Poczyńmy kilka założeń: dysponujemy 2 łączami w przykładzie będą to popularne DSL 4196/512kb/s.
jedno z łącz obsługiwać będzie www i pocztę, www i pop3 generują niskie i sporadyczne obciążenie uploadu w które idealnie wpasuje się wysyłka poczty po smtp,
Po drugim łączu pójdzie wszystko inne, będzie to w konfiguracji łącze podstawowe.

Routing

Na sam początek ustawiamy routing, najwygodniej jest utworzyć sobie plik wykonywalny i dodać go do skryptów startowych by routing był ładowany automatycznie na starcie systemu, oraz by mieć możliwość wygodnego przeładowywania regułek w czasie jego pracy.

Poniżej plik ale przed jego uruchomieniem należy dodać do /etc/iproute2/rt_tables wpisy:
240 T1
241 T2

Przechodzimy do treści skryptu:

```
#!/bin/sh

PATH="/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin"

echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter

echo "Czyszczenie"

ip route flush table T1
ip route flush table T2
ip rule del fwmark 1 table T1
ip rule del fwmark 2 table T2

echo "dsl #1"
ip route add 192.168.0.0/16 dev eth1 table T1 # sieć lokalna 192.168.0.0/16 na interfejsie eth1
ip route add default via X.X.X.X table T1 # brama dla DSL #1

echo "dsl #2"
ip route add 192.168.0.0/16 dev eth1 table T2 # sieć lokalna 192.168.0.0/16 na interfejsie eth1
ip route add default via Y.Y.Y.Y table T2 # brama dla DSL #2

echo "ip rule"
ip rule add fwmark 1 table T1
ip rule add fwmark 2 table T2

echo "przygotowanie iptables"

# zatrzymujemy niceshapera
niceshaper stop

iptables -t mangle -F PREROUTING

# markujemy numerkiem 1, wszystko co wpada do eth1 z naszej sieci lokalnej
iptables -t mangle -A PREROUTING -i eth1 -p tcp -s 192.168.0.0/16 -j MARK --set-mark 1

# nadpisujemy numerkiem 2, to co ma być routowane drugim łączem
iptables -t mangle -A PREROUTING -i eth1 -p tcp -s 192.168.0.0/16 --dport 80 -j MARK --set-mark 2
iptables -t mangle -A PREROUTING -i eth1 -p tcp -s 192.168.0.0/16 --dport 8080 -j MARK --set-mark 2
iptables -t mangle -A PREROUTING -i eth1 -p tcp -s 192.168.0.0/16 --dport 25 -j MARK --set-mark 2
iptables -t mangle -A PREROUTING -i eth1 -p tcp -s 192.168.0.0/16 --dport 110 -j MARK --set-mark 2

# ostatecznie jednak nadpisujemy numerkiem 1, to co nie będzie wychodziło w świat a wyżej mogło zostać oznaczone numerkiem 2
iptables -t mangle -A PREROUTING -i eth1 -p tcp -s 192.168.0.0/16 -d 192.168.0.0/16 -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -i eth1 -p tcp -s 192.168.0.0/16 -d X.X.X.X/m -j MARK --set-mark 1 # podsieć dsl #1
iptables -t mangle -A PREROUTING -i eth1 -p tcp -s 192.168.0.0/16 -d X.X.X.X/n -j MARK --set-mark 1 # podsieć dsl #2

# startujemy niceshapera
niceshaper start
```

Konfiguracja NiceShapera

Dla wygody do kontroli uploadu wykorzystamy interfejsy IMQ.
Jeśli sprzedajemy konkretne przepustowości downloadu, nie zwracamy uwagi na to którym łączem klient ściąga dane stąd konstruujemy pojedynczą sekcje dla downloadu i tam ustawiamy sztywne przydziały.
Jeśli zależy nam na maksymalnym wykorzystaniu łącz, konfigurujemy 2 sekcje downloadu, markujemy pakiety przychodzące ze świata i analogicznie jak z uploadem wrzucamy do odpowiedniej instancji, co nie powinno sprawić problemów.

Tak więc plik config:

```
<global>
  run download upload2 upload1
  stats unit kb/s file none
</global>

# upload przez DSL#1

<upload1>
  iface imq0 match srcip 192.168.0.0/16 mark 1
  section speed 64kB/s
  section shape 52kB/s
  default low 4kB/s
  default ceil 12kB/s
  default overtake 4kB/s
  default hold 90s
  default htb-scheduler sfq
```

```
  default imq-redirect
  mode upload
  reload 3s
</upload1>

# upload przez DSL#2.
# konfigurujemy troszkę większe przydziały,
# ruch www i poczty to szybkie wystrzały danych
# mniej konfliktowy i globalnie w mniejszym stopniu obciążający łącze niż ciągłe transfery przez ftp czy p2p.
# więc zapewniamy mu lepsze warunki.

<upload2>
  iface imq1 match srcip 192.168.0.0/16 mark 2
  section speed 64kB/s
  section shape 52kB/s
  default low 4kB/s
  default ceil 12kB/s
  default overtake 6kB/s
  default hold 90s
  default htb-scheduler sfq
  default imq-redirect
  mode upload
  reload 3s
</upload2>

# oczywiście sekcja dla downloadu.

<download>
  iface eth1 match dstip 192.168.0.0/16
  section speed 1024kB/s
  section shape 800kB/s
  default rate 64kB/s
  default overtake 20kB/s
  default htb-scheduler sfq
  default imq-redirect
  mode download
  reload 5s
</download>
```

Wycinek pliku class:

```
# Zyta
class download eth1 Zyta
  match dstip 192.168.0.10
class upload1 imq0 Zyta
  match srcip 192.168.0.10
class upload2 imq1 Zyta
  match srcip 192.168.0.10

# bonifacy
class download eth1 bonifacy
  match dstip 192.168.0.11
class upload1 imq0 bonifacy
  match srcip 192.168.0.11
class upload2 imq1 bonifacy
  match srcip 192.168.0.11
```