internet salgan de la red
- Ubicar las reglas adecuadamente

Capítulo 7: Firewall NAT

Network Address Translation

NAT (Network Address Translation) es un estándar de internet que permite que los hosts en una red local usen un conjunto de direcciones IP para las comunicaciones internas, y otro conjunto de direcciones IP para comunicación externa. Una LAN que usa NAT se la conoce como una “red nateada”. Para que funcione el NAT, debe existir un gateway NAT en cada red nateada. El gateway NAT (conocido como router NAT) ejecuta la reescritura de la dirección IP cuando el paquete viaja desde o hacia la LAN

Existen dos tipos de NAT

- **Source NAT (srcnat).-** Este tipo de NAT se realiza sobre paquetes que se originan a partir de una red nateada. Un router NAT reemplaza la dirección origen privada de un paquete IP con una nueva dirección IP pública a medida que viaja a través del router. Se aplica una operación inversa a los paquetes de respuesta que viajan en la otra dirección
  - re-escribe la dirección IP origen y/o el puerto = source NAT (src-nat)
  - procesa el trafico que se envía desde y a través del router, después de que se ha fusionado de los chains “output” y “forward” de los filtros de firewall
- **Destination NAT (dstnat).-** Este tipo de NAT se realiza sobre paquetes que tienen como destino la red nateada. Se utiliza mayormente para que los hosts de una red privada sean accesibles desde Internet. Un router NAT que realiza dstnat sustituye la dirección IP de destino de un paquete IP, ya que viajan a través del router hacia una red privada
  - re-escribe la dirección IP destino y/o el puerto = destination NAT (dst-nat)
  - procesa el trafico que se envía a y a través del router, antes de dividirlo en las cadenas “input” y “forward” de los filtros de firewall

Nota Importante: Las reglas de Firewall NAT solo procesan el primer paquete de cada conexión (Conecction State = new)

Los hosts que están detrás de un router con NAT habilitado no tienen una verdadera conectividad de extremo a extremo. Por lo tanto, algunos protocolos de Internet podrían no funcionar en escenarios con NAT.

Los servicios que requieren la iniciación de conexión TCP desde fuera de la red privada o protocolos sin estado, como UDP, pueden romperse. Por otra parte, algunos protocolos son inherentemente incompatibles con NAT, un ejemplo de esto es el protocolo AH de la suite IPsec

Para superar estas limitaciones RouterOS incluye los llamados ayudantes NAT (NAT helpers), que habilitan NAT transversal para varios protocolos.

Propiedades

- **`action`** (action name; Default: accept) – Especifica la acción que tomará el paquete si coincide con la regla
  - **`accept`** – Acepta el paquete. El paquete no se pasa a la siguiente regla de NAT.
  - **`add-dst-to-address-list`** – Se agrega una dirección destino a un address list especificado en el parámetro `address-list`
  - **`add-src-to-address-list`** – Se agrega una dirección origen a un address list especificado en el parámetro `address-list`
  - **`dst-nat`** – Reemplaza la dirección y /o puerto destino de un paquete IP por los valores especificados por los parámetros `to-addresses` y `to-ports`
  - **`jump`** – Se salta al chain definido por el usuario. Este valor se especifica en el parámetro `jump-target`
  - **`log`** – Se agrega un mensaje al log del sistema (system log) que contiene la siguiente información: `in-interface, out-interface, src-mac, protocol, src-ip:port->dst-ip:port` y longitud del paquete (`length of the packet`). Después de que el paquete coincide con esta regla, se pasa a la siguiente regla en la lista, similar a la acción `passthrough`
  - **`masquerade`** – Reemplaza la dirección origen de un paquete IP a una IP determinada por la facilidad de routing.
  - **`netmap`** – Crea un mapeo estático 1:1 de un conjunto de direcciones IP a otro conjunto de direcciones. Esta opción se usa frecuentemente para distribuir direcciones IP públicas a hosts en las redes privadas
  - **`passthrough`** – Ignora esta regla y el paquete pasa a la siguiente regla. Esta acción es útil para trabajar con estadísticas.
  - **`redirect`** – Reemplaza el puerto destino de un paquete IP a un puerto especificado por el parámetro `to-ports` y la dirección destino a una de las direcciones locales del router
  - **`return`** – Pasa el control de nuevo al `chain` donde tuvo lugar el `jump`
  - **`same`** – Entrega a un cliente particular la misma dirección IP origen/destino del rango proporcionado para cada conexión. Esto se utiliza con mayor frecuencia para los servicios que esperan la misma dirección del cliente para múltiples conexiones desde el mismo cliente.
  - **`src-nat`** – Reemplaza la dirección origen de un paquete IP a valores especificados en los parámetros `to-addresses` y `to-ports`
- **`address-list`** *(string; Default:)* - Nombre de la lista de direcciones (`address list`) que se utilizará. Se puede aplicar si `action=add-dst-to-address-list` o `action=add-src-to-address-list`

- **address-list-timeout** *(time; Default: 00:00:00)* – Especifica el intervalo de tiempo después del cual la dirección será removida del address list especificado en el parámetro `address-list`. Se usa en conjunto con las acciones `add-dst-to-address-list` o `add-src-to-address-list`. Cuando se especifica un valor de `00:00:00` significa que la dirección IP se dejará por siempre en el address list
- **chain** *(name; Default:)* – Especifica a que `chain` se agregará la regla. Si la entrada no coincide con el nombre de un `chain` ya definido, se creará un nuevo `chain`.
- **comment** *(string; Default:)* – Comentario descriptivo de esta regla.
- **connection-bytes** *(integer-integer; Default:)* - Coincide con los paquetes solamente si una determinada cantidad de bytes ha sido transferido a través de la conexión particular. 0 – significa infinito. Por ejemplo, `connection-bytes=2000000-0` significa que la regla hace coincidencia si más de 2MB han sido transferidos a través de esa conexión relevante.
- **connection-limit** *(integer,netmask; Default:)* - Restringir el límite de conexiones por dirección o bloque de direcciones hasta e incluyendo el valor dado
- **connection-mark** *(no-mark | string; Default:)* – Coincide con los paquetes marcados vía mangle con una marca de conexión particular (`connection mark`). Si se configura como `no-mark`, la regla coincidirá con cualquier conexión que no tenga marca.
- **connection-rate** *(Integer 0..4294967295; Default:)* – Permite capturar el tráfico basado en la velocidad actual de la conexión.
- **connection-type** *(ftp | h323 | irc | pptp | quake3 | sip | tftp; Default:)* - Coincide con los paquetes de conexiones relacionadas basado en la información de sus ayudantes de seguimiento de conexión (connection tracking helpers). Un ayudante (helper) de conexión relevante debe estar habilitado en `/ip firewall service-port`
- **content** *(string; Default:)* – Coincide con los paquetes que contienen un texto específico
- **dscp** *(integer: 0..63; Default:)* – Coincide con el campo de cabecera (header field) `DSCP IP`.
- **dst-address** *(IP/netmask | IP range; Default:)* - Coincide con los paquetes cuyo destino es igual a la IP especificada o cae dentro del rango IP especificado.
- **dst-address-list** *(name; Default:)* - Coincide con la dirección de destino de un paquete contra la lista de direcciones definido por el usuario
- **dst-address-type** *(unicast | local | broadcast | multicast; Default:)* – Coincide con el tipo de dirección destino:
  - **unicast** – Dirección IP usada para transmisión punto a punto
  - **local** – Si el `dst-address` está signado a una de las interfaces del router
  - **broadcast** – El paquete es enviado a todos los dispositivos en una subred
  - **multicast** – El paquete es enviado a un grupo definido de dispositivos
- **dst-limit** *(integer[/time],integer,dst-address | dst-port | src-address[/time]; Default:)* – Coincide con los paquetes hasta que una tasa (rate) dada es excedida. La tasa (rate) se define como paquetes por intervalo de. A diferencia del parámetro `limit`, cada flujo tiene su propio límite. El flujo se define por el parámtro `mode` (modo). Los parámetros se escriben en el siguiente formato: `count[/time],burst,mode[/expire]`.
  - **count** – Conteo de paquetes por intervalo de tiempo por flujo para que coincida
  - **time** – Especifica el intervalo de tiempo en el cual el conteo de paquetes por flujo no puede ser excedido (opcional, Si no se especifica nada se utilizará `1s`)
  - **burst** – Número inicial de paquete por flujo para coincidir: este número se recarga en uno cada vez/cuenta, hasta este número
  - **mode** – Este parámetro especifica qué campos únicos definen el flujo (`src-address, dst-address, src-and-dst-address, dst-address-and-port, addresses-and-dst-port`)
  - **expire** – Especifica el intervalo después del cual el flujo sin paquetes será permitido eliminar (opcional)
- **dst-port** *(integer[-integer]: 0..65535; Default:)* – Lista de números de puerto destino o rangos de número de puerto
- **fragment** (yes|no; Default:) – Coincide con los paquetes fragmentados. El primer paquete fragmentado (inicial) no cuenta. Si el `connection tracking` está habilitado entonces no habrá fragmentos ya que el sistema ensambla automáticamente cada paquete
- **hotspot** *(auth | from-client | http | local-dst | to-client; Default:)*
- **icmp-options** *(integer:integer; Default:)* – Coincide con los campos ICMP type:code
- **in-bridge-port** *(name; Default:)* – Interface real por la que el paquete ha ingresado al router, si la interface entrante es `bridge`. Trabaja únicamente si en la configuración de `bridge` se habilita `use-ip-firewall`.
- **in-interface** *(name; Default:)* – Interface por la que el paquete ha ingresado al router
- **ingress-priority** *(integer: 0..63; Default:)* – Coincide con la prioridad de ingreso del paquete. Prioridad se puede derivar de VLAN, WMM o MPLS EXP bit.
- **ipsec-policy** *(in | out, ipsec | none; Default:)* – Coincide con la política usada por `IPSec`. El valor es escrito en el siguiente formato: `direction, policy`. Direction se utiliza para seleccionar si coincide con la política usada para `decapsulation` o la política que será usada para `encapsulation`.
  - **in** – Válida en los chains `PREROUTING`, `INPUT` y `FORWARD`
  - **out** – Válida en los chains `POSTROUTING`, `OUTPUT` y `FORWARD`
  - **ipsec** – Coincide si el paquete está sujeto a procesamiento `IPsec`

    - **none** – Coincide con el paquete de transporte `IPsec`
    - Por ejemplo, si el router recibe `IPsec` con el paquete `GRE` encapsulado, entonces la regla `ipsec-policy=in,ipsec` coincidirá el paquete `GRE`, sino la regla `ipsec-policy=in,none` coincidirá el paquete `ESP`.
- **ipv4-options** *(any | loose-source-routing | no-record-route | no-router-alert | no-source-routing | no-timestamp | none | record-route | router-alert | strict-source-routing | timestamp; Default:)* – Coincide con las opciones de la cabecera IPv4.
    - **any** – Coincide con el paquete con al menos una de las opciones `IPv4`
    - **loose-source-routing** – Coinciden con los paquetes con opción `loose source routing`. Esta opción se utiliza para rutear el datagrama internet en base a la información suministrada por la fuente
    - **no-record-route** – Coincide con los paquetes con opción `no record route`. Esta opción se utiliza para rutear el datagrama internet en base a la información suministrada por la fuente
    - **no-router-alert** – Coincide con los paquetes con opción `no router alter`
    - **no-source-routing** - Coincide con los paquetes con opción `no source routing`
    - **no-timestamp** – Coincide con los paquetes con opción `no timestamp`
    - **record-route** – Coincide con los paquetes con opción `record route`
    - **router-alert** – Coincide con los paquetes con opción `router alter`
    - **strict-source-routing** – Coincide con los paquetes con opción `strict source routing`
    - **timestamp** - Coincide con los paquetes con `timestamp`
- **jump-target** *(name; Default:)* – Nombre del chain destino al cual debe saltar. Se aplica solo si `action=jump`
- **layer7-protocol** *(name; Default:)* – Nombre del filtro `Layer7` definido en menú de protocolo `layer7`.
- **limit** *(integer,time,integer; Default:)* – Coincide con los paquetes a una velocidad limitada. Regla que usa este matcher coincidirá hasta que se alcance este límite. Los parámetros se escriben en formato siguiente: `count[/time],burst`.
    - **count** – Conteo de paquetes por intervalo de tiempo para que coincida
    - **time** – Especifica el intervalo de tiempo en el cual el conteo de paquetes no puede ser excedido (opcional, si no especifica nada entonces se usará `1s`)
    - **burst** – Número inicial de paquetes para coincidir: este número se recarga en uno cada vez/cuenta, hasta este número
- **log-prefix** *(string; Default:)* - Añade texto especificado al principio de cada mensaje de registro (`log`). Se aplica si `action=log`
- **nth** *(integer,integer; Default:)* – Coincide cada n-ésimo (nth) paquete.
- **out-bridge-port** *(name; Default:)* - Interface real por la que el paquete abandona al router, si la interface de salida es `bridge`. Trabaja únicamente si en la configuración de `bridge` se habilita `use-ip-firewall`.
- **out-interface** *(; Default:)* – Interface por la que el paquete abandona el router
- **packet-mark** *(no-mark | string; Default:)* - Coincide con los paquetes marcados vía mangle con una marca de paquete particular (`packet mark`). Si se configura como `no-mark`, la regla coincidirá con cualquier paquete que no tenga marca.
- **packet-size** *(integer[-integer]:0..65535; Default:)* – Coincide con los paquetes de un tamaño específico o un rango de tamaño en bytes.
- **per-connection-classifier** *(ValuesToHash:Denominator/Remainder; Default:)* - Permite dividir el tráfico en streams iguales con la capacidad de mantener los paquetes con un conjunto específico de opciones en una stream particular.
- **port** *(integer[-integer]: 0..65535; Default:)* – Coincide si cualquier puerto (origen o destino) coincide con la lista específica de puertos o rangos de puertos. Se aplica solo si el protocolo es `TCP` o `UDP`
- **protocol** *(name or protocol ID; Default: tcp)* – Coincide con un protocolo IP específico por nombre de protocolo o por número de protocolo
- **psd** *(integer,time,integer,integer; Default:)* – Intenta detectar los escaneos TCP y UDP. Los parámetros están en el siguiente formato `WeightThreshold`, `DelayThreshold`, `LowPortWeight`, `HighPortWeight`
    - **WeightThreshold** – Peso total de los últimos paquetes `TCP/UDP` con diferentes puertos de destino procedentes del mismo host para ser tratado como una secuencia de escaneo de puertos
    - **DelayThreshold** – Retardo de los paquetes con diferentes puertos destino que proceden del mismo host paa ser tratados como una posible subsecuencia de escaneo de puertos
    - **LowPortWeight** – Peso de los paquetes con puerto de destino privilegiado (<=1024)
    - **HighPortWeight** – Peso de los paquetes con puerto de destino no-privilegiado
- **random** *(integer: 1..99; Default:)* – Coincide con los paquetes al azar con una probabilidad dada.
- **routing-mark** *(string; Default:)* – Coincide con los paquetes marcados por mangle con una marca de ruteo (`routing mark`) particular
- **same-not-by-dst** (yes | no; Default:) - Especifica si se toma en cuenta o no la dirección IP de destino cuando se selecciona una nueva dirección IP de origen. Se aplica si `action=same`
- **src-address** *(Ip/Netmaks, Ip range; Default:)* – Coincide con los paquetes cuya fuente es igual a la IP especificada o cae dentro de un rango IP específico.

- **src-address-list** *(name; Default:)* – Coincide con la dirección origen de un paquete contra el `address list` definido por el usuario
- **src-address-type** *(unicast | local | broadcast | multicast; Default:)* – Coincide con el tipo de dirección origen:
  - **unicast** – Dirección IP usada para transmisión punto a punto
  - **local** – Si la dirección es asignada a una de las interfaces del router
  - **broadcast** – El paquete es enviado a todos los dispositivos en una subred
  - **multicast** – El paquete es enviado a un grupo definido de dispositivos
- **src-port** *(integer[-integer]: 0..65535; Default:)* – Lista de puertos origen y rangos de puertos origen. Aplicable solo si el protocolo es `TCP` o `UDP`.
- **src-mac-address** *(MAC address; Default:)* – Coincide con la dirección MAC del paquete
- **tcp-flags** *(ack | cwr | ece | fin | psh | rst | syn | urg; Default:)* – Coincide con las banderas TCP específicas
  - **ack** – Reconocimiento de la data
  - **cwr** – Venta de congestión reducida
  - **ece** – Bandera `ECN-echo` (notificación de congestión explícita)
  - **fin** – Conexión cerrar (close)
  - **psh** – Función push (empujar)
  - **rst** – Conexión rechazar (drop)
  - **syn** – Nueva conexión (new)
  - **urg** – Data urgente
- **tcp-mss** *(integer: 0..65535; Default:)* – Coincide con el valor `TCP MSS` de un paquete IP
- **time** *(time-time,sat | fri | thu | wed | tue | mon | sun; Default:)* – Permite crear un filtro basado en el tiempo y fecha de arribo de un paquete, o para paquetes generados localmente, tiempo y fecha de partida
- **to-addresses** *(IP address[-IP address]; Default: 0.0.0.0)* – Reemplaza la dirección original con otra específica. Es aplicable si `action=dst-nat`, `action=netmap`, `action=same`, `action=src-nat`
- **to-ports** ***(integer**[-integer]: 0..255; Default:)* - Reemplaza el puerto original con otro específico. Es aplicable si `action=dst-nat`, `action=netmap`, `action=same`, `action=src-nat`
- **ttl** *(integer: 0..255; Default:)* – Coincide con el valor `TTL` de los paquetes

Estadísticas

`/ip firewall nat print stats` mostrará propiedades adicionales solo de lectura

- **bytes** *(integer)* – Cantidad total de bytes que coinciden con la regla
- **packets** *(integer)* – Cantidad total de paquetes que coinciden con la regla

Por default `print` es equivalente a `print static` y muestra únicamente las reglas estáticas

```
/ip firewall mangle print stats
Flags: X - disabled, I - invalid, D - dynamic
 #   CHAIN             ACTION                BYTES           PACKETS
 0   prerouting        mark-routing          17478158        127631
 1   prerouting        mark-routing          782505          4506
```

Para imprimir las reglas dinámicas se debe usar `print all`

```
/ip firewall mangle print all stats
Flags: X - disabled, I - invalid, D - dynamic
 #   CHAIN             ACTION                BYTES           PACKETS
 0   prerouting        mark-routing          17478158        127631
 1   prerouting        mark-routing          782505          4506
 2 D forward           change-mss            0               0
 3 D forward           change-mss            0               0
 4 D forward           change-mss            0               0
 5 D forward           change-mss            129372          2031
```

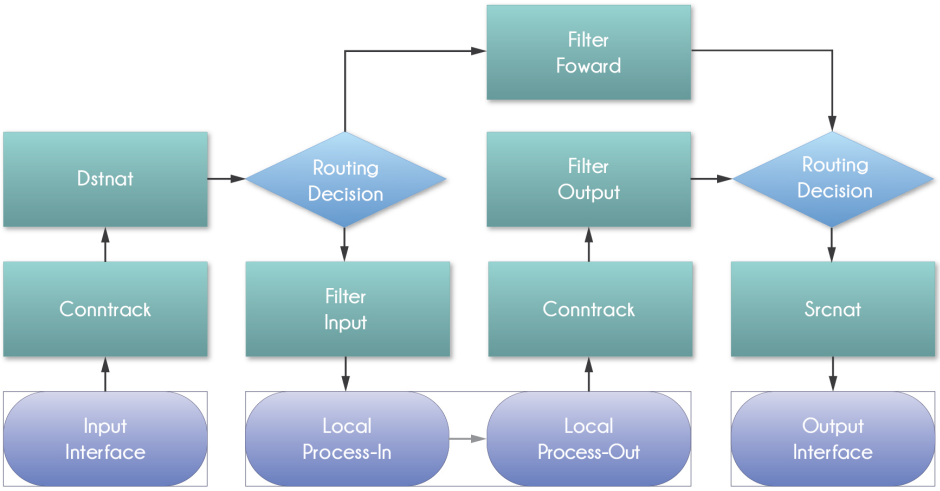
Para imprimir únicamente las reglas dinámicas se debe usar `print dynamic`

```
/ip firewall mangle> print stats dynamic
Flags: X - disabled, I - invalid, D - dynamic
 #   CHAIN             ACTION                BYTES           PACKETS
 0 D forward           change-mss            0               0
 1 D forward           change-mss            0               0
 2 D forward           change-mss            0               0
 3 D forward           change-mss            132444          2079
```

Estructura del Firewall NAT

- Dstnat – procesa el trafico que se envía a y a través del router, antes de dividirlo en las cadenas `input` y `forward` de los filtros de firewall

- Srcnat – procesa el trafico que se envía desde y a través del router, después de que se ha fusionado de los chains `output` y `forward` de los filtros de firewall
- Existen también chains definidos por el usuario

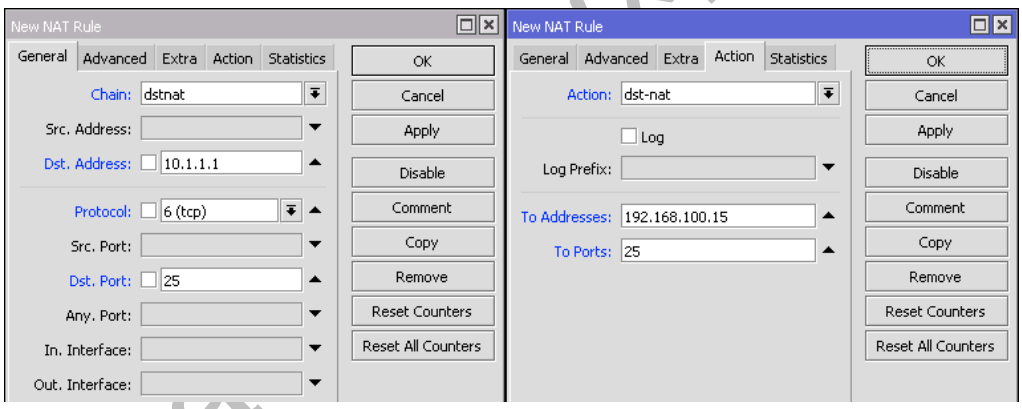


dst-nat

La acción “dst-nat” cambia la dirección y puerto destino del paquete a una dirección y puerto especificado

- Esta acción solo puede hacerse en el chain dstnat
- Aplicación típica: asegurar el acceso a los servicios de red local desde redes públicas

Ejemplo de `dst-nat`

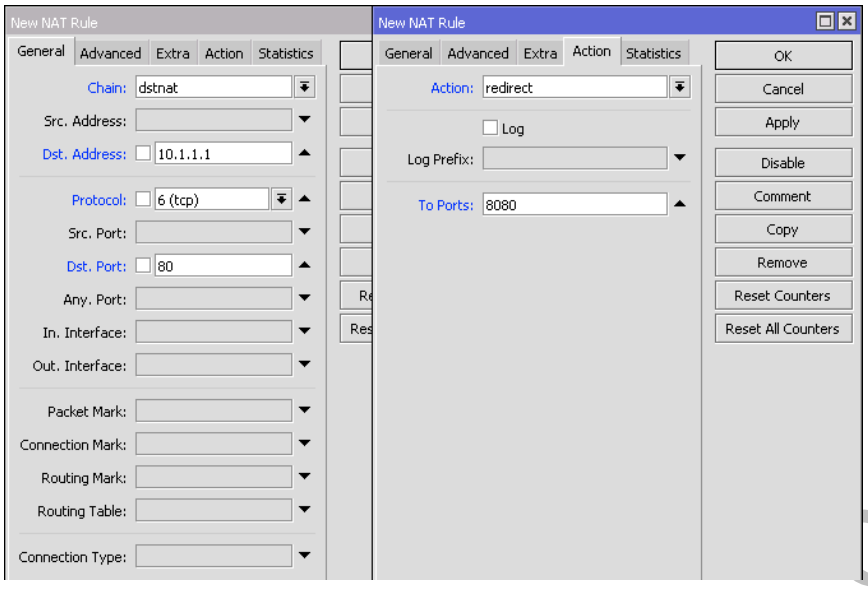


redirect

redirect cambia la dirección destino del paquete a la dirección y puerto especificado del router. Esta acción solo puede ejecutarse en el chain `dstnat`

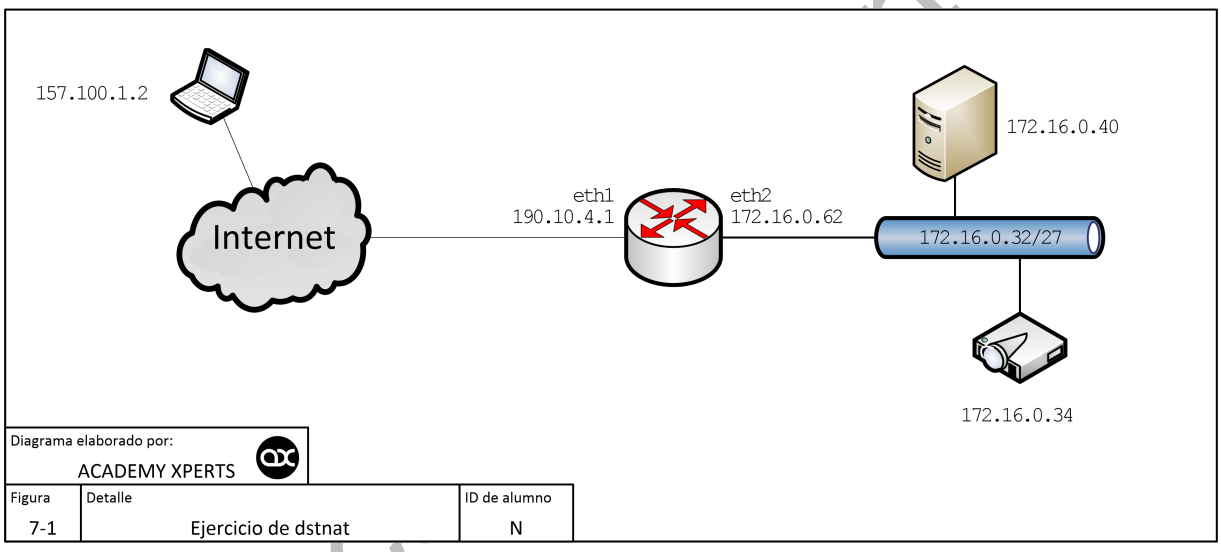
- Aplicación típica: hacer proxy transparente de servicios de red (DNS, HTTP)

Ejemplo de `redirect`



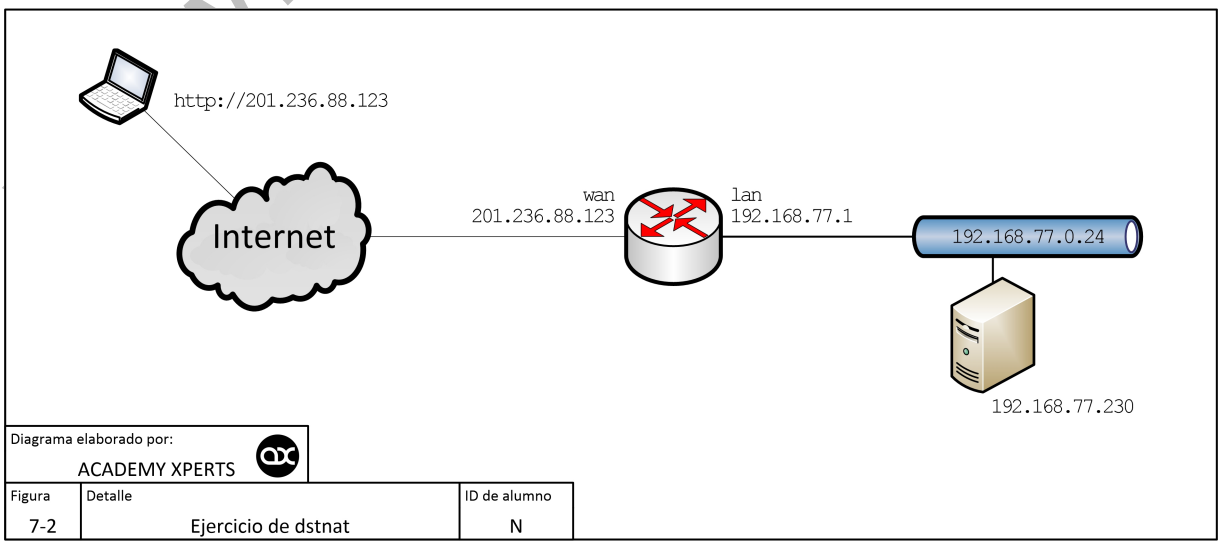
Laboratorio 7-1: Ejercicio de dstnat

El estudiante debe comprender el funcionamiento del firewall basado en el Diagrama de Flujo



Laboratorio 7-2: Ejercicio de dstnat

El estudiante debe comprender el funcionamiento del firewall basado en el Diagrama de Flujo



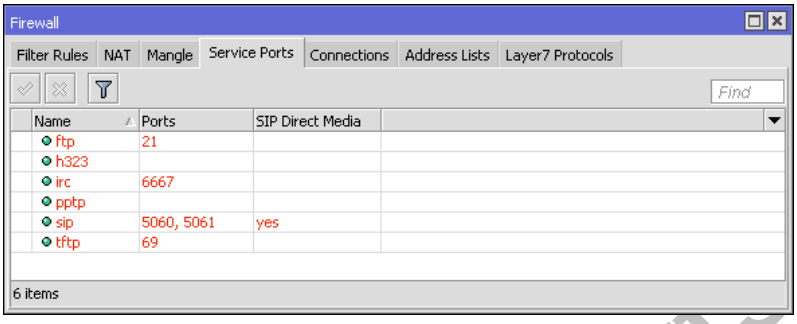
Inconvenientes del Source NAT

Los hosts atrás de un router NATeado no tienen una verdadera conectividad end-to-end

- No se puede iniciar una conexión desde afuera
- Algunos servicios TCP trabajarán en modo pasivo
- El src-nat detrás de varias direcciones IP es impredecible
- Algunos protocolos requerirán los llamados NAT Helpers para poder trabajar correctamente (NAT Traversal)

NAT Helpers (Service Ports)

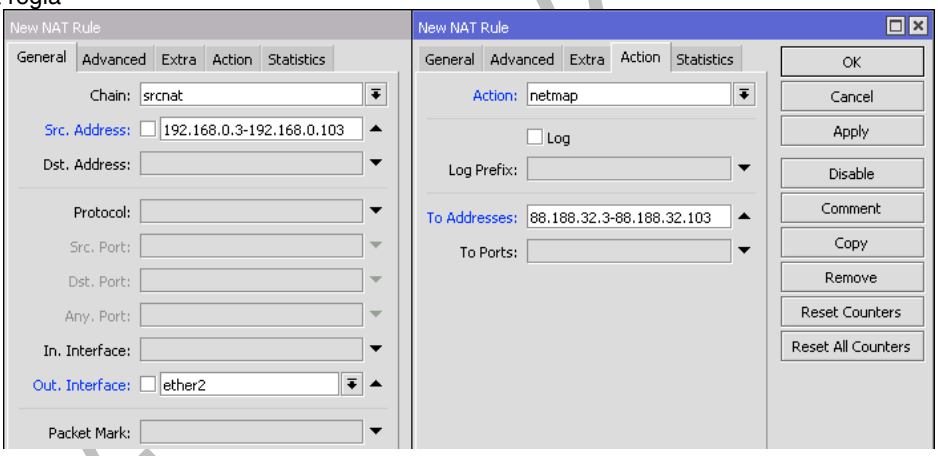
Se puede especificar puertos para los NAT Helpers que ya existen, pero no se puede añadir nuevos Helpers



NAT `action=netmap`

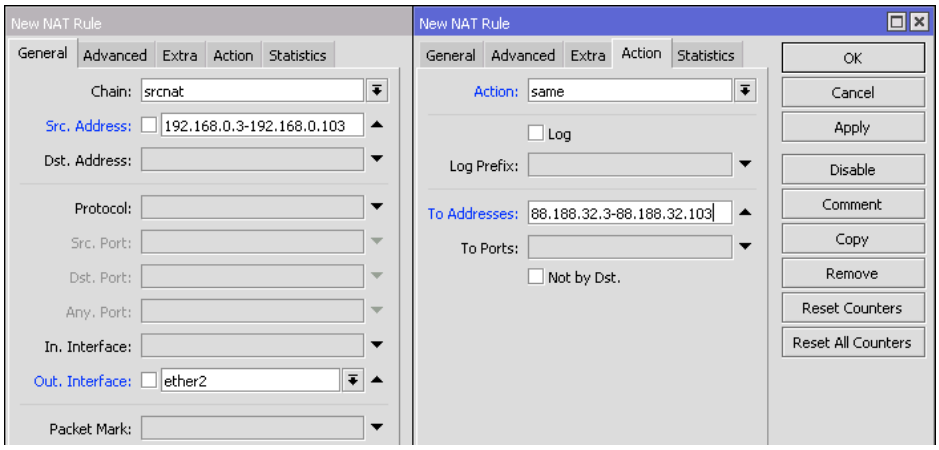
Puede ser usado en ambos chains `src-nat` y `dst-nat`

- Permite crear un NATeo de un rango de direcciones a otro rango de direcciones con solo una regla
- Se puede enmascarar `192.168.0.3-192.168.0.103` (100 direcciones) a `88.188.32.3-88.188.32.103` con solo una regla
- Es posible re direccionar `88.188.32.3-88.188.32.103` (100 direcciones) a `192.168.0.3-192.168.0.103` con una segunda regla



NAT `action=same`

Puede ser usado en ambos chains `src-nat` y `dst-nat`
Asegura que el cliente será NATeado a la misma dirección del rango especificado cada vez que intente comunicarse con un destino que fue usado antes
Si el cliente obtiene la IP `88.188.32.104` del rango cuando se comunicó a un server específico, cada vez que se comunique con ese Server usará la misma dirección



Capítulo 8: Firewall Mangle

Marcado de paquetes IP y ajuste de campos de cabecera IP

Mangle es una especie de “marcador” que etiqueta/marca paquetes para un futuro procesamiento con marcas especiales. Muchas facilidades/opciones en el RouterOS hacen uso de estas marcas como por ejemplo Queue Trees, NAT, routing. Las marcas de mangle únicamente existen dentro del mismo router, y estas marcas NO son transmitidas a través de la red.

Adicionalmente, el mangle se utiliza para modificar algunos campos en la cabecera IP como los campos TOS (DSCP) y TTL.

Propiedades

- **action** *(action name; Default: accept)* – Acción que se va a tomar si el paquete coincide con la regla:
  - **accept** – Acepta el paquete. El paquete no se pasa a la siguiente regla de firewall
  - **add-dst-to-address-list** – Se agrega una dirección destino a un address list especificado en el parámetro `address-list`
  - **add-src-to-address-list** – Se agrega una dirección origen a un address list especificado en el parámetro `address-list`
  - **change-dscp** – Cambia el valor del campo `DSCP` especificado por el parámetro `new-dscp` (DSCP = Differentiated Services Code Point)
  - **change-mss** – Cambia el valor del campo `MSS` (Maximum Segment Size) del paquete a un valor especificado por el parámetro `new-mss`
  - **change-ttl** – Cambia el valor del campo `TTL` (Time to Live) del paquete a un valor especificado por el parámetro `new-ttl`
  - **clear-df** – Limpia la bandera `'Do Not Fragment'`
  - **jump** – Se salta al chain definido por el usuario. Este valor se especifica en el parámetro `jump-target`
  - **log** – Se agrega un mensaje al log del sistema (system log) que contiene la siguiente información: `in-interface, out-interface, src-mac, protocol, src-ip:port->dst-ip:port` y longitud del paquete (`length of the packet`). Después de que el paquete coincide con esta regla, se pasa a la siguiente regla en la lista, similar a la acción `passthrough`
  - **mark-connection** – Ubica una marca especificada por el parámetro `new-connection-mark` en la conexión entera que coincide con la regla
  - **mark-packet** – Ubica una marca especificada por el parámetro `new-packet-mark` en un paquete que coincide con la regla
  - **mark-routing** – Ubica una marca especificada por el parámetro `new-routing-mark` en un paquete. Este tipo de marcas se utiliza únicamente para propósitos de políticas de ruteo.
  - **passthrough** – Ignora esta regla y el paquete pasa a la siguiente regla. Esta acción es útil para trabajar con estadísticas.
  - **return** – Pasa el control de nuevo al `chain` donde tuvo lugar el `jump`
  - **set-priority** – Establece la prioridad especificada por el parámetro `new-priority` en los paquetes enviados a través de un enlace que es capaz de transportar prioridad (`VLAN` o una interface wireless con `WMM` habilitado).
  - **sniff-pc**
  - **sniff-tzsp** – Se envía paquetes a un sistema remoto compatible `TZSP` (por ejemplo, Wireshark). Se establece el destino remoto con los parámetros `sniff-target` y `sniff-target-port` (Wireshark recomienda el puerto `37008`)
  - **strip-ipv4-options** – Desmonta los campos de opción IPv4 de la cabecera IP.
- **address-list** *(string; Default:)* - Nombre de la lista de direcciones (`address list`) que se utilizará. Se puede aplicar si `action=add-dst-to-address-list` o `action=add-src-to-address-list`
- **address-list-timeout** *(time; Default: 00:00:00)* – Especifica el intervalo de tiempo después del cual la dirección será removida del address list especificado en el parámetro `address-list`. Se usa en conjunto con las acciones `add-dst-to-address-list` o `add-src-to-address-list`. Cuando se especifica un valor de `00:00:00` significa que la dirección IP se dejará por siempre en el address list
- **chain** *(name; Default:)* – Especifica a que `chain` se agregará la regla. Si la entrada no coincide con el nombre de un `chain` ya definido, se creará un nuevo `chain`.
- **comment** *(string; Default:)* – Comentario descriptivo de esta regla.
- **connection-bytes** *(integer-integer; Default:)* - Coincide con los paquetes solamente si una determinada cantidad de bytes ha sido transferido a través de la conexión particular. 0 – significa infinito. Por ejemplo, `connection-bytes=2000000-0` significa que la regla hace coincidencia si más de 2MB han sido transferidos a través de esa conexión relevante.
- **connection-limit** *(integer,netmask; Default:)* - Restringir el límite de conexiones por dirección o bloque de direcciones hasta e incluyendo el valor dado
- **connection-mark** *(no-mark | string; Default:)* – Coincide con los paquetes marcados vía mangle con una marca de conexión particular (`connection mark`). Si se configura como `no-mark`, la regla coincidirá con cualquier conexión que no tenga marca.

- **connection-rate** *(Integer 0..4294967295; Default:)* – Permite capturar el tráfico basado en la velocidad actual de la conexión.
- **connection-state** *(estabilished | invalid | new | related; Default:)* – Interpreta el análisis de datos del `connection tracking` de un paquete en particular:
  - **established** – Un paquete que pertenece a una conexión existente
  - **invalid** – Un paquete que no pudo ser identificado por algún motivo
  - **new** – El paquete ha iniciado una nueva conexión, o de otra manera asociado con una conexión de la que no se ha visto paquetes en ambas direcciones.
  - **related** – Un paquete que se relaciona con, pero que no es parte de una conexión existente, tales como errores ICMP o un paquete que inicia la conexión de datos
- **connection-type** *(ftp | h323 | irc | pptp | quake3 | sip | tftp; Default:)* - Coincide con los paquetes de conexiones relacionadas basado en la información de sus ayudantes de seguimiento de conexión (connection tracking helpers). Un ayudante (helper) de conexión relevante debe estar habilitado en `/ip firewall service-port`
- **content** *(string; Default:)* – Coincide con los paquetes que contienen un texto específico
- **dscp** *(integer: 0..63; Default:)* – Coincide con el campo de cabecera (header field) `DSCP IP`.
- **dst-address** *(IP/netmask | IP range; Default:)* - Coincide con los paquetes cuyo destino es igual a la IP especificada o cae dentro del rango IP especificado.
- **dst-address-list** *(name; Default:)* - Coincide con la dirección de destino de un paquete contra la lista de direcciones definido por el usuario
- **dst-address-type** *(unicast | local | broadcast | multicast; Default:)* – Coincide con el tipo de dirección destino:
  - **unicast** – Dirección IP usada para transmisión punto a punto
  - **local** – Si el `dst-address` está signado a una de las interfaces del router
  - **broadcast** – El paquete es enviado a todos los dispositivos en una subred
  - **multicast** – El paquete es enviado a un grupo definido de dispositivos
- **dst-limit** *(integer[/time],integer,dst-address | dst-port | src-address[/time]; Default:)* – Coincide con los paquetes hasta que una tasa (rate) dada es excedida. La tasa (rate) se define como paquetes por intervalo de. A diferencia del parámetro `limit`, cada flujo tiene su propio límite. El flujo se define por el parámetro `mode` (modo). Los parámetros se escriben en el siguiente formato: `count[/time],burst,mode[/expire]`.
  - **count** – Conteo de paquetes por intervalo de tiempo por flujo para que coincida
  - **time** – Especifica el intervalo de tiempo en el cual el conteo de paquetes por flujo no puede ser excedido (opcional, Si no se especifica nada se utilizará `1s`)
  - **burst** – Número inicial de paquete por flujo para coincidir: este número se recarga en uno cada vez/cuenta, hasta este número
  - **mode** – Este parámetro especifica qué campos únicos definen el flujo (`src-address, dst-address, src-and-dst-address, dst-address-and-port, addresses-and-dst-port`)
  - **expire** – Especifica el intervalo después del cual el flujo sin paquetes será permitido eliminar (opcional)
- **dst-port** *(integer[-integer]: 0..65535; Default:)* – Lista de números de puerto destino o rangos de número de puerto
- **fragment** (yeslno; Default:) – Coincide con los paquetes fragmentados. El primer paquete fragmentado (inicial) no cuenta. Si el `connection tracking` está habilitado entonces no habrá fragmentos ya que el sistema ensambla automáticamente cada paquete
- **hotspot** *(auth | from-client | http | local-dst | to-client; Default:)*
- **icmp-options** *(integer:integer; Default:)* – Coincide con los campos ICMP type:code
- **in-bridge-port** *(name; Default:)* – Interface real por la que el paquete ha ingresado al router, si la interface entrante es `bridge`. Trabaja únicamente si en la configuración de `bridge` se habilita `use-ip-firewall`.
- **in-interface** *(name; Default:)* – Interface por la que el paquete ha ingresado al router
- **ingress-priority** *(integer: 0..63; Default:)* – Coincide con la prioridad de ingreso del paquete. Prioridad se puede derivar de VLAN, WMM o MPLS EXP bit.
- **ipsec-policy** *(in | out, ipsec | none; Default:)* – Coincide con la política usada por `IPSec`. El valor es escrito en el siguiente formato: `direction, policy`. Direction se utiliza para seleccionar si coincide con la política usada para `decapsulation` o la política que será usada para `encapsulation`.
  - **in** – Válida en los chains `PREROUTING`, `INPUT` y `FORWARD`
  - **out** – Válida en los chains `POSTROUTING`, `OUTPUT` y `FORWARD`
  - **ipsec** – Coincide si el paquete está sujeto a procesamiento `IPsec`
  - **none** – Coincide con el paquete de transporte `IPsec`
  - Por ejemplo, si el router recibe `IPsec` con el paquete `GRE` encapsulado, entonces la regla `ipsec-policy=in,ipsec` coincidirá el paquete `GRE`, sino la regla `ipsec-policy=in,none` coincidirá el paquete `ESP`.
- **ipv4-options** *(any | loose-source-routing | no-record-route | no-router-alert | no-source-routing | no-timestamp | none | record-route | router-alert | strict-source-routing | timestamp; Default:)* – Coincide con las opciones de la cabecera IPv4.
  - **any** – Coincide con el paquete con al menos una de las opciones `IPv4`

    - **loose-source-routing** – Coinciden con los paquete con opción `loose source routing`. Esta opción se utiliza para rutear el datagrama internet en base a la información suministrada por la fuente
    - **no-record-route** – Coincide con los paquetes con opción `no record route`. Esta opción se utiliza para rutear el datagrama internet en base a la información suministrada por la fuente
    - **no-router-alert** – Coincide con los paquetes con opción `no router alter`
    - **no-source-routing** - Coincide con los paquetes con opción `no source routing`
    - **no-timestamp** – Coincide con los paquetes con opción `no timestamp`
    - **record-route** – Coincide con los paquetes con opción `record route`
    - **router-alert** – Coincide con los paquetes con opción `router alter`
    - **strict-source-routing** – Coincide con los paquetes con opción `strict source routing`
    - **timestamp** - Coincide con los paquetes con `timestamp`
- **jump-target** *(name; Default:)* – Nombre del chain destino al cual debe saltar. Se aplica solo si `action=jump`
- **layer7-protocol** *(name; Default:)* – Nombre del filtro `Layer7` definido en menú de protocolo `layer7`.
- **limit** *(integer,time,integer; Default:)* – Coincide con los paquetes a una velocidad limitada. Regla que usa este matcher coincidirá hasta que se alcance este límite. Los parámetros se escriben en formato siguiente: `count[/time],burst`.
    - **count** – Conteo de paquetes por intervalo de tiempo para que coincida
    - **time** – Especifica el intervalo de tiempo en el cual el conteo de paquetes no puede ser excedido (opcional, si no especifica nada entonces se usará `1s`)
    - **burst** – Número inicial de paquetes para coincidir: este número se recarga en uno cada vez/cuenta, hasta este número
- **log-prefix** *(string; Default:)* - Añade texto especificado al principio de cada mensaje de registro (`log`). Se aplica si `action=log`
- **new-connection-mark** (string; Default:)
- **new-dscp** (integer: 0..63; Default:)
- **new-mss** (integer; Default:)
- **new-packet-mark** (string; Default:)
- **new-priority** (integer; Default:)
- **new-routing-mark** (string; Default:)
- **new-ttl** (decrement | increment | set:integer; Default:)
- **nth** *(integer,integer; Default:)* – Coincide cada n-ésimo (nth) paquete.
- **out-bridge-port** *(name; Default:)* - Interface real por la que el paquete abandona al router, si la interface de salida es `bridge`. Trabaja únicamente si en la configuración de `bridge` se habilita `use-ip-firewall`.
- **out-interface** *(; Default:)* – Interface por la que el paquete abandona el router
- **p2p** (all-p2p | bit-torrent | blubster | direct-connect | edonkey | fasttrack | gnutella | soulseek | warez | winmx; Default:) - Coincide con los paquetes de parios protocolos peer-to-peer (P2P). No trabaja en paquetes p2p encriptados.
- **packet-mark** *(no-mark | string; Default:)* - Coincide con los paquetes marcados vía mangle con una marca de paquete particular (`packet mark`). Si se configura como `no-mark`, la regla coincidirá con cualquier paquete que no tenga marca.
- **packet-size** *(integer[-integer]:0..65535; Default:)* – Coincide con los paquetes de un tamaño específico o un rango de tamaño en bytes.
- **per-connection-classifier** *(ValuesToHash:Denominator/Remainder; Default:)* - Permite dividir el tráfico en streams iguales con la capacidad de mantener los paquetes con un conjunto específico de opciones en una stream particular.
- **port** *(integer[-integer]: 0..65535; Default:)* – Coincide si cualquier puerto (origen o destino) coincide con la lista específica de puertos o rangos de puertos. Se aplica solo si el protocolo es `TCP` o `UDP`
- **protocol** *(name or protocol ID; Default: tcp)* – Coincide con un protocolo IP específico por nombre de protocolo o por número de protocolo
- **psd** *(integer,time,integer,integer; Default:)* – Intenta detectar los escaneos TCP y UDP. Los parámetros están en el siguiente formato `WeightThreshold`, `DelayThreshold`, `LowPortWeight`, `HighPortWeight`
    - **WeightThreshold** – Peso total de los últimos paquetes `TCP/UDP` con diferentes puertos de destino procedentes del mismo host para ser tratado como una secuencia de escaneo de puertos
    - **DelayThreshold** – Retardo de los paquetes con diferentes puertos destino que proceden del mismo host paa ser tratados como una posible subsecuencia de escaneo de puertos
    - **LowPortWeight** – Peso de los paquetes con puerto de destino privilegiado (<=1024)
    - **HighPortWeight** – Peso de los paquetes con puerto de destino no-privilegiado
- **random** *(integer: 1..99; Default:)* – Coincide con los paquetes al azar con una probabilidad dada.
- **routing-mark** *(string; Default:)* – Coincide con los paquetes marcados por mangle con una marca de ruteo (`routing mark`) particular
- **src-address** *(Ip/Netmaks, Ip range; Default:)* – Coincide con los paquetes cuya fuente es igual a la IP especificada o cae dentro de un rango IP específico.

- **src-address-list** *(name; Default:)* – Coincide con la dirección origen de un paquete contra el `address list` definido por el usuario
- **src-address-type** *(unicast | local | broadcast | multicast; Default:)* – Coincide con el tipo de dirección origen:
  - **unicast** – Dirección IP usada para transmisión punto a punto
  - **local** – Si la dirección es asignada a una de las interfaces del router
  - **broadcast** – El paquete es enviado a todos los dispositivos en una subred
  - **multicast** – El paquete es enviado a un grupo definido de dispositivos
- **src-port** *(integer[-integer]: 0..65535; Default:)* – Lista de puertos origen y rangos de puertos origen. Aplicable solo si el protocolo es `TCP` o `UDP`.
- **src-mac-address** *(MAC address; Default:)* – Coincide con la dirección MAC del paquete
- **tcp-flags** *(ack | cwr | ece | fin | psh | rst | syn | urg; Default:)* – Coincide con las banderas TCP específicas
  - **ack** – Reconocimiento de la data
  - **cwr** – Venta de congestión reducida
  - **ece** – Bandera `ECN-echo` (notificación de congestión explícita)
  - **fin** – Conexión cerrar (close)
  - **psh** – Función push (empujar)
  - **rst** – Conexión rechazar (drop)
  - **syn** – Nueva conexión (new)
  - **urg** – Data urgente
- **tcp-mss** *(integer: 0..65535; Default:)* – Coincide con el valor `TCP MSS` de un paquete IP
- **time** *(time-time,sat | fri | thu | wed | tue | mon | sun; Default:)* – Permite crear un filtro basado en el tiempo y fecha de arribo de un paquete, o para paquetes generados localmente, tiempo y fecha de partida
- **ttl** *(integer: 0..255; Default:)* – Coincide con el valor `TTL` de los paquetes

Estadísticas

`/ip firewall mangle print stats` mostrará propiedades adicionales solo de lectura

- **bytes** (integer) – Cantidad total de bytes que coinciden con la regla
- **packets** (integer) – Cantidad total de paquetes que coinciden con la regla

Por default `print` es equivalente a `print static` y muestra únicamente las reglas estáticas

```
/ip firewall mangle> print stats
Flags: X - disabled, I - invalid, D - dynamic
 #   CHAIN              ACTION                BYTES          PACKETS
 0   prerouting         mark-routing          17478158       127631
 1   prerouting         mark-routing          782505         4506
```

Para imprimir las reglas dinámicas se debe usar `print all`

```
/ip firewall mangle print all stats
Flags: X - disabled, I - invalid, D - dynamic
 #   CHAIN              ACTION                BYTES          PACKETS
 0   prerouting         mark-routing          17478158       127631
 1   prerouting         mark-routing          782505         4506
 2 D forward            change-mss            0              0
 3 D forward            change-mss            0              0
 4 D forward            change-mss            0              0
 5 D forward            change-mss            129372         2031
```

Para imprimir únicamente las reglas dinámicas se debe usar `print dynamic`

```
/ip firewall mangle print stats dynamic
Flags: X - disabled, I - invalid, D - dynamic
 #   CHAIN              ACTION                BYTES          PACKETS
 0 D forward            change-mss            0              0
 1 D forward            change-mss            0              0
 2 D forward            change-mss            0              0
 3 D forward            change-mss            132444         2079
```

Change MSS

Es un hecho bien conocido de que los enlaces VPN tiene un tamaño de paquete más pequeño debido al overhead de la encapsulación. Un paquete largo con MSS que excede el MSS del enlace VPN sebe ser fragmentado antes de enviarse a través de esa conexión. Sin embargo, si el paquete tiene configurada la bandera DF, no puede ser fragmentado y debe descartarse.

En los enlaces que se ha roto el Path MTU Discovery (PMTUD) puede conducir a una serie de problemas, incluyendo problemas con transferencias de datos FTP y HTTP y servicios de e-mail

En el caso de un enlace con PMTUD roto, un decremento del MSS de los paquetes que vienen a través del enlace VPN resuelve el problema. El siguiente ejemplo demuestra como decrementar el valor del MSS vía mangle:

```
/ip firewall mangle
add out-interface=pppoe-out protocol=tcp tcp-flags=syn action=change-mss new-mss=1300 chain=forward
```

Marcado de Paquetes

El marcado de cada paquete es un recurso bastante caro, especialmente si la regla tiene que coincidir contra muchos parámetros de la cabecera IP o de la lista de direcciones que contiene cientos de entradas

Por ejemplo, se desea lo siguiente:

1. Marcar todos los **paquetes** excepto los paquetes TCP/80, y hacer coincidir estos paquetes contra la primera lista de direcciones (`address-list`)
2. Marcar todos los **paquetes** UDP y hacerlos coincidir contra la segunda lista de direcciones (`address-list`)

```
/ip firewall mangle
  add chain=forward protocol=tcp port=!80 dst-address-list=first action=mark-packet new-packet-mark=first
  add chain=forward protocol=udp dst-address-list=second action=mark-packet new-packet-mark=second
```

La configuración anterior se ve bastante simple y probablemente trabajará sin problemas en redes pequeñas. Pero si multiplicamos la cantidad de reglas por 10, y se agregan unos pocos cientos de entradas en la lista de direcciones (`address-list`), se ejecuta tráfico de 100 Mbps sobre este router, se podrá observar cuan rápido se incrementa el uso del CPU. Esto se debe a que cada regla lee la cabecera IP de cada paquete y trata de coincidir la data obtenida contra los parámetros especificados en la regla de firewall.

Afortunadamente, si el `connection tracking` está habilitado, se pueden usar marcas de conexión (`connection mark`) para optimizar esta configuración.

```
/ip firewall mangle
  add chain=forward protocol=tcp port=!80 dst-address-list=first connection-state=new action=mark-connection \
new-connection-mark=first
  add chain=forward connection-mark=first action=mark-packet new-packet-mark=first passthrough=no

  add chain=forward protocol=udp dst-address-list=second connection-state=new action=mark-connection \
new-connection-mark=second
  add chain=forward connection-mark=second action=mark-packet new-packet-mark=second passthrough=no
```

En esta nueva configuración la primera regla intenta coincidir la data de la cabecera IP únicamente del primer paquete de una nueva conexión y agrega una marca de conexión (`connection mark`). La siguiente regla ya no volverá a chequear la cabecera IP de cada paquete, tan solo comparará las marcas de conexión, dando como resultado un consumo de CPU mucho más bajo.

Adicionalmente se agregó `passthrough=no` para ayudar a reducir el consumo del CPU aún más.

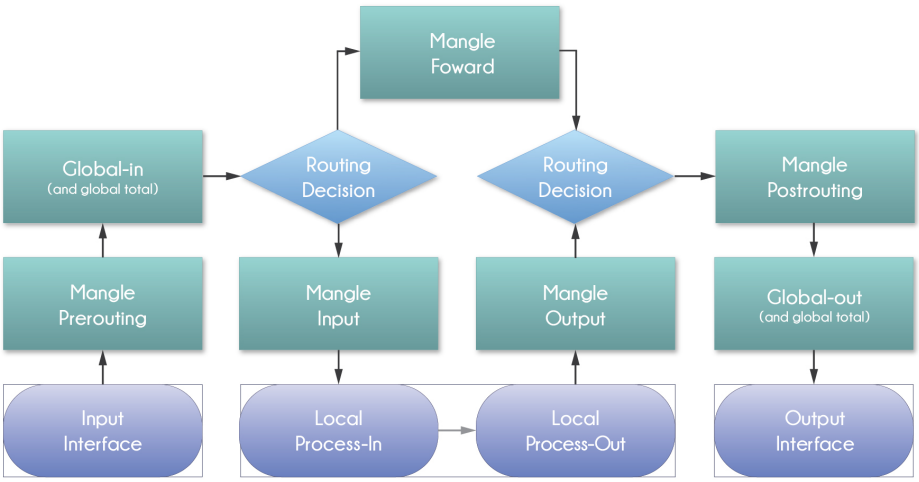
Estructura del Mangle

Las reglas del mangle están organizadas en cadenas. Existen 5 cadenas predeterminadas

- **`prerouting`**: hace una marca antes del queue Global-In
- **`postrouting`**: hace una marca antes del queue Global-Out
- **`input`**: hace una marca antes del filtro Input
- **`output`**: hace una marca antes del filtro Output
- **`forward`**: hace una marca antes del filtro Forward

Se pueden añadir nuevos chains definidos por el usuario, tantos como sean necesarios

Diagrama de Mangle y Queue



Acciones del Mangle

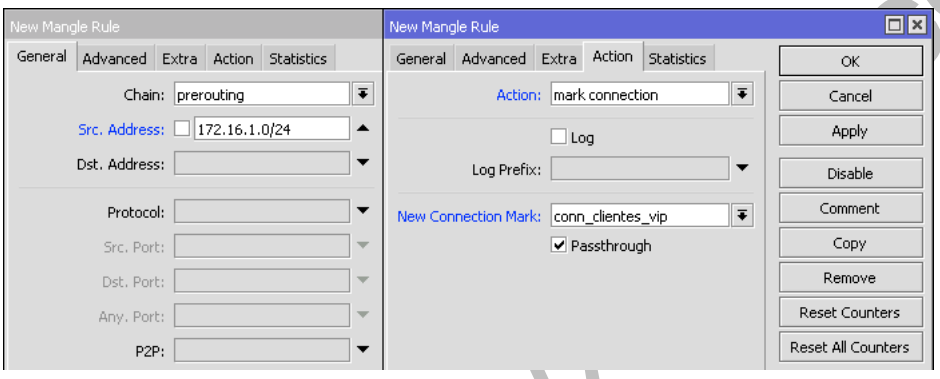
- **`mark-connection`**: marca la conexión (solo el primer paquete)
- **`mark-packet`**: marca un flujo (todos los paquetes)
- **`mark-routing`**: marca los paquetes para políticas de ruteo
- **`change-mss`**: cambia el máximo tamaño del segmento del paquete
- **`change-tos`**: cambia el tipo de servicio
- **`change-ttl`**: cambia el time-to-live
- **`Strip IPv4 options`**

Marcado de Conexiones

Se usa `mark-connection` para identificar uno o un grupo de conexiones

- Las marcas de conexiones son almacenadas en la tabla de connection tracking
- Solo puede haber una marca de conexión para una conexión
- Connection Tracking ayuda a asociar cada paquete a una conexión específica (connection mark)

Regla de Marcado de Conexión

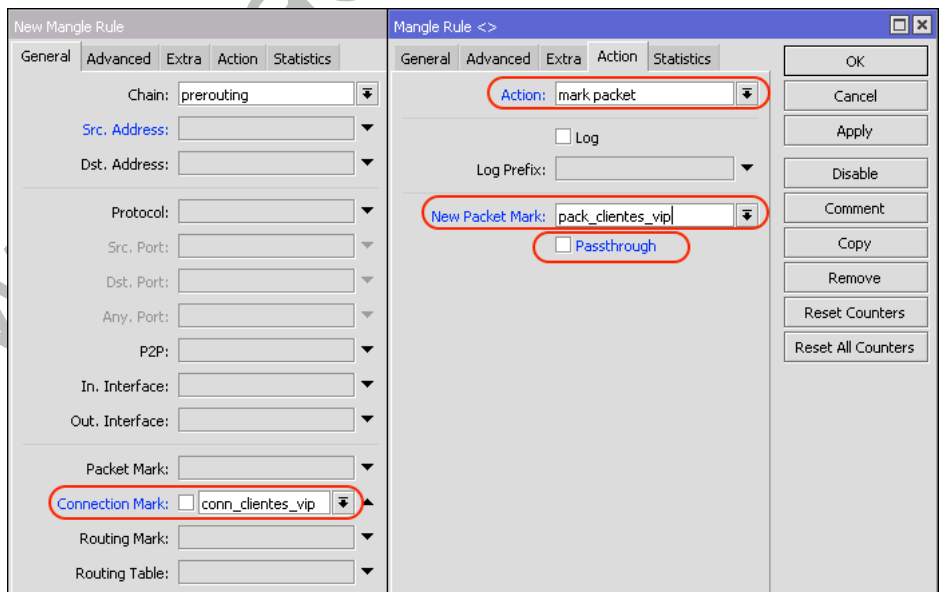


Marcado de Paquetes

Los paquetes pueden ser marcados

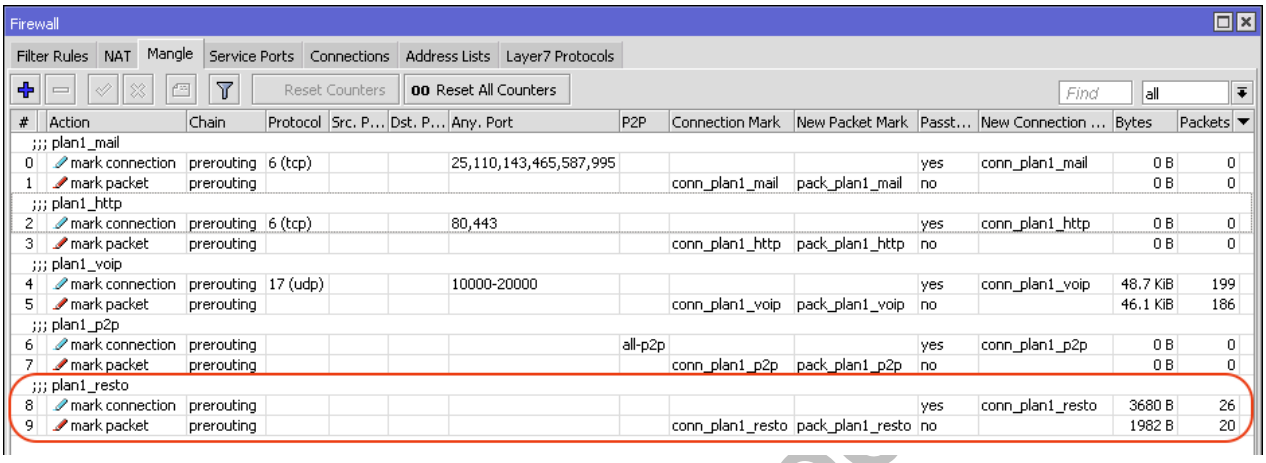
- **Indirectamente:** usando el recurso Connection Tracking, basado en conecction marks creados previamente (es más rápido)
- **Directamente:** sin el Connection Tracking. No se necesitan connection marks, el router comparará cada paquete en base a una conexión dada. Este proceso imita algo de las características del connection tracking. El router utiliza más capacidad de procesamiento del CPU

Regla del Marcado de Paquetes



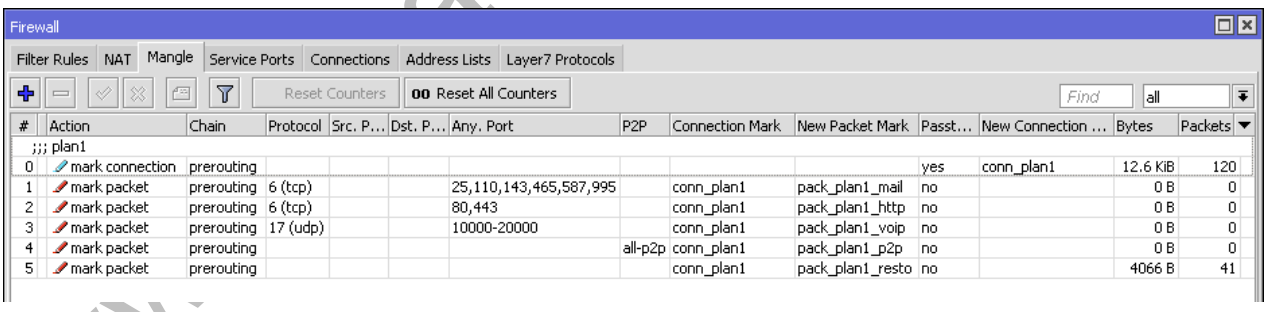
Laboratorio 8-1: Ejercicio de mangle #1

- Configurar las reglas de mangle (conexión y paquetes) para realizar QoS
- Este grupo de reglas se realizará por cada servicio, aprovechando cada marca de paquete la marca de conexión previa
- Los servicios sobre los que se desea hacer `connection-mark` y `packet-mark` son los siguientes:
    - Mail (TCP: POP 110, POP SSL 995, IMAP 143, IMAP SSL 993, SMTP 25, 465, 587)
    - VoIP (UDP: 10000 al 20000)
    - http/https (80, 443)
    - P2P
    - Resto del tráfico



Laboratorio 8-2: Ejercicio de mangle #2

- Configurar las reglas de mangle (conexión y paquetes) para realizar QoS
- Este grupo de reglas se realizará haciendo una sola marca de conexión. Las demás reglas de marcado de paquetes se armarán basados en la marca de conexión principal
- Se utilizará el mismo `address-list` del Laboratorio 8-1
- Los servicios sobre los que se desea hacer `connection-mark` y `packet-mark` son los siguientes:
    - Mail (TCP: POP 110, POP SSL 995, IMAP 143, IMAP SSL 993, SMTP 25, 465, 587)
    - VoIP (UDP: 10000 al 20000)
    - http/https (80, 443)
    - P2P
    - Resto del tráfico



Capítulo 9: HTB

Hierarchical Token Bucket

HTB (Hierarchical Token Bucket) es un método de encolamiento de clase que es útil para manejar diferente tipo de tráfico. Se deben seguir tres pasos básicos para crear una estructura HTB:

- **Coincidir y marcar tráfico. -** Clasificar el tráfico para uso posterior. Consiste de uno o más parámetros de coincidencia para seleccionar paquetes de la clase específica
- **Crear reglas (políticas) para marcar tráfico. -** Poner la clase de tráfico específico dentro de una cola específica y definir las acciones que se tomarán para cada clase.
- **Adjuntar la política para la(s) interface(s) específica(s).-** Agregar la política para todas las interfaces (`global-in`, `global-out` o `global-total` hasta v5.x; `global` a partir de v6), para la interface específica, o para la cola padre específica.

HTB permite crear una estructura de cola jerárquica y determina las relaciones entre las colas, como por ejemplo “padre-hijo” o “hijo-hijo”

Tan pronto como una cola tiene al menos un hijo se convierte en una cola interna, todas las colas sin hijos son colas-hojas. Las colas-hojas son las que realizan el consumo real de tráfico. Las colas internas únicamente son responsables de la distribución del tráfico. Todas las colas-hojas son tratadas de la misma manera.

En RouterOS es necesario especificar la opción padre (`parent`) para asignar una cola como hijo de otra cola

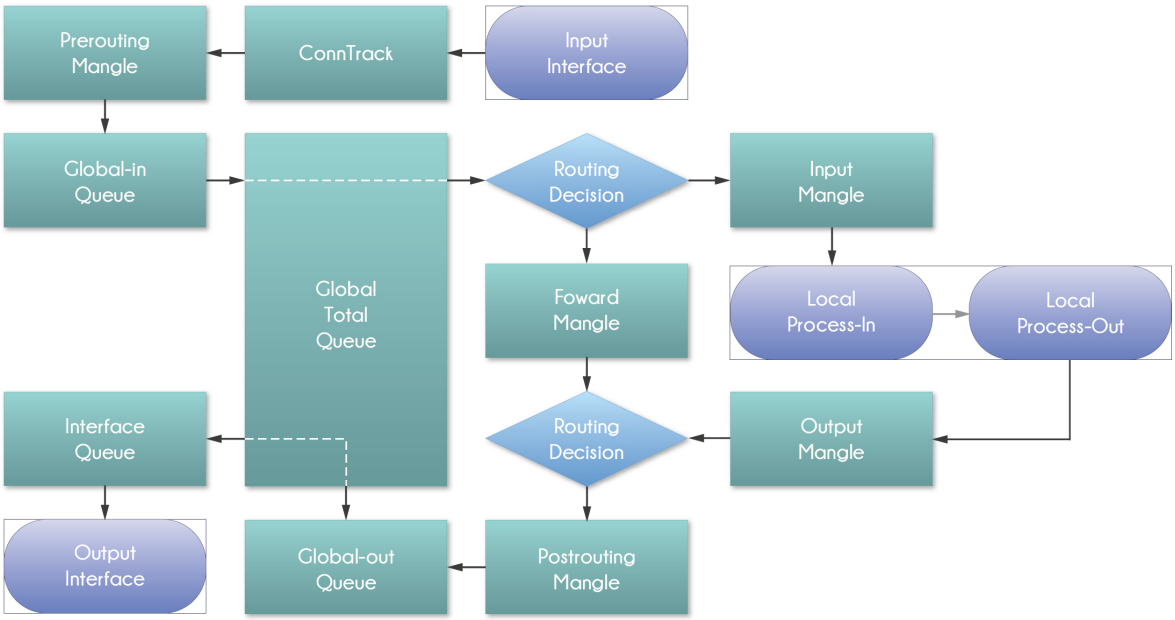
Colores de las colas en Winbox

- Verde = 0% - 50% del tráfico usado
- Amarillo = 51% - 75% del tráfico usado
- Rojo = 76% - 100% del tráfico usado

Resumen sobre HTB

- Toda la implementación de Calidad de Servicio (QoS) en RouterOS está basado en HTB
- HTB permite crear una estructura de cola jerárquica y determinar relaciones entre colas padres e hijos, y la relación entre las colas hijos
- RouterOS hasta la v5 soporta 3 HTBs virtuales (global-in, global-total, global-out) y una más justo antes de cada interface
- A partir de la v6, RouterOS soporta 1 HTB virtual (global) y una más justo antes de cada interface

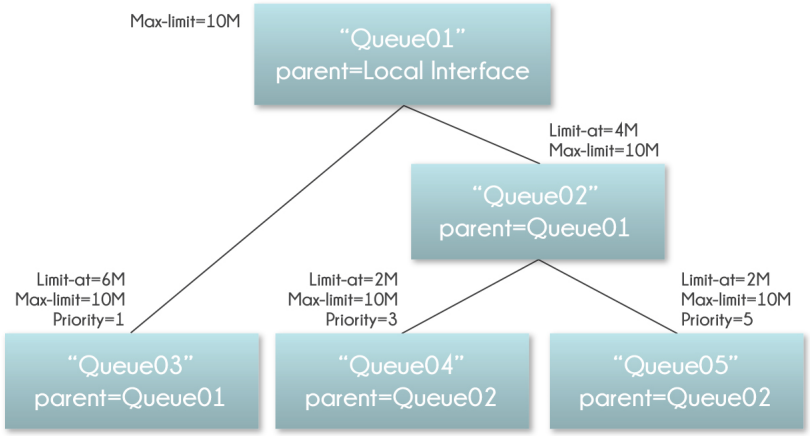
Mangle y HTBs



- Cuando el paquete viaja a través del router, pasa todos los 4 árboles HTB
- Cuando el paquete viaja hacia el router, pasa solamente los HTB global-in y global-total
- Cuando el paquete viaja desde el router pasa por global-out, global-total y la interface HTB

Estructura HTB

- Tan pronto como una cola tiene por lo menos un hijo, se vuelve una cola padre
- Todas las colas hijos (no importa cuantos niveles de padres ellos tengan) están en el mismo nivel bajo de HTB
- Las colas hijo son las que efectúan el consumo de tráfico, las colas padres son responsables únicamente de la distribución del tráfico
- Las colas hijo obtendrán primero el limit-at y luego el resto del tráfico será distribuido por los padres



Limitación Dual

Cada cola en HTB tiene dos límites de tasa:

- **CIR** (Committed Information Rate) – (**`limit-at`** en RouterOS) en el peor escenario posible, el flujo obtendrá esta cantidad de tráfico sin importar qué (asumiendo que se puede enviar realmente esa cantidad de data)
- **MIR** (Maximal Information Rate) – (**`max-limit`** en RouterOS) en el mejor escenario posible, la velocidad que se puede alcanzar si es que el padre de la cola tiene ancho de banda para repartir.

En otras palabras, al inicio se satisfará el `limit-at` (CIR) de todas las colas, solamente entonces las colas hijo tratarán de obtener la tasa de datos que necesita de sus padres a fin de alcanzar su `max-limit` (MIR)

Nota: El `limit-at` (CIR) será asignado a la cola correspondiente sin importar nada (incluso si se excede el `max-limit` del padre)

Esta es la razón por la que, para asegurar el uso óptimo (tal como fue diseñado) de la característica de limitación dual, se sugiere seguir las siguientes normas:

- La suma de los limit-at (CIR) de los hijos debe ser menor o igual que la cantidad de tráfico que está disponible para el padre:
  - `limit-at` (padre) ≥ `limit-at` (hijo1) +...+ `limit-at` (hijoN)
  - CIR (padre) ≥ CIR (hijo1) +...+ CIR (hijoN)
  - Cuando el padre es el padre principal
    - `limit-at` (padre) = `limit-at` (padre)
    - CIR(padre)=MIR(padre)
- El max-limit de cualquier hijo debe ser menor o igual que el max-limit del padre
  - `max-limit` (padre) ≥ `max-limit` (hijo1) … `max-limit` (padre) ≥ `max-limit` (hijoN)
  - MIR (padre) ≥ MIR(hijo1)... MIR (padre) ≥ MIR(hijoN)

Prioridad

Ya se conoce que a todas las colas se les entregará el `limit-at` (CIR) sin importar qué.

La prioridad es responsable de distribuir el tráfico remanente de las colas padre hacia las colas hijo de tal forma que sean capaces de alcanzar el `max-limit`.

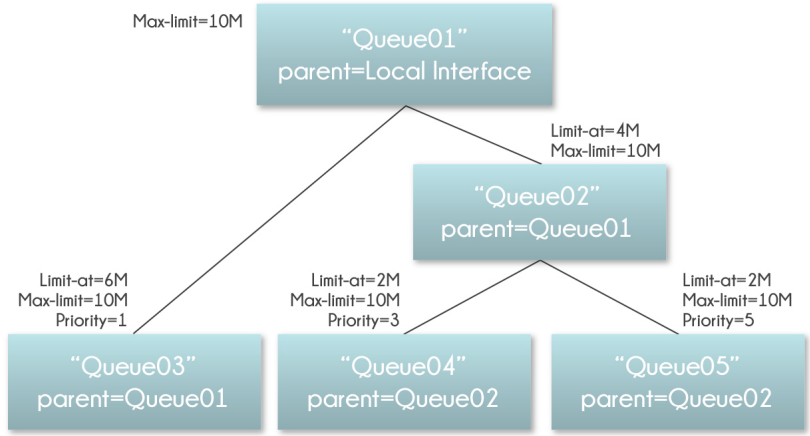
La cola con la prioridad más alta alcanzará su `max-limit` antes que la cola con prioridad más baja

- 8 es la prioridad más baja
- 1 es la prioridad más alta

Es importante recordar que la prioridad trabaja únicamente en las siguientes situaciones:

- Para las colas-hojas, la prioridad en las colas interiores no tiene significado
- Si se especifica un valor en `max-limit`. Es decir, que el `max-limit` no puede ser cero

Ejemplo 1 – caso típico



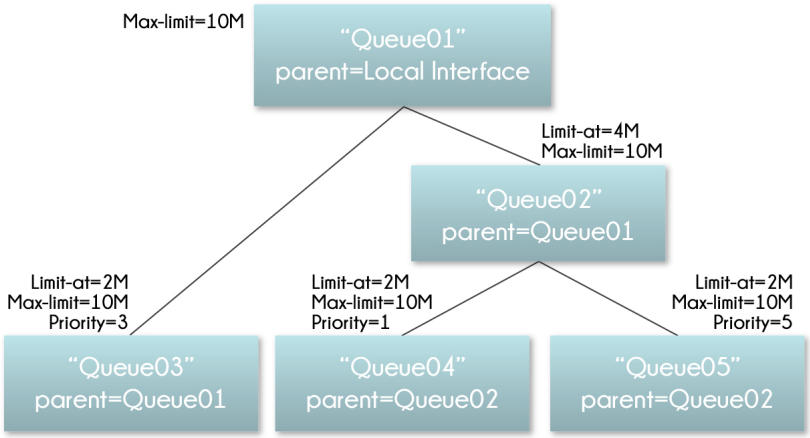

```
Queue01 limit-at=0Mbps max-limit=10Mbps
Queue02 limit-at=4Mbps max-limit=10Mbps
Queue03 limit-at=6Mbps max-limit=10Mbps priority=1
Queue04 limit-at=2Mbps max-limit=10Mbps priority=3
Queue05 limit-at=2Mbps max-limit=10Mbps priority=5
```

Resultado

- Queue03 recibirá 6Mbps
- Queue04 recibirá 2Mbps
- Queue05 recibirá 2Mbps

- Aclaración: HTB fue construido de tal forma que, cuando se satisfacen todos los limit-at, la cola principal no tiene más throughput para distribuir

Ejemplo 2 – caso típico con max-limit

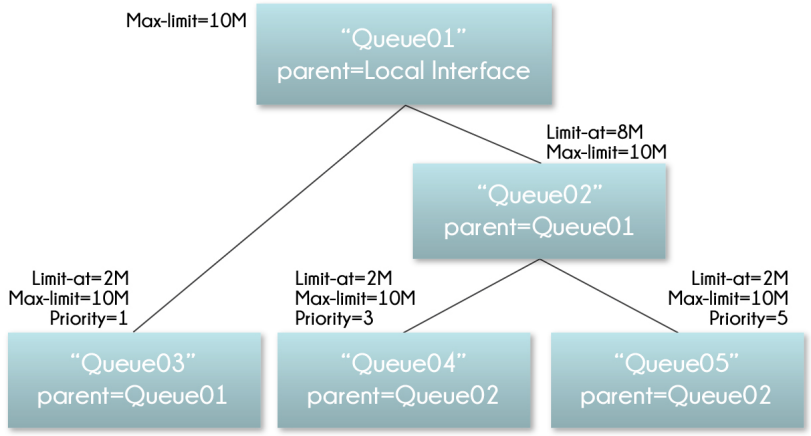



```
Queue01 limit-at=0Mbps max-limit=10Mbps
Queue02 limit-at=4Mbps max-limit=10Mbps
Queue03 limit-at=2Mbps max-limit=10Mbps priority=3
Queue04 limit-at=2Mbps max-limit=10Mbps priority=1
Queue05 limit-at=2Mbps max-limit=10Mbps priority=5
```

Resultado

- Queue03 recibirá 2Mbps
- Queue04 recibirá 6Mbps
- Queue05 recibirá 2Mbps
- Aclaración: Después de satisfacer todos los limit-at, el HTB entregará el throughput a la cola con más alta prioridad.

Ejemplo 3 – limit-at de la cola más interna



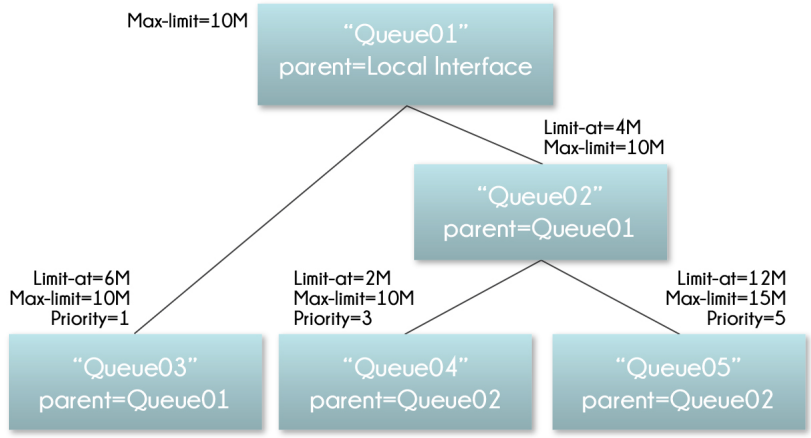

```
Queue01 limit-at=0Mbps max-limit=10Mbps
Queue02 limit-at=8Mbps max-limit=10Mbps
Queue03 limit-at=2Mbps max-limit=10Mbps priority=1
Queue04 limit-at=2Mbps max-limit=10Mbps priority=3
Queue05 limit-at=2Mbps max-limit=10Mbps priority=5
```

Resultado

- Queue03 recibirá 2Mbps
- Queue04 recibirá 6Mbps
- Queue05 recibirá 2Mbps

- Aclaración: Después de satisfacer todos los limit-at, la estructura HTB entregará el throughput a la cola con la prioridad más alta. En este caso la cola más interna Queue02 tiene especificado un limit-at, por lo que, tiene reservado 8Mbps de throughput para las colas Queue04 y Queue05. De estas dos colas, Queue04 tiene la prioridad más alta, por lo que obtiene el Throughput adicional.

Ejemplo 4 – limit-at de la cola hoja




```
Queue01 limit-at=0Mbps max-limit=10Mbps
Queue02 limit-at=4Mbps max-limit=10Mbps
Queue03 limit-at=6Mbps max-limit=10Mbps priority=1
Queue04 limit-at=2Mbps max-limit=10Mbps priority=3
Queue05 limit-at=12Mbps max-limit=15Mbps priority=5
```

Resultado

- Queue03 recibirá ~3Mbps
- Queue04 recibirá ~1Mbps
- Queue05 recibirá ~6Mbps
- Aclaración: Para satisfacer todos los limit-at, la estructura HTB está forzada a colocar 20Mbps (6Mbps para Queue03, 2Mbps para Queue04, 12Mbps para Queue05), pero la interface de salida solo puede manejar 10Mbps. Puesto que la interface de salida es usualmente FIFO, la asignación del throughput mantendrá el radio 6:2:12 o 3:1:6

Capítulo 10: Queues

Las colas se utilizan para limitar y priorizar el tráfico:

- Limitar la velocidad para ciertas direcciones IP, subredes, protocolos, puertos y otros parámetros
- Limitar el tráfico peer-to-peer
- Priorizar el flujo de algunos paquetes sobre otros
- Configurar el tráfico Burst para una navegación (browsing) más rápida
- Aplicar diferentes límites basados en tiempo
- Compartir el tráfico disponible de manera equitativa entre usuarios, o dependiendo de la carga del canal

La implementación de colas en RouterOS se basa en HTB (Hierarchical Token Bucket). HTB permite crear una estructura de cola jerárquica y determina las relaciones entre las colas.

En RouterOS, hasta la **v5.x**, estas estructuras jerárquicas se las puede encontrar en 4 lugares diferentes:

- **`global-in`** – Representa a todas las interfaces de entrada en general (INGRESS queue). Las colas unidas a `global-in` aplican al tráfico que es recibido por el router antes del filtrado de paquetes.
- **`global-out`** – Representa a todas las interfaces de salida en general (EGRESS queue).
- **`global-total`** – Representa a todas las interfaces juntas de entrada y de salida (en otras palabras, es la agregación de `global-in` y `global-out`). Se utiliza en los casos en que los clientes tienen un límite único para ambas, subida y bajada (upload y download).
- **`<interface name>`** – Representa una interface de salida en particular. Únicamente el tráfico que se designa para salir a través de esta interface pasará esta cola HTB.

Existen dos formas diferentes de configurar colas en RouterOS:

- **`/queue simple`** *(colas simples)* – Diseñado para facilitar la configuración de las tareas de gestión de colas simples y cotidianas (tales como la limitación de un solo cliente de upload/download, limitación de tráfico p2p, etc.).
- **`/queue tree`** *(árbol de colas)* – Para implementar tareas de encolamiento avanzadas (tales como políticas de priorización global, limitaciones de grupos de usuarios). Requiere el flujo de paquetes marcados en `/ip firewall mangle`.

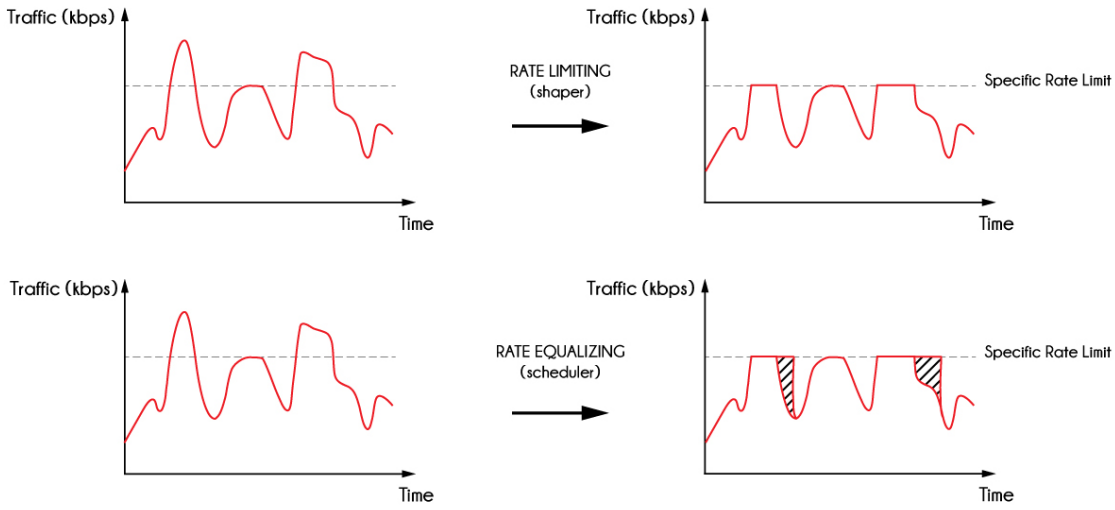
Principios de limitación de velocidad

La limitación de velocidad se utiliza para controlar la velocidad de flujo de tráfico enviado o recibido en una interfaz de red. El tráfico en el que la tasa es menor o igual a la tasa especificada se envía, mientras que el tráfico que supera la tasa se descarta o se retrasa.

La limitación de velocidad se puede realizar en dos formas:

- Descartar todos los paquetes que exceden el límite de velocidad – Limitación de velocidad (dropper o shaper) (Limitador de velocidad al 100% cuando el tamaño de la cola = 0)
- Retardo de los paquetes que exceden el límite de velocidad específico en la cola y transmiten cuando es posible – Igualar la velocidad (scheduler) (Igualación tasa al 100% cuando el tamaño de la cola=ilimitado)

La siguiente figura explica la diferencia entre la limitación de velocidad y la ecualización de tasa:



Como se puede ver en el primer caso todo el tráfico es superior a la tasa específica y es descartado. En el otro caso el tráfico excede tasa específica y se retrasa en la cola y se transmite más tarde, cuando sea posible, pero tenga en cuenta que el paquete sólo puede ser retrasado hasta que la cola no esté llena. Si no hay más espacio en el tope de la cola, los paquetes se descartan.

Para cada cola se pueden definir dos límites de tasa:

- **CIR** *(Committed Information Rate = Velocidad de Información Comprometida)* – (`limit-at` en RouterOS) En el peor escenario, el flujo recibirá esta cantidad de tasa de tráfico, independientemente de otros flujos de tráfico. En cualquier momento dado, el ancho de banda no debe caer por debajo de esta tasa comprometida.
- **MIR** *(Maximum Information Rate = Máxima Velocidad de Información)* – (`max-limit` en RouterOS) En el mejor escenario, velocidad máxima de datos disponible para el flujo, si está libre cualquier parte del ancho de banda.

Simple Queues (Colas Simples)

Sub-menu: `/queue simple`

La forma más sencilla de limitar la velocidad de datos para direcciones y / o subredes IP específicas, es usar colas simples.

También puede utilizar las colas simples para crear aplicaciones avanzadas de QoS. Tienen características integradas útiles:

- Colas de tráfico peer-to-peer
- Aplicación de reglas de colas en intervalos de tiempo elegidos
- Prioridades
- Uso de múltiples marcas de paquetes de `/firewall ip mangle`
- Shaping (programación) de tráfico bidireccional (un límite para el total de subida + bajada)

Un elemento de configuración en `/queue simple` puede crear de 0 a 3 colas separadas (una cola en global-in, una cola en global-out y una cola en global-total). Si todas las propiedades de una cola tienen valores por defecto (no hay límites establecidos, tipo de cola es por defecto), y la cola no tiene hijos, entonces no se crea realmente. De esta manera, por ejemplo, la creación de colas global-total se puede evitar si únicamente se utiliza limitación basada en upload/download.

Las colas simples tienen orden estricto - cada paquete debe pasar por cada cola hasta que cumpla las condiciones. (En el caso de 1000 colas, un paquete para alcanzar el final de cola tendrá que pasar a través de 999 colas antes de que llegue a su destino).

Ejemplo de configuración

Asumamos que tenemos topología como la de la red como en la Figura siguiente y queremos limitar la bajada y la subida para la red privada (`upload = 256kbps`, `download = 512kbps`).

Agregue una regla de cola simple, la cual limitará el tráfico de bajada a 512kbps y subida a 256kbps para la red `10.1.1.0/24`, configurada en la interface `Ether2`:

```
/queue simple add name=private target=10.1.1.0/24 max-limit=256K/512K interface=ether2
```

En este caso la declaración funciona bien incluso si indicamos solamente uno de los parámetros: `target=` o `interface=`, porque ambos definen dónde y para cuál tráfico se implementará esta cola.

Para chequear la configuración:

```
/queue simple print
Flags: X - disabled, I - invalid, D - dynamic
 0    name="private" target=10.1.1.0/24 dst-address=0.0.0.0/0
      interface=ether2 parent=none direction=both priority=8
      queue=default-small/default-small limit-at=0/0 max-limit=256k/512k
      burst-limit=0/0 burst-threshold=0/0 burst-time=0s/0s
      total-queue=default-small
```

El parámetro `max-limit` reduce el ancho de banda máximo disponible. El valor `max-limit=256k/512k` significa que los clientes de la red privada recibirán máximo `512kbps` para descarga y `256 kbps` para subida. El `target` permite definir la fuente de direcciones IP a la que se aplicará la regla de colas.

Si se desea excluir el servidor de que tenga límite, se debe agregar una cola para sin ninguna limitación (`max-limit = 0/0`, significa que no hay limitación). Se debe mover esta regla al principio de la lista, ya que los elementos de las cola simples se ejecutan en orden uno por uno. Si el router encuentra una regla que satisface ciertos paquetes, las siguientes reglas ya no se comparan:

```
/queue simple add name=server target=10.1.1.1/32 max-limit=0/0 interface=ether2
```

Identificadores de Flujo

- **`target`** *(multiple choice: IP address/netmask)* : Lista de rangos de direcciones IP que serán limitadas por esta cola.
- **`interface`** *(Name of the interface, or all)* : Identifica la interface en el que el objetivo (`target`) está conectado. Esta opción es útil cuando no es posible especificar las direcciones objetivo (`target`).

Nota Importante: A partir de **`RouterOS v6`** estos valores se combinan en la opción `target` donde se puede especificar cualquiera de los anteriores. `target` es para ser visto desde la perspectiva del objetivo. Si desea limitar la capacidad de carga de sus usuarios, debe configurar `target-upload`.

Cada una de estas dos propiedades se puede utilizar para determinar en cual dirección es subida (upload) y cual de bajada (download).

Debe tener cuidado al configurar ambas opciones para la misma cola. En caso de que se apuntan a direcciones opuestas la cola no funcionará.

Si no se especifica ni el valor del objetivo ni de la interface, la cola no será capaz de hacer diferencia entre carga y descarga, y limitará todo el tráfico en dos veces.

Otras Propiedades

- **name** *(Text)* : Identificador de cola único que puede ser usado valor de opción padre para otras colas
- **direction** *(One of both, upload, download, none; default: both)* : (solo hasta V5.x) Permite habilitar la limitación unidireccional de colas simples (deshabilita la otra dirección)
- **both** – Limita el tráfico de la descarga y la subida
- **upload** – Limita únicamente el tráfico hacia el objetivo (target)
- **download** - Limita únicamente el tráfico desde el objetivo (target)
- **time** *(TIME-TIME,sun,mon,tue,wed,thu,fri,sat - TIME es el tiempo local, todos los nombres de días son opcionales; default: not set)* : permitirá especificar el momento en que la cola determinada estará activa. El router debe tener las configuraciones de tiempo correctas.
- **dst-address** *(IP address/netmask)* : Permite seleccionar únicamente streams específicos (desde `target address` a este `destination address`) para limitar lo que es `target` y lo que es `dst`, y lo que es `upload` y lo que no es.
- **p2p** *(one of all-p2p, bit-torrent, blubster, direct-connect, edonkey, fasttrack, gnutella, soulseek, winmx; default: not set)* : (solo hasta V5.x) Permite seleccionar los paquetes NO encriptados de un particular p2p para limitar el ancho de banda
- **packet-marks** *(Comma separated list of packet mark names)* : Permite usar los paquetes marcados en `/ip firewall mangle`. Revisar el diagrama de flujo de paquete de RouterOS. Es necesario marcar los paquetes antes de las colas (antes de la cola `global-in` HTB) o la limitación de la descarga (download) del `target` no funcionará. El único `chain` de mangle antes de `global-in` es `prerouting`.

Nota Importante: En `RouterOS v6` se eliminan las opciones `direction` y `p2p`, pero se puede usar `mangle` para sustituirlas. Adicionalmente `dst-address` se une en la nueva opción `target`

Propiedades HTB

- **parent** *(Name of parent simple queue, or none)* : Asignan esta cola como una cola hijo para el `target` seleccionado {{{...}}}. La cola `target` puede ser la cola HTB o cualquier otra cola simple creada. Con la finalidad de que el tráfico alcance a las colas hijos, la cola padre debe capturar todo el tráfico necesario.
- **priority** *(1..8) :* Prioriza una cola hijo sobre otra cola. Este parámetro no trabaja en colas padres (si la cola tiene por lo menos un hijo). Uno (1) es la prioridad más alta, y ocho (8) es la prioridad más. La cola hijo con la prioridad más alta tendrá oportunidad de alcanzar su `max-limit` antes que una cola hijo con prioridad más. La prioridad no tiene nada que ver con los `bursts`.
- **queue** *(SOMETHING/SOMETHING)* : Elige el tipo de cola upload/download. Los tipos de colas pueden ser creados en `/queue type`.
- **limit-at** *(NUMBER/NUMBER)* : Representa la tasa de datos normal de upload/download que se garantiza al `target`
- **max-limit** *(NUMBER/NUMBER)* : Tasa de datos máxima de upload/download que es permitida que un `target` pueda alcanzar
- **burst-limit** *(NUMBER/NUMBER)* : Tasa de datos máxima de upload/download que se puede alcanzar mientras el `burst` está activo
- **burst-time** *(TIME/TIME)* : Período de tiempo, en segundos, sobre el cual se calcula la tasa de datos promedio de datos upload/download. (Este parámetro NO ES el tiempo actual del burst)
- **burst-threshold** *(NUMBER/NUMBER)* : Cuando la tasa de datos promedio está por debajo de este valor, se permite el burst, tan pronto como la tasa promedio de datos alcanza este valor el burst se niega. (básicamente este es un switch burst on/off). Para un comportamiento óptimo de burst, este valor debe estar sobre el valor del `limit-at` y bajo el valor de `max-limit`

A continuación las opciones correspondientes a la cola `global-total` HTB:

- **total-queue** *(SOMETHING/SOMETHING)*: Corresponde a la cola
- **total-limit-at** *(NUMBER/NUMBER)*: Corresponde al `limit-at`
- **total-max-limit** *(NUMBER/NUMBER)*: Corresponde al `max-limit`
- **total-burst-limit** *(NUMBER/NUMBER)*: Sorresponde al `burst-limit`
- **total-burst-time** *(TIME/TIME)*: Corresponde al `burst-time`
- **total-burst-threshold** *(NUMBER/NUMBER)*: Corresponde al `burst-threshold`

Las buenas prácticas sugieren que:

- La suma de los valores de los `limit-at` de los hijos debe ser menor o igual que el `max-limit` del padre.
- El `max-limit` de cada hijo debe ser menor que el `max-limit` del padre. De esta manera se dejará algo de tráfico para las otras colas hijos, y será capaz de conseguir tráfico sin luchar por ella con otras colas hijos.

Estadísticas

- **rate** *(read-only/read-only)* : Promedio de la tasa de datos de la cola en bytes por segundo
- **packet-rate** *(read-only/read-only)* : Promedio de la tasa de datos de la cola en paquetes por segundo
- **bytes** *(read-only/read-only)* : Número de bytes procesados por esta cola

- **packets** *(read-only/read-only)* : Número de paquetes procesados por esta cola
- **queued-bytes** *(read-only/read-only)* : Número de bytes esperando en la cola
- **queued-packets** *(read-only/read-only)* : Número de paquetes esperando en la cola
- **dropped** *(read-only/read-only)* : Número de paquetes perdidos
- **borrows** *(read-only/read-only)* : Paquetes que sobrepasaron su valor limit-at (y que estaban sin usar y se quitaron de otras colas)
- **lends** *(read-only/read-only)* : Paquetes que pasaron el valor bajo el limit-at, o si la cola es un padre, la suma de todos los paquetes prestados al hijo
- **pcq-queues** *(read-only/read-only)* : Número de substreams PCQ, si el tipo de cola (`queue type`) es `PCQ`

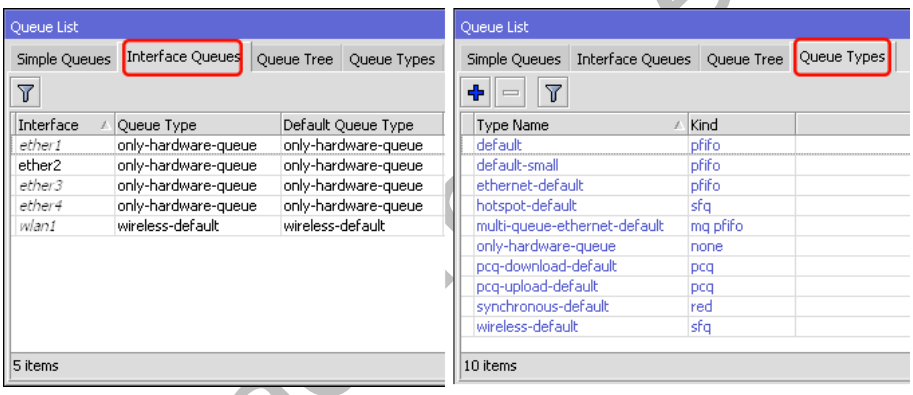
A continuación, las opciones correspondientes a la cola `global-total` HTB:

- **total-rate** *(read-only)*: Corresponde a la velocidad (`rate`)
- **total-packet**-rate *(read-only)*: Corresponde al `packet-rate`
- **total-bytes** *(read-only)*: Corresponde a `bytes`
- **total-packets** *(read-only)*: Corresponde a `packets`
- **total-queued-bytes** *(read-only)*: Corresponde a `queued-bytes`
- **total-queued-packets** *(read-only)*: Corresponde a `queued-packets`
- **total-dropped** *(read-only)*: Corresponde a `dropped`
- **total-lends** *(read-only)*: Corresponde a `lends`
- **total-borrows** *(read-only)*: Corresponde a `borrows`
- **total-pcq-queues** *(read-only)*: Corresponde a `pcq-queues`

Tipos de Colas (Queue Types)

Sub-menu: `/queue type`

En este sub-menú se listan los tipos de colas creados por default y permite añadir nuevos tipos creados por el usuario



Por default RouterOS crea los siguientes tipos de colas pre-definidos:

```
/queue type print
0 name="default" kind=pfifo pfifo-limit=50
1 name="ethernet-default" kind=pfifo pfifo-limit=50
2 name="wireless-default" kind=sfq sfq-perturb=5 sfq-allot=1514
3 name="synchronous-default" kind=red red-limit=60 red-min-threshold=10 red-max-threshold=50 red-
burst=20
   red-avg-packet=1000
4 name="hotspot-default" kind=sfq sfq-perturb=5 sfq-allot=1514
5 name="only-hardware-queue" kind=none
6 name="multi-queue-ethernet-default" kind=mq-pfifo mq-pfifo-limit=50
7 name="default-small" kind=pfifo pfifo-limit=10
```

Nota importante: A partir de la versión v5.8 existe una nueva cola por default: `only-hardware-queue`. Todos los RouterBOARDs tendrán este nuevo tipo de encolamiento configurado como la cola de interface por default.

only-hardware-queue deja la interface únicamente con un buffer de anillo descriptor transmisor de hardware que actúa como una cola por si mismo. Usualmente al menos 100 paquetes pueden ser encolados para transmitir en el "`transmit descriptor ring buffer`". El tamaño del "`transmit descriptor ring buffer`" y la cantidad de paquetes que pueden ser encolados en él, varía para diferentes tipos de MACs ethernet.

Al no tener cola de software es especialmente beneficioso en sistemas SMP, ya que elimina el requisito para sincronizar el acceso a la misma desde diferentes CPUs/núcleos que son costosos.

multi-queue-ethernet-default puede ser beneficioso en sistemas SMP con interfaces Ethernet que tienen soporte para múltiples colas de transmisión y tienen una compatibilidad de controladores Linux para múltiples colas de transmisión. Al tener la cola de un software para cada cola de hardware podría haber menos tiempo gastado para sincronizar el acceso a ellos.

Nota Importante: Al tener la posibilidad de configurar `only-hardware-queue` se requiere requiere apoyo en el driver de Ethernet por lo que sólo está disponible para algunas interfaces de Ethernet en su mayoría que aparecen en RBs.

Nota Importante: La mejora `only-hardware-queue` y `multi-queue-ethernet-default` está presente únicamente cuando no hay una entrada `/queue tree` con una interface particular como padre.

Tipos de colas

Los tipos de cola o encolamiento (scheduling) describen cuál paquete será transmitido en la siguiente línea. RouterOS soporta varios algoritmos de encolamiento:

- BFIFO, PFIFO, MQ PFIFO
- RED
- SFQ
- PCQ

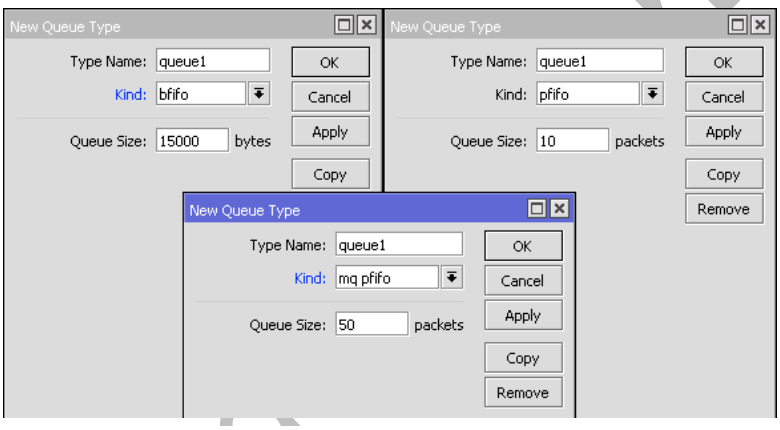
PFIFO, BFIFO y MQ PFIFO

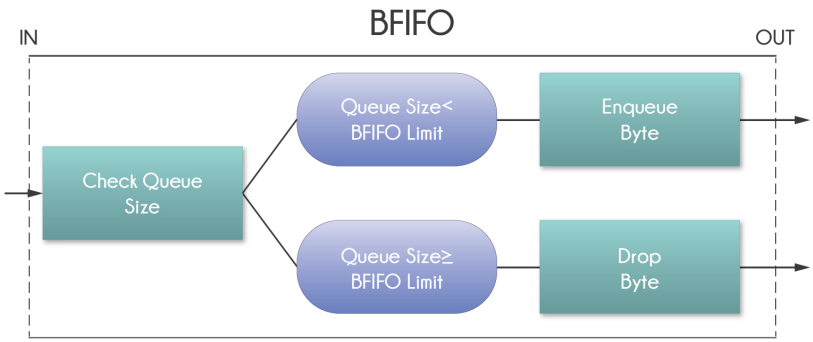
Estas disciplinas de encolamiento están basada el algoritmo FIFO (First-In First-Out, Primero que entra primero que sale).La diferencia PFIFO y BFIFO es que el primero está medido en paquetes y el segundo en bytes.

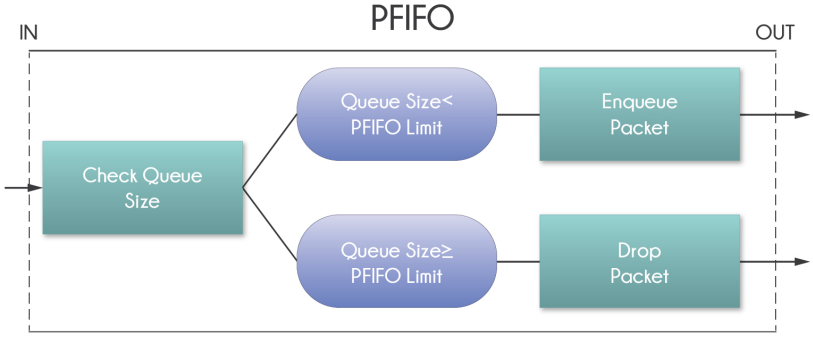
Cada paquete que no puede ser encolado (si es que la cola está llena), es descartado. Los tamaños de colas grandes pueden incrementar la latencia, pero utilizan mejor el canal.

Estas colas usan los parámetros `pfifo-limit` y `bfifo-limit`.

`mq-pfifo` es PFIFO con soporte para múltiples colas de transmisión. Esta cola es beneficiosa en sistemas SMP con interfaces Ethernet que tienen soporte para múltiples colas de transmisión y tienen una compatibilidad de controladores Linux para múltiples colas de transmisión. `mq-pfifo` usa el parámetro `mq-pfifo-limit`.







RED

Random Early Drop es un mecanismo de encolamiento que trata de evitar la congestión de la red controlando el tamaño promedio de la cola.

El tamaño promedio de la cola se compara con dos umbrales: un umbral mínimo (`minth`) y un umbral máximo (`maxth`). Si el tamaño promedio de la cola (`avgq`) es menor que el umbral mínimo, ningún paquete es descartado. Cuando el tamaño promedio de la cola es mayor que el umbral máximo, todos los paquetes entrantes se descartan. Pero si el tamaño promedio de la cola está entre los umbrales mínimos y máximos los paquetes se descartaron al azar con probabilidad **Pd**, donde la probabilidad es exacta en función del tamaño promedio de la cola:

```
Pd = Pmax(avgq - minth) / (maxth - minth)
```

Si la cola promedio crece, la probabilidad de abandonar los paquetes entrantes también crece.

`Pmax` es el radio, que puede ajustar la probabilidad de descarte de paquetes abruptamente, (en el caso más simple `Pmax` puede ser igual a uno. El siguiente diagrama muestra la probabilidad de descarte de paquetes en el algoritmo RED.

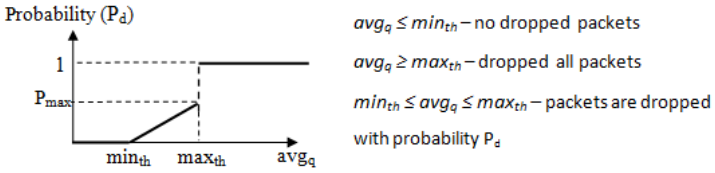
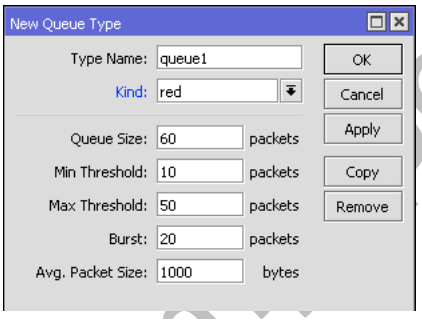
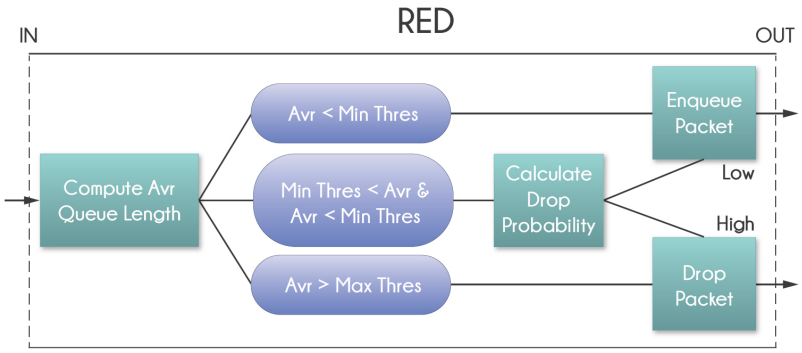


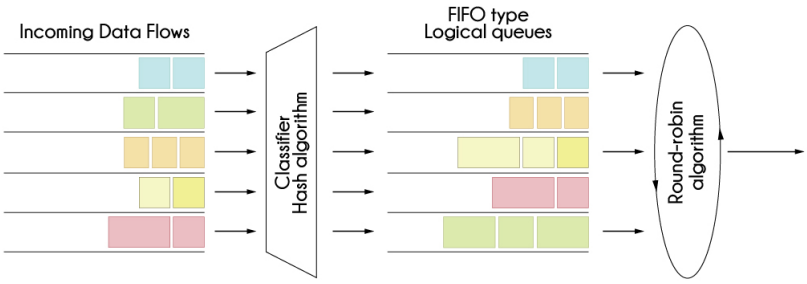
Figure 8.2. RED operation





SFQ

Stochastic Fairness Queuing (SFQ) está asegurada por algoritmos de `hashing` y `round-robin`. Un flujo de tráfico puede ser identificado por 4 opciones (`src-address`, `dst-address`, `src-port` y `dst-port`), por lo que estos paquetes son usados por el algoritmo de `hashing` SFQ para clasificar los paquetes en uno de los 1024 posibles sub-streams.



El algoritmo de round-robin empezará a distribuir el ancho de banda disponible a todos los sub-streams, en cada ronda se entrega `sfq-allot` bytes de tráfico. La cola completa SFQ puede contener 128 paquetes y hay 1024 sub-streams disponibles.

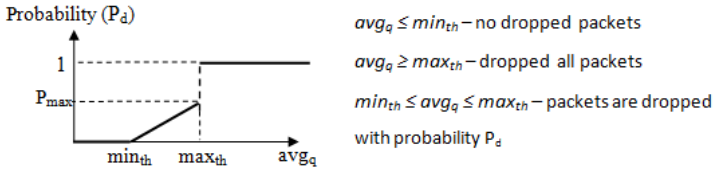
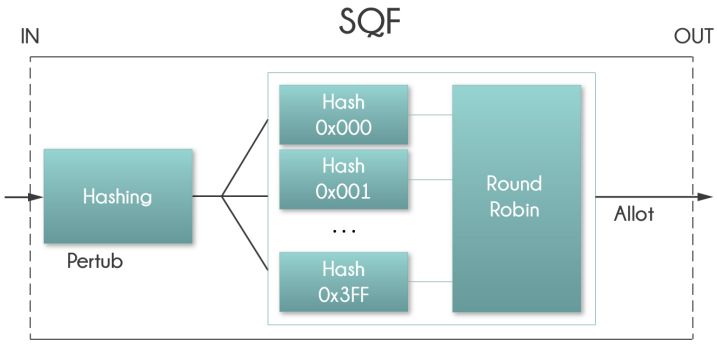
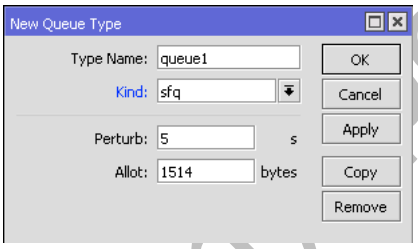


Figure 8.2. RED operation



SFQ es conocido como Estocástico (Stochastic) porque en realidad no se asigna una cola para cada flujo, tiene un algoritmo que divide el tráfico sobre un número limitado de colas (1024) utilizando un algoritmo de hashing.

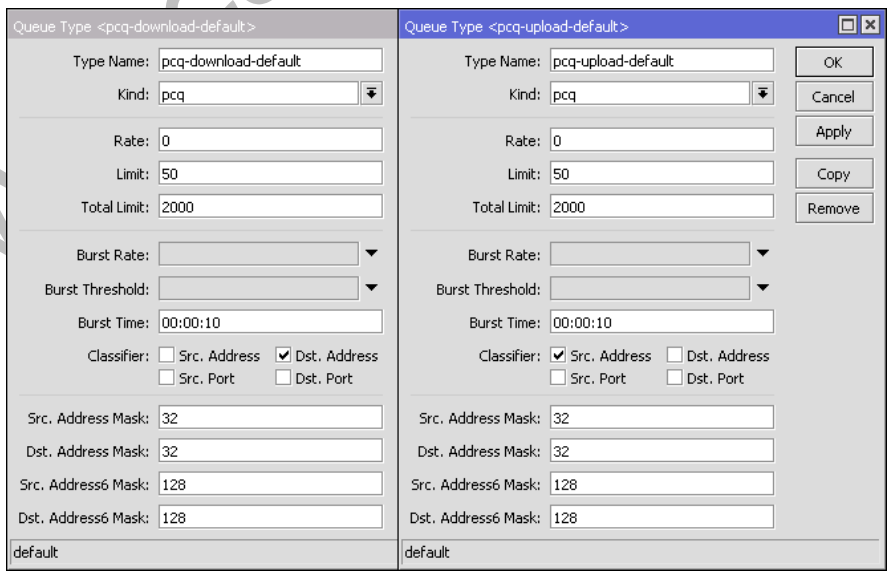


PCQ

Per Connection Queuing (PCQ) es similar a SFQ, pero tiene características adicionales.

Es posible elegir identificadores de flujo (desde `dst-address` | `dst-port` | `src-address` | `src-port`). Por ejemplo, si se clasifica el flujo por `src-address` en la interface local (interface con los clientes), cada sub-stream PCQ será una subida de un cliente particular.

Se puede asignar una limitación de velocidad a los sub-streams con la opción `pcq-rate`. Si pcq-rate=0 los sub-streams dividirán equitativamente el tráfico disponible.



Propiedades

Las propiedades que inician con un nombre de tipo de encolamiento particular, se aplican únicamente a un tipo particular. Por ejemplo, todas las propiedades que inician con PCQ, aplican únicamente al tipo de cola PCQ.

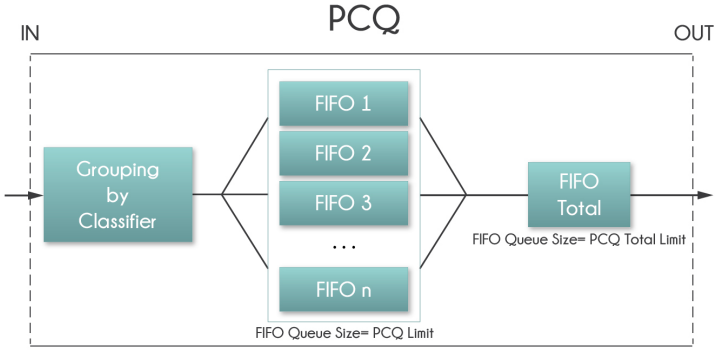
- **bfifo-limit** *(integer [1000..4294967295]; Default: 15000)* – Máximo número de bytes que la cola FIFO puede mantener. Se aplica si el tipo es BFIFO.
- **kind** *(bfifo | mq-pfifo | none | pcq | pfifo | red | sfq; Default:)* – Clase especial de tipo de cola.
- **mq-pfifo-limit** *(integer [1..4294967295]; Default: 50)* – Límite `multi-queue` PFIFO.
- **name** *(string; Default:)* – Nombre descriptivo del tipo de cola
- **pcq-burst-rate** *(integer [0..4294967295]; Default: 0)* – Máxima tasa de datos de upload/download que puede ser alcanzada mientras el burst para el substream es permitido
- **pcq-burst-threshold** *(integer [0..4294967295]; Default: 0)* – Este es el valor del switch burst on/off
- **pcq-burst-time** *(time; Default: 10s)* – Período de tiempo, en segundos, sobre el cual se calcula la tasa de datos promedio. (Este NO es el tiempo real del burst)
- **pcq-classifier** *(lista de src-address|dst-address|src-port|dst-port; Default: "")* – Selección de identificadores de sub-stream
- **pcq-dst-address-mask** *(integer [0..32] | IPNetmask; Default: 32)* – Tamaño de la red IPv4 que será usada como un identificador `dst-address sub-stream`
- **pcq-dst-address6-mask** *(integer [0..128]; Default: 128)* - Tamaño de la red IPv6 que será usada como un identificador `dst-address sub-stream`
- **pcq-limit** *(integer [1..4294967295]; Default: 50)* – Tamaño de cola de un simple sub-stream (en kilobytes)
- **pcq-rate** *(integer [0..4294967295]; Default: 0)* – Máxima tasa de datos disponible de cada substream
- **pcq-src-address-mask** *(integer [0..32] | IPNetmask; Default: 32)* - Tamaño de la red IPv4 que será usada como un identificador `src-address sub-stream`
- **pcq-src-address6-mask** *(integer [0..128]; Default: 128)* - Tamaño de la red IPv6 que será usada como un identificador `src-address sub-stream`
- **pcq-total-limit** *(integer [1..4294967295]; Default: 2000)* – Tamaño total de la cola de todos los sub-streams (en kilobytes)
- **pfifo-limit** *(integer [1..4294967295]; Default: 50)* – Máximo número de paquetes que la cola PFIFO puede mantener. Aplica si el tipo es PFIFO.
- **red-avg-packet** *(integer [1..65535]; Default: 1000)* – Usado por RED para el cálculo del tamaño promedio de la cola (para la traducción de packet a byte)
- **red-burst** *(integer [0..4294967295]; Default: 20)* – Número de paquetes permitidos para bursts de paquetes cuando no hay paquetes en la cola
- **red-limit** *(integer [0..4294967295]; Default: 60)* – Límite de cola RED en paquetes
- **red-max-threshold** *(integer [0..4294967295]; Default: 50)* – El tamaño de cola promedio en el cual la probabilidad de marcar el paquete es la más alta.
- **red-min-threshold** *(integer [0..4294967295]; Default: 10)* – Tamaño promedio de la cola en bytes.
- **sfq-allot** *(integer [0..32767]; Default: 1514)* – Cantidad de datos en bytes que pueden ser enviados en una ronda round-robin
- **sfq-perturb** *(integer [0..4294967295]; Default: 5)* – Cuan frecuente debe ser refrescada la función hash

PCQ se introdujo para optimizar los sistemas de calidad de servicio masivos, donde la mayoría de las colas son exactamente las mismas para los diferentes sub-streams. Por ejemplo, un sub-stream puede ser de descarga (download) o de subida (upload) para un cliente en particular (IP) o para una conexión con un servidor.

El algoritmo PCQ es muy simple, en un primer momento se utiliza clasificadores seleccionados para distinguir un sub-stream de otro, entonces se aplica un tamaño de la cola FIFO individual y limitación en todos los sub-streams, luego se agrupa todos los sub-streams y se aplica una limitación y tamaño de cola global.

Parámetros de PCQ:

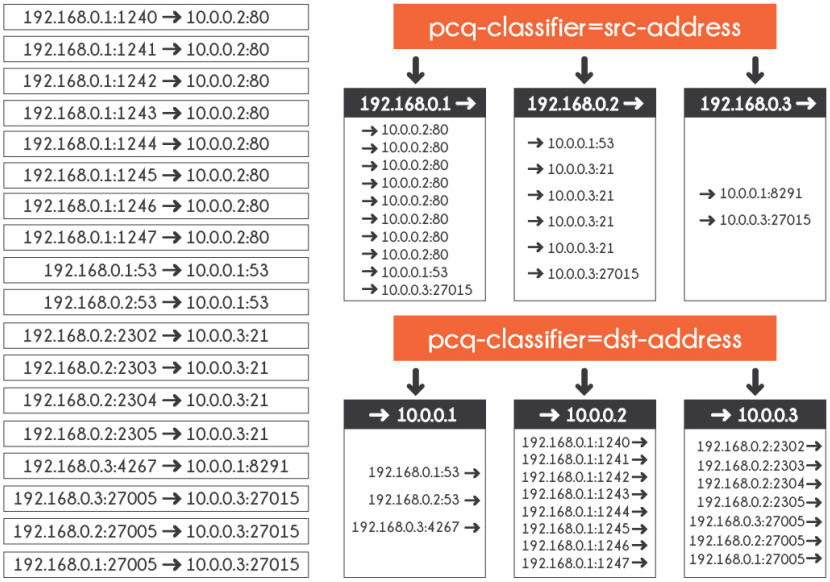
- **pcq-classifier** *(dst-address | dst-port | src-address | src-port; default: "")* – Selección de identificadores de sub-stream
- **pcq-rate** *(number)* – Máxima tasa de datos disponible de cada substream
- **pcq-limit** *(number)* – Tamaño de cola de un simple sub-stream (en KB)
- **pcq-total-limit** *(number)* – Cantidad máxima de data encolada en todos los sub-streams (en KB)

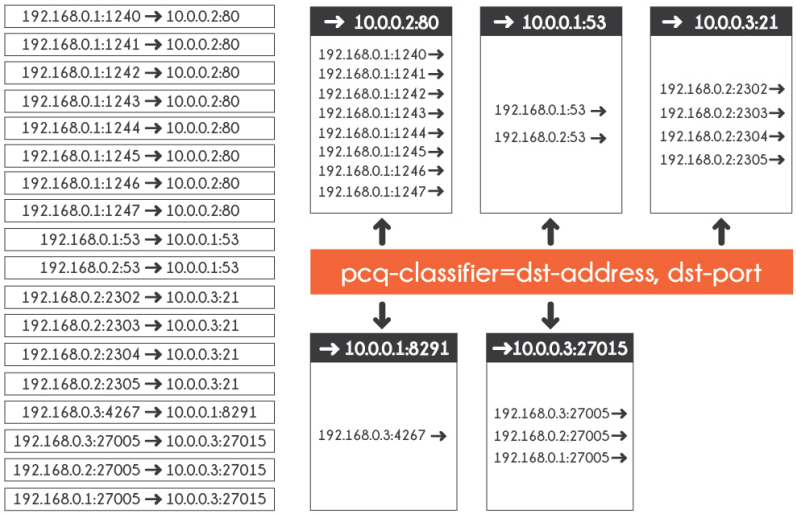


Por lo tanto, en vez de tener 100 colas con limitación de 1,000 Kbps para descarga (download), podemos tener una sola cola PCQ con 100 sub-streams.

Ejemplos de clasificación

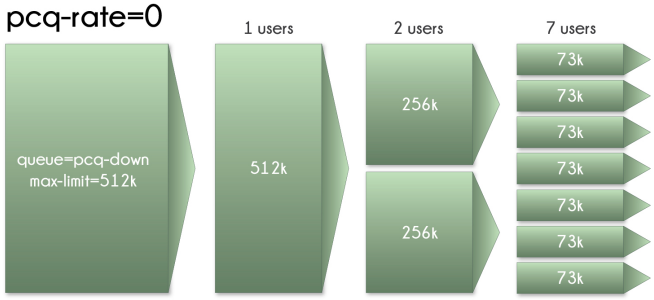
Para entender mejor la clasificación tomaremos una lista de 18 streams de paquetes con una dirección y puerto específico, hacia una dirección y puerto específico. Entonces vamos a elegir un clasificador y dividir los 18 streams de paquetes en sub-streams PCQ

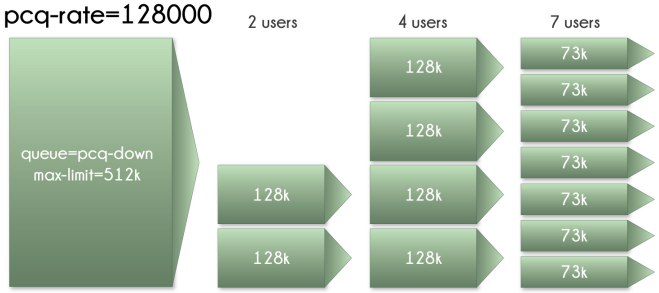




Ejemplos de PCQ Rate

Aquí es posible ver qué sucede si se especifica o no el PCQ-rate. Es importante notar que si ambos límites (pcq-rate y max-limit) no se especifican, el comportamiento de la cola puede ser impreciso. Por lo tanto, se sugiere fuertemente tener por lo menos una de estas opciones configuradas.





Nueva implementación de PCQ (a partir de la v5.0RC5)

PCQ fue reescrito en la v5.0RC4 para optimizarlo con alto rendimiento tanto en Mbps y PPS (paquetes por segundo). Esta implementación utiliza adecuadamente todas las nuevas características de Linux Kernel, esto hace que PCQ sea más rápido y que demande menos recursos.

Tan pronto como se activa un nuevo stream se obtendrá ¼ de la tasa con más alta prioridad. Si el rate=0 el sub-stream no tendrá esta característica (ya que ¼ de 0 es 0)

Es necesario conocer esto por una buena razón: Asumamos que la tasa del sub-stream es 10Mbps, por lo que en el momento en que el nuevo sub-stream solicite tráfico obtendrá primero 2500k de tráfico sin limitación. Esto puede resultar en una mayor velocidad que lo que se espera como resultado en programas como Speedtest.net. Para evitar asegurarse que Speedtest.net no es el primer programa que utiliza el ancho de banda que se ejecuta en el PC.

También a partir de la v5.0RC5, PCQ tiene las siguientes características:

- PCQ Burst para sub-streams. PCQ tiene una implementación de burst idéntica a Simple Queues y Queue Tree

Parámetros PCQ:

- **pcq-burst-rate** *(number)* – Máxima tasa de datos de upload/download que se alcanzará mientras es burst para el sub-stream es permitido
- **pcq-burst-threshold** *(number)* – Este es el valor de burst en el switch on/off
- **pcq-burst-time** *(time)* – Período de tiempo, en segundos, sobre el cual se calcula la tasa de datos promedio. (Este NO es el tiempo actual de burst)

PCQ también permite utilizar diferentes tamaños de redes IPv4 e IPv6 como identificadores de sub-stream. Antes estaba cerrada a una dirección IP única. Esto se hace principalmente para IPv6 como clientes desde el punto de vista ISP estarán representados por red /64, pero los dispositivos en la red de los clientes serán /128. PCQ se puede utilizar para ambos de estos escenarios y mucho más.

Parámetros PCQ:

- **pcq-dst-address-mask** *(number)* – Tamaño de la red IPv4 que se usará como identificador del sub-stream `dst-address`
- **pcq-src-address-mask** *(number)* – Tamaño de la red IPv4 que se usará como identificador del sub-stream `src-address`
- **pcq-dst-address6-mask** *(number)* – Tamaño de la red IPv6 que se usará como identificador del sub-stream `dst-address`
- **pcq-src-address6-mask** *(number)* – Tamaño de la red IPv6 que se usará como identificador del sub-stream `src-address`

Interface Queue

Sub-menu: `/queue interface`

Antes de enviar datos a través de una interfaz, ésta es procesada por la cola. Este submenú lista todas las interfaces disponibles en RouterOS y permite cambiar el tipo de cola para determinada interfaz.

Nota Importante: No se puede agregar nuevas interfaces a este menú. La lista es generada automáticamente.

Propiedades

- **interface** *(string)* – Nombre de la interface a la cual se aplica la cola. Es un parámetro de solo-lectura.
- **queue** *(string; Default:)* – Tipo de cola asignado a una interface en particular.

Capítulo 11: Burst

Burst es una característica que permite satisfacer los requerimientos de cola para ancho de banda adicional, incluso si la tasa requerida es más grande que el MIR (`max-limit`) durante un periodo de tiempo limitado

El `burst` puede ocurrir sólo si el consumo promedio (`average-rate`) de la cola, de los últimos segundos del `burst-time` es más pequeño que el `burst-threshold`. El burst se detendrá si el consumo promedio (`average-rate`) de la cola de los últimos segundos del `burst-time` es más grande o igual al `burst-threshold`

El mecanismo del `burst` es simple: si el burst es permitido, el valor del `max-limit` se reemplaza por el valor del `burst-limit`. Cuando el `burst` está deshabilitado el valor del `max-limit` permanece sin cambio.

- **`burst-limit`** *(NUMBER)* – Máxima tasa de datos upload/download que se puede alcanzar mientras el burst es permitido
- **`burst-time`** *(TIME)* – Período de tiempo, en segundos, sobre el cual se calcula la tasa promedio (Este NO es el tiempo de burst actual)
- **`burst-threshold`** *(NUMBER)* – Este es el valor del switch on/off del burst
- **`average-rate`** *(read-only)* – Cada 1/n partes del burst-time, el router calcula la tasa de datos promedio de cada clase sobre los últimos segundos `burst-time`
- **`actual-rate`** *(read-only)* – Tasa de transferencia de tráfico actual de la cola

El burst es una de las mejores formas de incrementar el desempeño HTTP

Los Bursts son usados para permitir altas tasas de datos por un período corto de tiempo

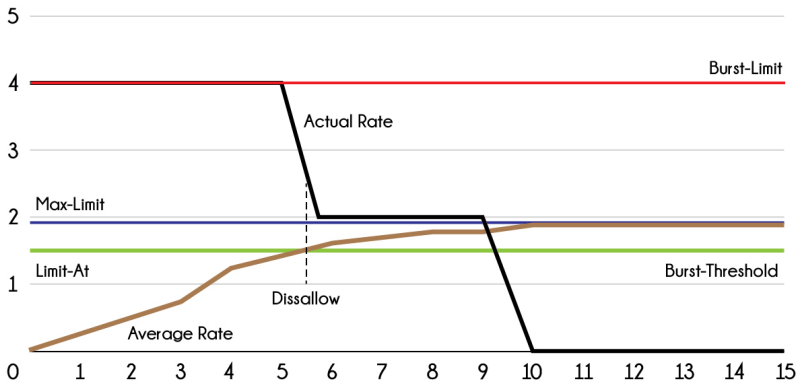
- Si una tasa de datos promedio es menor que el burst-threshold, el burst puede ser usado (la tasa de datos actual puede alcanzar el `burst-limit`)
- La tasa de datos promedio se calcula de los últimos segundos de `burst-time`

La tasa de datos promedio (average data rate) se calcula de la siguiente forma:

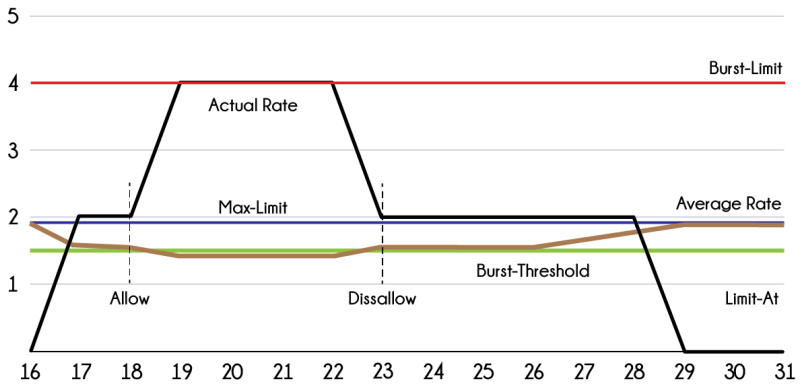
El `burst-time` se divide en N períodos

- El router calcula la tasa de datos promedio de cada clase sobre esos pequeños períodos
- Note que el actual burst-period no es igual al burst-time. Puede ser varias veces más corto que el burst-time dependiendo del histórico de los parámetros max limit, burst-limit, burst-threshold, y actual data rate (ver el gráfico ejemplo en el siguiente slide)
- `longest-burst-time = burst-threshold * burst-time / burst-limit`

Burst (primeros 16 segundos)



Burst (siguientes 16 segundos)



Capítulo 12: Web proxy

MikroTik RouterOS ejecuta proxy para requerimientos `HTTP` y `HTTP-proxy` (para protocolos `FTP` y `HTTP`).

El web-proxy de RouterOS tiene 3 características principales

- Caching de tráfico HTTP y FTP
- Filtro de nombres DNS
- Redireccionamiento de DNS

El servidor proxy ejecuta funciones de caché de objetos de internet almacenando dichos objetos, por ejemplo: datos disponibles vía protocolos `HTTP` y `FTP` en un sistema posicionado lo más cercano al recipiente con la finalidad de acelerar la navegación del cliente entregándole copias de los archivos requeridos desde el caché del proxy y a la velocidad de la red local.

MikroTik RouterOS implementa las siguientes características de proxy server:

- **Proxy HTTP regular.-** El cliente, por si mismo, debe especificar cuál es su proxy server
- **Proxy transparente.-** El cliente no sabe que un proxy está habilitado y no existe ninguna necesidad de configuración adicional en el browser del cliente
- **Lista de acceso por origen, destino, URL y método solicitado (firewall HTTP)**
- **Lista de acceso de caché.-** Para especificar qué objetos se deben almacenar en caché y cuales no
- **Lista de acceso directa.-** Para especificar cuáles recursos deben ser accesados directamente, y cuáles a través de otro proxy server (proxy padre)
- **Facilidad de registro (log).-** Permite obtener y almacenar información sobre el funcionamiento de proxy
- **Soporte de proxy padre.-** Permite especificar otro servidor proxy (si no se tiene el objeto solicitado se debe pedir a sus padres, o al servidor original)

El servidor proxy usualmente se coloca en varios puntos entre los usuarios y el servidor destino (conocido como el servidor origen) en internet.

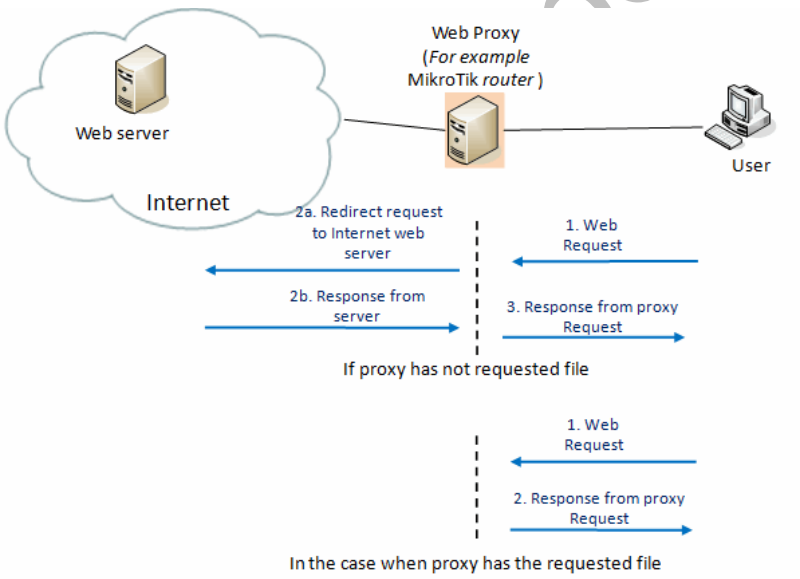


Figure 10.1. Web proxy basic operation scheme

Un Web-Proxy-Caché observa los requerimientos que provienen del cliente, y almacena copia de las respuestas para si mismo. Luego, si existe otro requerimiento para el mismo URL, puede usar la respuesta que ya tiene almacenada, en lugar de preguntar nuevamente al servidor de origen. Si el Proxy no posee el archivo solicitado, lo descarga del servidor original

Puede haber muchos efectos potenciales del servidor proxy:

- Decrementar la velocidad de acceso a los recursos (le toma menos tiempo al cliente conseguir el objeto)
- Trabaja como firewall HTTP (denegando acceso a página no deseables)
- Permitir filtrar contenido web (especificando parámetros como direcciones origen, direcciones y puerto destino, URL, métodos de requerimientos HTTP) y escanear el contenido de salida, por ejemplo, para la protección de fuga de datos

Nota: Puede ser útil mantener corriendo el Web Proxy (incluso sin caché) cuando se desea usarlo como firewall HTTP y FTP (por ejemplo, para denegar acceso a páginas no deseadas o para tipos específicos de archivos por ejemplo archivos *.mp3) o para re direccionar solicitudes transparentemente a un Proxy Externo (por ejemplo, a un proxy con funciones de caché)

Ejemplo de configuración de Proxy

En MikroTik RouterOS la configuración se realiza en el menú /ip proxy. A continuación, un ejemplo donde se habilita el proxy en el puerto 8080 y se configura la IP 195.10.10.1 como la dirección origen

```
/ip proxy set enabled=yes port=8080 src-address=195.10.10.1

/ip proxy print
                    enabled: yes
                src-address: 195.10.10.1
                       port: 8080
              parent-proxy: 0.0.0.0:0
               cache-drive: system
        cache-administrator: "admin@mikrotik.com"
        max-disk-cache-size: none
         max-ram-cache-size: 100000KiB
         cache-only-on-disk: yes
maximal-client-connections: 1000
maximal-server-connections: 1000
           max-fresh-time: 3d
```

Cuando se configura un servicio de proxy regular, se debe asegurar de que sirve únicamente a sus clientes y se debe prevenir el acceso no autorizado creando reglas de firewall que permitan que únicamente sus clientes puedan usar el proxy. Sino, puede ser usado como un Proxy abierto.

Debe recordar que cuando se usa como Proxy Regular, se requiere también la configuración del browser del cliente

Ejemplo de configuración de Proxy Transparente

RouterOS también puede actuar como un servidor de Caché Transparente, sin requerir configuración en el browser del cliente. El Proxy Transparente no modifica la URL solicitada ni tampoco la respuesta. RouterOS tomará todas las solicitudes HTTP y las re direccionará al servicio de proxy local. Este proceso será enteramente transparente para el usuario (los usuarios pueden no saber nada acerca de servidor proxy que se encuentra entre ellos y servidor original), y la única diferencia para ellos será el incremento en la velocidad de navegación.

Para habilitar el modo transparente, se debe agregar una regla de firewall en Destination NAT, especificando cuáles conexiones (a que puertos) debe ser re direccionado transparentemente al proxy. Se debe chequear las configuraciones y re direccionar los usuarios al Proxy Server

```
/ip firewall nat add chain=dstnat protocol=tcp src-address=192.168.1.0/24 \
dst-port=80 action=redirect to-ports=8080

/ip firewall nat print
 Flags: X - disabled, I - invalid, D - dynamic
  0   chain=dstnat protocol=tcp dst-port=80 action=redirect to-ports=8000
```

El Web Proxy puede ser usado como un Web Proxy Normal y como un Web Proxy Transparente al mismo tiempo. En modo transparente se puede usar como un Web Proxy estándar también. Sin embargo, en este caso, los usuarios del Proxy podrían tener problemas para alcanzar las páginas web que son accesadas transparentemente.

Firewall basado en Proxy

La lista de acceso (Access List) se implementa en la misma forma en que se procesan las reglas de firewall en MikroTik, es decir desde la parte superior a la inferior, en orden secuencial. La primera regla de coincidencia especifica la decisión de qué hacer con esta conexión.

Las conexiones pueden coincidir por su dirección de origen, dirección de destino, puerto de destino, sub-cadena de dirección URL solicitada (Uniform Resource Locator) o método de solicitud. Si no se especifica ninguno de estos parámetros, cada conexión coincidirá con esta regla.

Si la conexión coincide con una regla, la propiedad de acción de esta regla especifica si la conexión se permitirá o no (negar). Si la conexión no coincide con ninguna regla, entonces se permitirá.

En este ejemplo suponemos que hemos configurado el servidor proxy transparente que figura en el ejemplo anterior

Bloquear Websites específicos

```
/ip proxy access add dst-host=www.facebook.com action=deny
```

Esta regla bloqueará el Website `http://www.facebook.com`, siempre podemos bloquea lo mismo para diferentes redes usando `src-address`.

```
/ip proxy access add src-address=192.168.1.0/24 dst-host=www.facebook.com action=deny
```

Los usuarios de la red `192.168.1.0/24` no podrán acceder al Website `www.facebook.com`.

También se puede bloquear Websites que contienen palabras específicas en el URL:

```
/ip proxy access add dst-host=:mail action=deny
```

Esta declaración bloqueará todos los sitios web que contienen la palabra "`mail`" en el URL. Por ejemplo `www.mail.com`, `www.hotmail.com`, `mail.yahoo.com`, etc.

También se puede detener la descarga de tipos específicos de archivos, por ejemplo: .flv, .avi, .mp4, .mp3, .exe, .dat, …etc.

```
/ip proxy access
 add path=*.flv action=deny
 add path=*.avi action=deny
 add path=*.mp4 action=deny
 add path=*.mp3 action=deny
 add path=*.zip action=deny
 add path=*.rar action=deny.
```

En este ejemplo también están disponibles diferentes caracteres comodines (wildcard), para crear condiciones específicas y para que coincida con la lista de acceso de proxy

Las propiedades comodín (dst-host y dst-path) coinciden con una cadena completa (por ejemplo, no coincidirán "example.com" si está configurada como "example"). Los comodines disponibles son:

- * – Coincide con cualquier número de caracteres
- ? – Coincide con cualquier carácter

Las expresiones regulares también son aceptadas aquí, pero si la propiedad debe ser tratada como una expresión regular, se debe comenzar con dos puntos (':').

Para demostrar que ningún símbolo se permite antes del patrón dado, usamos el símbolo ^ al comienzo del patrón.

Para demostrar que ningún símbolo se permite después del patrón dado, usamos el símbolo $ al final del patrón.

Habilitación de RAM o Caché basado en almacenamiento

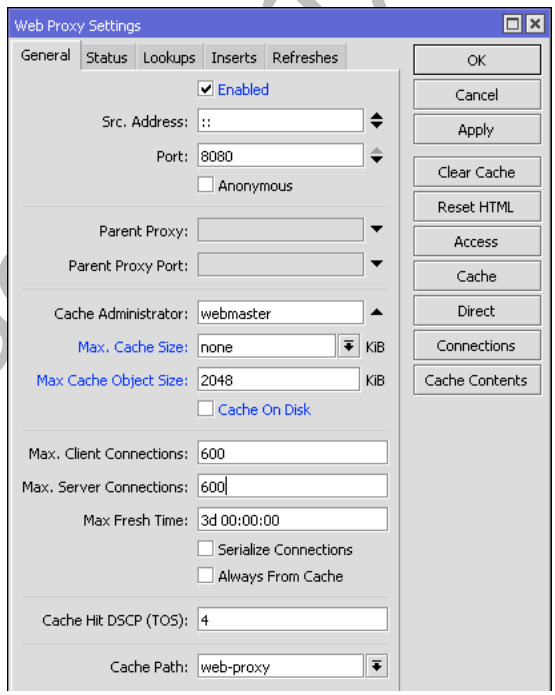
En el siguiente ejemplo se asumirá que ya se tiene un proxy configurado y que únicamente se habilitará el caché.

Caché basado en RAM

- Esta opción es efectiva si se dispone de un dispositivo con una considerable cantidad de RAM disponible para caché.
- Si se habilita esto en un dispositivo con 256MB de RAM o menos, simplemente no le dará ningún beneficio a la red
- Es una forma mucho más rápida de lectura/escritura de caché, que cuando está almacenada en medios conectados USB o SATA

Caché basado en almacenamiento

- Están disponibles caché de proxy más grandes debido las diferentes capacidades del medio



No hace caché

- max-cache-size = none

Caché en RAM

- max-cache-size ≠ none
- cache-on-disk = no

Caché en HDD

- max-cache-size ≠ none

- cache-on-disk = yes

Caché de proxy en RAM:

Comandos importantes: max-cache-size, max-cache-object-size, cache-on-disk

```
/ip proxy set max-cache-size=unlimited max-cache-object-size=50000KiB cache-on-disk=no
/ip proxy print
                 enabled: yes
             src-address: ::
                    port: 8080
               anonymous: no
            parent-proxy: 0.0.0.0
      parent-proxy-port: 0
    cache-administrator: webmaster
          max-cache-size: unlimited  <-------
  max-cache-object-size: 500000KiB  <-------
           cache-on-disk: no  <-------
  max-client-connections: 600
  max-server-connections: 600
          max-fresh-time: 3d
   serialize-connections: no
       always-from-cache: no
          cache-hit-dscp: 4
              cache-path: proxy-cache
```

Caché de proxy con almacenamiento:

Comandos importantes: max-cache-size, max-cache-object-size, cache-on-disk, cache-path

```
/ip proxy set cache-on-disk=yes cache-path=/usb1/proxy/cache
/ip proxy print
                 enabled: yes
             src-address: ::
                    port: 8080
               anonymous: no
            parent-proxy: 0.0.0.0
      parent-proxy-port: 0
    cache-administrator: webmaster
          max-cache-size: unlimited  <-------
  max-cache-object-size: 50000KiB  <-------
           cache-on-disk: yes  <-------
  max-client-connections: 600
  max-server-connections: 600
          max-fresh-time: 3d
   serialize-connections: no
       always-from-cache: no
          cache-hit-dscp: 4
              cache-path: usb1/proxy/cache  <-------

/file print
 # NAME                                                TYPE
 0 skins                                               directory
 5 usb1/proxy                                          directory
 6 usb1/proxy/cache                                    web-proxy store   <-------
 7 usb1/lost+found                                     directory
```

Nota: Este ejemplo muestra como configurar el caché para las versiones desde la v6.20 ya que el almacenamiento está ahora en el menú /file como directorios

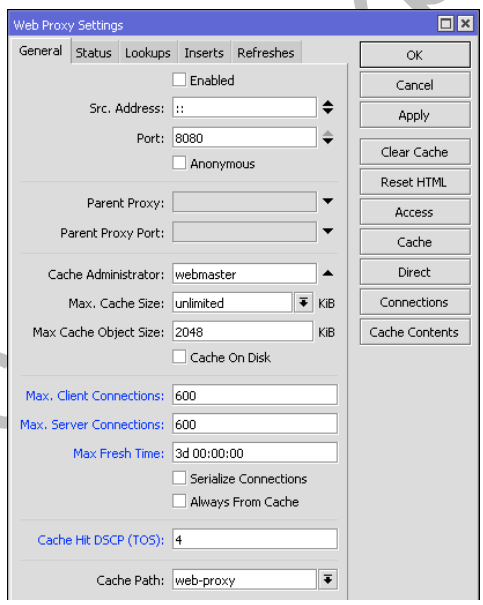
Para verificar si el caché está trabajando:

```
/ip proxy monitor
                 status: running
                 uptime: 2w20h28m25s
     client-connections: 15
     server-connections: 7
               requests: 79772
                   hits: 30513
             cache-used: 481KiB
         total-ram-used: 1207KiB
  received-from-servers: 4042536KiB
       sent-to-clients: 4399757KiB
   hits-sent-to-clients: 176934KiB
```

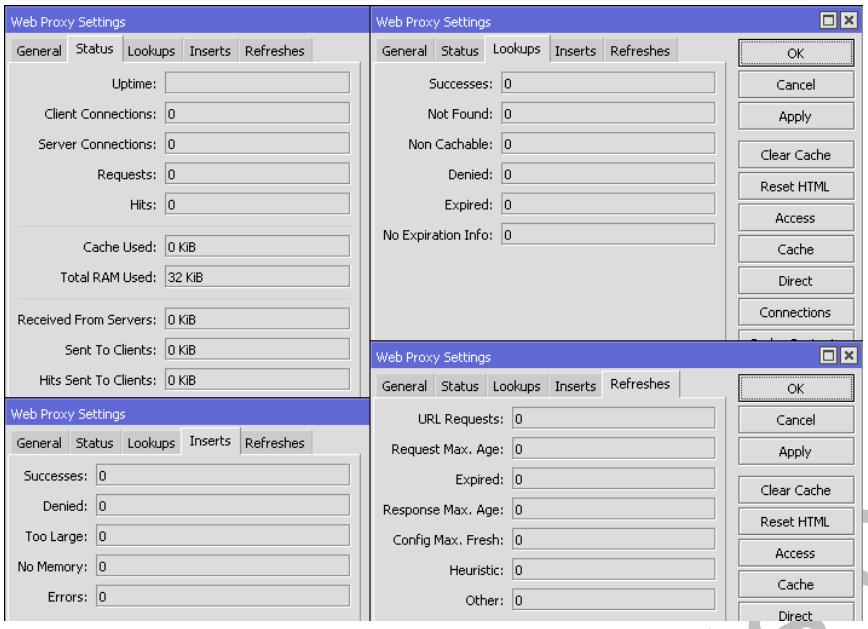
Parámetros

Sub-menu: `/ip proxy`

- **`always-from-cache`** *(yes | no; Default: no)*
- **`cache-administrator`** *(string; Default: webmaster)* – e-mail del Administrador que se muestra en la página de error del proxy
- **`cache-hit-dscp`** *(integer: 0..63; Default: 4)*
- **`cache-on-disk`** *(yes | no; Default: no)*
- **`max-cache-size`** *(none | unlimited | integer: 0..4294967295; Default: none)* – Especifica el tamaño máximo del caché, medido en kilobytes
- **`max-client-connections`** *(integer: 1..5000; Default: 600)* – Número máximo de conexiones aceptadas de clientes (más conexiones serán rechazadas)
- **`max-fresh-time`** *(time; Default: 3d)* – Tiempo máximo para almacenar un objeto en caché. El período de validez de un objeto es usualmente definido por el objeto mismo, pero en caso de que es demasiado alto, puede anular el valor máximo
- **`max-server-connections`** *(integer: 1..5000; Default: 600)* – Número máximo de conexiones realizadas a servidores (más conexiones de clientes se pondrán en espera hasta que algunas conexiones de servidor finalicen)
- **`parent-proxy`** *(ip4 | ip6; Default: 0.0.0.0)* – Dirección IP y puerto de otro proxy HTTP para redirigir todas las peticiones. Si se configura como `0.0.0.0` entonces no se utiliza el Proxy padre.
- **`parent-proxy-port`** *(integer: 0..65535; Default: 0)* – Puerto en que el Proxy Padre está escuchando.
- **`port`** *(integer: 0..65535; Default: 8080)* – Puerto TCP en que el Proxy Server estará escuchando. Este puerto tiene que ser especificado en todos los clientes que desean usar el servidor como Proxy HTTP. Se puede realizar la configuración de Proxy Transparente (en la que los clientes no realizan ninguna configuración) redireccionando las solicitudes HTTP a este puerto en `IP firewall` usando `destination NAT`
- **`serialize-connections`** *(yes | no; Default: no)*
- **`src-address`** *(ip4 | ip6; Default: 0.0.0.0)* – El proxy usará la dirección especificada cuando se conecte a un Proxy Padre o Website. Si se configura como `0.0.0.0` entonces la dirección IP apropiada será tomada de la tabla de ruteo.



Estadísticas de Web Proxy



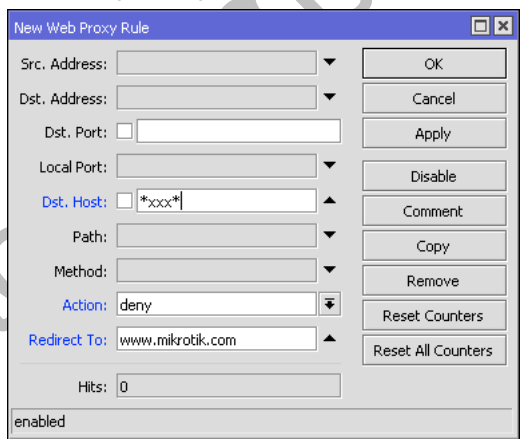
Access List

Sub-menu: `/ip proxy access`

La reglas de la Lista de Acceso (Access List) se implementan en la misma forma en que se procesan las reglas de firewall en MikroTik, es decir desde la parte superior a la inferior, en orden secuencial. La primera regla de coincidencia especifica la decisión de qué hacer con esta conexión.

Hay un total de 6 clasificadores que especifican restricciones coincidentes. Si no se especifica ninguno de estos clasificadores, la regla particular coincidirá con cada conexión..

Si la conexión se corresponde con una regla, la propiedad de acción de esta regla especifica si la conexión se permitirá o no. Si la conexión particular no coincide con ninguna regla, entonces se permitirá.



Parámetros:

- **action** *(allow | deny; Default: allow)* – Especifica si se pasan o se niegan los paquetes coincidentes
- **dst-address** *(Ip4[-Ip4 | /0..32] | Ip6/0..128; Default:)* – Dirección destino del servidor destino.
- **dst-host** *(string; Default:)* – Dirección IP o nombre DNS utilizados para hacer la conexión al servidor de destino (esta es la cadena que el usuario escribió en el navegador antes de especificar el puerto y la ruta de acceso a una página web en particular)
- **dst-port** *(integer[-integer[,integer[,...]]]: 0..65535; Default:)* – Lista o rango de puertos a los que el paquete está destinado
- **local-port** *(integer: 0..65535; Default:)* – Especifica el puerto del proxy web a través del cual se recibió el paquete. Este valor debe coincidir con uno de los puertos por los que está escuchando el Web Proxy.
- **method** *(any | connect | delete | get | head | options | post | put | trace; Default:)* – Método HTTP usado en la solicitud (ver la sección de métodos HTTP)
- **path** *(string; Default:)* – Nombre de la página solicitada en el servidor de destino (es decir, el nombre de una página web en particular o documento sin el nombre del servidor en el que reside)
- **redirect-to** *(string; Default:)* – En caso de que el acceso es negado para esta regla, el usuario será redirigido a la URL especificada aquí

- **src-address** *(Ip4[-Ip4 | /0..32] | Ip6/0..128; Default:)* – Dirección de origen del emisor de la conexión.

Propiedades solo lectura

- **hits** *(integer)* – Conteo de las solicitudes que coincidieron con esta regla

Las propiedades comodín (`dst-host` y `dst-path`) coinciden con una cadena completa (por ejemplo, no coincidirán "example.com" si está configurada como "example"). Los comodines disponibles son:

- * – Coincide con cualquier número de caracteres
- ? – Coincide con cualquier carácter

Las expresiones regulares también son aceptados aquí, pero si la propiedad debe ser tratada como una expresión regular, se debe comenzar con dos puntos (':').

Pequeños consejos en el uso de expresiones regulares:

- La secuencia del símbolo \\ se utiliza para ingresar el carácter \ en consola
- El patrón \. Significa únicamente . (en expresiones regulares un simple punto en el patrón significa cualquier símbolo)
- Para demostrar que ningún símbolo se permite antes que el patrón dado, Se usa el símbolo ^ al inicio del patrón
- Para demostrar que ningún símbolo se permite después que el patrón dado, Se usa el símbolo $ al final del patrón
- Para ingresar los símbolos [o], se debe hacer un "escape" con el backslash \.

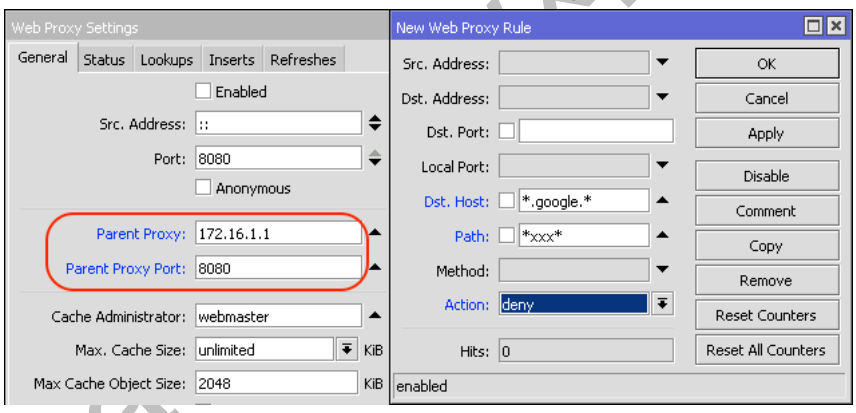
Se recomienda fuertemente denegar todas las direcciones IP excepto las que están detrás del router ya que el proxy todavía se puede utilizar para acceder a sus servidores web de uso-interno-solamente (intranet). También, se debe consultar ejemplos en el Manual de Firewall sobre cómo proteger a su router.

Direct Access

Sub-menu: `/ip proxy direct`

Si se especifica la propiedad parent-proxy, es posible decirle al servidor proxy si debe tratar de pasar la solicitud al Proxy Padre o para resolverlo conectarse al servidor solicitado directamente. Direct Access List se maneja similar al Proxy Access List descrito anteriormente, a excepción del argumento de acción.

A diferencia de Access List, el Direct Access List tiene una acción por default igual a denegar. Toma lugar cuando no se especifican reglas o una solicitud en particular no coindice con cualquier regla.



Parámetros:

- **action** *(allow | deny; Default: allow)* – Especifica la acción a realizar en los paquetes coincidentes:
  - **allow** – Siempre resuelve las solicitudes coincidentes sin pasar directamente al router padre
  - **deny** – Resuelve las peticiones coincidentes a través del proxy padre. Si no se especifica ninguna tiene el mismo efecto que "permitir" (`allow`).
- **dst-address** *(Ip4[-Ip4 | /0..32] | Ip6/0..128; Default:)* – Dirección destino del servidor destino.
- **dst-host** *(string; Default:)* – Dirección IP o nombre DNS utilizados para hacer la conexión al servidor de destino (esta es la cadena que el usuario escribió en el navegador antes de especificar el puerto y la ruta de acceso a una página web en particular)
- **dst-port** *(integer[-integer[,integer[,...]]]: 0..65535; Default:)* – Lista o rango de puertos a los que el paquete está destinado
- **local-port** *(integer: 0..65535; Default:)* – Especifica el puerto del proxy web a través del cual se recibió el paquete. Este valor debe coincidir con uno de los puertos por los que está escuchando el Web Proxy.
- **method** *(any | connect | delete | get | head | options | post | put | trace; Default:)* – Método HTTP usado en la solicitud (ver la sección de métodos HTTP)
- **path** *(string; Default:)* – Nombre de la página solicitada en el servidor de destino (es decir, el nombre de una página web en particular o documento sin el nombre del servidor en el que reside)
- **src-address** *(Ip4[-Ip4 | /0..32] | Ip6/0..128; Default:)* – Dirección de origen del emisor de la conexión.

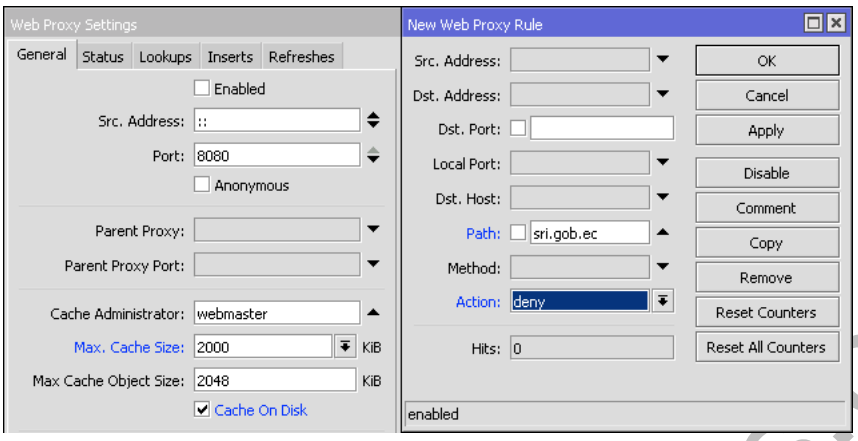
Propiedades solo lectura

- **hits** *(integer)* – Conteo de las solicitudes que coincidieron con esta regla

Administración del Caché

Sub-menu: `/ip proxy cache`

El Caché Access List especifica, que requerimientos (dominios, servidores, páginas) tienen que ser almacenados en caché localmente por el Web Proxy, y cuales no. Esta lista se implementa exactamente del mismo modo que el Access List del Web proxy. La acción predeterminada es almacenar en caché los objetos (si no se encuentra una regla que coincida).



Parámetros:

- **`action`** *(allow | deny; Default: allow)* – Especifica la acción a realizar en los paquetes coincidentes:
  - `allow` – Caché de objetos que coinciden con el requerimiento
  - `deny` – No hace caché de objetos que coinciden con el requerimiento.
- **`dst-address`** *(Ip4[-Ip4 | /0..32] | Ip6/0..128; Default:)* – Dirección destino del servidor destino.
- **`dst-host`** *(string; Default:)* – Dirección IP o nombre DNS utilizados para hacer la conexión al servidor de destino (esta es la cadena que el usuario escribió en el navegador antes de especificar el puerto y la ruta de acceso a una página web en particular)
- **`dst-port`** *(integer[-integer[,integer[,...]]]: 0..65535; Default:)* – Lista o rango de puertos a los que el paquete está destinado
- **`local-port`** *(integer: 0..65535; Default:)* – Especifica el puerto del proxy web a través del cual se recibió el paquete. Este valor debe coincidir con uno de los puertos por los que está escuchando el Web Proxy.
- **`method`** *(any | connect | delete | get | head | options | post | put | trace; Default:)* – Método HTTP usado en la solicitud (ver la sección de métodos HTTP)
- **`path`** *(string; Default:)* – Nombre de la página solicitada en el servidor de destino (es decir, el nombre de una página web en particular o documento sin el nombre del servidor en el que reside)
- **`src-address`** *(Ip4[-Ip4 | /0..32] | Ip6/0..128; Default:)* – Dirección de origen del emisor de la conexión.

Propiedades solo lectura

- **`hits`** *(integer)* – Conteo de las solicitudes que coincidieron con esta regla

Connections

Sub-menu: `/ip proxy connections`

Este menú contiene la lista de las conexiones reales a las que el proxy está atendiendo

- **`client`** *()*
- **`dst-address`** *(Ip4 | Ip6)* – Dirección destino IPv4/Ipv6 de la conexión destino
- **`protocol`** *(string)* – Nombre del protocolo
- **`rx-bytes`** *(integer)* – La cantidad de bytes recibidos por el cliente
- **`server`** *()*
- **`src-address`** *(Ip4 | Ip6)* – Dirección destino IPv4/Ipv6 de la conexión origen
- **`state`** *(closing | connecting | converting | hotspot | idle | resolving | rx-header | tx-body | tx-eof | tx-header | waiting)* Estado de la conexión:
  - **`closing`** – La transferencia de datos ha terminado, y la conexión se está finalizando
  - **`connecting`** – Estableciendo la conexión
  - **`converting`** – Reemplazando los campos de encabezado (header) y pie de página (footer) en respuesta o solicitud de paquetes
  - **`hotspot`** – Comprobar si la autenticación de punto de acceso (hotspot) permite continuar (par hotspot proxy)
  - **`idle`** – Permanece inactivo
  - **`resolving`** – Resolviendo el nombre DNS del servidor
  - **`rx-header`** – Recibiendo la cabecera HTTP
  - **`tx-body`** – Transmitiendo el cuerpo HTTP al cliente

    - **tx-eof** – Escribiendo chunk-end (al convertir a una respuesta fragmentada)
    - **tx-header** – Transmitiendo la cabecera HTTP al cliente
    - **waiting** – Esperando por la transmisión de un compañero (peer)
- **tx-bytes** *(integer)* – La cantidad de bytes enviados por el cliente

Métodos HTTP

Opciones

Este método es una solicitud de información acerca de las opciones de comunicación disponibles en la cadena entre el cliente y el servidor identificado por el Request-URI. El método permite al cliente determinar las opciones y (o) los requerimientos asociados con un recurso sin iniciar cualquier recuperación de recursos.

GET

- Este método recupera toda la información identificada por el Request-URI. Si el Request-URI se refiere a un proceso de procesamiento de datos entonces la respuesta al método GET debe contener los datos producidos por el proceso, no el código fuente de los procedimientos de proceso, a menos que la fuente es el resultado del proceso.
- El método GET puede convertirse en un GET condicional si el mensaje de solicitud incluye un campo de cabecera If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match, o If-Range. El método GET condicional se usa para reducir el tráfico de red especificando que la transferencia de la entidad debería ocurrir sólo en circunstancias descritas por los campos de cabecera condicional.
- El método GET puede convertirse en un GET parcial si el mensaje de solicitud incluye un campo campo de cabecera Range. El método GET parcial tiene la intención de reducir el uso innecesario de la red mediante la solicitud solamente de partes de entidades sin transferir los datos ya en poder del cliente.
- La respuesta a una petición GET es cacheable si y sólo si cumple con los requisitos para el almacenamiento de caché HTTP.

HEAD

- Este método comparte todas las características del método GET, excepto que el servidor no tiene que volver un cuerpo de mensaje en la respuesta. Este recupera la metainformación de la entidad implicada por la solicitud que conduce a un amplio uso de la misma para la prueba de enlaces hipertexto para la validez, la accesibilidad, y la modificación reciente.
- La respuesta a una petición HEAD puede ser cacheable en la forma en que la información contenida en la respuesta se puede utilizar para actualizar entidad previamente almacenada en caché identificado por ese Request-URI.

POST

- Este método solicita que el servidor de origen acepte la entidad incluida en la solicitud como un nuevo subordinado del recurso identificado por el Request-URI.
- La acción real realizada por el método POST se determina por el servidor de origen y por lo general es dependiente del Request-URI.
- Las respuestas al método POST no son cacheable, a menos que la respuesta incluya los apropiados campos de cabecera Cache-Control o Expires.

PUT

- Este método pide que la entidad adjunta sea almacenada bajo el Request-URI suministrado. Si existe otra entidad bajo el Request-URI especificado, la entidad adjunta debe ser considerada como la versión actualizada (más reciente) de la que reside en el servidor de origen. Si la Solicitud-URI no está apuntando a un recurso existente, el servidor de origen debe crear un recurso con ese URI.
- Si la solicitud pasa a través de una memoria caché y el Request-URI identifica una o más entidades actualmente almacenada en caché, las entradas deben ser tratadas como obsoletos. Las respuestas a este método no son cacheables.

TRACE

- Este método invoca un requerimiento de mensaje remoto de la capa de aplicación. El destinatario final de la solicitud debe reflejar el mensaje recibido de vuelta al cliente como el cuerpo de la entidad de una respuesta 200 (OK). El destinatario final es el servidor de origen o el primer proxy o gateway para recibir un valor Max-Forwards de 0 en la solicitud. Una petición TRACE no debe incluir una entidad.
- Las respuestas a este método, no deberán ser almacenadas en caché.

Capítulo 13: TTL

El TTL (Time To Live = Tiempo de vida) es un mecanismo que limita la duración o el tiempo de vida de los datos en un computadora o red. El TTL pueden implementarse como un contador o marca de tiempo unido/incrustado en los datos. Una vez que el conteo de evento prescribe o ha transcurrido un intervalo de tiempo, los datos se descartan. En las redes de computadoras, el TTL evita que un paquete de datos circule indefinidamente. En las aplicaciones de cómputo, el TTL se utiliza para mejorar el rendimiento del almacenamiento de caché o para mejorar la privacidad.

Datos sobre TTL en el Protocolo Internet (Internet Protocol)

- TTL es un campo de 8-bit
- En la cabecera IPv4, el TTL es el 9th octeto (de 20 octetos)
- En la cabecera IPv6, es el 8th octeto (de 40 octetos)
- El valor máximo de TTL es 255, que es el máximo valor de un simple octeto
- Un valor inicial recomendado de TTL es 64
- Si el TTL=0 el datagrama debe ser destruido

El valor `time-to-live` puede ser considerado como un límite superior en el momento en que puede existir un datagrama IP en un sistema de Internet. El campo TTL es establecido por el remitente (sender) del datagrama, y se reduce en cada router en el camino hacia su destino. Si el campo TTL llega a cero antes que el datagrama llegue a su destino, entonces el datagrama es descartado y un datagrama de error ICMP (`11 - Tiempo Excedido`) se devuelve al remitente. El propósito del campo TTL es evitar una situación en la que, un datagrama que no se puede entregar, se mantiene en circulación en un sistema de Internet, y tal sistema eventualmente llega a ser inundado por tales paquetes "inmortales".

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				Type Of Service								Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source Address																															
16	128	Destination Address																															
20	160	Option + Padding																															
24	192	Data																															
		IP Header Structure																															

En teoría, bajo IPv4, el TTL se mide en segundos, aunque cada host que pasa el datagrama debe reducir el TTL por lo menos en una unidad. En la práctica, el campo TTL se reduce en uno en cada salto. Para reflejar esta práctica, el campo es rebautizado como `hop limit` en IPv6.

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				Priority				Flow Lebel																							
4	32	Payload Length																Next Header								Hop Limit							
8	64	Source Address (128 bits)																															
12	96																																
16	128																																
20	160																																
24	192	Destination Address (128 bits)																															
28	224																																
32	256																																
36	288																																
		IPv6 Header Structure																															

Valor de TTL para los principales Dispositivos y Sistemas Operativos

OS/Device	Version	Protocol	TTL
AIX		TCP	60
AIX		UDP	30
AIX	3.2, 4.1	ICMP	255
BSDI	BSD/OS 3.1 and 4.0	ICMP	255
Compa	Tru64 v5.0	ICMP	64
Cisco		ICMP	254
DEC Pathworks	V5	TCP and UDP	30

OS/Device	Version	Protocol	TTL
Solaris	2.5.1, 2.6, 2.7, 2.8	ICMP	255
Solaris	2.8	TCP	64
Stratus	TCP_OS	ICMP	255
Stratus	TCP_OS (14.2-)	TCP and UDP	30
Stratus	TCP_OS (14.3+)	TCP and UDP	64
Stratus	STCP	ICMP/TCP/UDP	60
SunOS	4.1.3/4.1.4	TCP and UDP	60

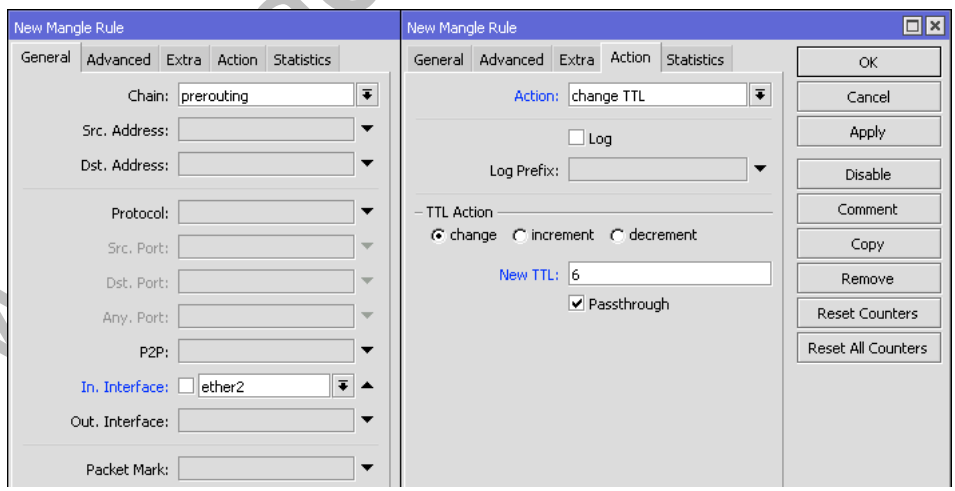
OS/Device	Version	Protocol	TTL
Foundry		ICMP	64
FreeBSD	2.1R	TCP and UDP	64
FreeBSD	3.4, 4.0	ICMP	255
FreeBSD	5	ICMP	64
HP-UX	9.0x	TCP and UDP	30
HP-UX	10.01	TCP and UDP	64
HP-UX	10.2	ICMP	255
HP-UX	11	ICMP	255
HP-UX	11	TCP	64
Irix	5.3	TCP and UDP	60
Irix	6.x	TCP and UDP	60
Irix	6.5.3, 6.5.8	ICMP	255
juniper		ICMP	64
MPE/IX (HP)		ICMP	200
Linux	2.0.x kernel	ICMP	64
Linux	2.2.14 kernel	ICMP	255
Linux	2.4 kernel	ICMP	255
Linux	Red Hat 9	ICMP and TCP	64
MacOS/MacTCP	2.0.x	TCP and UDP	60
MacOS/MacTCP	X (10.5.6)	ICMP/TCP/UDP	64
NetBSD		ICMP	255
Netgear FVG318		ICMP and UDP	64
OpenBSD	2.6 & 2.7	ICMP	255
OpenVMS	07.01.2002	ICMP	255
OS/2	TCP/IP 3.0		64
OSF/1	V3.2A	TCP	60
OSF/1	V3.2A	UDP	30

OS/Device	Version	Protocol	TTL
SunOS	5.7	ICMP and TCP	255
Ultrix	V4.1/V4.2A	TCP	60
Ultrix	V4.1/V4.2A	UDP	30
Ultrix	V4.2 – 4.5	ICMP	255
VMS/Multinet		TCP and UDP	64
VMS/TCPware		TCP	60
VMS/TCPware		UDP	64
VMS/Wollongong	1.1.1.1	TCP	128
VMS/Wollongong	1.1.1.1	UDP	30
VMS/UCX		TCP and UDP	128
Windows	for Workgroups	TCP and UDP	32
Windows	95	TCP and UDP	32
Windows	98	ICMP	32
Windows	98, 98 SE	ICMP	128
Windows	98	TCP	128
Windows	NT 3.51	TCP and UDP	32
Windows	NT 4.0	TCP and UDP	128
Windows	NT 4.0 SP5-		32
Windows	NT 4.0 SP6+		128
Windows	NT 4 WRKS SP 3, SP 6a	ICMP	128
Windows	NT 4 Server SP4	ICMP	128
Windows	ME	ICMP	128
Windows	2000 pro	ICMP/TCP/UDP	128
Windows	2000 family	ICMP	128
Windows	Server 2003		128
Windows	XP	ICMP/TCP/UDP	128

Mangle action=change-ttl

- TTL es un límite de dispositivos de Capa 3 (L3) que el paquete IP puede experimentar antes de que sea descartado
- El valor por default del TTL es 64 y cada router reduce el valor en uno justo antes de la hacer el forwarding
- El router no pasará el tráfico al siguiente dispositivo si recibe un paquete IP con TTL=1
- Aplicación útil: eliminar la posibilidad a los clientes de crear redes enmascaradas

Cambiando el TTL

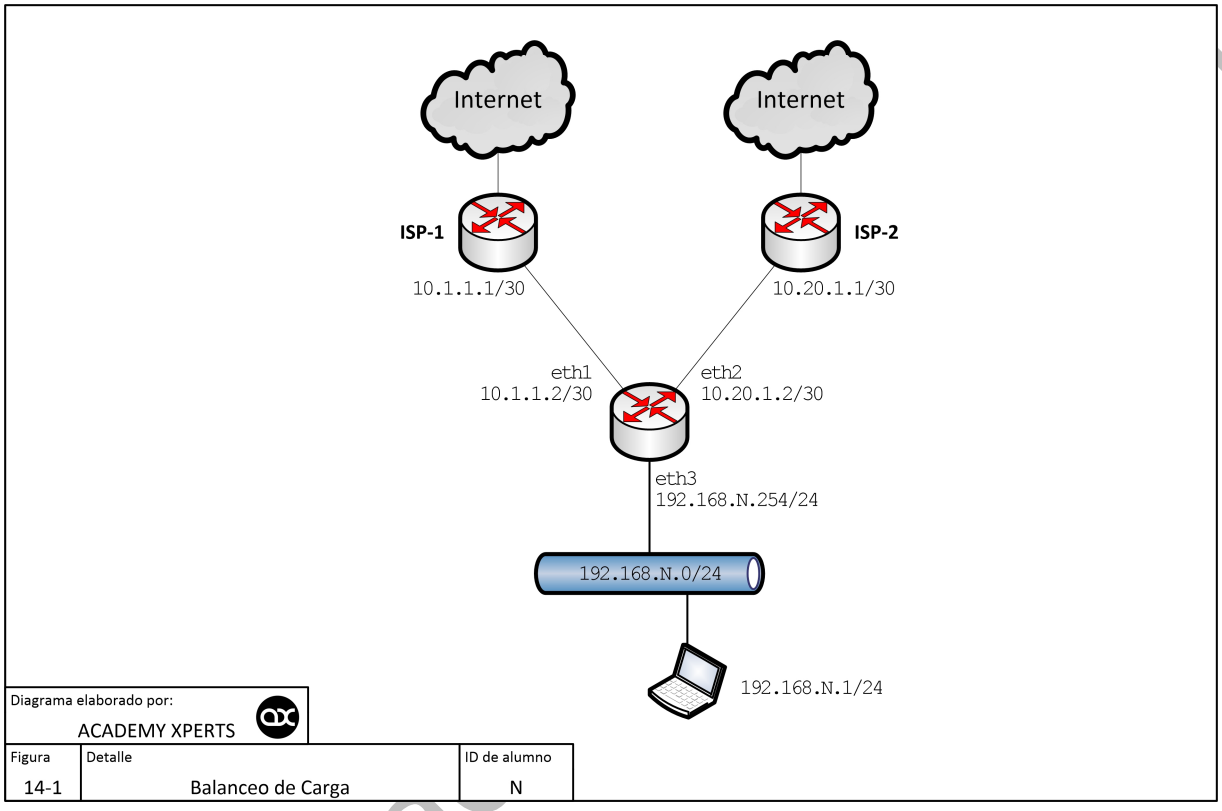


Capítulo 14: Balanceo de Carga

Existen 3 métodos conocidos para balanceo de carga en RouterOS:

- Usando PCC (per Connection Classifier)
- Usando ECMP (Equal Cost Multi-Path)
- Usando Nésimo Paquete (Nth Packet)

Balanceo de Carga Usando PCC (Per Connection Classifier)



Direcciones IP

```
/ip address
add address=192.168.0.1/24 interface=eth3
add address=10.1.1.2/30 interface=eth1
add address=10.20.1.2/30 interface=eth2
```

Políticas de ruteo

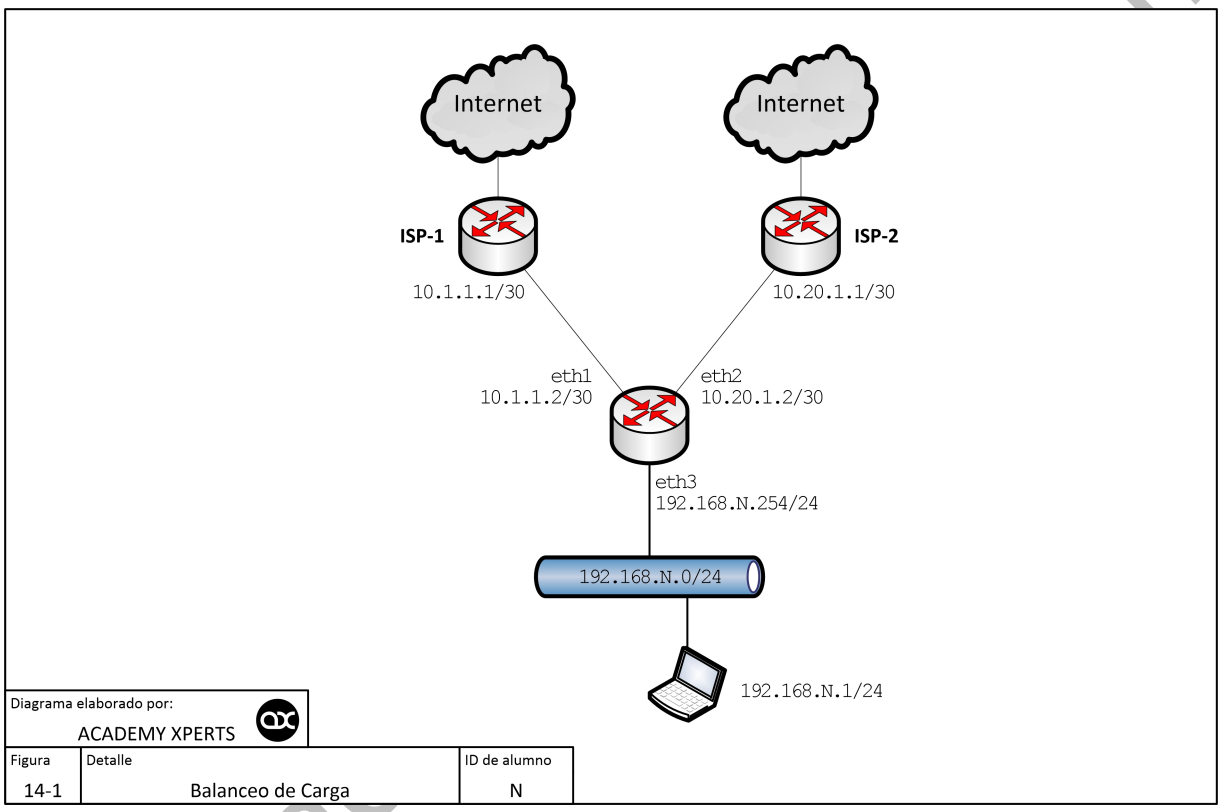
```
/ip firewall mangle
add chain=prerouting dst-address=10.1.1.0/30 action=accept in-interface=eth3
add chain=prerouting dst-address=10.20.1.0/30 action=accept in-interface=eth3

add chain=prerouting in-interface=eth1 connection-mark=no-mark action=mark-connection \
        new-connection-mark=ISP1_conn
add chain=prerouting in-interface=eth2 connection-mark=no-mark action=mark-connection \
        new-connection-mark=ISP2_conn

add chain=prerouting in-interface=eth3 connection-mark=no-mark dst-address-type=!local \
        per-connection-classifier=both-addresses:2/0 action=mark-connection new-connection-mark=ISP1_conn
add chain=prerouting in-interface=eth3 connection-mark=no-mark dst-address-type=!local \
        per-connection-classifier=both-addresses:2/1 action=mark-connection new-connection-mark=ISP2_conn

add chain=prerouting connection-mark=ISP1_conn in-interface=eth3 action=mark-routing \
        new-routing-mark=to_ISP1 passthrough=no
add chain=prerouting connection-mark=ISP2_conn in-interface=eth3 action=mark-routing \
        new-routing-mark=to_ISP2 passthrough=no

add chain=output connection-mark=ISP1_conn action=mark-routing new-routing-mark=to_ISP1 passthrough=no
```

```
add chain=output connection-mark=ISP2_conn action=mark-routing new-routing-mark=to_ISP2 passthrough=no

/ip route
add dst-address=0.0.0.0/0 gateway=10.1.1.1 routing-mark=to_ISP1 check-gateway=ping
add dst-address=0.0.0.0/0 gateway=10.20.1.1 routing-mark=to_ISP2 check-gateway=ping

add dst-address=0.0.0.0/0 gateway=10.1.1.1 distance=1 check-gateway=ping
add dst-address=0.0.0.0/0 gateway=10.20.1.1 distance=2 check-gateway=ping
```

NAT

```
/ip firewall nat
add chain=srcnat out-interface=eth1 action=masquerade
add chain=srcnat out-interface=eth2 action=masquerade
```

Balanceo de Carga Usando ECMP (Equal Cost Multi-Path)



Direcciones IP

```
/ip address
add address=192.168.0.1/24 interface=eth3
add address=10.1.1.2/30 interface=eth1
add address=10.20.1.2/30 interface=eth2
```

NAT

```
/ip firewall nat
add chain=srcnat out-interface=eth1 action=masquerade
add chain=srcnat out-interface=eth2 action=masquerade
```

Routing

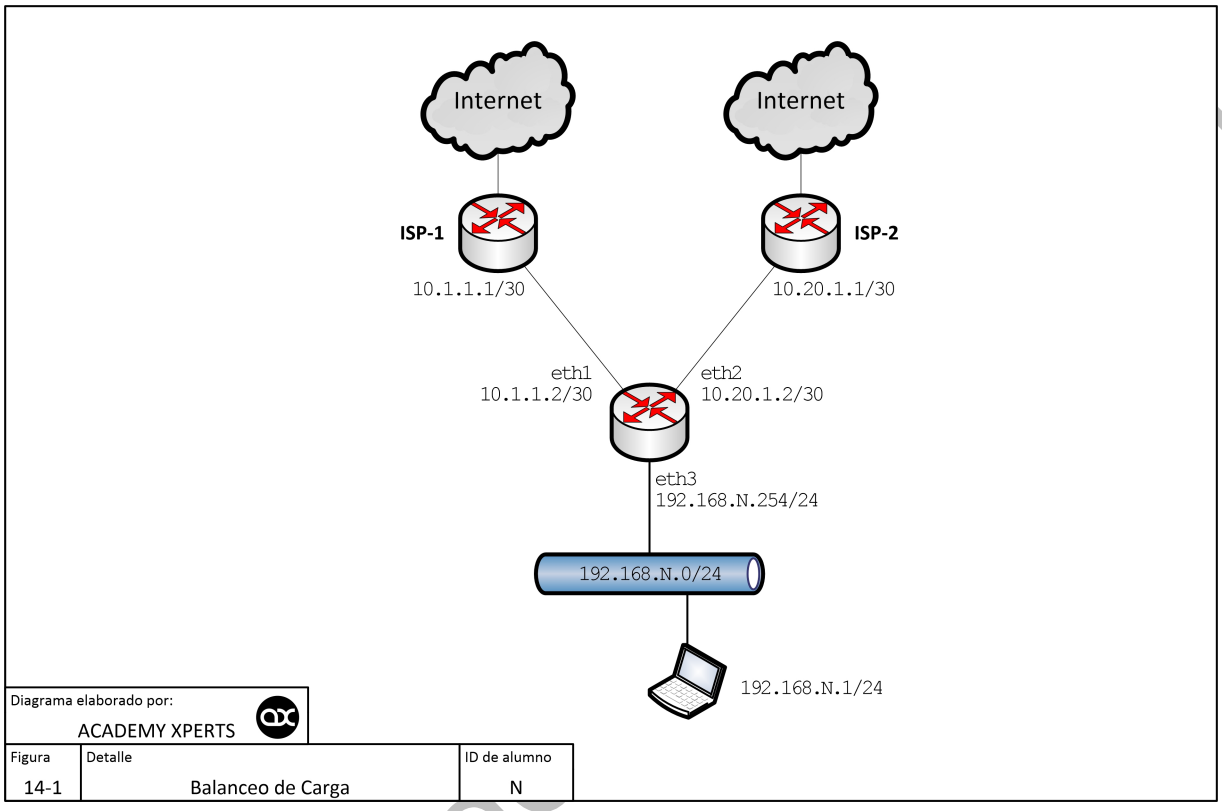
```
/ip route
add dst-address=0.0.0.0/0 gateway=10.1.1.1,10.20.1.1 check-gateway=ping
/ip route
add dst-address=0.0.0.0/0 gateway=10.1.1.1,10.20.1.1,10.20.1.1,10.20.1.1,10.20.1.1,10.20.1.1 \
        check-gateway=ping
```

Conexiones usadas por propio router

```
/ip firewall mangle
add chain=input in-interface=eth1 action=mark-connection new-connection-mark=ISP1_conn
add chain=input in-interface=eth2 action=mark-connection new-connection-mark=ISP1_conn
```

```
add chain=output connection-mark=ISP1_conn action=mark-routing new-routing-mark=to_ISP1 passthrough=no
add chain=output connection-mark=ISP2_conn action=mark-routing new-routing-mark=to_ISP2 passthrough=no

/ip route
add dst-address=0.0.0.0/0 gateway=10.1.1.1 routing-mark=to_ISP1
add dst-address=0.0.0.0/0 gateway=10.20.1.1 routing-mark=to_ISP2
```

Balanceo de Carga usando Nésimo Paquete (Nth Packet)



IP Addresses

```
/ip address
add address=192.168.0.1/24 interface=eth3
add address=10.1.1.2/30 interface=eth1
add address=10.20.1.2/30 interface=eth2
```

Mangle

```
/ip firewall mangle
add chain=prerouting src-address-list=odd in-interface=eth3 action=mark-connection \
        new-connection-mark=odd passthrough=yes
add chain=prerouting src-address-list=odd in-interface=eth3 action=mark-routing \
        new-routing-mark=odd

/ip firewall mangle
add chain=prerouting src-address-list=even in-interface=eth3 action=mark-connection \
        new-connection-mark=even passthrough=yes
add chain=prerouting src-address-list=even in-interface=eth3 action=mark-routing \
        new-routing-mark=even

/ip firewall mangle
add chain=prerouting in-interface=eth3 connection-state=new nth=2,1 \
        action=mark-connection new-connection-mark=odd passthrough=yes
add chain=prerouting in-interface=eth3 action=add-src-to-address-list \
        address-list=odd address-list-timeout=1d connection-mark=odd passthrough=yes
add chain=prerouting in-interface=eth3 connection-mark=odd action=mark-routing \
        new-routing-mark=odd passthrough=no


/ip firewall mangle
```

```
add chain=prerouting in-interface=eth3 connection-state=new nth=2,2 \
        action=mark-connection new-connection-mark=even passthrough=yes
add chain=prerouting in-interface=eth3 action=add-src-to-address-list \
        address-list=even address-list-timeout=1d connection-mark=even passthrough=yes
add chain=prerouting in-interface=eth3 connection-mark=even action=mark-routing \
        new-routing-mark=even passthrough=no

add chain=prerouting in-interface=eth3 connection-state=new nth=2,2 \
        src-address-list=!odd action=mark-connection new-connection-mark=even \
        passthrough=yes
```

NAT

```
/ip firewall nat
add chain=srcnat out-interface=eth1 action=masquerade
add chain=srcnat out-interface=eth2 action=masquerade
```

Routing

```
/ip route
add dst-address=0.0.0.0/0 gateway=10.1.1.1 scope=255 target-scope=10 routing-mark=odd
add dst-address=0.0.0.0/0 gateway=10.20.1.1 scope=255 target-scope=10 routing-mark=even

/ip route
add dst-address=0.0.0.0/0 gateway=10.20.1.1 scope=255 target-scope=10
```

Control de Tráfico, Firewall y QoS con MikroTik RouterOS

por Mauro Escalante

academy xperts