

Содержание

| | |
|--|---|
| 1 Optimizing uplink channel | 3 |
| SCAT helps reduce uplink expenses by 25% and enhance QoE at the same time. How is it possible? | 3 |

1 Optimizing uplink channel

SCAT helps reduce uplink expenses by 25% and enhance QoE at the same time. How is it possible?

Initial conditions:



Let us examine a weekly chart of channel usage by the common household operator. One can see that the maximum typically takes place on Saturday's evening. It's the time when all people take their weekend rest. Many people spend this time in Internet. They download and watch movies, download music, visit some entertaining sites and so on. The peak load typically happens about 23:00 and lasts for less than one hour. It is 30% higher than normal load. The torrent traffic can take up to 50% of bandwidth at this time. Online video consumes about 20%. All other activities take about 30% only.

What do we want to get:

One is tempted not to pay for the "extra" bandwidth that is used during one hour only. However, subscribers would face problems during peak hours, if one just refuses to pay for the extra bandwidth. The problems look like: delays in video, denial in opening some sites, online games are stalled and difficulties in IP calls. The operator can not accept this. It is a "prime-time" and subscribers demand high quality Internet right at this time. What can we do? We desire to pay less while our subscribers do not notice any troubles and are even grateful.

Tools:

SCAT options "Channel analysis" and "QOS management". NfSen to view and analyse Netflow5 statistics.

Configuration:

Let us split the traffic into channels defined by protocols in use. One can assign up to 8 classes. SCAT uses them to mark packets in DSCP/TOS of IP headers as well as VLAN/MPLS priority field. An external platform can further control QOS. We shall consider a case where SCAT itself controls it. Let us place services that we can restrict during peak hours to the lowest priority class. They are typically those services that depend on external factors by their speed; services that are used in background mode. They are: torrent download, SW update services and may be something else in your case.

Further, we place services of online broadcasting in a class with higher priority. Such services typically can adapt themselves to an available bandwidth and to adjust their quality. The interactive services that are used in an interactive mode we place into the highest priority class. Problems with such services are immediately noticed by subscribers. The typical examples of such services are: WEB browsing, online games, IP calls (SIP, Skype). We can just place everything what remains to this class

for simplicity.

We limit the maximum bandwidth available for each class traffic. Alternatively, we can rely on the method of bandwidth borrowing with automatic control. SCAT supports such functionality. SCAT would start to limit the traffic for some classes only when the inbound traffic is close to its threshold value. We also limit the minimal bandwidth. This allows to avoid damaging effect of these controls on low priority traffic during peak hours.

SCAT starts limiting the outbound traffic for a class when the bandwidth for this class exceeds the upper limit. The bandwidth limitations are shared in a uniform way between subscribers. Limitations implied on the outbound traffic lead to restriction on the inbound one due to the bi-directional operation of most of protocols: query - reply, transmission - confirmation). In the automatic control mode, the degree of this influence is determined by SCAT by the feedback method. First, the lowest priority traffic is restricted until it reaches the minimal threshold set. In case this restriction is not enough, limitations are applied to the traffic of the next priority and so on. Both modes allow using of the borrowing method, when the available bandwidth is distributed between classes according to the current needs and it is redistributed when it is close to the limit.

Configuration example to implement the method described:

First, we create the file protocols.txt:

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

Then we convert protocols.txt to protocols.dscp

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

We add the strict limits by classes into the configuration file /etc/dpi/fastdpi.conf, taking the statistics into account. This is a simple solution but it is far from an optimal one.

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

Alternatively, we allow DPI to prioritize protocols by the classes' hierarchy within the limit of total bandwidth available. This method requires to adjust upper limit parameters to take the traffic peaking into account. It is required to apply restrictions by DPI priorities rather than the threshold set by the physical channel capacity. We take the full bandwidth minus 10% and that minus 50% for torrents as an initial approach.

```
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
```

Then we need to restart DPI:

```
service fastdpi restart
```

The outcome:

Only the traffic that is less critical and important for subscribers is restricted during peak hours. Operation of online services that are important for customers (they immediately notice problems in such services) is faster and has less delays - thanks to prioritization. We manage to reduce the channel bandwidth requirements while Internet becomes more nimble for the users. An additional advantage is: other operator's equipment is also badly loaded during peak hours. This equipment may introduce its own problems, like lost packets increased latency and so on. Using SCAT shaping allows to avoid exceeding the critical equipment load. The lates causes problems in its performance. Therefore SCAT allows to delay hardware upgrade.

Do you wish to get further savings on the channel? Please read ["Cach option"](#) description.