

# Table of Contents

Персистентная очередь .....	3
-----------------------------	---



# Персистентная очередь

Очередь организуется в файловой системе в каталоге `/var/spool/dpi/pcrf`. Для каждого fastDPI-сервера в этом каталоге создается отдельный файл с именем `pq-<fastDPI-address>:<port>`, например, если в `fastpcrf.conf` описаны два fastDPI-сервера:

```
fdpi_server=127.0.0.1%lo:29000
fdpi_server=10.20.30.40:eth1:29000
```

то каталог `/var/spool/dpi/pcrf` будет содержать два файла:

```
pq-127.0.0.1:29000
pq-10.20.30.40:29000
```

Можно изменить каталог очереди в `fastpcrf.conf` параметром `fdpi_pqueue_dir`:

```
# Каталог, в котором создаются персистентные очереди
fdpi_pqueue_dir=/var/spool/dpi/pcrf
```

Очередь страничная, размер одной страницы 2М. Можно задать максимальный размер очереди в страницах `fastpcrf.conf`-параметром `fdpi_pqueue_max_pagecount`:

```
# Макс число страниц в очереди.
# Размер каждой страницы = 2М
# 0 - число страниц не ограничено.
# Минимальное значение: 2
# Следует учитывать, что размер файла очереди ограничен сверху параметром
rlimit_fsize
# больше этого размера очередь не может быть.
fdpi_pqueue_max_pagecount=4
```

Если fastDPI-сервер online и его очередь пуста, данные шлются непосредственно серверу, минуя очередь. Если же сервер становится недоступным, данные записываются в его очередь. Если очередь переполняется, то есть невозможно распределить новую страницу, самая старая страница стирается из очереди, и на её месте создается новая страница; таким образом, очередь представляет собой циклический буфер.

Как только fastDPI-сервер становится доступным, ему посылаются все данные из его очереди. FastPCRF периодически пытается соединиться с отвалившимися fastDPI-серверами, тайм-аут этих попыток задается `fastpcrf.conf`-параметром `fdpi_reconnect_timeout`, по умолчанию 2 секунды:

```
# Тайм-аут попытки подсоединения к fastdpi-серверам, секунд
#fdpi_reconnect_timeout=2
```

Можно вручную **при остановленном fastpcrf** удалять файлы очередей.

[CLI-команды управления очередями](#)