

Pierwsza wersja, bardzo uboga, sklecona na kolanie;)

Przykład 1

wystarczający download, bardzo mały i przeciążony upload.

W przykładzie dysponujemy asymetrycznym łączem o dużej przepustowości downloadu, niech to będzie 4mbit/s lecz malutkim pasmem wychodzącym 512kbit/s, które "ledwo zipi".

Sprawa konfiguracji downloadu jest tu banalna i można by tu nawet zrezygnować z dynamicznego podziału, użyć czystego htb i sprzedawać klientom konkretne przepustowości.

Jednak o upload trzeba szczególnie zadbać gdyż od niego będzie zależało czy klienci uzyskają zakupione przez siebie pasmo.

Oto konfiguracja a poniżej komentarz:

```
<global>
run ul
support mark on eth0
stats unit kB/s
stats file /var/www/stats/nsstats.txt
stats owner root group root mode 644
</global>
```

```
<ul>
iface eth0 src network 192.168.0.0/24
iface eth0 src network 192.168.1.0/24
section speed 512kb/s
section shape 400kb/s
default low 40kb/s
default ceil 128kb/s
default overtake 20kb/s
default htb-scheduler sfq
mode upload
reload 3s
</ul>
```

Sekcja global jest wymagana i tutaj mówi które z kolejnych skonfigurowanych sekcji należy uruchomić a można ich mieć kilka, i dla wygody przez jeden parametr określać które w danym momencie mają zostać uruchomione.

```
support mark on eth0
```

włącza filtr fw czyli w skrócie markowanie pakietów na podanym interfejsie (nie jest to poprawne określenie ale dobrze oddaje idee), opcja ta jest wymagana gdy podsieć jest natowana.

Tworzymy przykładową sekcję dla uploadu nazwijmy ją `ul`, która to nazwa jest w przeciwieństwie do starej wersji programu całkowicie dowolna i równie dobrze może się nazywać upload, sekcja 1 czy przewrotnie download;)

Dyrektywa `iface` ma za zadanie wskazać co dana sekcja ma obsługiwać, w naszym przypadku wszystko wychodzące do internetu z podsieci `192.168.0.0/24` oraz `192.168.1.0/24` i pojawiające się na interfejsie wyjściowym `eth0`

Dalej określamy przepustowość pasma wychodzącego przyporządkowanego konfigurowanej sekcji, oraz poziom na którym oczekujemy utrzymywać jego wykorzystanie

```
section speed 512kb/s shape 400kb/s
```

Wartość `shape` przy dobrej jakości łączy musi być bezwzględnie!! kilkanaście procent niższa od `speed`

Zbyt wysoka wartość zwiększy ryzyko zwiększenia pingów na łącza aż do całkowitego niedziałania podziału w ekstremalnej sytuacji.

Dyrektywa `default` określa domyślne wartości podanych parametrów dla klas którym nie przypisano "indywidualnych" warunków pracy.

Po opis każdego z powyższych parametrów odsyłam do dokumentacji.

A co jeśli zdecydujemy się jednak kontrolować dynamicznie także download?

Oto konfiguracja kolejnej sekcji a poniżej komentarz:

```
<dl>
iface eth1 dst network 192.168.0.0/24
iface eth2 dst network 192.168.1.0/24
section speed 4096kb/s
section shape 36000kb/s
default low 64kb/s
default ceil 2000kb/s
default overtake 256kb/s
default htb-scheduler sfq
mode download
reload 3s
</dl>
```

By uroznać przykład przyjmujemy że nasza sieć jest podzielona na 2 podsieci obsługiwane przez niezależne karty sieciowe.

Odmienne w sytuacji gdy obydwie podsieci współdzielą jedną kartę mamy:

```
iface eth1 dst network 192.168.0.0/24
iface eth1 dst network 192.168.1.0/24
```

lub jeszcze wygodniej:

```
iface eth1 dst network 192.168.0.0/23
```

Pozostałe parametry zostały opisane wcześniej a ich ustawienia dokonujemy kierując się konsekwentnie tymi samymi zasadami.

Czym jest klasa.

Teraz nie pozostaje nam nic innego jak skonfigurować klasy.

Samą klasę zdefiniował bym jako wycinek ruchu traktowany w określony sposób, widziany jako jedna całość wydzielona za pomocą filtrów wskazujących hosty, porty, podsieci czy też inne pakiety oznaczone przez inne kryteria.

Przykładowe dwie konstrukcje:

```
class dl eth1 przykładowa1
    match srcip 212.77.100.101 dstip 192.168.0.10
```

```
class dl eth1 przykładowa2
    match srcip 212.77.100.101
    match dstip 192.168.0.10
```

Czym się one różnią?

Do pierwszej klasy zostaną zakwalifikowane pakiety pochodzące z adresu 212.77.100.101 i jednocześnie skierowanego do hosta lokalnego 192.168.0.10.

Do drugiej pakiety pochodzące z adresu 212.77.100.101 oraz pakiety skierowane do 192.168.0.10 jednak filtry te są niezależne od siebie, więc dana klasa będzie obsługiwać wszystko co otrzymujemy z pierwszego adresu bez znaczenia co do adresu docelowego oraz wszystko to co zmierza do drugiego adresu bez względu na pochodzenie.

W skrócie w pierwszym przypadku obydwie parametry muszą zostać spełnione jednocześnie a w drugim którykolwiek z podanych dwóch.

Pamiętać należy że każdy pakiet testowany jest przez kolejno zdefiniowane filtry i nieodwołalnie wpada do pierwszego który jest spełniony, stąd w wielu przypadkach kolejność definiowania klas ma kluczowe znaczenie.

Dodajemy klasy.

```
class dl eth1 www
    match srcport 80 proto tcp
    match srcport 8080 proto tcp
```

```
class dl eth1 Krzysiek
    match dstip 192.168.0.10
```

```
class dl eth1 Janek
    match dstip 192.168.0.11
```

```
class dl eth1 dwa_Komputery_Oli
    match dstip 192.168.0.12
    match dstip 192.168.0.13
```

```
class ul eth0 Krzysiek
    match srcip 192.168.0.10
```

```
class ul eth0 Janek
    match srcip 192.168.0.11
```

```
class ul eth0 dwa_Komputery_Oli
    match srcip 192.168.0.12
    match srcip 192.168.0.13
```

Utworzyliśmy tu kilka klas dla klientów, przy czym pierwsza obsługuje wyłącznie ruch www ale za to widziany jako całość bez podziału na hosty, a ostatnia przydziela dwóm komputerom wspólne pasmo.

Co trzeba zauważyć każda klasa zostaje przyporządkowana wyłącznie jednej sekcji a pozostałe nie wiedzą o jej istnieniu, fakt że klasy zostaną ze sobą wymieszane w ramach pliku konfiguracyjnego nie ma żadnego znaczenia, dla każdej sekcji ważna jest kolejność klas własnych pozostałe nie istnieją.