

Содержание

Администрирование Баз Данных: UDR и SDR	3
Активация встроенной UDR	4
Увеличение размера БД UDR и SDR	5
Восстановление БД с переносом данных	5
Удаление БД и загрузка в нее данных заново из внешнего источника (биллинга и т.п.)	5
Экспериментальный раздел	6
Восстановление БД в формат команд fdpi_ctrl	6
Восстановление отдельных таблиц БД в формат команд fdpi_ctrl	6

Администрирование Баз Данных: UDR и SDR

До 14 версии используется только одна встроенная база данных **UDR (User Data Repository)** предназначена для постоянного хранения данных об услугах, полисингах и других настройках FastDPI.

С 14 Версии вводится разделение UDR на UDR и SDR. Разделение происходит автоматически при обновлении версии.

SDR (System Data Repository) предназначена для хранения настроек FastDPI, не связанных с абонентами. Можно считать, что SDR является продолжением fastdpi.conf. Никакой специальной активации SDR не требуется - необходимые .mdb-файлы создаются автоматически при включении соответствующего режима в fastdpi.conf.



Глобальные правила для vlan drop/pass/hide/permit, заданные прежней CLI-командой vlan group, будут сконвертированы и перенесены из UDR в SDR с удалением из UDR. При даунгрейде с 14 на 13 версию данные из SDR будут потеряны, нужно делать ручной перенос через сохранение копии SDR перед обновлением.

Список таблиц в UDR

Таблица	Назначение
bindings	Связь login и адреса
bindings_multi	Связь login и адреса для мультиабонентов (с несколькими IP)
ip_props	Свойства абонентов BRAS
policing	Настройки полисинга для абонентов
profile_names	Имена профилей
profiles	Профили услуг и полисинга
services	Настройки услуг для абонентов
vchannel_policing	Настройки полисинга для каналов

Структура SDR

В каталоге SDR /var/db/dpisdr, в зависимости от включенных режимов, находятся файлы базы данных:

- fdpi.mdb: общие данные
- bras.mdb: данные, относящиеся к режиму BRAS
- router.mdb: данные, относящиеся к режиму роутера

Содержимое этих БД для 14 версии:

fdpi.mdb:

- `acl_vlan_rule` - хранит VLAN правила (правила задаются CLI-командами группы `vlan rule`)
- `acl_qinq_rule` - хранит QinQ правила (правила задаются CLI-командами группы `vlan rule`)

`bras.mdb`:

- `l2tp_server_props` - L2TP-сервера, обслуживаемые BRASом, и их свойства. (правила задаются CLI-командами группы `l2tp server`)


`router.mdb`:

- `neighbor4` - IPv4 neighborDB, [подробнее в разделе "ARP менеджмент"](#)
- `neighbor6` - IPv6 neighborDB, [подробнее в разделе "ARP менеджмент"](#)



Максимальный размер каждого из `.mdb`-файлов задается параметром `udr_size`.

Бэкап и восстановление

1.  Остановить `fastDPI`
2. **Создание бэкапа:** сохранить копии `.mdb`-файлов из каталога `SDR`
3. **Восстановление бэкапа:** скопировать сохраненные `.mdb`-файлы в каталог `SDR`

Активация встроенной UDR



Активация UDR требуется в обязательном порядке при использовании динамических IP адресов и для поддержки абонентов с несколькими IP. Никакой специальной активации `SDR` не требуется - необходимые `.mdb`-файлы создаются автоматически при включении соответствующего режима в `fastdpi.conf`.

Активация UDR осуществляется настроечным параметром в файле `/etc/dpi/fastdpi.conf`

```
udr=1
```

Созданная база данных размещается в каталоге `/var/db/dpi`



Сделать копию БД можно без остановки DPI командой резервного копирования

```
mdb_copy /var/db/dpi ./DB
```

и выполнять дальнейшие манипуляции над копией БД.

Увеличение размера БД UDR и SDR

По умолчанию размер БД UDR и SDR ограничен 1GB, если у вас количество профилей полисинга превышает 1 млн, то потребуется увеличить размер по умолчанию:

```
udr_size=2147483648
```

установит размер БД в 2GB



Встроенная БД не требует администрирования и устойчива к отказам. Тем не менее в редких случаях происходило повреждение встроенной БД. В этом случае возможны следующие варианты действий:

Восстановление БД с переносом данных

Останавливаем fastDPI

```
service fastdpi stop
```

Выполняем скрипт

```
rm -rf /var/db/dpi.recover/*
mkdir -p /var/db/dpi.recover/tmp
for table in $(mdb_dump -l /var/db/dpi); do
  mdb_dump -f /var/db/dpi.recover/tmp/dump.$table.load -s $table /var/db/dpi
  mdb_load -f /var/db/dpi.recover/tmp/dump.$table.load /var/db/dpi.recover
done
rm /var/db/dpi/lock.mdb
mv /var/db/dpi/data.mdb /var/db/dpi.recover/data.mdb.backup
cp -f /var/db/dpi.recover/data.mdb /var/db/dpi/
```

Запускаем fastDPI

```
service fastdpi start
```

Удаление БД и загрузка в нее данных заново из внешнего источника (биллинга и т.п.)

Останавливаем fastDPI

```
service fastdpi stop
```

Удаляем БД

```
/bin/rm /var/db/dpi/*
```

Запускаем fastDPI

```
service fastdpi start
```

Загружаем в БД заново все настройки с помощью своих скриптов

Экспериментальный раздел

Восстановление БД в формат команд fdpi_ctrl

Останавливаем fastDPI

```
service fastdpi stop
```

Выполняем скрипт

```
mdb_dump -p -a -f dump.sh /var/db/dpi  
/bin/rm /var/db/dpi/*
```

Запускаем fastDPI

```
service fastdpi start
```

Запускаем скрипт

```
chmod +x dump.sh  
./dump.sh
```

Восстановление отдельных таблиц БД в формат команд fdpi_ctrl

Останавливаем fastDPI

```
service fastdpi stop
```

Выполняем скрипт

```
for table in $(mdb_dump -l /var/db/dpi); do  
mdb_dump -p -f dump.$table.sh -s $table /var/db/dpi  
done  
/bin/rm /var/db/dpi/*
```

Запускаем fastDPI

```
service fastdpi start
```

Выбираем нужные нам скрипты и запускаем, **пример**

```
chmod +x dump.bindings.sh  
./dump.bindings.sh
```