

Table of Contents

Policing of Virtual Channel (vChannel)	3
Setting up	4
For physical interfaces	4
For VLANs	5
For CIDR	7
Policing Configuration	8
Block List Setup - Service 4	9
Allow List Setup - Service 5	9

Policing of Virtual Channel (vChannel)

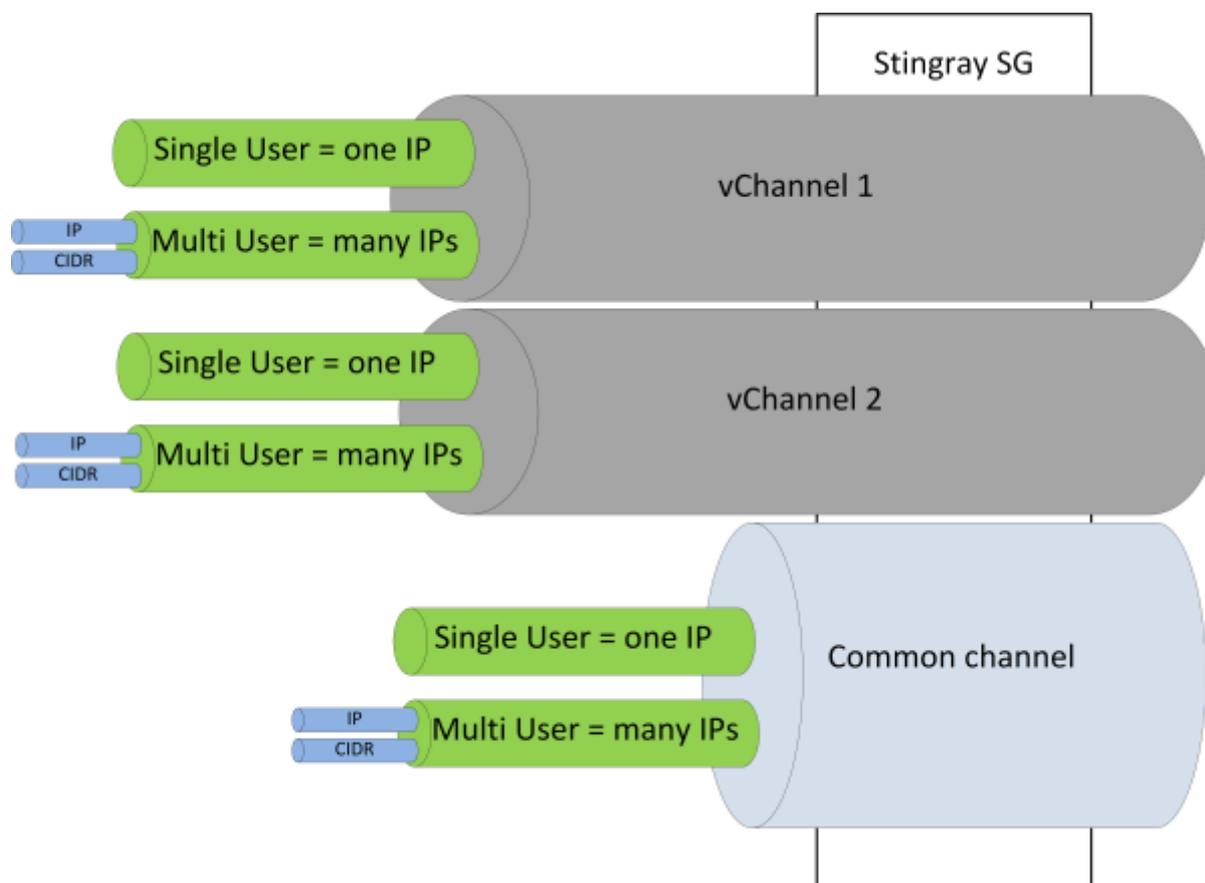
More about VChannels configuration:



In case if the operator has several external (uplink) or internal channels often it is needed to control Channel Upper Boundary and limit low priority traffic independently of one another. Channel traffic is usually uneven and channels often are unequal.



By using vChannel, you can separate traffic from the common channel based on physical ports / VLANs / CIDRs. Policing and services are applied on vChannel [similarly to subscriber rate plans](#). Channel parameters and restrictions in the channel are configured by a special identifier.



Setting up

In the DPI setting `/etc/dpi/fastdpi.conf` there is a parameter to identify the way of channels differentiation:

```
vchannels_type=1
```

where

- 1 different NICs will be used for different channels
- 2 different VLANs will be used for different channels
- 3 they will be defined via CIDR



SSG 12.4+ Added setting `vchannels_default=` to put traffic unallocated on other channels into a separate channel **(but not 0!)**

Next, you need to specify which physical interfaces, VLANs, CIDRs form the virtual channel.

For physical interfaces

```
vchannels_list=60-00.0:60-00.1|61-00.0:61-00.1
```



Traffic through network interfaces **60-00.0** and **60-00.1** refers to the **first virtual channel**, and traffic through **61-00.0** and **61-00.1** refers to the **second**.

If the Stingray SG has other network interfaces specified in the `in_dev/out_dev` setting, the traffic through them will refer to the common channel, which is managed and limited in [the configuration file](#).

For VLANs

Method 1

```
vchannels_list=100:101-115|200:201:240-250
```



VLAN 100 and 101-115 - virtual channel 1, VLAN 200 and 240-250 - virtual channel 2. Traffic that is not in VLANs defined in `vchannels_list` will be in common channel as before. It is managed as usual in [configuration file](#)

Method 2



When applying the settings below, the value set in `fastdpi.conf` ([method 1](#)) will be overwritten. However, upon service restart, the parameter value will be applied from the `fastdpi.conf` file.

1. Setting a VLAN list:

```
fdpi_cli dpi config set vchannels_list='100|101|102|103' --json
[
  {
    result: {
      result_code: 0,
      message: "All parameters successfully reloaded"
    },
    params: [
      {
        opt_name: "vchannels_list",
        result: true,
        err_msg: ""
      }
    ]
  }
]
```

Where: 100 is vchannel 1, 101 is vchannel 2, and so on.

2. Retrieving the VLAN list:

```
fdpi_cli dpi config get vchannels_list --json
[
  {
    result: {
      result_code: 0
    },
    params: [
      {
        opt_name: "vchannels_list",
        opt_value: "100|101|102|103",
        result: true
      }
    ]
  }
]
```

3. Applying a service plan:

```
fdpi_ctrl load --policing --profile.name 1M --vchannel 2 --outformat
json
{
  "query" : { "cmd":"load", "param":"policing", "all":0, "profile":0},
  "lddl" : [
  ],
  "stat_ld" : { "total" : 1, "ok" : 1, "ierr" : 0, "isset" : 0, "noset"
: 0}
}
```

Where:

- total — the number of executed queries
- ok — the number of successful executions
- ierr — the number of errors
- isset — the number of commands that were issued but not executed due to missing settings or restrictions
- noset — the number of commands that were not executed for other reasons

1. Checking the applied service plan on vchannel:

```
fdpi_ctrl list --policing --vchannel 2 --outformat json
{
  "query" : { "cmd":"list", "param":"policing", "all":0, "profile":0},
  "lpolicings" : [
    { "nvc":2, "policing":{ "type":"HTB", "outbound":{
"root_rate":"1mbit","root_ceil":"1mbit","root_burst":"500kbit","root_cb
urst":"500kbit" , "classes" : [ {"cte":"8bit",
"ceil":"1mbit","burst":"8bit","cburst":"500kbit" },{"class":1,
"rate":"8bit", "ceil":"1mbit","burst":"8bit","cburst":"500kbit"
},{"class":2, "rate":"1mbit","burst":"8bit","cburst":"500kbit"
},{"class":3, "rate":"8bit",
"ceil":"1mbit","burst":"8bit","cburst":"500kbit" },{"class":4,
```

```

"rate":"8bit", "ceil":"1mbit", "burst":"8bit", "cburst":"500kbit"
},{ "class":5, "rate":"8bit",
"ceil":"1mbit", "burst":"8bit", "cburst":"500kbit" }, {"class":6,
"rate":"8bit", "ceil":"8bit", "burst":"8bit" }, {"class":7,
"rate":"8bit", "ceil":"1mbit", "burst":"8bit", "cburst":"500kbit" } ]},
"inbound":{ "bw_rate":"0bit", "bw_ceil":"0bit" ,
"root_rate":"1mbit", "root_burst":"500kbit", "root_cburst":"500kbit" ,
"classes" : [ {"class":0, "rate":"8bit",
"ceil":"1mbit", "burst":"8bit", "cburst":"500kbit" }, {"class":1,
"rate":"8bit", "ceil":"1mbit", "burst":"8bit", "cburst":"500kbit"
}, {"class":2, "rate":"8bit",
"ceil":"1mbit", "burst":"8bit", "cburst":"500kbit" }, {"class":3,
"rate":"8bit", "ceil":"1mbit", "burst":"8bit", "cburst":"500kbit"
}, {"class":4, "rate":"8bit",
"ceil":"1mbit", "burst":"8bit", "cburst":"500kbit" }, {"class":5,
"rate":"8bit", "ceil":"1mbit", "burst":"8bit", "cburst":"500kbit"
}, {"class":6, "rate":"8bit",
"ceil":"8bit", "burst":"8bit", "cburst":"8bit" }, {"class":7,
"rate":"8bit", "ceil":"1mbit", "burst":"8bit", "cburst":"500kbit" } ],
"description":{ "name":"1M" }}
],
"stat_lst" : { "total" : 1, "isset" : 1, "noset" : 0}
}

```

Name - the name of the service plan.

2. Deleting a service plan from vchannel:

```

fdpi_ctrl del --policing --vchannel 2 --outformat json
{
  "query" : { "cmd":"del", "param":"policing", "all":0, "profile":0},
  "lddls" : [
  ],
  "stat_ld" : { "total" : 1, "ok" : 1, "ierr" : 0, "isset" : 0, "noset"
: 0}
}

```

For CIDR

Creating ipchannels.txt file:

```

10.0.1.0/24 1
10.0.2.0/24 1
10.1.0.0/16 2

```

To convert:

```

cat ipchannels.txt | as2bin /etc/dpi/ipchannels.bin

```

CKAT 12.4+ Added support for IPv6 channels (with reload).

Example of an assignment:



```
fe80::0/8 1  
cat ipchannels6.txt | as2bin6 /etc/dpi/ipchannels6.bin
```

To list vChannels in /etc/dpi/fastdpi.conf:

```
vchannels_list=1|2
```



Traffic from **CIDR 10.0.1.0/24 and 10.0.2.0/24 refers to the first** virtual channel, **from CIDR 10.1.0.0/16 refers to the second** channel. Traffic that is not in the defined CIDR will refer to the common channel, which is managed and limited in the [configuration file](#).

Policing Configuration

Load the policing setting for channels 1 and 2

```
fdpi_ctrl load --policing vchannel1.cfg --vchannel 1  
fdpi_ctrl load --policing vchannel2.cfg --vchannel 2
```

where vchannel1.cfg - file with policing definition (same as for the general channel, virtual channels and subscriber rate plans)

Example of the configuration file:

```
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
```

Let's check what setup is loaded:

```
fdpi_ctrl list all vchannel --policing
```

The settings for the virtual channels are stored in DPI UDR (internal database) and can be changed on the fly during operation as well as for subscribers. Configuration of policing for the channel can be loaded by the named profile either.

```
fdpi_ctrl load --policing --profile.name vchannel_1 --vchannel 1
```

The tariff plan can be set in [JSON format](#) as well.

Block List Setup - Service 4

The profile is created similarly to creating Subscriber's profile: [Filtering Management](#).

```
fdpi_ctrl load --service 4 --profile.name test_blocked --vchannel 2
```

Allow List Setup - Service 5

The profile is created similarly to creating Subscriber's profile: [Allow List Management](#).

```
fdpi_ctrl load --service 5 --profile.name test_white --vchannel 1
```