Table of Contents

Authorization Pending Queue

FastDPI sends authorization requests to the fastPCRF server, which, in turn, transmits these requests to the Radius server. The Radius server bandwidth is limited, so fastPCRF has an internal authorization request queue to smooth out peaks of requests.

The maximum number of requests to the Radius server is regulated by the fastpcrf.conf parameter `radius_max_connect_count`. This parameter sets the maximum number of connections to the Radius server, default value = 16. Each connection can simultaneously process no more than 256 requests. To set the number of simultaneously processed requests, for example, 1024, you need to specify in fastpcrf.conf:

```
# 4 connections with 256 requests = 1024 simultaneous requests
radius_max_connect_count=4
```

If an authorization request comes from fastDPI and all connections to the Radius server are busy, the request is placed in the internal FIFO waiting queue. When a response is received from the Radius server and the slot becomes free, fastPCRF extracts the pending request from the queue and sends it to the Radius server.

There are two fastpcrf.conf parameters that control queue behavior:

- `radius_max_pending_requests` - maximum queue size, default value is 1000. When this limit is reached, a new request pops the oldest one from the queue.
- `radius_pending_timeout` - maximum time a request can be in the queue, seconds, default value is 10. Expired requests are not sent to the Radius server.

The authorization pending queue can also be managed by CLI commands `pcrf auth queue`.