

# Содержание

Утилиты управления .....	3
<i>dpdkinfo</i> .....	3
<i>bpctl_util</i> .....	3
<i>driverctl</i> .....	3
<i>checklock</i> .....	4
<i>checkproto</i> .....	4
<i>checknat</i> .....	4



# Утилиты управления

## dpdkinfo

Получение диагностических данных с SFP-модулей.

Параметры:

- -h – подсказка
- module\_eeprom – информация по оптической диагностике модуля SFP (если она поддерживается модулем).

## bpctl\_util

Ручное управление bypass.

DPI управляет bypass самостоятельно, но в случае необходимости ручное управление осуществляется данной утилитой.

Параметры:

- get\_bypass – получить состояние bypass
- set\_bypass on – активировать bypass
- set\_bypass off – деактивировать bypass
- get\_std\_nic – диагностика
- set\_std\_nic off – установка карты в режим bypass (переключает режим в НЕстандартный, то есть с bypass режимом)

## driverctl

Управление DPDK.

Параметры:

- list-overrides – проверить список карт, находящихся в режиме DPDK
- unset-override 0000:04:00.0 – вывести карту из режима DPDK  
**Предварительно необходимо остановить процесс fastDPI командой `service fastdpi stop`!**
- -v set-override 0000:04:00.0 vfio-pci – вернуть карты обратно под управление DPDK после работ со штатным драйвером  
**При переводе карт в режим DPDK будьте внимательны и не переведите случайно управляющий интерфейс сервера в режим DPDK — связь с сервером сразу прервется!**



Конфигурирование DPDK в Hyper-V подробно описано в соответствующем [разделе](#).

## checklock

Проверка вхождения адреса или порта в черный список.

Пример:

```
checklock 188.114.97.28
```

## checkproto

Проверка вхождения адреса или порта в custom протокол.

Пример проверки порта:

```
checkproto youtube
```

## checknat

Проверка распределения белых адресов. Показывает, как распределяется сеть для NAT между воркерами-процессами.

Формат записи:

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

Параметры:

- nthr - количество рабочих потоков на кластер
- algo - **0** - hashmask (по умолчанию), **1** - crc, **2** - rxdsp\_2
- cidrs - список белых адресов cidr
- check\_correct\_hash - проверка хэш-функции
  - gr\_cidrs='список серых адресов cidr для проверки'
  - dst\_cidrs='список адресов назначения cidr для проверки'

Примеры:

```
# Пример 1
nthr=16 algo=0 cidrs='16.35.120.0/24,91.210.24.128/26'
# Пример 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'
```