# Table of Contents

# Optimizing uplink channel

How the Stingray Service Gateway can help you to reduce uplink costs by 25% while increasing QoE?

### Entry conditions:

If you look at a weekly channel usage graph of a typical home telecommunication operator, you can see that the peek value usually is reached on the most busy hours (MBH), when everyone is resting after working hours and spend their leisure time on the Internet: download and watch movies, listen to music, visit entertainment portals, etc.

The channel utilization peak occurs around 11 PM and usually lasts less than an hour, exceeding normal levels by 30%. During this time, torrent traffic can account for up to 30% of the channel, about 50% is taken up by online video streaming, and only 20% is used for everything else.

### What we want to get:

To suspend the increasing of channel bandwidth by optimizing usage of available bandwidth by different types of traffic.

There is a strong temptation to avoid paying for the “extra” bandwidth, which is needed only for an hour and the rest of the time is merely not used. But if you just don’t pay, then your users will notice problems at a rush hour such as: glitches may occur when watching videos, websites may become unavailable, online games may become running slow and IP calls can not be held. The operator will never accept this strategy because, it is essentially prime time and users need high-quality Internet at exactly this time. How about this situation here? Operator wants to save money, and at the same time it is required to deliver high-quality Internet to the end user, and to make him pleased with services provided.

### Tools:

The Stingray Service Gateway options to use:

- Gathering and analysis of statistics on protocols and directions
- Optimizing the use of external access channels

Additional Modules:

- For receiving, processing and storing NetFlow - QoE Store statistics software.
- For visualization and generating reports - DPIUI2 graphical interface.

### Configuration:

Let’s divide the traffic into classes based on the protocol used. Up to 8 classes can be assigned in total, which SSG can use to mark packets either in the DSCP/TOS field of IP headers or in the VLAN/MPLS priority field. After that, QoS can be managed by an external platform. But we will consider the case when the SSG does it on its own. We place in the lowest priority class all the services that we can afford to be limited during peak hours. Usually these are services having the speed depending on external factors, it's the case when the user doesn't use the service in interactive mode but instead in the background one. The examples of such a services are: downloading files via torrent trackers, software update services and maybe something else specific to your case. Online broadcasting services being able to adapt to the available bandwidth and select the quality needed should be placed in a class with a higher priority, and, finally, in the highest priority class we should place interactive services, which operation is controlled by the user in online mode, so when the problems occur, they will become immediately perceptible; these are web browsing, online games, IP-telephony (SIP, Skype). And to put it simple, we will place everything else in this class. There are exist two approaches for limiting traffic for each class: to limit the maximum bandwidth that the traffic of corresponding class can occupy, or rely upon the mechanism originally provided by the SSG allowing to borrow bandwidth automatically. In the latter case, the SSG will begin to limit traffic based on classes only when the usage of the incoming traffic bandwidth approaches its threshold value. And furthermore, the bandwidth should be bounded from the bottom side in order to get a certain traffic class be affected by latencies only within the controlled limits during peak loads.

If the upper bandwidth limit for a given class is exceeded, the outgoing traffic of this class is limited by the SSG, in this case the bandwidth is decreased evenly for all the subscribers. Due to the "two-way" nature of the most protocols (request-response and subsequent transmission-acknowledgement are used), the restriction of outgoing traffic leads to the restriction of incoming one. When the SSG automatic control mode is used, the degree of this influence is determined using the feedback mechanism, i.e. at first the traffic of the minimum priority will be affected to restriction until the minimum threshold is reached. If this restriction is not enough, then the restriction will be applied to the next priority traffic, and so on and so forth. Both modes allow you to use the mechanism of borrowing bandwidth when unused bandwidth is allocated between classes according to current needs and is reallocated when needed.

## Configuration example for the description given above

Create protocols.txt file:

```
default cs0
mpeg    cs1
bittorrent cs7
```

Convert protocols.txt to protocols.dscp

```
cat protocols.txt|lst2dscp /etc/dpi/protocols.dscp
```

Put hard class-based restrictions to the /etc/dpi/fastdpi.conf configuration file, according to the statistics graph (simple solution, but far from being perfect)

```
#Limit outbound torrent
```

```
tbf_class7=rate 50mbit
#Limit inbound torrent
tbf_inbound_class7=rate 50mbit
```

Another way is to let the DPI to prioritize protocols in a class hierarchy independently within the entire available bandwidth (it requires upper bounds parameters to be selected to account for steep traffic increases, to ensure the restriction occurs based on dpi priorities, rather than because of the physical channel exhaustion, here, for the starting point the channel width minus 10% can be taken, in case of torrents, 50% should be subtracted)

```
htb_inbound_root=rate 180mbit
htb_inbound_class0=rate 100mbit ceil 180mbit
htb_inbound_class1=rate 50mbit ceil 180mbit
htb_inbound_class2=rate 8bit ceil 180mbit
htb_inbound_class3=rate 8bit ceil 180mbit
htb_inbound_class4=rate 8bit ceil 180mbit
htb_inbound_class5=rate 8bit ceil 180mbit
htb_inbound_class6=rate 8bit ceil 180mbit
htb_inbound_class7=rate 10mbit ceil 100mbit
htb_root=rate 180mbit
htb_class0=rate 100mbit ceil 180mbit
htb_class1=rate 50mbit ceil 180mbit
htb_class2=rate 8bit ceil 180mbit
htb_class3=rate 8bit ceil 180mbit
htb_class4=rate 8bit ceil 180mbit
htb_class5=rate 8bit ceil 180mbit
htb_class6=rate 8bit ceil 180mbit
htb_class7=rate 10mbit ceil 100mbit
```

Restart the DPI

```
service fastdpi restart
```

## Result:

Only traffic that is less critical to the end users will be affected by latencies, and the time-sensible online services that are important for the user, will be delivered faster and with minimal delays as a result of prioritization. We managed to reduce the requirements for channel width, and, from the user the point of view, the Internet acces has become faster.
Bonus: A large load also lies down on other operator equipment, so being under the load it can cause additional challenges: packet loss, increasing of response time, etc. Using the SSG shaping feature will prevent operator equipment from being under the critical load, beyound which the issues arise. Also the usage of SSG shaping feature allows operator to postpone its equipment upgrade to a later date.

If you want to save on the channel even more, then follow the description of the “Caching” option .