Policy Routing

hat man mehrere Wege kann man über rules definieren, welchen Weg ein Paket nehmen soll. In meinem Fall habe ich mit 2 Internet-Anschlüssen (telekom+bambit) rumprobiert

separate Routingtabelle in /etc/iproute2/rt_tables anlegen (am Ende hinzufügen)

```
1 telekom
2 bambit
```

PPP-UP-Script (erstellt default-Routen in separater Routing-Tabelle für ppp-Verbindung)

```
ip route flush table bambit

ip route add default via 192.168.178.1 dev wan table bambit
ip rule add from 192.168.178.10 lookup bambit

#route specific devices
ip rule add from 192.168.0.80/32 table bambit
ip rule add from 192.168.0.26/32 table bambit

#exit on local adresses
ip route add throw 192.168.10.0/24 table bambit
ip route add throw 192.168.11.0/24 table bambit
ip route add throw 192.168.0.0/24 table bambit

ip route flush cache
```

auch sollte man nicht den DNS des Providers nehmen, da dieser ggf. vom anderen Provider keine DNS-Auflösung macht. Diese Erfahrung habe ich gemacht...anpingbar war er, aber es gab keine DNS-Auflösung. Dazu in der peers-Datei (/etc/ppp/peers/dateiname) die option usepeerdns ausschalten.

Load-Balancing

Loadbalancing mit NAT hat ein paar Probleme besonders im Zusammenhang mit Captchas, da diese oft an eine IP-Adresse gebunden sind und es sein kann, dass die nachfolgende Verbindung nicht den gleichen Weg nimmt (=andere IP-Quell-Adresse). Der nachfolgende Teil dokumentiert nur meinen bisherigen Ansatz und ist nicht für den Produktiveinsatz bestimmt.

der zweite step ist, dass ankommender Traffic markiert wird um die Antwort zum gleichen interface wieder raus zu schicken. Das ist wichtig, da der client die Antwort von der Adresse erwartet an die er die Anfrage geschickt hat. Weiterhin existiert bei vielen Providern ein sog. Reverse Path Filtering, es wird also geschaut, ob die Quelladresse zum sendenden Host passt. Das ist besonders bei lokalem Traffic wichtig, da dieser nicht in der Prerouting-Chain landet sondern erst das NAT greift (default route) und dann nur in der Output Chain greifbar ist.

```
wan1=ppp8
wan2=ppp0

iptables -A PREROUTING -t mangle -j CONNMARK --restore-mark
#iptables -A PREROUTING -t mangle --match mark --mark 1 -j ACCEPT
iptables -A PREROUTING -t mangle -i $wan1 -j MARK --set-mark 1
#iptables -A PREROUTING -t mangle --match mark --mark 2 -j ACCEPT
iptables -A PREROUTING -t mangle -i $wan2 -j MARK --set-mark 2

#ggf. weitere markings

iptables -A PREROUTING -t mangle -j CONNMARK --save-mark
```

danach kommt der interne Traffic, damit auch dieser durchgängig den gleichen Uplink verwendet (solange die TCP-Session besteht)

```
iptables -t mangle -N MARKING

iptables -A PREROUTING -t mangle -m mark --mark 0x0 -j MARKING #without
mark move to new chain

#for local packets to get in prerouting-chain
#needs sysctl -w net.ipv4.conf.lan0.rp_filter=0

iptables -t mangle -A MARKING -j MARK --set-mark 3 #bambit
iptables -t mangle -A MARKING -m statistic --mode random --probability 0.3
-j MARK --set-mark 4 #telekom
#iptables -t mangle -A MARKING -m mark --mark 3 -j LOG --log-prefix "fwmark
3: "
#iptables -t mangle -A MARKING -m mark --mark 4 -j LOG --log-prefix "fwmark
4: "

iptables -A PREROUTING -t mangle -j CONNMARK --save-mark
```

hier wird jdes 3. unmarkierte Paket mit 4 markiert, der Rest (die anderen 2) bleibt bei der vorher gesetzten Markierung 3

zum Schluss muss man sich noch um den lokal generierten Traffic kümmern (kein forwarded ⇒ kein prerouting). Diesen bekommt man nur in der OUTPUT-Chain (des ausgehenden Interfaces...hier meine 2 ppp) zu packen

```
iptables -t mangle -N MARKING_LOCAL

iptables -t mangle -A OUTPUT -j CONNMARK --restore-mark
iptables -t mangle -A OUTPUT -o ppp0 -m mark --mark 0x0 -j MARKING_LOCAL
#without mark move to new chain
iptables -t mangle -A OUTPUT -o ppp8 -m mark --mark 0x0 -j MARKING_LOCAL
#without mark move to new chain
iptables -t mangle -A MARKING_LOCAL -j HMARK --hmark-offset 3 --hmark-tuple
sport,dport --hmark-mod 2 --hmark-rnd 0xdeb1a4f0
#iptables -t mangle -A MARKING_LOCAL -m mark --mark 3 -j LOG --log-prefix
```

```
"fwmark 3 (l): "
#iptables -t mangle -A MARKING_LOCAL -m mark --mark 4 -j LOG --log-prefix
"fwmark 4 (l): "
iptables -t mangle -A MARKING_LOCAL -j CONNMARK --save-mark

iptables --table nat --append POSTROUTING --out-interface ppp8 -j
MASQUERADE
iptables --table nat --append POSTROUTING --out-interface ppp0 -j
MASQUERADE
```

nun kann man über „ip rule“ festlegen welcher traffic (nach Markierung) über welche Routing-Tabelle geschickt wird:

```
ip rule add fwmark 1 table telekom #incoming from telekom
ip rule add fwmark 2 table bambit #incoming from bambit
ip rule add fwmark 3 table telekom #outgoing
ip rule add fwmark 4 table bambit #outgoing
```