Table of Contents

Management utilities

dpdkinfo

Receive diagnostic data from SFP modules. Parameters:

- `-h` – help
- `module_eeprom` – SFP module optical diagnostics information (if supported by the module).

bpctl_util

Manual bypass control. DPI controls bypass independently, but manual bypass control is performed by this utility if necessary:

- `get_bypass` – get bypass status
- `set_bypass on` – activate bypass
- `set_bypass off` – deactivate bypass
- `get_std_nic` – diagnostic
- `set_std_nic off` – setting the card to bypass mode (switches the mode to NON-standard mode, i.e. with bypass mode)

driverctl

DPDK management. Parameters:

- `list-overrides` – check the list of cards in DPDK mode
- `unset-override 0000:04:00.0` – exit the card from the DPDK mode
  The fastDPI process must first be stopped with the command `service fastdpi stop`!
- `-v set-override 0000:04:00.0 vfio-pci` – return the cards back under DPDK control after working with the standard driver
  When switching cards to DPDK mode, be careful not to accidentally switch the server control interface to DPDK mode - communication with the server will be immediately interrupted!

Please note that Mellanox network interface cards cannot be switched to DPDK using the driverctl utility — their driver is installed in a different way. They also remain under operating system control, therefore the interfaces will still appear in the output of ip/ipconfig utilities. If it is necessary to install a driver to support DPDK on Mellanox network cards, please contact technical support.

Configuring DPDK in Hyper-V is described in detail in the appropriate section.

checklock

Check if an address or port is blacklisted.
Example of a port check:

```
checklock 188.114.97.28
```

checkproto

Check if the address or port is included in the custom protocol.
Example of a address check:

```
checkproto youtube.com
```

checknat

White address allocation check. Shows how the network for NAT is distributed among process vorkers.
Command entry format:

```
nthr=x, algo=0|1|2, cidrs='list cidrs'
[,tcheck_correct_hash=0:1,gr_cidrs='list gray cidrs',dst_cidrs='list
destination cidrs']
```

Parameters:

- `nthr` – num work threads per cluster. Maximum value - 128
- `algo` – **0** - hashmask (default), **1** - crc, **2** - rxdsp_2
- `cidrs` – list cidr white address
- `check_correct_hash` – check hash function
    - `gr_cidrs='list cidr gray address for check'`
    - `dst_cidrs='list cidr destination address for check'`

Examples:

```
# Example 1
nthr=16 algo=0 cidrs='16.35.120.0/24,91.210.24.128/26'
# Example 2
nthr=16 algo=0 cidrs='16.35.120.0/24,91.210.24.128/26' check_correct_hash=1
gr_cidrs='10.0.0.0/24,192.168.4.0/28' dst_cidrs='30.0.0.0/24,50.0.0.0/24'
```