

# Table of Contents

<b>Configuration</b> .....	3
<b><i>Individual Profile</i></b> .....	3
Example 1: torrent limiting .....	3
Example 2: maximum speed for peering .....	4
Example 3: assigning policing for multisubscribers .....	5



# Configuration

Policing has two connection options:

1. Individual profile: connects without specifying a profile, an individual profile is created for each subscriber.
2. Named profile: connects with the profile name.



For BRAS, named profiles must be used **name specified in the Radius-Accept attributes**.

## Individual Profile

An individual configuration file is used to define each bandwidth control policy. This file specifies bandwidth limits for protocol classes (groups). The method is similar to [overall bandwidth control](#)).

`fdpi_ctrl` utility applies the configured policies to subscribers.

The format of the instruction:

```
fdpi_ctrl command --policing policing_description_file [IP_list]
```

Instructions' syntax and IP addresses specification methods are described in details here: [Control instructions](#).



Note the use of [named policing profiles](#)  
The tariff plan can be set in [JSON format](#) as well.

## Example 1: torrent limiting



For advanced users: We recommend that you read "[Outgoing traffic management via feedback](#)".

We intend to offer our subscribers the plan for 10 Mb/s with torrent bandwidth limit of 3 Mb/s.

To accomplish the goal we create a class for torrents as described in [Configuring priorities](#).

```
bittorrent    cs1  
default       cs0
```

We're splitting the traffic into 2 classes for this example:

- cs0 - corresponds to DSCP=0 QOS(IPP)=0 Best Effort
- cs1 - corresponds to DSCP=8 QOS(IPP)=1 Priority

We're creating the configuration file `rateplan_1.cfg`. It specifies bandwidth limits for each of 8 protocol classes (groups). We use HTB (the method of borrowing available bandwidth) and specify the bandwidth limit for torrents 3 Mb, but not smaller than 1 Mb. Unlike torrents, other traffic may take the whole available bandwidth.

```
htb_inbound_root=rate 10mbit
htb_inbound_class0=rate 8bit ceil 10mbit
htb_inbound_class1=rate 1mbit ceil 3mbit
htb_inbound_class2=rate 8bit ceil 10mbit
htb_inbound_class3=rate 8bit ceil 10mbit
htb_inbound_class4=rate 8bit ceil 10mbit
htb_inbound_class5=rate 8bit ceil 10mbit
htb_inbound_class6=rate 8bit ceil 10mbit
htb_inbound_class7=rate 8bit ceil 10mbit
htb_root=rate 10mbit
htb_class0=rate 8bit ceil 10mbit
htb_class1=rate 1mbit ceil 3mbit
htb_class2=rate 8bit ceil 10mbit
htb_class3=rate 8bit ceil 10mbit
htb_class4=rate 8bit ceil 10mbit
htb_class5=rate 8bit ceil 10mbit
htb_class6=rate 8bit ceil 10mbit
htb_class7=rate 8bit ceil 10mbit
```

- `htp_inbound_root`, `htb_root` are the root classes that define the overall bandwidth for inbound and outbound traffic. The bandwidth is distributed within these classes.
- `rate` - is the minimal bandwidth
- `ceil` - is the maximum bandwidth that can be borrowed from the root class if available
- `class2-7` would not be used as we configure two classes only: 0 and 1.

Here we assign the configured policy to subscribers that use this plan:

```
fdpi_ctrl load --policing rateplan_1.cfg --file
subscribers_with_rateplan_1.txt
```

## Example 2: maximum speed for peering

Bandwidth allocation for several classes (for example containing peer-to-peer traffic) can be deduced from the HTB (Hierarchical Token Bucket) hierarchy. It can be done by specifying the keyword **static** in the description. In this case, the restriction for this class will be applied independently regardless to `htb_root`. For example under the conditions noted above, we separately limit the class 6 to 100 Mbps.

```
htb_inbound_root=rate 10mbit
htb_inbound_class0=rate 8bit ceil 10mbit
htb_inbound_class1=rate 1mbit ceil 3mbit
htb_inbound_class2=rate 8bit ceil 10mbit
htb_inbound_class3=rate 8bit ceil 10mbit
```

```
htb_inbound_class4=rate 8bit ceil 10mbit
htb_inbound_class5=rate 8bit ceil 10mbit
htb_inbound_class6=rate 100mbit static
htb_inbound_class7=rate 8bit ceil 10mbit
htb_root=rate 10mbit
htb_class0=rate 8bit ceil 10mbit
htb_class1=rate 1mbit ceil 3mbit
htb_class2=rate 8bit ceil 10mbit
htb_class3=rate 8bit ceil 10mbit
htb_class4=rate 8bit ceil 10mbit
htb_class5=rate 8bit ceil 10mbit
htb_class6=rate 100mbit static
htb_class7=rate 8bit ceil 10mbit
```

### Example 3: assigning policing for multisubscribers

Let's assign the plan from the previous example to a subscriber with several IPs.

Check that database support is enabled in DPI **/etc/dpi/fastdpi.conf**:

```
udr=1
```

If it is not enabled: we enable it and restart DPI: `service fastdpi restart`

Reserve for corporative subscriber all his IPs:

```
fdpi_ctrl load --bind_multi --user
000_PizzaJohnes:192.168.0.1-192.168.0.5,192.168.1.10-192.168.1.25
```

The subscriber's IP list can be modified [dynamically](#) (i.e. add new IPs and delete it).

Let's assign the bandwidth limits according to the plan:

```
fdpi_ctrl load --policing rateplan_1.cfg --login PizzaJohnes_LLC
```

Setting the schedule for tariff plans:

