

Содержание

- Резервирование BRAS L2 в режиме Active-Standby 3
 - Репликация данных авторизации и Синхронизация базы данных нескольких BRAS* 3
 - Применение данных на fastDPI 4
 - Описание алгоритма* 4
 - Скрипт для активного резервирования DPI* 5

Резервирование BRAS L2 в режиме Active-Standby

Вебинар по теме:



Video

Резервирование BRAS в режиме L2 предполагает включение двух СКАТ в один широкополосный L2 домен. Один в режиме Master, другой в режиме Slave. Master осуществляет обработку трафика, авторизацию пользователей через PCRF сервер. Slave не пропускает трафик через себя, интерфейсы DPDK находятся в режиме ожидания трафика (down). Синхронизация информации об абонентах происходит через PCRF сервер. Slave отслеживает доступность и работоспособность Master, при сбое в работе Slave в автоматическом или ручном режиме активирует (up) DPDK интерфейсы и начинает обрабатывать трафик. Пример включения и прохождения трафика представлено на схеме.



Репликация данных авторизации и Синхронизация базы данных нескольких BRAS

В СКАТ BRAS состоит из компонент

- fastDPI - обработка трафика абонентов.
- fastPCRF - интеграция по протоколу Radius между fastDPI и Radius сервером. Один fastPCRF может обслуживать несколько fastDPI серверов.

Применяется следующая схема репликации для согласования данных об абонентах на всех fastDPI-серверах: fastPCRF шлет ответы авторизации и CoA-запросы на все fastDPI серверы, перечисленные в параметрах [fdpi_server](#).



Отправка параметров авторизации производится через персистентную очередь, так что даже если какой-то из серверов fastDPI был отключен на момент отправки данных, при включении он получит все данные за время своего



Применение данных на fastDPI

При приеме данных авторизации сервер fastDPI видит, его это был запрос или же это ответ на чужой запрос (для этого в пакете есть специальная метка). Если это ответ на свой запрос, данные применяются "по полной": создается DHCP или PPPoE-сессия, если это был запрос DHCP или PPPoE-авторизации, данные запоминаются в UDR. Если же это ответ на чужой запрос, fastDPI просто запоминает "чужие" данные у себя в UDR. Тем самым при отключении основного fastDPI-сервера и переводе нагрузки на резервный, у резервного fastDPI-сервера в UDR уже будут все свойства абонента - его услуги, полисинг, L2-свойства - MAC-адрес, VLAN и пр. То есть UDR у основного и резервного серверов будут по большому счету согласованы.

Описание алгоритма

Концепция резервирования SKAT — MASTER-SLAVE (L2-BRAS):

MASTER является активным сервером и обрабатывает трафик во время нормальной работы сети. SLAVE в свою очередь находится в состоянии ожидания с выключенными интерфейсами и активно опрашивает состояние MASTER и в случае проблем на нем включается в работу.

Нормальная работа сети:

Через MASTER проходит обработка всего трафика, авторизация и т.д, SLAVE в обработке трафика не участвует, но через mgmt интерфейс опрашивает состояние MASTER. Происходит несколько проверок состояния:

- Проверка, что процесс fastDPI запущен;
- Проверка, что процесс fastDPI работает стабильно и не находится в состоянии циклического рестарта;
- *Опционально* — проверка состояния интерфейсов. В зависимости от схемы подключения такая проверка может быть не нужна, поэтому данная проверка опциональна.

Если все проверки прошли успешно, то SLAVE остается в режиме ожидания и не предпринимает никаких действий.

В случае обнаружения ошибки SLAVE берет на себя роль MASTER.

Переключение MASTER→SLAVE:

При обнаружении на MASTER ошибки SLAVE начинает обрабатывать трафик самостоятельно, для этого отправляется команда на отключение MASTER, отключение интерфейсов SKAT и остановки процесса fastDPI. В это же время SLAVE включает свои интерфейсы. Следует отметить, что автоматического переключения обратно не реализовано, так как необходима проверка инженером что проблема на основном сервере устранена и можно переключать трафик обратно.

Переключение SLAVE→MASTER:

После устранения проблем на основном сервере необходимо выключить интерфейсы резервного сервера, запустить обработку трафика основным сервером и включить сервис резервирования на SLAVE.

Скрипт для активного резервирования DPI

Скрипт должен быть установлен на резервном DPI, где он работает в непрерывном цикле, мониторя состояние основного DPI через SSH.

Этот скрипт использует **4 проверки** для подтверждения того, что основной DPI работает:

1. Сервер доступен по сети (pingcheck)
2. Процесс fastDPI присутствует
3. PID процесса fastDPI не изменился
4. Состояние ссылки на основном DPI не изменилось (необязательная проверка). Эта проверка отключена по умолчанию, так как может быть не нужна в некоторых топологиях.

Процесс установки:

1. Скачать все файлы из [архива](#) на целевой резервный сервер.
2. Настроить IP-адрес основного сервера в скрипте `SRS.sh`.
3. Создать пару SSH-ключей на резервном сервере с помощью команды

```
ssh-keygen -t ed25519
```

4. Создать нового пользователя с правами `sudo` на основном сервере.
5. Скопировать приватные SSH-ключи с резервного сервера в файл `authorized_keys` нового аккаунта на основном сервере.
6. Добавить права на выполнение установочного скрипта с помощью команды

```
chmod +x install.sh
```

7. Запустить `install.sh`.

Управление сервисом:

1. Запуск сервиса:

```
systemctl start fastsrs
```

2. Проверка статуса сервиса:

```
systemctl status fastsrs
```

3. Остановка сервиса:

```
systemctl stop fastsrs
```

4. Проверка логов сервиса:

```
journalctl -u fastsrs
```