PCC y Balanceo de Carga en Mikrotik RouterOS

Por Freddy Bohorquez Quevedo

Objetivo

- Conocer y comprender el funcionamiento del Balanceo de Carga por Conexiones (PCC en RouterOS)

- Comparar los distintos métodos de Balanceo de Carga y sus ámbitos de aplicación práctica.

Contenido

Introducción

Métodos de Balanceo

PCC: Per Connection Classifier

Análisis

“

El bosque no sería el mismo si sólo cantaran los pájaros que mejor lo hacen.

Rabindranath Tagore; *premio Nobel de Literatura 1913.*

Balanceo de Carga

- Técnica que distribuye carga de trabajo en dos o más enlaces de red.

- Permite maximizar el rendimiento, minimizar el tiempo de respuesta y evitar la sobrecarga.

- Múltiples enlaces de red balanceados, en lugar de enlaces simples, incrementan la disponibilidad: redundancia.

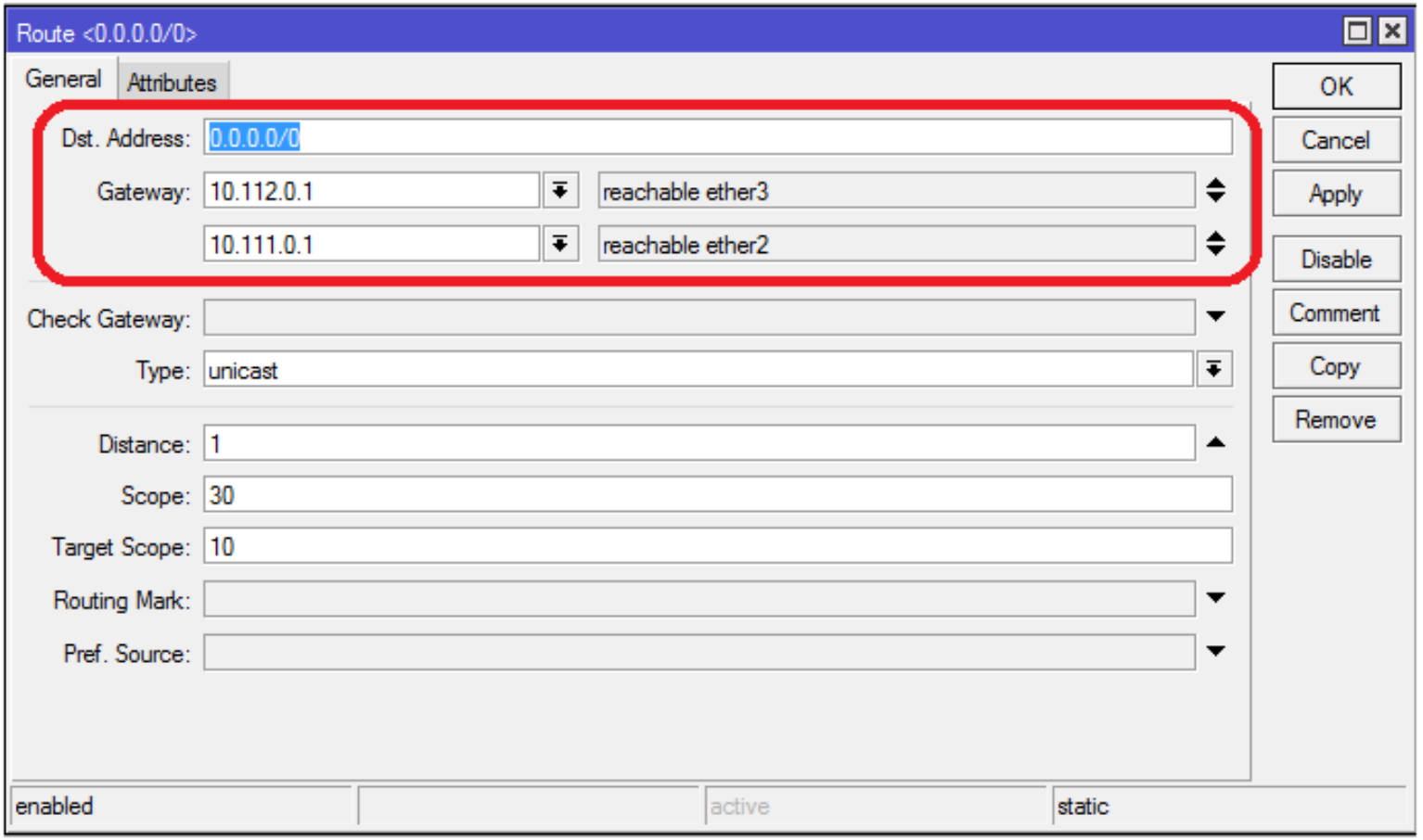
Métodos de Balanceo

- ECMP (Equal Cost Multi-Path).
- Etiquetado y políticas de enrutamiento.
- Bonding.
- NTH (N-esimo).
- PCC Per Connection Classifier.

ECMP (Equal Cost Multi-Path)

- Utiliza varios gateway con el mismo costo.
- Simple de implementar.
- Balanceo persistente por conexión.
- No se puede controlar la forma de balanceo.
- No funciona con puertas de enlace iguales.
- Failover automático.
- Mismo gateway puede estar varias veces.

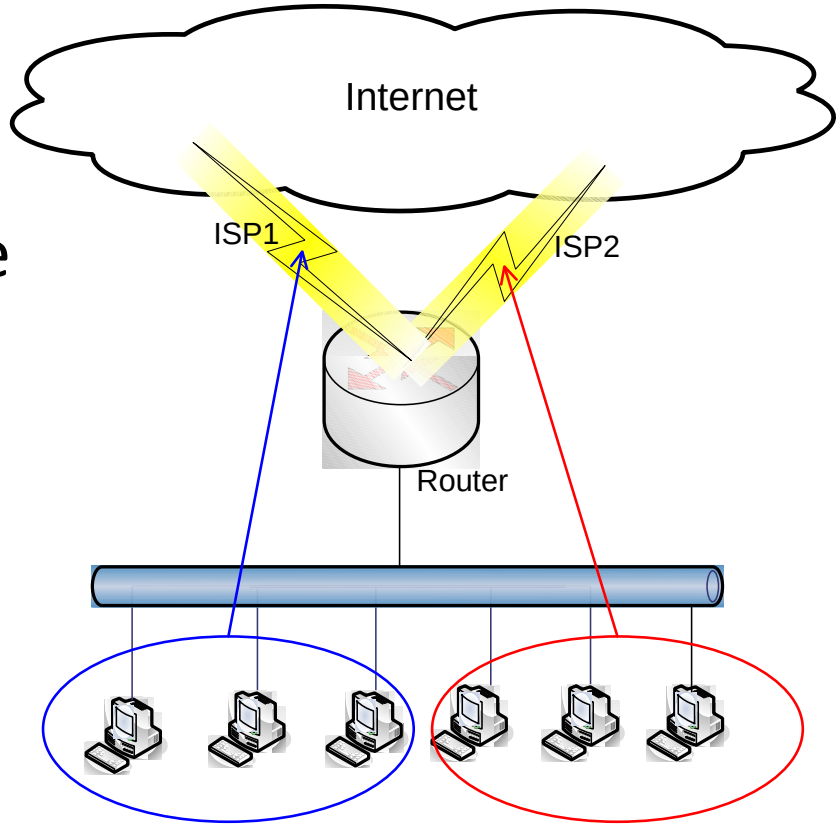
ECMP (Equal Cost Multi-Path)

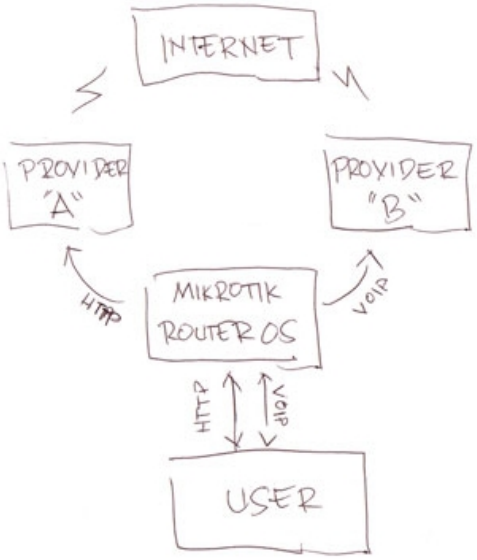



```
> ip route add gateway=10.111.0.1,10.112.0.1
```

Etiquetado y políticas de enrutamiento

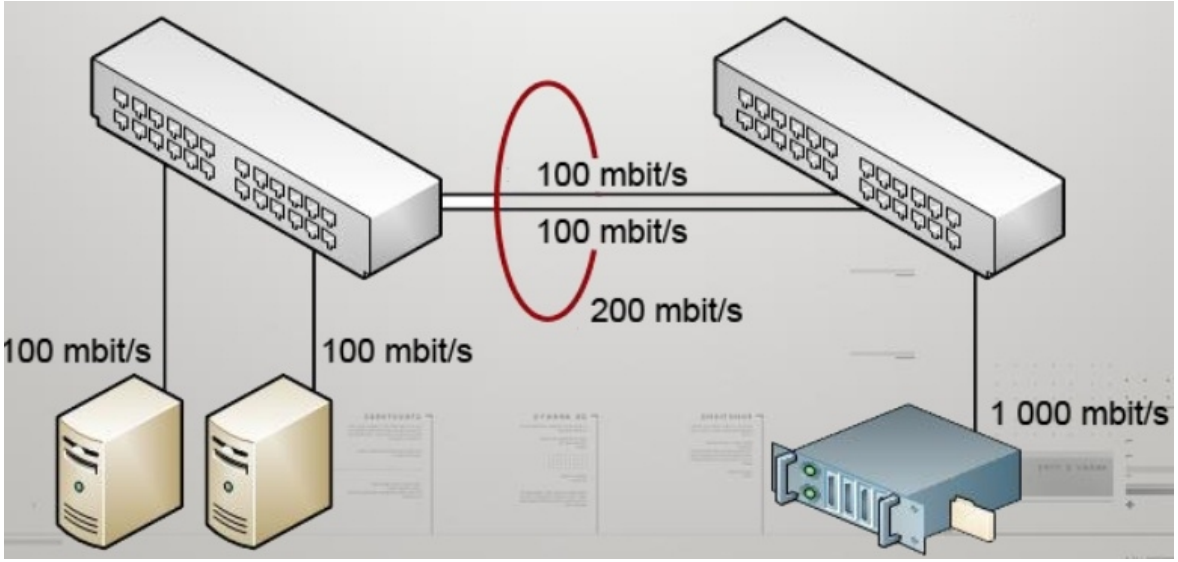
- Fácil de implementar
- Se tiene control exacto del tráfico
- No dinámico (problemas de escalabilidad)





Bonding

- Fácil de implementar.
- Redundancia automática con failover.
- Es necesario controlar ambos extremos del enlace.



NTH (N-th)

- Cada regla tiene su propio contador.
- De cada M paquetes inspeccionados (every) hace match con el N-esimo paquete.

1

```
/ip firewall mangle
add action=mark-packet chain=prerouting new-packet-mark=AAA nth=3,1 passthrough=no;
add action=mark-packet chain=prerouting new-packet-mark=BBB nth=2,1 passthrough=no;
add action=mark-packet chain=prerouting new-packet-mark=CCC ;
```

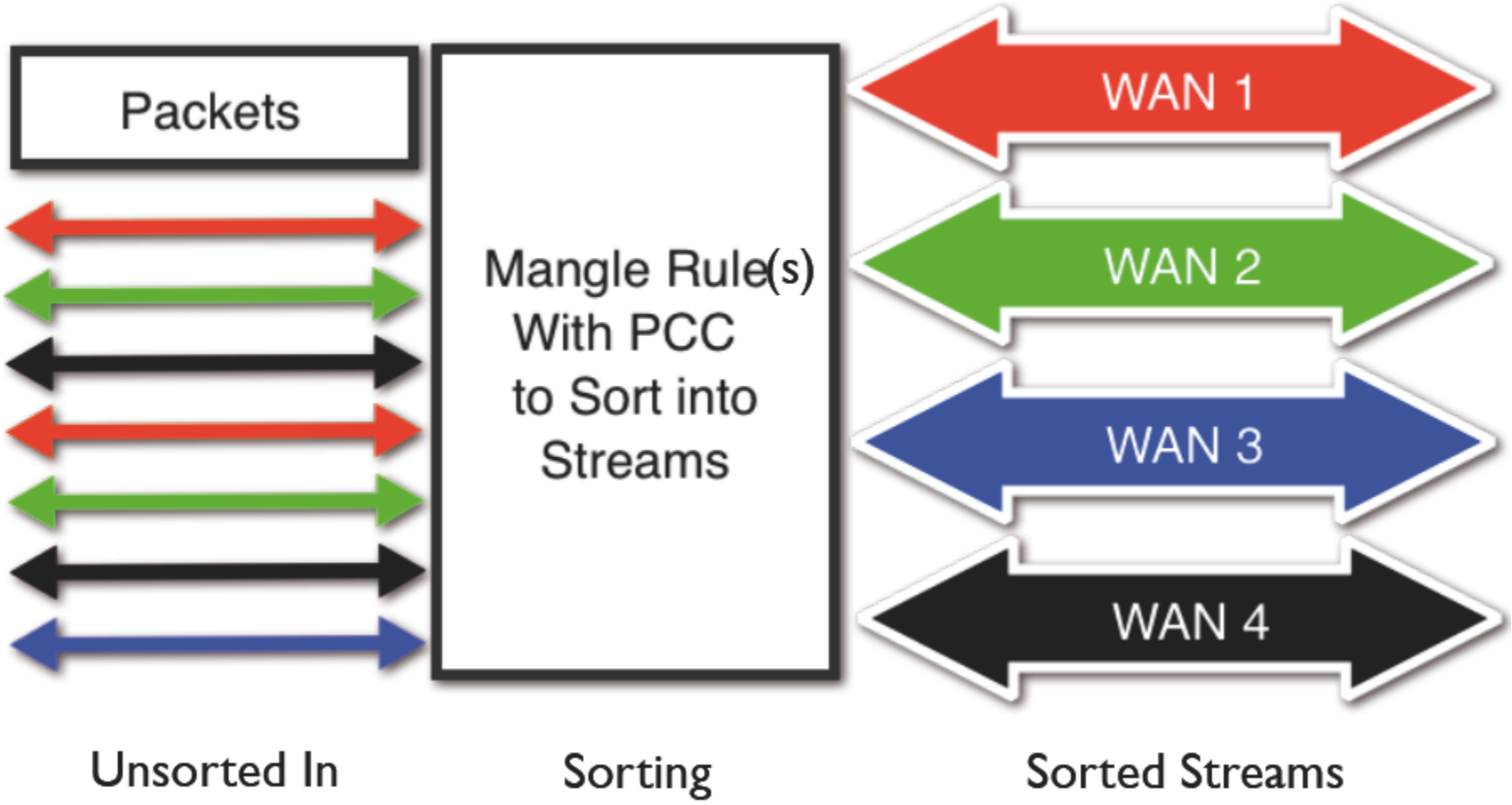
2

```
/ip firewall mangle
add action=mark-packet chain=prerouting new-packet-mark=AAA nth=3,1 passthrough=yes;
add action=mark-packet chain=prerouting new-packet-mark=BBB nth=3,2 passthrough=yes;
add action=mark-packet chain=prerouting new-packet-mark=CCC nth=3,3 passthrough=yes;
```

PCC: Per Connection Classifier

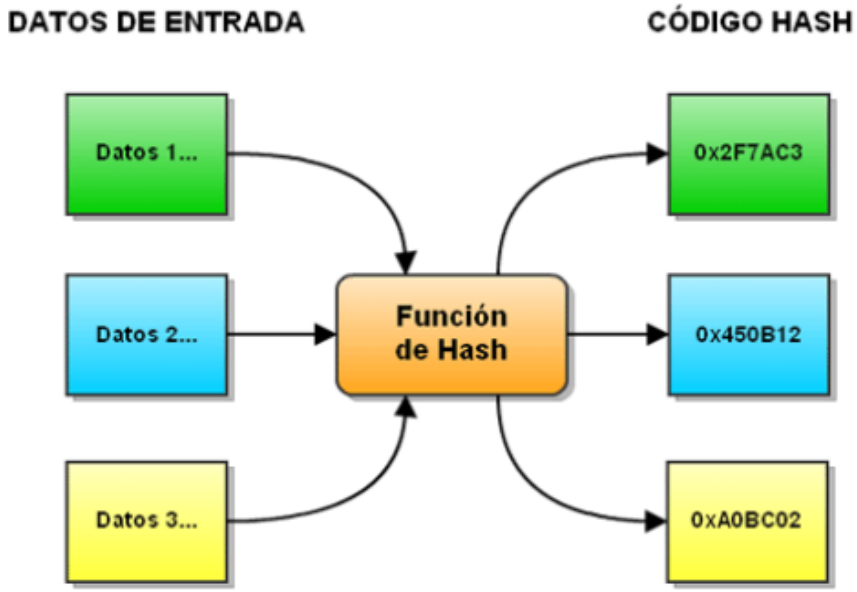
- Opción tipo **match** usada normalmente en mangle.
- **PCC matcher** permite clasificar el tráfico en flujos “similares”.
- Puede **mantener** los paquetes en un flujo específico basado en parámetros.
- **Conjunto de opciones:** src-address, src-port, dst-address, dst-port no del todo excluyentes entre si.

PCC: Per Connection Classifier



PCC: función Hash

- Función computable mediante un **algoritmo**.



- **Ejemplos: CRC, SHA1,SHA-1,SHA-512,MD5…**

PCC: función Hash propiedades

- Unidireccionalidad.
- Compresión.
- Facilidad de cálculo.
- Difusión.
- ….
- **Uniformidad:** valores hash distribuidos con la misma probabilidad
- **Determinista:** misma cadena de entrada devuelve el mismo valor hash.

PCC: Aritmética modular

- Algoritmo PCC usa división modular.
- División de números enteros:

$$a = \mathbf{bq} + \mathbf{r}\ con\ a, b, q, r\ \in \mathbb{I};\ a = \mathbf{D}q + \mathbf{R}$$

- ***R*** (valor que nos interesa) suele llamarse residuo o resto y se escribe como:

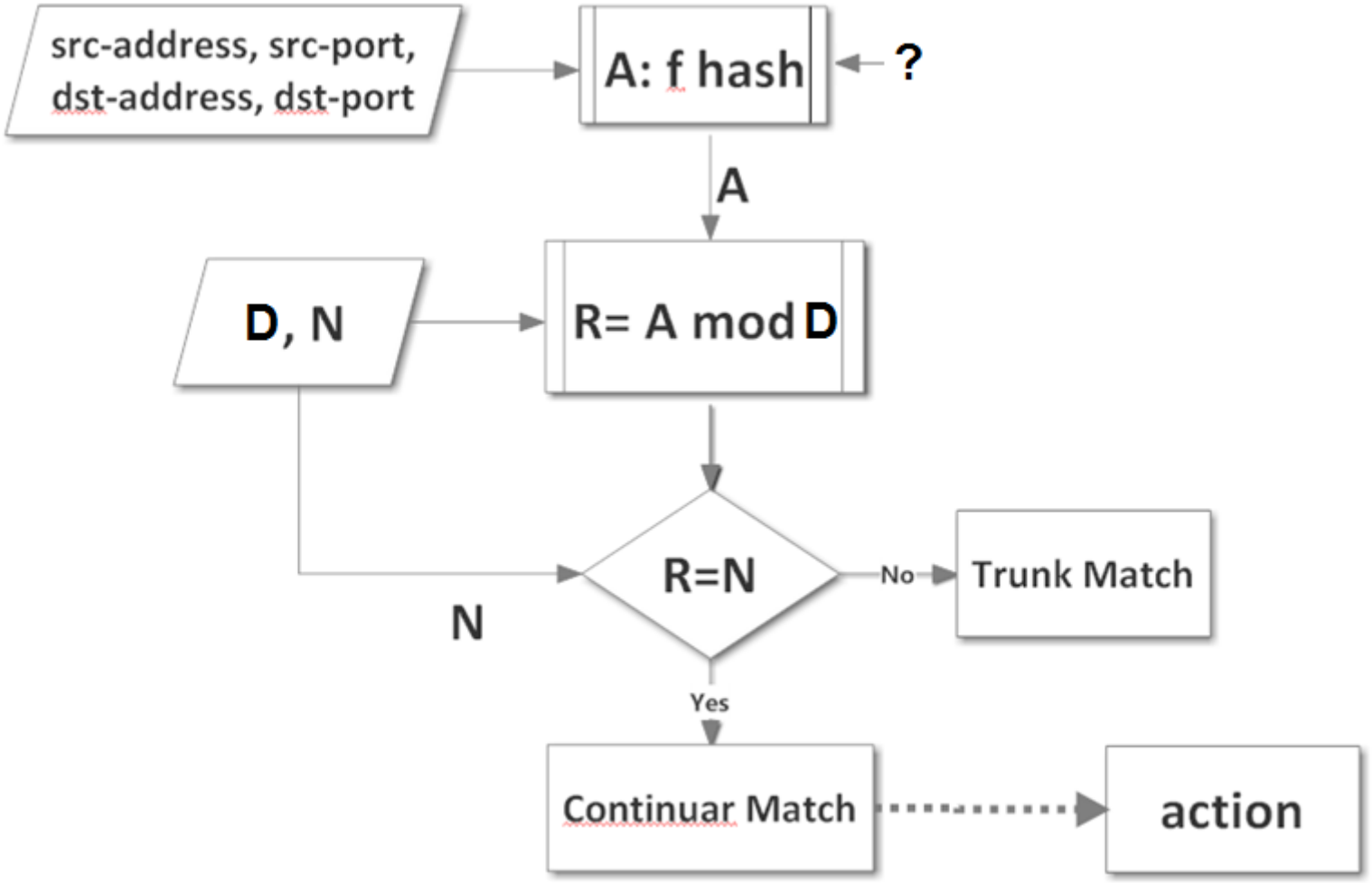
$$R = A\ \boldsymbol{mod}\ B \text{ o tambien } \mathrm{R} = \mathrm{A}\ \%\ \mathrm{B}$$

- Ejemplos:

  127 **mod** 2 = 1

  151567 **mod** 5 = 2

PCC: Funcionamiento (match)

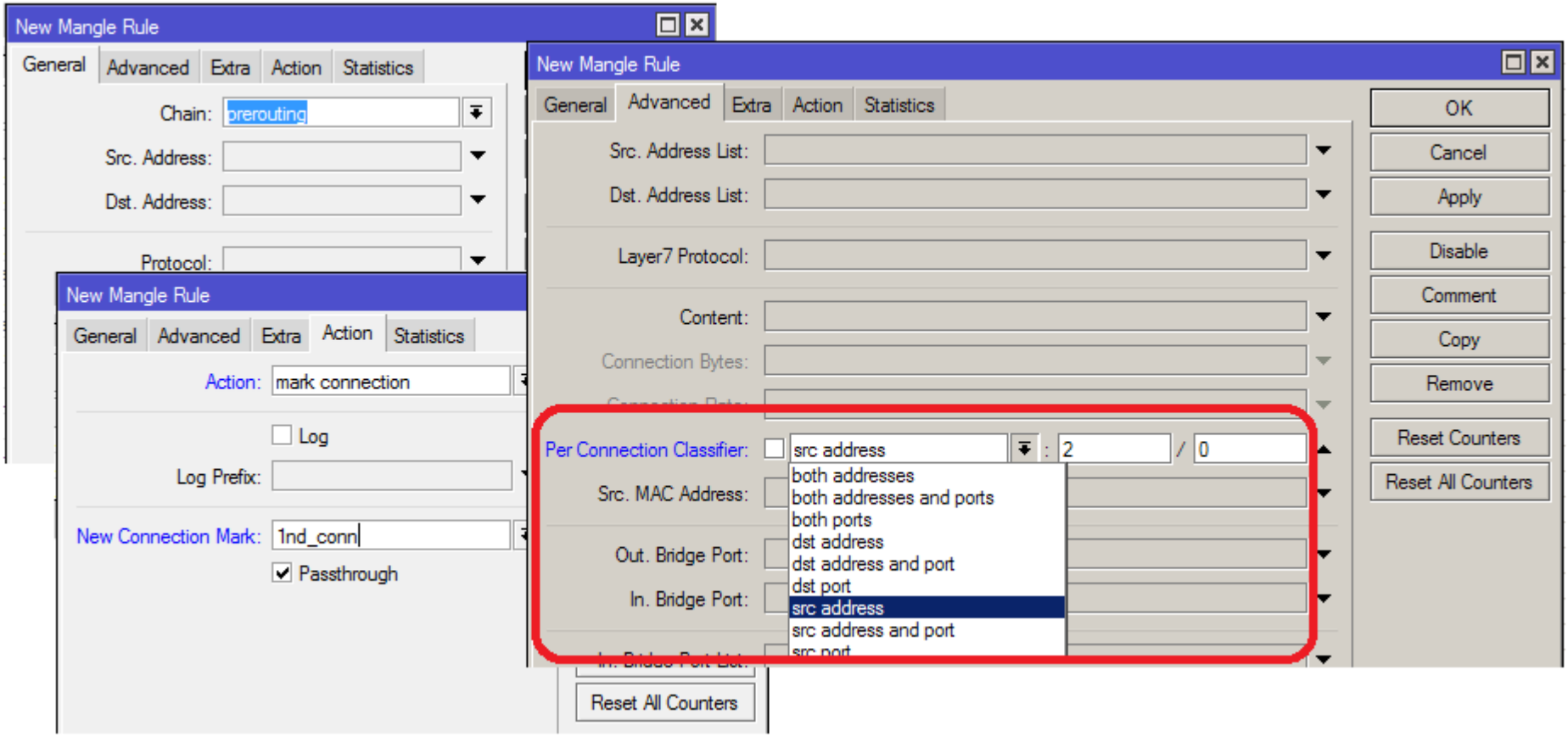


PCC: ValuesToHash

```
per-connection-classifier=
PerConnectionClassifier ::= [!]ValuesToHash:Denominator/Remainder
  Remainder ::= 0..4294967295 (integer number)
  Denominator ::= 1..4294967295 (integer number)
  ValuesToHash ::= both-addresses|both-ports|dst-address-and-port| src-address|
  src-port|both-addresses-and-ports|dst-address|dst-port|src-address-and-port
```

- **both-addresses|both-ports|dst-address-and-port|src-address|src-port|both-addresses-and-ports|dst-address|dst-port|src-address-and-port**

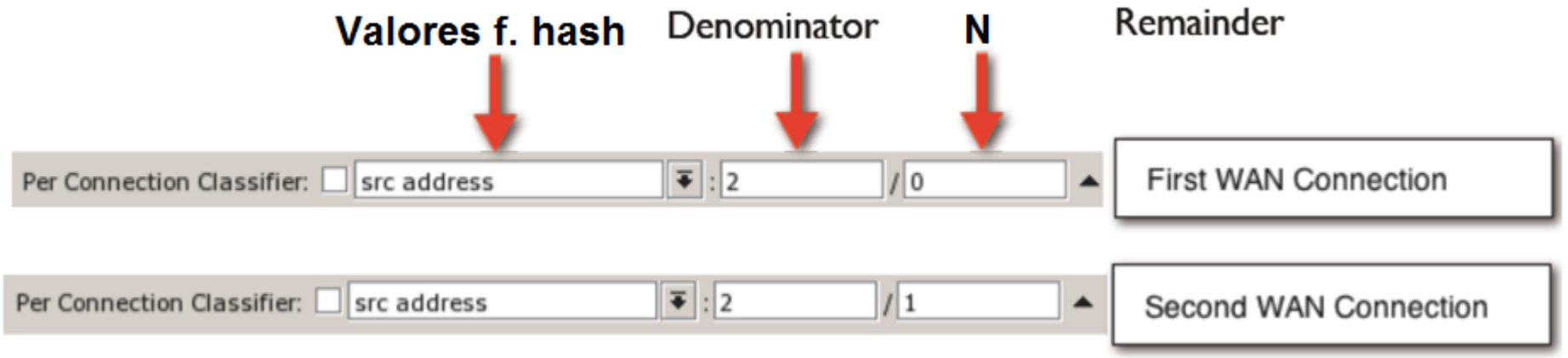
PCC: Dónde se encuentra?




```
> ip firewall mangle add chain=prerouting action=mark-connection new-connection-mark=2nd_conn per-connection-classifier=src-address:2/0
```

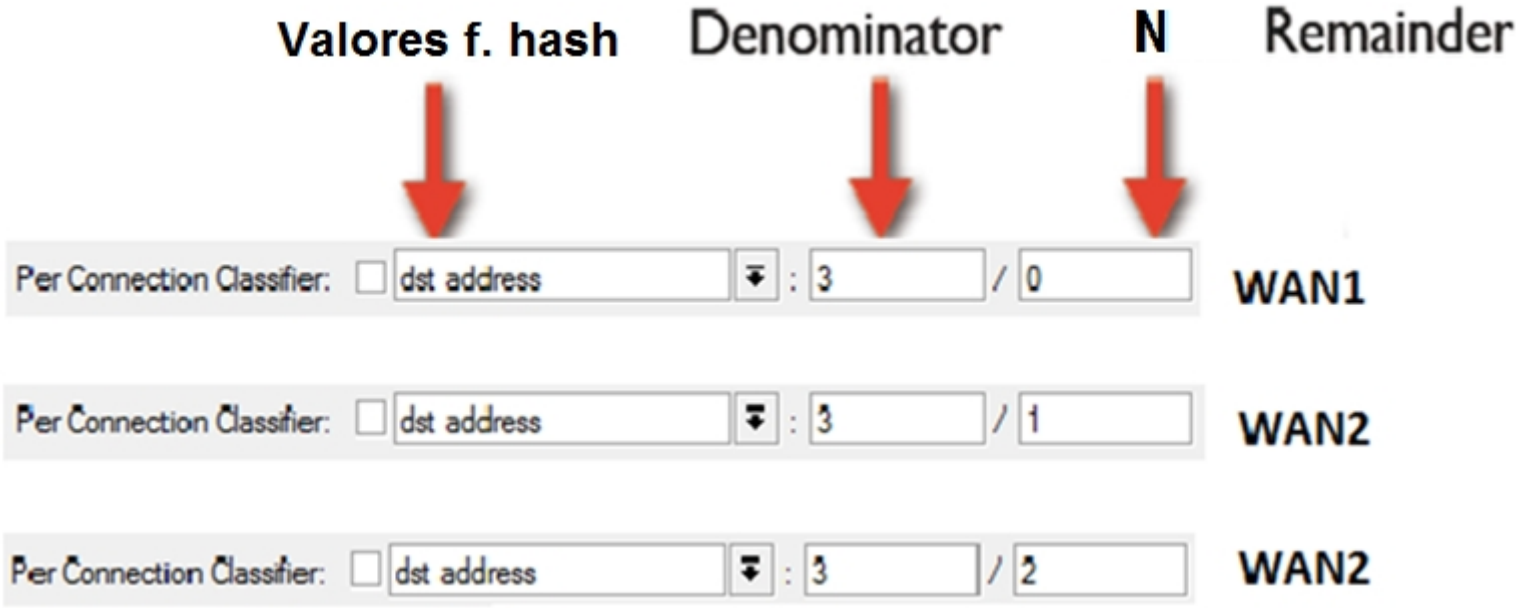
PCC: Ejemplo 1

- Clasificación con 2 flujos simétricos requiere 2 reglas:



PCC: Ejemplo 2

- Para clasificación de 2 flujos asimétricos de 1M y 2M se requerirán 3 reglas:



PCC: Ejemplo 2

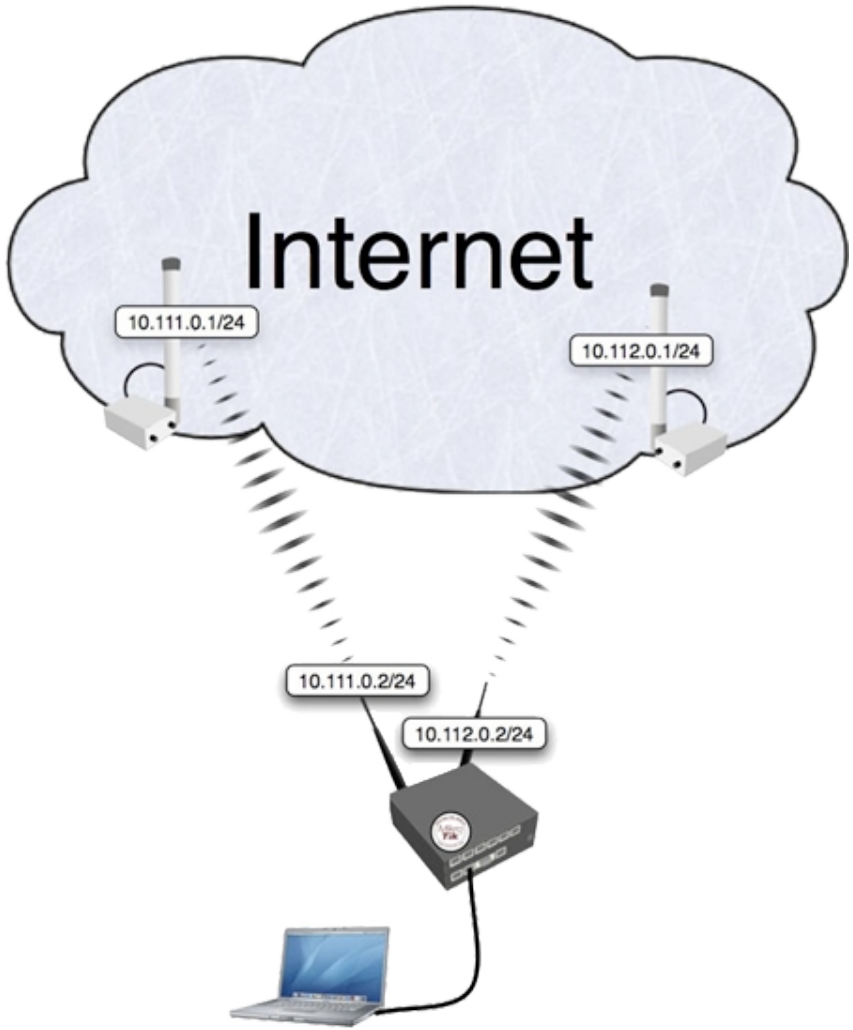
- En general si se tienen “n” flujos de a_1, a_2,... , a_n velocidades siendo a_i enteros.
- Se tendrán m reglas donde $m=(a_1+a_2+... +a_n)/D$ con $D=MCD(a_1, a_2,... , a_n)$.
- Cada flujo tendrá a_i/D reglas (i=1,2,…,n)
- Se deben realizar consideraciones prácticas para no incrementar el número de reglas (mayor tiempo de procesamiento)

Ejercicio

- Se tiene 3 WAN con 4,5Mbps, 3Mbps y 9Mbps. ¿Cuántas reglas recomendaría para clasificar el tráfico con PCC y cuáles para cada WAN? .
- Similar considerando 7 Mbps, 6 Mbps y 3 Mbps respectivamente.

Balanceo de carga usando PCC

- Escenario:



Pasos Requeridos

- Configuración base:
  - Direcciones IP.
  - Default gateway**’s**.
  - Enmascaramiento.
- Configuración Mangle.
- Configuración tablas de enrutamiento.

Configuración Base

```
/interface ethernet
set 0 name=ISP1
set 1 name=ISP2
set 4 name=LAN

/ip address
add address=192.168.0.1/24 interface=LAN
add address=10.111.0.2/24 interface=ISP1
add address=10.112.0.2/24 interface=ISP2

/ip route
add gateway=10.111.0.1 distance=1 check-gateway=ping
add gateway=10.112.0.1 distance=2 check-gateway=ping

/ip firewall nat
add chain=srcnat out-interface=ISP1 action=masquerade
add chain=srcnat out-interface=ISP2 action=masquerade
```

Políticas de Enrutamiento

- Método que permite crear políticas distintas para tráficos diferenciados mediante tablas de enrutamiento personalizadas.
- En RouterOS las tablas de enrutamiento se crean:
  - Por cada tabla especificada en **`/ip route rule`**.
  - Por cada **`routing-mark`** en mangle.
- El tráfico “marcado” es asignado de forma automática a la tabla de enrutamiento correspondiente.

routing-mark

- Atributo asignado a cada paquete (seguido en conntrack).
- **`routing-mark`** se cambia con un `action` en `/ip firewall mangle` antes de cualquier decisión de enrutamiento:
  - **chain prerouting** para todo el tráfico entrante.
  - **chain output** para el tráfico saliente del router.
- Cada nueva **`routing-mark`** tiene su propia tabla de enrutamiento del mismo nombre.
- De forma predeterminada, todos los paquetes tienen la marca de enrutamiento **`main`**.

Mangle: Tráfico a redes conectadas

- Las rutas conectadas sólo están disponibles en la tabla “main”.
- Debemos asegurarnos que el tráfico a estas redes permanezca en la tabla “main” (“no-mark”).
- Esto permite la comunicación adecuada entre clientes conectados local y remotamente.

```
/ip firewall mangle
add chain=prerouting dst-address=10.111.0.0/24 action=accept in-interface=LAN
add chain=prerouting dst-address=10.112.0.0/24 action=accept in-interface=LAN
```

Mangle: Conexiones remotas

- Conexiones iniciadas desde una interface pública, deben responderse por esa misma interface.
- Marcar estas conexiones para luego ponerlas en la tabla de enrutamiento adecuada (**`chain output`**).

```
/ip firewall mangle
add chain=prerouting in-interface=ISP1 connection-mark=no-mark
action=mark-connection new-connection-mark=ISP1_conn
add chain=prerouting in-interface=ISP2 connection-mark=no-mark
action=mark-connection new-connection-mark=ISP2_conn
```

Mangle: PCC

- Para reducir el consumo de recursos y el tiempo de procesamiento se identifican las conexiones (`no-mark` o `new`).
  - Primero se identifica el tráfico (PCC) y con **`action`** se marca la conexión.

```
/ip firewall mangle
add chain=prerouting  in-interface=LAN 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=LAN connection-mark=no-mark dst-address-type=!local
per-connection-classifier=both-addresses:2/1 action=mark-connection new-connection-mark=ISP2_conn
```

Mangle: Marcado de rutas

- Segundo, en cada paquete de la conexión se procede al `mark-routing`.

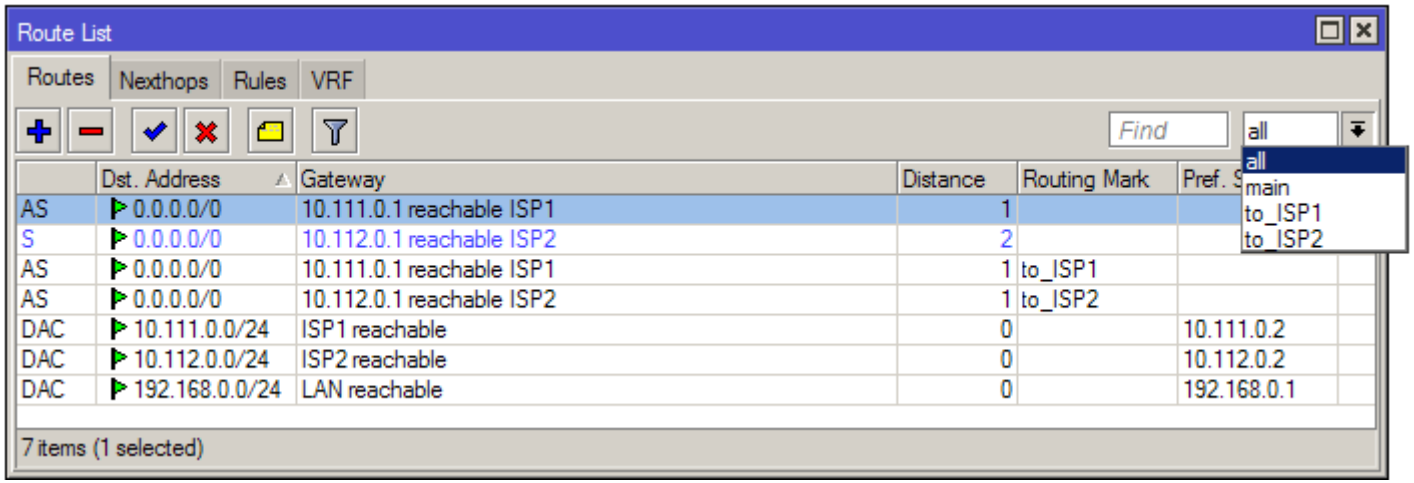
```
/ip firewall mangle
add chain=prerouting connection-mark=ISP1_conn in-interface=LAN action=mark-routing
new-routing-mark=to_ISP1
add chain=prerouting connection-mark=ISP2_conn in-interface=LAN action=mark-routing
new-routing-mark=to ISP2
```

```
/ip firewall mangle
add chain=output connection-mark=ISP1_conn action=mark-routing new-routing-mark=to_ISP1
add chain=output connection-mark=ISP2 conn action=mark-routing new-routing-mark=to ISP2
```

Route: Tabla de Enrutamiento

- Se agrega las rutas por defecto en la tabla de enrutamiento correspondiente:

```
/ip route
add dst-address=0.0.0.0/0 gateway=10.111.0.1 routing-mark=to_ISP1 check-gateway=ping
add dst-address=0.0.0.0/0 gateway=10.112.0.1 routing-mark=to_ISP2 check-gateway=ping
```



Consideración de excepciones

- De acuerdo a la configuración propuesta en PCC, se aplican políticas de enrutamiento especificas para los paquetes originados en la LAN, esto puede traer problemas.
- Un enfoque simple y extensible consiste en crear primero una lista de direcciones que contenga todas las redes locales, incluidas las redes de enlace WAN, y luego aceptar todo el tráfico entre redes en la lista de direcciones.

Solución: excepciones con `accept`

- Address list:

```
/ip firewall address-list
add list=local-networks address=10.111.0.0/24
add list=local-networks address=10.112.0.0/24
add list=local-networks address=192.168.0.0/24
add list=local-networks address=172.16.0.0/24
```

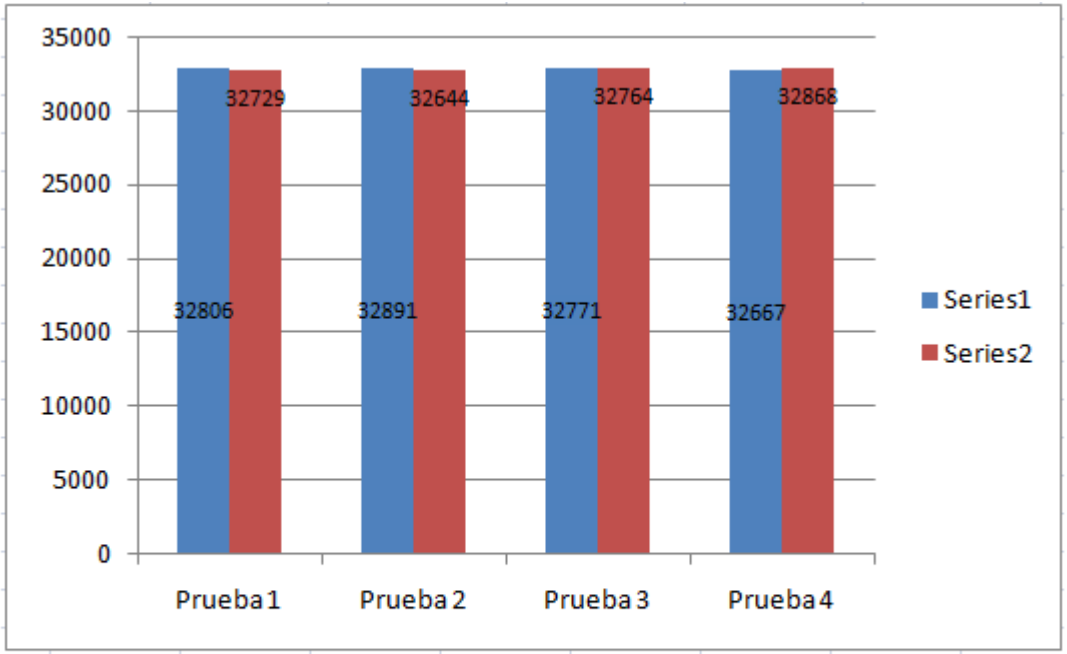
- Reemplazar las reglas accept por:

```
/ip firewall mangle
add chain=prerouting src-address-list=local-networks dst-address-list=local-networks action=accept
```

Route: Fail-Over

- Ciertamente no basta el **`check-gateway`**.
- El uso de scripts para este caso puede complejizar la configuración.
- Usar rutas recursivas, esta suficientemente utilizado y documentado y escapa a esta presentación por tiempo.

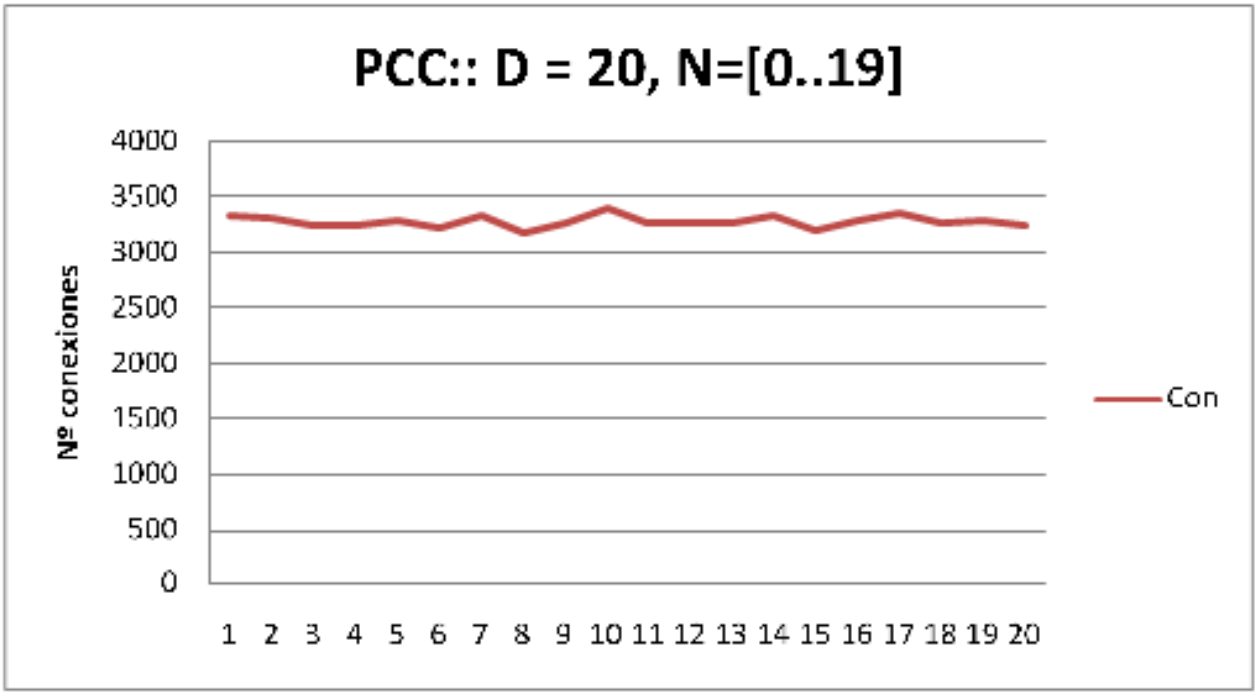
Análisis: D=2, dst-port, 4 test



	Prueba 1		Prueba 2		Prueba 3		Prueba 4	
Regla	N	%	N	%	N	%	N	%
0	32806	50.0587	32891	50.1884	32771	50.0053	32667	49.84665
1	32729	49.9413	32644	49.8116	32764	49.9947	32868	50.15335
Total	65535	100	65535	100	65535	100	65535	100

Análisis: D=20, dst-port, 1 test

N	Con	%
0	3334	5.08736
1	3307	5.04616
2	3232	4.93172
3	3235	4.93629
4	3280	5.00496
5	3226	4.92256
6	3330	5.08125
7	3168	4.83406
8	3260	4.97444
9	3403	5.19265
10	3272	4.99275
11	3256	4.96834
12	3273	4.99428
13	3328	5.07820
14	3185	4.86000
15	3292	5.02327
16	3364	5.13313
17	3259	4.97292
18	3295	5.02785
19	3236	4.93782
	65535	100



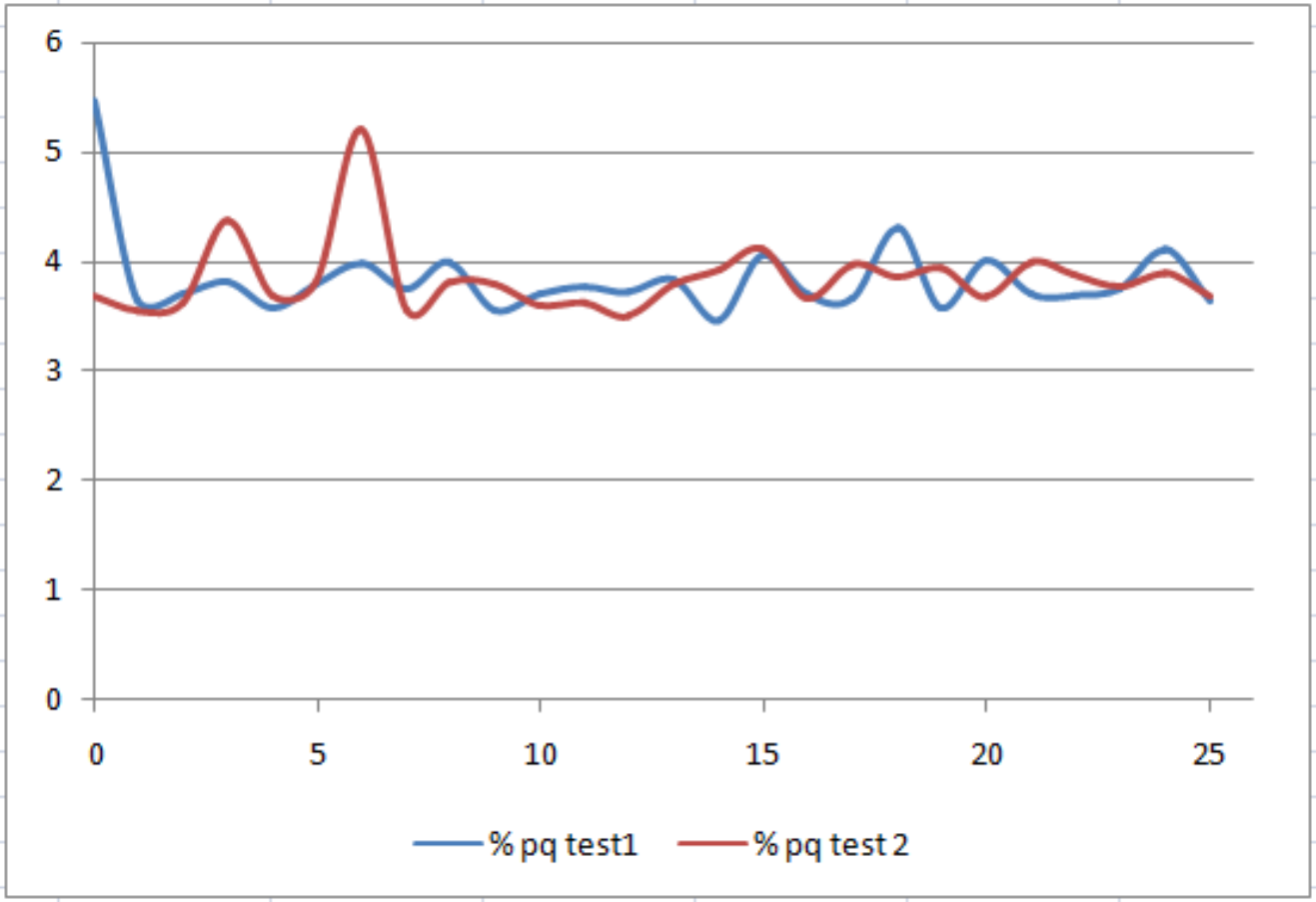
Análisis: D=3. both addresses. real

#		Action	Chain	Src. Address	Ds...	In. Inter...	Per Connection Cl...	Bytes	Packets
0		acc...	prerouting		19...	bridge1		5.7 MiB	100 443
1		acc...	prerouting		19...	bridge1		0 B	0
2		acc...	prerouting		19...	bridge1		222.2 KiB	2 894
3		mar...	prerouting			wan1		1322.5 KiB	12 206
4		mar...	prerouting			wan2		1828.9 KiB	14 581
5		mar...	prerouting			wan3		1701.1 KiB	12 667
6		mar...	prerouting			bridge1	both addresses:3/0	36.5 MiB	448 930
7		mar...	prerouting			bridge1	both addresses:3/1	26.7 MiB	317 490
8		mar...	prerouting			bridge1	both addresses:3/2	44.9 MiB	572 889
9		mar...	prerouting			bridge1		2504.4 MiB	23 305 949
10		mar...	prerouting			bridge1		3478.6 MiB	24 292 316
11		mar...	prerouting			bridge1		3049.3 MiB	31 105 782
12		mar...	output					283.3 KiB	3 258
13		mar...	output					203.4 KiB	2 384
14		mar...	output					3288.4 KiB	11 356

Terminal

```
                name:          ether5          lan          wan1          wan2          wan3
      driver-rx-byte:  10 681 242 729  40 634 313  51 964 215 495  45 031 310 159  63 488 593 431
    driver-rx-packet:      78 828 862      532 191      40 831 842      35 967 617      49 097 726
      driver-tx-byte: 160 461 318 638 236 651 330   3 046 763 935   3 899 861 302   3 644 366 769
    driver-tx-packet:     125 791 693      233 386      24 585 954      23 394 785      31 170 149
            rx-bytes:  10 996 623 426  42 762 537  52 146 055 818  45 330 967 419  63 703 332 802
          rx-packet:      78 826 500            0      40 844 155      36 071 807      49 110 440
       rx-too-short:               0            0               0               0               0
             rx-64:      17 723 660           55         871 554         856 956       1 055 632
          rx-65-127:      52 625 487      462 279       2 954 471       2 465 198       3 392 013
        rx-128-255:       3 424 149       69 249         952 493       1 690 685       1 172 658
        rx-256-511:       1 045 183          529         530 421         497 791         537 578
       rx-512-1023:       1 335 518           17         546 128         663 452         410 594
      rx-1024-1518:       2 675 405           54      34 989 238      29 897 903      42 542 145
        rx-too-long:               0            0               0               0               0
      rx-broadcast:             998      525 730               0               1               0
          rx-pause:               0            0               0               0               0
      rx-multicast:           1 904        6 453             148             177             180
-- [Q quit|D dump|down]
```

Análisis: D=25, both addresses, real



Rememoremos

- Balanceo de Carga.
- Diferentes problemas-diferentes soluciones.
- PCC solución propietaria de Mikrotik.
- Análisis.

Material de Consulta

- Manual:PCC - MikroTik Wiki. http://wiki.mikrotik.com/wiki/Manual:PCC
- Load Balancing Using PCC & RouterOS - MUM US12 - MikroTik, Steve Discher.
- MikroTik RouterOS Workshop Load Balancing Best Practice. Las Vegas, MUM USA 2011, Jānis Meģis

Gracias...



“

El bosque no sería el mismo si sólo cantaran los pájaros que mejor lo hacen.

Rabindranath Tagore; *premio Nobel de Literatura 1913.*