

## Zarządzanie pasmem (QoS)

---

### Wybór skryptu i instalacja

---

Instalacja pakietów i skryptów odpowiedzialnych za QoS w OpenWrt jak wszystko (prawie) jest prosta i banalna. Trochę bardziej problematycznym jest wybór skryptu odpowiedzialnego za QoS. W OpenWrt dominują dwa skrypty: autorem jednego jest niejaki Rudi zaś drugiego nbd - jeden z developerów OpenWrt. Analizując wątek na oficjalnym forum OpenWrt poświęcony QoS zauważyć można, że zarówno jeden jak i drugi skrypt ma swoich gorących zwolenników jak i przeciwników. Można spotkać się z opinią że skrypt autorstwa nbd jest gorszy niż skrypt Rudi'ego jako, że słabiej priorytetyzuje ruch VoIP. Trudno to jednoznacznie ocenić; najlepiej samemu zainstalować jedną z wersji i przeprowadzić odpowiednie testy. Skrypty nbd w przeciwieństwie do skryptów Rudi'ego posiadają możliwość konfiguracji za pośrednictwem X-Wrt.

### Skrypt nbd

Instalacja skryptu nbd jest prosta i banalna i sprowadza się do kliknięcia przycisku znajdującego się na stronie X-Wrt Network\QoS . Spowoduje to, iż system samodzielnie dociągnie sobie wszystkie potrzebne mu pakiety i skrypty oraz przeładuje stronę X-Wrt.

Jeżeli używamy Kamikaze to instalacja pakietu sprowadza się do jednej komendy

```
opkg install qos-scripts
```

### Podstawowa konfiguracja QoS

Kamikaze

Ustawiamy wartości maksymalne UL/DL zależne od posiadanego łącza internetowego.

```
uci set qos.wan.upload=256
uci set qos.wan.download=1024
uci commit qos
```

Automatyczny start QoS przy starcie systemu

```
/etc/init.d/qos enable
```

Whiterussian X-Wrt

Po pomyślnym zainstalowaniu skryptów odpowiedzialnych za QoS na stronie Network\Qos wypełniamy poszczególne opcje tj.

- włączamy QoS → QoS Service: Enabled
- podajemy parametry posiadanego łącza (w kbps):
  - szybkość uploadu → WAN Upload Speed
  - szybkość downloadu → WAN Download Speed
- ostatnią pozycję tj. Show Advanced Rules należy zostawić wyłączoną tj. Disabled chyba, że

ktoś jest bardzo ciekawy.

## Ręczne sterowanie QoS

```
qos-start - uruchomienie usługi QoS
qos-stop - zatrzymanie QoS
qos-stat - podgląd priorytetów
```

Zaawansowana konfiguracja QoS zawarta jest w pliku `/etc/config/qos`

## Skrypt Rudy'ego

Mimo nieposiadania konfiguratora graficznego, sama konfiguracja skryptu jest stosunkowo łatwa, nie powinna nastreczyć problemów nawet początkującym użytkownikom o ile nie boją się oni linii poleceń.

Pierwszym etapem będzie zainstalowanie skryptu za pomocą `ipkg`:

```
opkg install http://files.eschauzier.org/qos-re_1.05_all.ipk # nietestowane na 8.09
```

Następnie należy otworzyć plik `/etc/qos.conf` w dowolnym edytorze i odnaleźć linie:

```
# Download speed in kilobits per second
# Set 5% - 10% lower than *measured* line speed (set to zero to disable)
DOWNLOAD=100
```

```
# Upload speed in kilobits per second
# Set 5% - 10% lower than *measured* line speed (set to zero to disable)
UPLOAD=50
```

Zmieniamy wartości `DOWNLOAD` i `UPLOAD` na te nam odpowiadające pamiętając o zmniejszeniu rzeczywistych wartości o 5 do 10%. Jeśli ustawimy te wartości na zbyt wysokie to znacznie nam się zapychać kolejka pakietów i wtedy cały QoS na nic nam się nie zda. Po tej operacji mamy właściwie poprawnie skonfigurowane zarządzanie pasmem.

## Kamikaze

Jako, że nie pojawiają się nowe wersje skryptu Rudyego, a co za tym idzie nie ma wersji na Kamikaze. Poniżej przedstawiono skuteczny sposób na zmuszenie tego skryptu do pracy z Kamikaze.

- Instalujemy pakiety

```
opkg install qos-script (potrzeba nam zależności tego pakietu, **potem go usuwamy!!**)
```

```
opkg remove qos-scripts
```

```
opkg install ip
```

```
opkg install http://files.eschauzier.org/qos-re_1.05_all.ipk -nodeps (wersja HTB)
```

lub

```
opkg install http://files.eschauzier.org/qos-re-hfsc_1.05_all.ipk -nodeps (wersja HFSC)
```

## - Konfiguracja

```
nano /etc/qos.conf
```

Na dole zmieniamy parametr QOS\_IF na interfejs do którego podłączony jest net.

```
# Define custom QoS interface. Defaults to wan interface.
#QOS_IF=br0
```

W przypadku PPPoE to będzie „ppp0”, w innych przypadkach vlan1(**informacja niepewna**). Oczywiście przed zapisaniem usuwamy znak hasha(#)

## - Skrypt startowy

```
nano /etc/rc.d/S99qos

#!/bin/sh
qos-start
```

## Filtry Level7

---

Modyfikując QoS pod własne potrzeby wykorzystać możemy zaawansowane filtry klasyfikujące ruch sieciowy m.in. na podstawie nagłówek pakietów (L7-filters). Przestrzegam tutaj przed nadmiernym optymizmem w wykorzystaniu tych filtrów, jednakże w przypadku niektórych rodzajów ruchu sieciowego sprawdzają się całkiem przyzwoicie.

Standardowo skrypt nbd oferuje dosyć ograniczoną liczbę tych filtrów. Aby poszerzyć sobie możliwości klasyfikacji i priorytetyzowaniu ruchu sieciowego za pomocą filtrów L7 musimy ściągnąć sobie sygnatury interesującego nas ruchu ze strony <http://l7-filter.sourceforge.net/protocols> i zapisać je w katalogu /etc/l7-protocols. Tak zapisane sygnatury możemy później wykorzystać w samodzielnie tworzonych przez nas regułach QoS.

## SimpleShaper

---

### Najnowsza wersja: v.0.4-4

Autor pakietu uruchomił git - system zarządzania wersjami pod adresem <http://artekw.net/gitweb>

Instalujemy pakiet poprzez dodanie do pliku /etc/ipkg.conf linijki przed „dest root /”

```
src shaper ftp://artekw.cerbero.pl/simpleshaper/old # dla 7.09 tylko brcm-2.4
src shaper ftp://artekw.cerbero.pl/simpleshaper/8.09/<architektura> # dla 8.09
```

Następnie koniecznie jest wykonanie aktualizacji drzewa pakietów

```
opkg update
```

Ostatecznie instalujemy pakiet

```
opkg install simpleshaper
```

Pakiet zainstaluje wszystkie potrzebne inne pakiety do poprawnego działania. Program zawiera dwa pliki konfiguracyjne: ip.conf oraz ts.conf w katalogu /etc/

Konfiguracja adresów IP(przykład pliku ip.conf)

```
192.168.1.2
128
128
40
115
1
11
100

192.168.1.3
256
750
40
115
2
12
100

192.168.1.4
256
750
40
115
1
13
100
```

Opis zawartości pliku ip.conf:

- 1 - adres ip komputera, któremu nadajemy limit
- 2 - minimalne ściąganie(download)
- 3 - maksymalne ściąganie
- 4 - minimalne wysyłanie(upload)
- 5 - maksymalne wysyłanie
- 6 - priorytet(im mniejszy tym dany komputer jest ważniejszy)
- 7 - klasa przydziału(każdy przydział musi mieć inną klasę, zawierającą się od 11 wzwyż; klasa 10 jest domyślną i zawiera ustalony limit 1kb/s dla DI i UI i służy do zabezpieczenia przed zmianą adresu IP przez klienta, innymi słowy każdy komputer nie przydzielony do podziału pasma dostaje 1kb/s)
- 8 - ilość połączeń na dane IP

Przykład pliku ts.conf:

```
DOWNLOAD=768
UPLOAD=128
IFACE=br-lan
P2P-UL-RATIO=30
P2P-UP-RATIO=30
```

Zawartość pliku raczej nie wymaga komentarza. Ustalamy tutaj interfejs na którym jest stawiany przydział pasma(domyślnie br-lan).

Skrypt posiada regułę dotyczącą ograniczania pasma dla sieci P2P przy pomocy IPP2P. Domyślnie ustawienia powodują, że ruch P2P(zarówno download i upload) osiągnie maksymalnie 30% łącza całego.

Zarządzanie pasmem aktywujemy komendą

```
ss-start
```

Natomiast wyłączamy

```
ss-stop
```

Dostępne są również proste statystyki

```
ss-stat
```

Domyślnie dostarczany jest skrypt init, który aktywuje usługę na starcie OpenWrt.

[qos](#), [simpleshaper](#)

## Dyskusja

---

software/pasmo.txt · ostatnio zmienione: 2008/11/27 19:06 przez arteq