Table of Contents

Setting

Setting up subnets for balancing

Balancing applies only to IP addresses belonging to the AS defined as `local` in `asnum.dscp`.

1. Define an autonomous system with IP addresses that are used by subscribers:

   ```
   vi aslocal.txt
   10.0.0.0/8 64511
   172.16.0.0/12 64511
   192.168.0.0/16 64511
   cat aslocal.txt | as2bin /etc/dpi/aslocal.bin
   ```

   1. Examples for IPv4
   2. Example for IPv6
2. Mark a given autonomous system as local:

   ```
   vi my_as_dscp.txt
   64511 local
   10415 local
   cat my_as_dscp.txt | as2dscp /etc/dpi/asnum.dscp
   ```

   You can use either a custom AS or a public AS as the AS.
   Read more at the link

3. A reload must be performed to apply the parameters:

   ```
   service fastdpi reload
   ```

Setting /etc/dpi/fastdpi.conf

1. Define the input and output interfaces in the configuration file.
   The input interfaces to which the traffic mirror is fed are specified in the `in_dev` parameter, and the output interfaces are specified in `out_dev`.
   Interfaces do not form pairs and species constructs are allowed:

   ```
   in_dev=05-00.0:05-00.1:05-00.2:05-00.3:0b-00.0:0b-00.1:0b-00.2:0b-00.3
   out_dev=08-00.0:08-00.1:08-00.2:08-00.3
   ```

   Or

   ```
   in_dev=05-00.0
   out_dev=out_dev=08-00.0:08-00.1:08-00.2:08-00.3
   ```

2. Enable balancing mode in the `enable_l2_lb` parameter, where:
   - `0` — deactivate balancing;

    - `1` — activate balancing.
3. Determine by what value to initialize the hash table in the `lb_hash_out_dev_type` parameter, where:
    - `0` — use the internal index of the output interface;
    - `1` — use the interface name from [in|out]_dev.
4. Select an engine to handle thread dispatchers in the `dpdk_engine` parameter, where:
    - `0` — read/write **default** engine, one dispatcher for everything;
    - `1` — read/write engine with two dispatcher threads: a dispatcher for each direction;
    - `2` — Read/write engine with RSS support: for each direction, `dpdk_rss` dispatchers are created (default `dpdk_rss`=2), so total number of dispatchers = 2 * `dpdk_rss`— to work with in_dev, and a separate dispatcher to work with out_dev.
      When the parameter value is 2, the `mqrx_lb_engine` engine is activated. The principle of operation is the same as in the usual mode `dpdk_engine`=2, only rss is enabled on in_dev, and only one rx queue is created on out_dev.

      Learn more about the `dpdk_engine` parameter by clicking here.
5. Select a balancing algorithm. The `maglev` algorithm with fixed hash table size is used for traffic balancing, where:
    - `1` — if src and dst ip are both local, then hash is calculated based on these two addresses;
    - `2` — if only src ip local, then hash is calculated based on src ip;
    - `3` — if only dst ip local, hash is calculated on the basis of dst ip;
    - `4` — hash is calculated based on src and dst ip.

      Based on the calculated hash value, the output interface is determined by determining the index of the hash table cell containing the interface index from the array of output interfaces.