

Ruteo *Avanzado*

CDD2

iproute2

- Conjunto integrado de utilitarios para administrar el networking de Linux
- Reemplaza a net-tools (ifconfig, arp, route)
- Más eficiente (acceso al kernel a través de sockets netlink)
- Separa los diferentes conceptos (objetos) relativos a networking (direcciones, dispositivos, rutas, etc)
- Sintaxis común para todos los objetos
- Incorpora conceptos nuevos (p. ej. Túneles)
- Soporta balanceo de carga, control de tráfico, etc
- Documentacion: Linux Advanced Routing and Traffic Control, <https://www.lartc.org/lartc.pdf>

Comandos iproute2 (1)

link (man ip link)

- un link es un dispositivo de red; puede ser físico o lógico
- tiene solo dos comandos: show y set (para configurarlo)

address (man ip address)

- direcciones del protocolo de nivel 3 (ipv4 o ipv6) asignadas a los dispositivos
- permite agregar, eliminar o modificar direcciones

neighbour (man ip neighbour)

- cache conteniendo información relativa a los nodos vecinos (en la misma red física) para cada dispositivo existe una tabla
- permite agregar, borrar y cambiar parámetros de las entradas

ntable (man ip ntable)

- permite ver y cambiar parámetros de las tablas con info de los nodos vecinos (neighbour)

token (man ip token)

- permite especificar el host ID para configuración SLAAC (radvd)

monitor (man ip monitor)

- muestra la actividad en los diferentes objetos (rutas, dispositivos, etc)

Comandos iproute2 (2)

route (man ip route)

- entradas en las tablas de ruteo
- permite agregar, borrar y modificar las rutas que conoce el equipo

rule (man ip rule)

- reglas en la base de datos de políticas de ruteo

maddress (man ip maddress)

- permite administrar las direcciones multicast asociadas al equipo

mroute (man mroute)

- cache de ruteo multicast
- permite ver las entradas que se utilizan cuando el equipo actúa como router (multicast)
- para eso debe estar corriendo algún protocolo de ruteo multicast, como pimd, etc.
- por el momento, solo es posible verlas pero no modificarlas ni agregar.

tcp_metrics (man tcp_metrics)

- permite ver y borrar las entradas relativas a las conexiones TCP mantenidas por el kernel
- contienen parámetros específicos TCP

tunnel (man ip tunnel)

- permite crear y administrar varios tipos de túneles

Tipos de rutas (1)

Unicast

- Indica el camino (nexthop) a destinos reales
- Por defecto una ruta es unicast
 - `ip -6 route add 2001:db8:1::2/64 dev eth0`
 - `ip -6 route add unicast 2001:db8:1::2/64 via 2001:db8:3::4`

Unreachable

- Para cualquier dirección comprendida en la red, se elimina el paquete y se envía al origen un ICMP host unreachable.
 - `ip -6 route add unreachable 2001:db8:8::/64`

Blackhole

- Para cualquier dirección comprendida en la red, se elimina el paquete sin enviar ningún aviso
 - `ip -6 route add blackhole 2001:db8:8::/64`

Prohibit

- Para cualquier dirección comprendida en la ruta, se elimina el paquete y se envía al origen un ICMP communication administratively prohibited
 - `ip -6 route add prohibit 2001:db8:8::/64`

Tipos de rutas (2)

Local

- Se refiere a destinos en el mismo equipo. Los paquetes son reenviados al mismo host
- Tener en cuenta: tabla local diferente a ruta local
- Solo válido en IPv6
- Para cualquier tipo de ruta se puede especificar un host -prefijo 128 - o ausencia de prefijo
 - `ip -6 route add local 2001:10::/64 dev eth0` – agrega ruta local en tabla local
 - `ip -6 route add 2001:10::/64 dev eth0 table local` - agrega una dirección unicast en tabla local
 - `ip -6 route add local 2001:10::/64 dev eth0 table local` – agrega ruta local en tabla local
 - `ip -6 route add local 2001:10::/64 dev eth0 table 100` – agrega ruta local en tabla 100

Broadcast

- Indica los destinos que deben ser enviados como broadcast en la red física (link layer).
- Solo válido en IPv4
 - `ip route add broadcast 10.1.1.0/24 dev eth0` – agrega por defecto en tabla local
 - `ip route add broadcast 10.1.1.0/24 dev eth0 table 100` – agrega en tabla 100

Tipos de rutas (3)

Anycast

- Son direcciones unicast que serán utilizadas como anycast. La diferencia es que no se pueden utilizar como direcciones de origen.
- No se pueden definir desde ip2
- Se generan automáticamente en el router, al definir una dirección (y su red)
 - `ip -6 addr add 2001:999::3/64 dev eth0` generará la entrada en la tabla local
 - `anycast 2001:999:: dev eth0 proto kernel scope global metric 0 pref medium`

throw

- Ruta de control utilizada en conjunto con policy routing.
- Si un destino coincide con una ruta de tipo throw, el efecto es el mismo que si no hubiera ruta para ese destino
- Si no usamos policy routing, se descarta el paquete y se genera ICMP net unreachable
- Si usamos policy routing, se evalúa la siguiente regla en la RPDB
- `ip -6 route add throw 2001:9898::/64`

Políticas de ruteo

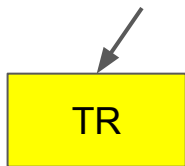
Ruteo estándar

- Selección del next-hop en función de la dirección de destino del paquete (longest match prefix)
- No permite seleccionar rutas en base a otras características del tráfico

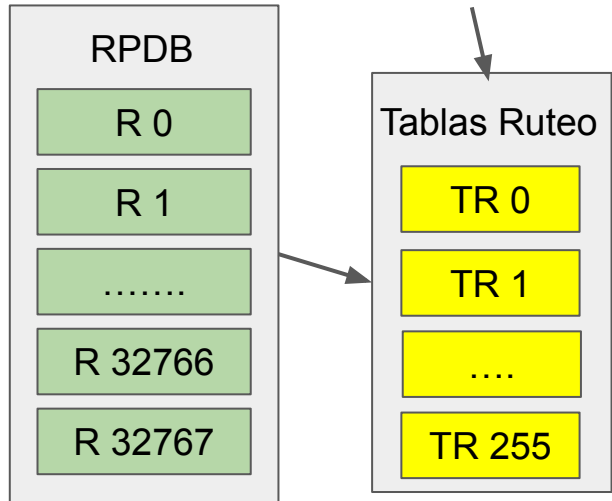
Ruteo avanzado

- Utilización de Routing Policy Data Base
- Permite elección de ruta según otras características de los paquetes Dirección de origen
 - Tipo de servicio
 - Tamaño del paquete
 - Tipo de tráfico
 - Protocolos de nivel aplicación
 - Otros

longest match prefix

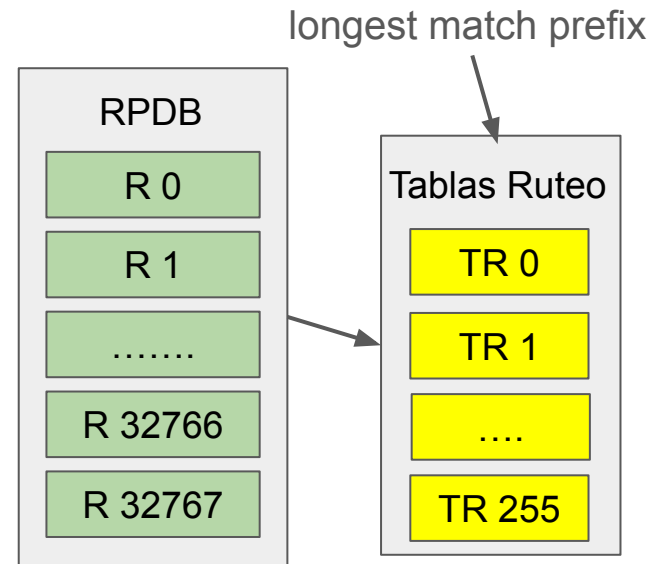


longest match prefix



Funcionamiento ruteo avanzado

- Componentes
 - Routing Policy Data Base
 - reglas
 - tablas de ruteo
- Procedimiento
 - Chequear reglas en orden de prioridad (0 a 32767)
 - Si matching, decidir según la regla y terminar
- Condiciones de matching
 - Tipo de Servicio
 - Dir de origen y/o destino
 - Otros
- Posibles decisiones de las reglas
 - Utilizar una tabla
 - Blackhole
 - Otros



Reglas en la RPDB

- Formato Regla: selectores target
- Selectores (sobre campos del paquete y marks):
 - Dir origen (from)
 - Dir destino (to)
 - Interfaz de entrada (iif)
 - Interfaz de salida (oif)
 - Tipo de servicio (tos, dsfield)
 - fwmark
- Targets (Tipos de reglas)
 - Unicast: la ruta al destino se busca en la tabla indicada
 - Blackhole: se descarta el paquete sin enviar ninguna respuesta
 - Unreachable: se genera un error "Network unreachable"
 - Prohibit: se genera un error 'Communication is administratively prohibited'
- Ejemplos
 - `ip rule add blackhole to 10.182.17.64/28`
 - `ip rule add prohibit from 209.10.26.51`
 - `ip rule add unreachable iif wan0 fwmark 5` (Previamente: `iptables -t mangle -A INPUT -m mark --mark 5`)
 - `ip rule add unicast from 192.168.100.17 table 5`

Funcionamiento de ruteo avanzado

ip -6 rule add priority 100 from 2001:2::/64 table 25

ip -6 route add default via fd::1 table 25

ip -6 route add default via fd::2

TR 25 (usuario)


Default via fd::1

TR 254 (main)

Default via fd::2

REGLA	CONDICIÓN	ACCIÓN
0	Todos los datagrams	Consultar tabla de ruteo local (255)
1	Condición	Acción
...
100	ip.src == 2001:2::/64	Consultar tabla de ruteo 25
...
32765	Condición	Acción
32766	Todos los datagrams	Consultar tabla de ruteo main (254)
32767	Todos los datagrams	Consultar tabla de ruteo default (253)

Reglas y tablas de ruteo (v4)



vcmd

Archivo Editar Pestañas Ayuda

```
root@n1:/tmp/pycore.46233/n1.conf# ip rule show
0:      from all lookup local
32766:  from all lookup main
32767:  from all lookup default

root@n1:/tmp/pycore.46233/n1.conf# ip route show table local
broadcast 10.0.0.0 dev eth0 proto kernel scope link src 10.0.0.1
local 10.0.0.1 dev eth0 proto kernel scope host src 10.0.0.1
broadcast 10.0.0.255 dev eth0 proto kernel scope link src 10.0.0.1
broadcast 127.0.0.0 dev lo proto kernel scope link src 127.0.0.1
local 127.0.0.0/8 dev lo proto kernel scope host src 127.0.0.1
local 127.0.0.1 dev lo proto kernel scope host src 127.0.0.1
broadcast 127.255.255.255 dev lo proto kernel scope link src 127.0.0.1

root@n1:/tmp/pycore.46233/n1.conf# ip route show table main
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.1

root@n1:/tmp/pycore.46233/n1.conf# ip route show table default
root@n1:/tmp/pycore.46233/n1.conf#
```

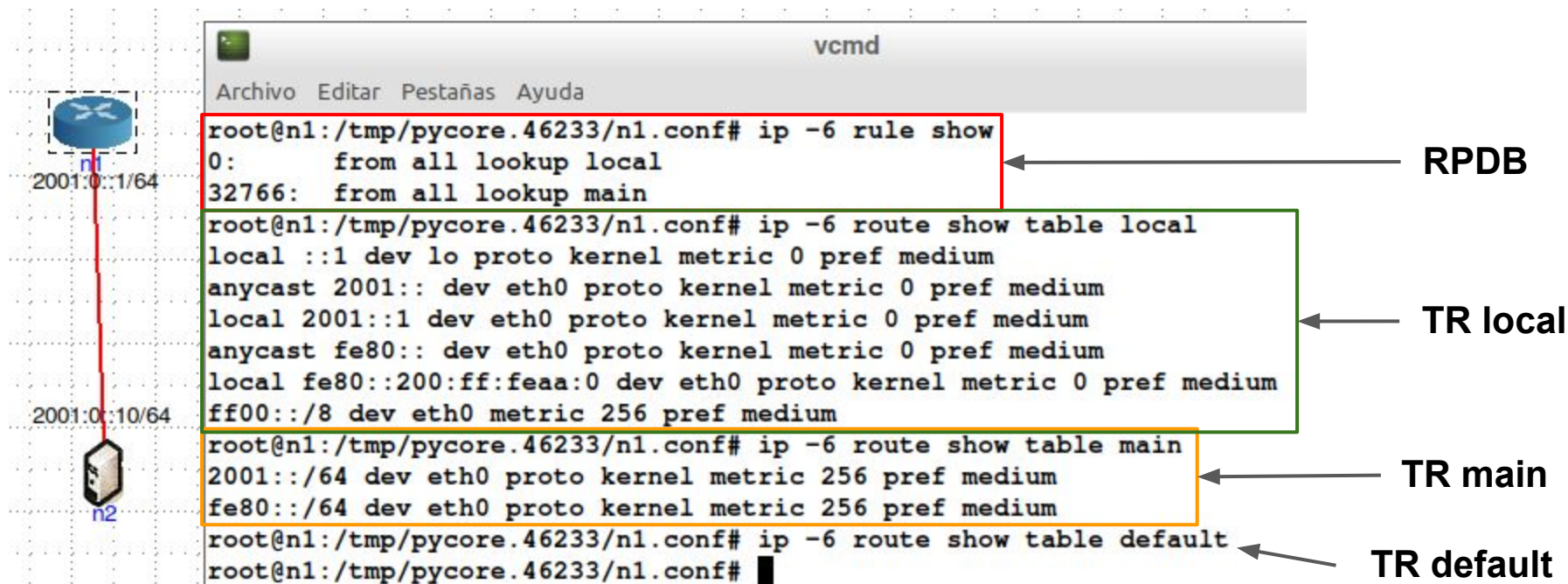
RPDB

TR local

TR main

TR default

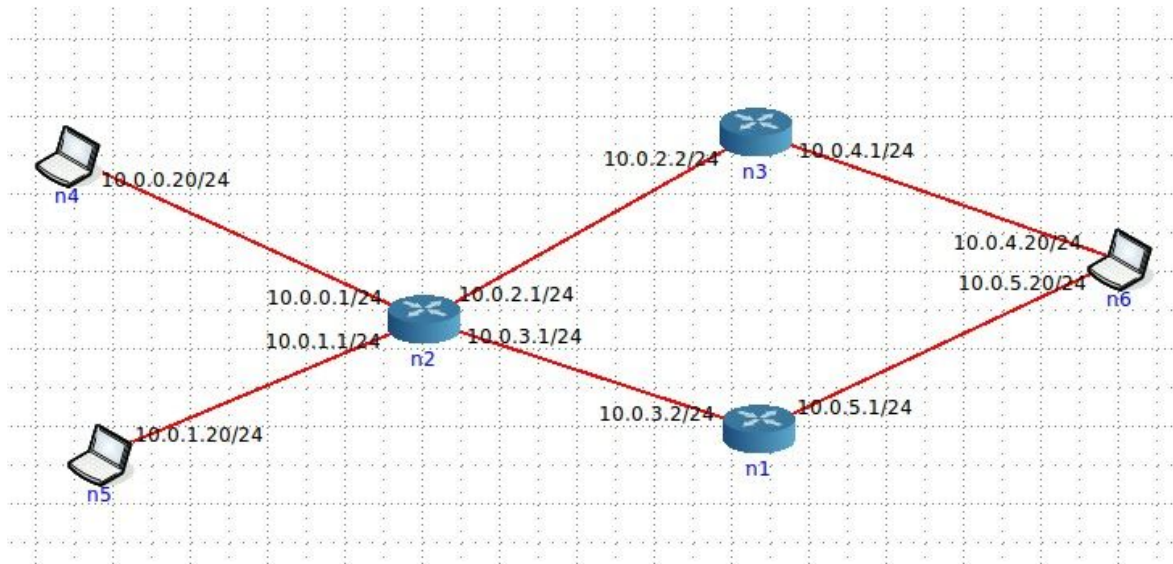
Reglas y tablas de ruteo (v6)



Ejemplo de uso de policy routing (v4)

Ruteo en n2

- usar ruta por defecto 10.0.2.2 (n3)
- si recibe de la red 10.0.0.0/24, usar 10.0.3.2 (n1)



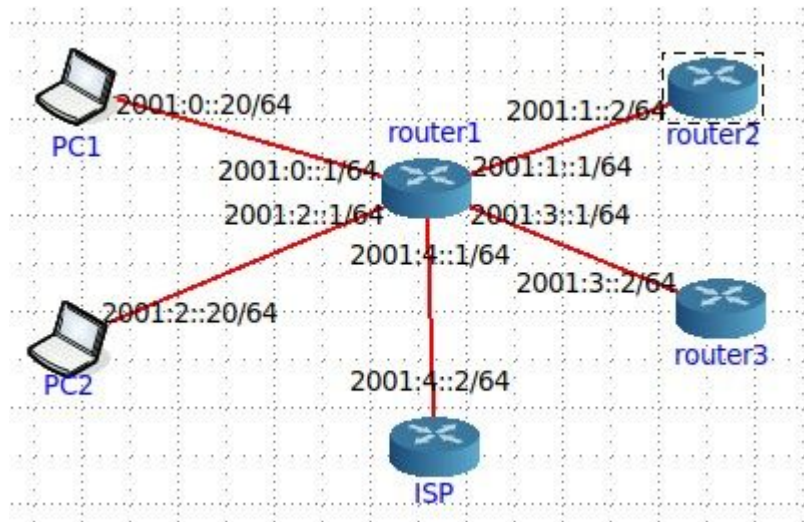
- cargar default route en tabla main, vía n3 (ip route add default via 10.0.2.2)
- en tabla 100, poner ruta por defecto vía n1 (ip route add default via 10.0.3.2 table 100)
- crear una regla de ruteo: usar tabla 100 si el origen es red 10.0.0.0/24 (ip rule add from 10.0.0.0/24 table 100)

Ejemplo de uso de policy routing (v6)

Si los datagrams provienen de la red 2001::/64, salen por router2 (2001:1::2)

Si los datagrams provienen de la red 2001:2::/64 salen por router3 (2001:3::2)

El resto de los datagrams, salen por el ISP (2001:4::2)



RPDB

R 0: para cualquier datagram, utilizar tabla 255 (local)

R 100: si dirección de origen es 2001::/64, utilizar tabla 10

R 120: si dirección de origen es 2001:2::/64 utilizar tabla 11

R 32766: para cualquier datagram, utilizar tabla 254 (main)

```
ip -6 rule add unicast pri 100 from 2001::64 table 10
```

```
ip -6 rule add unicast pri 120 from 2001:2::64 table 11
```

Tablas de Ruteo

T 10: ruta por defecto, via 2001:1::2

T 11: ruta por defecto via 2001:3::2

T 254: ruta por defecto via 2001:4::2

```
ip -6 route add default via 2001:4::2
```

```
ip -6 route add default via 2001:1::2 table 10
```

```
ip -6 route add default via 2001:3::2 table 11
```