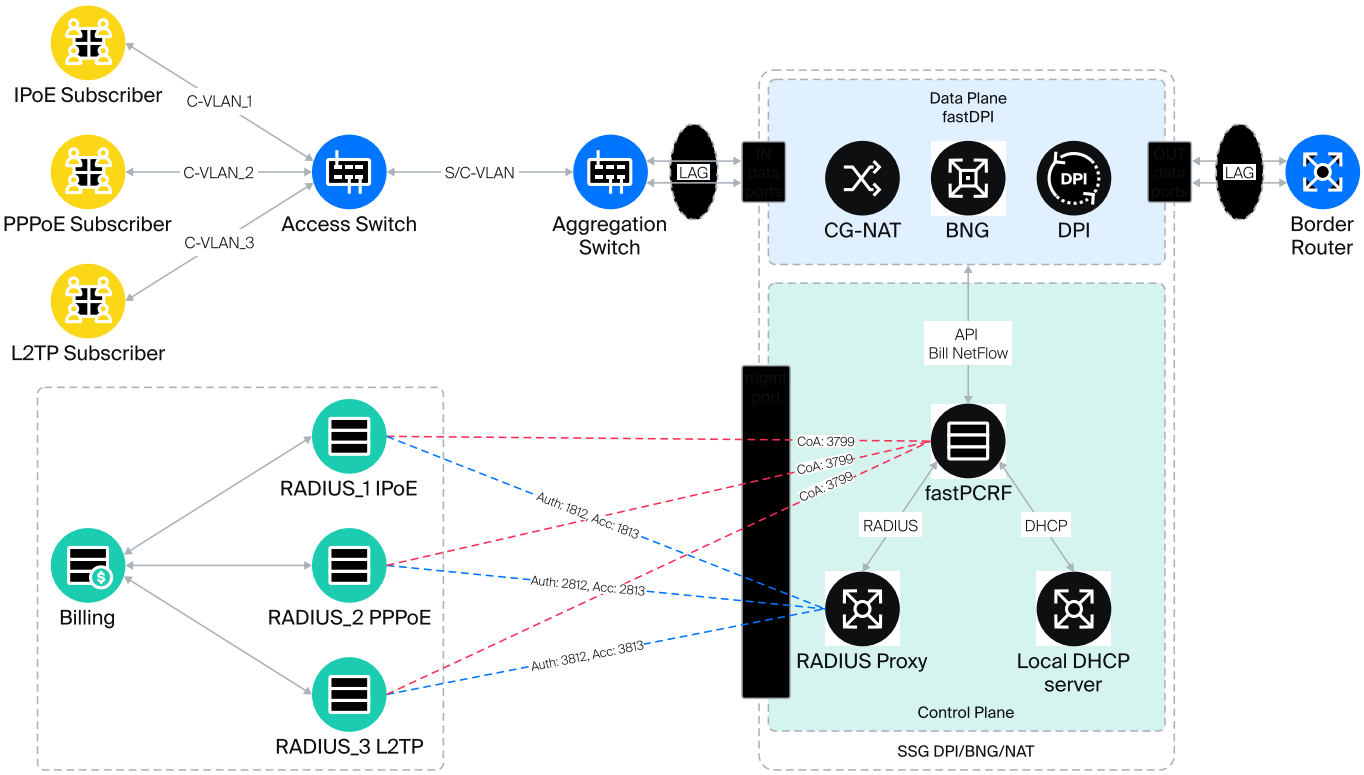
Table of Contents

Configuration of load balancing and distribution across RADIUS server groups

Used for scenarios where it is necessary to distribute subscriber RADIUS traffic with different access types (IPoE, PPPoE, L2TP) across different groups of RADIUS servers.



In this example, we consider the distribution of authorization and accounting in load-balance mode for different access types across RADIUS server groups:

1. IPoE to RADIUS_1 (1812, 1813), RADIUS_1.1 (1822, 1823)
2. PPPoE to RADIUS_2 (2812, 2813), RADIUS_2.1 (2822, 2823)
3. L2TP to RADIUS_3 (3812, 3813), RADIUS_3.1 (3822, 3823)

Separation of auth/accounting for different access types is performed based on the `VasExperts-Service-Type` attribute using branching operators (if-else).

RADIUS CoA messages can be sent from RADIUS servers to BNG (fastPCRF) in the following ways:

1. directly from RADIUS (billing) to fastPCRF. it is required to add separate CoA clients in fastPCRF.
2. via proxy: RADIUS (billing) → proxy → fastPCRF. an example is described below.

FreeRADIUS proxy configuration

The proxy configuration is located in `/etc/raddb/proxy.conf`. it defines the main sections:

```
proxy server{
```

```
  retry_delay = 5
  retry_count = 3
  default_fallback = no
  #dead_time = 120
  wake_all_if_all_dead = yes
}
```

- `retry_delay` — waiting interval (in seconds) after a failed attempt to establish a connection with the server.
- `retry_count` — maximum number of attempts to send a request to the server, after which it is considered "dead".
- `default_fallback` — parameter that determines whether a reject response is sent to the client if all servers are in the "dead" state.
- `wake_all_if_all_dead` — parameter that defines periodic availability checks for servers marked as "dead".

Section that defines parameters of "home" servers (which store subscriber data).

```
home_server rad_1 {
        type = auth+acct
        ipaddr = 10.166.220.232
        port = 1812
        secret = secret
        # proto = udp
# optional items
        src_ipaddr = 10.16.20.117
        response_window = 6
        zombie_period = 40
        status_check = status-server
        check_interval = 6
        check_timeout = 4
        num_answers_to_alive = 2
        max_outstanding = 65536
        coa {
                irt = 2
                mrt = 16
                mrc = 5
                mrd = 30
        }
        limit {
              max_connections = 16
              max_requests = 0
              lifetime = 0
              idle_timeout = 0
        }
}
```

- `type` — server role; most often used for authorization (`auth`) or for authorization and accounting (`auth+acct`).
- `ipaddr` — IPv4 address of the RADIUS server; `ipv6addr` can be used if required.
- `port` — port to which requests are proxied (usually 1812). in `auth+acct` mode, only the

authorization port is specified, and the accounting port is determined as `port+1`.
- `proto` — transport protocol; default is `udp`.
- `secret` — shared secret used to sign and protect packets between the RADIUS server and the proxy.
- `src_ipaddr` — source IP address from which the proxy sends requests.
- `response_window` — response waiting interval; after it expires, the server is marked as "zombie" and gets minimal priority during selection.
- `zombie_period` — maximum period to wait for a response to any packet, after which the server is considered "dead".
- `status_check` — method used to check server state.
- `check_interval` — interval between sending status check packets.
- `check_timeout` — timeout for waiting for a status check response.
- `num_answers_to_alive` — number of consecutive successful checks after which the server is considered "alive".
- `max_outstanding` — maximum number of unacknowledged packets (difference between sent and received); when exceeded, sending new packets is paused to prevent RADIUS server overload.
- `coa` — section describing retry intervals and counts for Change of Authorization.
- `limit` — section applicable only when using TCP. includes parameters:
    - `max_connections` — maximum number of connections;
    - `max_requests` — maximum number of requests within one connection;
    - `lifetime` — connection lifetime in seconds;
    - `idle_timeout` — maximum idle time within a connection after which it is closed.

      a value of `0` for all parameters means no limits.

For the proxy, define the required number of servers. servers responsible for the same authorization type should be grouped into a `pool`. use the `home_server_pool` section for load balancing and failover between servers.
This article provides an example of a minimal configuration; the full configuration is available in the archive.

```
home_server_pool pool_rad_servers {
    type = load-balance
    home_server = rad_1
}
```

It is important that all `home_server` entries are of the same type, i.e., all `auth` or all `auth+acct`.

In the `realm` section, specify which server pool should be used for this realm.

```
realm rlm_prod_servers {
    pool = pool_rad_servers
    nostrip
}
```

When using a pool that contains only authorization servers, apply `auth_pool`; when the pool includes only accounting servers, use `acct_pool`. it is recommended to use a universal `pool` that combines both options.
Proxying of CoA packets should be performed based on the `Operator-Name` attribute via `coa_pool`. to configure CoA, use the file `/etc/raddb/sites-available/coa`, where CoA proxying

parameters are defined. specify the conditions under which a request is routed to the corresponding `realm`, and in each `home_server` define the parameters used for CoA (as a rule, they are set by default).
For working with CoA, it is recommended to use direct interaction between PCRF and RADIUS, since all parameters are configured in `fastpcrf.conf`, and data exchange is performed directly. The description of configuring these functions is provided in the article article.

The `nostrip` parameter is used to disable stripping of the `User-Name` value.

FreeRADIUS virtual server configuration

The configuration file is located at `/etc/raddb/sites-available/default`.
All parameters should be specified for `default`. first, configure the `listen` section, where subnets and ports for listening are defined, as well as the types of accepted messages.

```
server default {
listen {
        type = auth
        ipaddr = *
        port = 0
        limit {
              max_connections = 16
              lifetime = 0
              idle_timeout = 30
        }
}
listen {
        ipaddr = *
        port = 0
        type = acct
        limit {
        }
}
```

Configure the IPv6 section:

```
listen {
        type = auth
        ipv6addr = ::   # any.  ::1 == localhost
        port = 0
        limit {
              max_connections = 16
              lifetime = 0
              idle_timeout = 30
        }
}
listen {
        ipv6addr = ::
        port = 0
```

```
        type = acct
        limit {
        }
}
```

Next, proceed to the authorization section, which lists supported authentication protocols.
In this section, define request routing rules: all authorization requests with the `VasExperts-Service-Type` attribute with values `0` and `1` should be sent to the DHCP realm; values 2, 3, and 4 — to the PPPoE realm; all other requests should be rejected. attribute values for other authentication types with explanations are provided in the vasexperts dictionary and in the article.

```
authorize {
        preprocess
        chap
        mschap
        digest
        suffix
        files
        -sql
        -ldap
        expiration
        logintime
        if (Tunnel-Type) {
            update control {
                Proxy-To-Realm := "rlm_prod_servers_3"
           }
        }

        else {
              if (VasExperts-Service-Type == 0  || VasExperts-Service-Type
== 1) {
          update control {
                Proxy-To-Realm := "rlm_prod_servers_1"
          }
        }
  else ( VasExperts-Service-Type == 2 || VasExperts-Service-Type == 3 ||
VasExperts-Service-Type == 4 ) {
                update control {
                      Proxy-To-Realm := "rlm_prod_servers_2"
                }
              }

              else {
                  reject
              }
        }
}
```

This section defines the authentication methods that FreeRADIUS will use. since they are empty, default modules will be applied.

```
authenticate {
        Auth-Type PAP {
        }
        Auth-Type CHAP {
        }
        Auth-Type MS-CHAP {
        }
}

preacct {
        preprocess
        acct_unique
        suffix
        files
}
```

The billing and accounting section, which defines packet handling rules with different `VasExperts-Service-Type` values, similarly to the authorization module. a key difference is the first rule for `Acct-Status-Type`, based on which the proxy forwards general accounting start packets to RADIUS instead of dropping them.

```
accounting {
        -sql
        if (Acct-Status-Type == Accounting-On || Acct-Status-Type ==
Accounting-Off) {
            update control {
                Proxy-To-Realm := "rlm_acct_servers"
           }
         }

        else {
           if (Tunnel-Type) {
            update control {
                Proxy-To-Realm := "rlm_prod_servers_3"
           }
        }
        else {
              if (VasExperts-Service-Type == 0 || VasExperts-Service-Type ==
1) {
                update control {
                   Proxy-To-Realm := "rlm_prod_servers_1"
                }
               }
               else   (VasExperts-Service-Type == 2 || VasExperts-Service-
Type == 3 || VasExperts-Service-Type == 4 ) {
                  update control {
                    Proxy-To-Realm := "rlm_prod_servers_2"
               }
               }
          }
     }
```

```
}
session {
}
```

Final sections. set the session timeout parameter and define system behavior on reject.

```
post-auth {
        update reply {
                Session-Timeout := 4294967295
        }
        -sql
        Post-Auth-Type REJECT {
        }
        Post-Auth-Type Challenge {
        }
}
pre-proxy {
}
post-proxy {
}
}
```