

Руководство пользователя УСОТ

В документе приводятся основные настройки и параметры запуска программного обеспечения «Узел специализированной обработки трафика (УСОТ)»

Общие сведения

Программное обеспечение УСОТ предназначено для обработки сетевых пакетов в режиме реального времени. Основные функции программы:

- Удаление содержимого пакета
- Сохранение первых N байт в пакете
- Добавление временных меток в пакет (timestamping)
- Удаление повторяющихся пакетов (дедупликация)
- Маскирование данных в пакете
- Удаление GRE и ERSPAN заголовков

Программа выполняется на стандартных серверах с сетевыми картами Intel, которые поддерживают DPDK (Data Plane Development Kit). Использование DPDK обеспечивает кратный рост производительности по сравнению с обычными сетевыми картами.

Управление работой программы осуществляется с помощью командного интерфейса (CLI) операционной системы.

Подготовительные действия

Перед запуском программы необходимо провести ряд подготовительных действий.

Скачать и собрать DPDK версии 22.11.4:

```
cd /opt
wget 'http://static.dpdk.org/rel/dpdk-22.11.4.tar.gz'
tar xf dpdk-22.11.4.tar.gz
cd dpdk-stable-22.11.4
meson setup build
в файле config/rte_config.h
добавить #define RTE_LIBRTE_IEEE1588
cd build
ninja
meson install
ldconfig
```

Для того чтобы приложение могло работать с сетевыми картами, их надо назначить выбранному драйверу. Для этого в DPDK есть утилита

```
/opt/dpdk-stable-22.11.4/usertools/dpdk-devbind.py
```

Посмотреть статус сетевых карт: `dpdk-devbind.py --status`

Назначить устройство: `dpdk-devbind.py -b vfio-pci 0000:00:00.0,`

где `vfio-pci` - выбранный драйвер, `0000:00:00.0` - pci адрес устройства.

Требуемое количество ядер процессора: минимум 1 + encap_rx_cores + rx_cores (см. файл настройки ниже). Приложение использует ядра начиная с №1.

При запуске приложения требуется указать в качестве аргумента путь к конфигурационному файлу. Конфигурационный файл в формате JSON представлен ниже.

```
{
  "l4_payload_off": 100,
  "l4_payload_mask": "ffffffffffffffffffffffffffffffff",
  "trim_to_size" : 128,
  "trim_by_l4" : true,
  "timesync" : true,
  "encap_rcv_nic": "03000",
  "decap_snd_nic": "03000",
  "decap_dst_mac": "40:a6:b7:4b:5a:31",
  "encap_rx_cores" : 8,
  "rcv_nic": "03001",
  "snd_nic": "03001",
  "dst_mac": "40:a6:b7:4b:5a:31",
  "rx_cores" : 8,
  "hash_table_size" : 786432,
  "pkt_pool_size" : 786432,
  "timeout": 10,
  "nic_code": {"03000": "25187a0f36e29180", "03001": "9ad9e333f9efb4a8"}
},
```

где

- l4_payload_off - опциональный параметр, задает смещение после L4 заголовка для маскирования payload (если не указан, то = 0)
- l4_payload_mask - опциональный параметр, задает маску payload в hex (в данном примере 20 байт = 0xff)
- trim_to_size - опциональный параметр, если задан, то пакет обрезается таким образом, чтобы суммарная длина, включая Ethernet заголовок, не превышала заданное значение
- trim_by_l4 - опциональный boolean параметр, если указан и равен true, то пакет обрезается по L4 заголовку
- timesync - опциональный boolean параметр, если указан и равен true, то на сетевых интерфейсах активируются установка меток времени (timestamp'ы)
- encap_rcv_nic - опциональный строковый параметр, задает имя интерфейса для приема GRE/ERSPAN. Пакеты после снятия GRE/ERSPAN будут отправлены через decap_snd_nic по адресу decap_dst_mac и количество задействованных ядер encap_rx_cores)
- rcv_nic - опциональный строковый параметр, задает имя интерфейса для дедупликации. Принятые пакеты после дедупликации будут отправлены через snd_nic по адресу dst_mac. Количество задействованных ядер указывается в rx_cores
- hash_table_size - размер hash таблицы в пакетах для дедупликации
- pkt_pool_size - обязательный параметр, задает размер пула пакетов
- timeout - опциональный параметр, задает время жизни в ms записи в hash таблице
- nic_code - задает коды активации для интерфейсов, коды запроса активации выводятся при запуске tap_service_node, пример:

```
Port's '03000' activation id 5c41a0b1e8ebfff4
Port's '03001' activation id 4ade57494d0dd08c
```

Запуск и остановка приложения

1. Приложение находится по адресу
/opt/tap_service_node/tap-service-node
2. Запустить приложение можно следующим образом
sudo /opt/tap_service_node/tap_service_node cfg.json,
где cfg.json – файл конфигурации, который нужно обязательно указывать
3. После запуска приложения будет выведен экран со статистикой, отображающей количество полученных, обработанных и отправленных пакетов. Пример экрана статистики представлен ниже.

```
Port statistics =====
Statistics for port 03000 (MAC 3C:FD:FE:C7:F0:4C)-----
Up 10000 (AutoNeg) Full Duplex
tx:          50000 hw_tx:          50000
tx_drops:    0 hw_tx_errors:    0
rx:          50035 hw_rx:          50035
rx_drops:    35 hw_rx_missed/errors: 0 /
rx[0]:       0
rx[1]:      15000
rx[2]:      10035
rx[3]:      15000
rx[4]:       5000
rx[5]:       0
rx[6]:      5000
rx[7]:       0
Statistics for port 03001 (MAC 3C:FD:FE:C7:F0:4D)-----
Up 10000 (AutoNeg) Full Duplex
tx:          10 hw_tx:          10
tx_drops:    0 hw_tx_errors:    0
rx:          50000 hw_rx:          50000
rx_drops:    0 hw_rx_missed/errors: 0 /
rx[0]:       0
rx[1]:      15000
rx[2]:      10000
rx[3]:      15000
rx[4]:       5000
rx[5]:       0
rx[6]:      5000
rx[7]:       0
tx[0]:       0
tx[1]:       3
tx[2]:       2
tx[3]:       3
tx[4]:       1
tx[5]:       0
tx[6]:       1
tx[7]:       0
=====
pkt_mbuf_avail_count: 700406
pkt_mbuf_in_use_count: 86026
pkt_node_avail_count: 786432
pkt_node_in_use_count: 0
MaxHashList: 1
```

Рис. 1 Экран после запуска tap_service_node

4. Остановить выполнение команды можно, используя комбинацию клавиш Ctrl + C

Лицензирование

Контроль распространения программного обеспечения осуществляется привязкой к установленной сетевой карте. Для работы на новом сервере или новой сетевой картой необходимо обратиться к разработчику программного обеспечения для генерации новых лицензий.