

OmniPGW

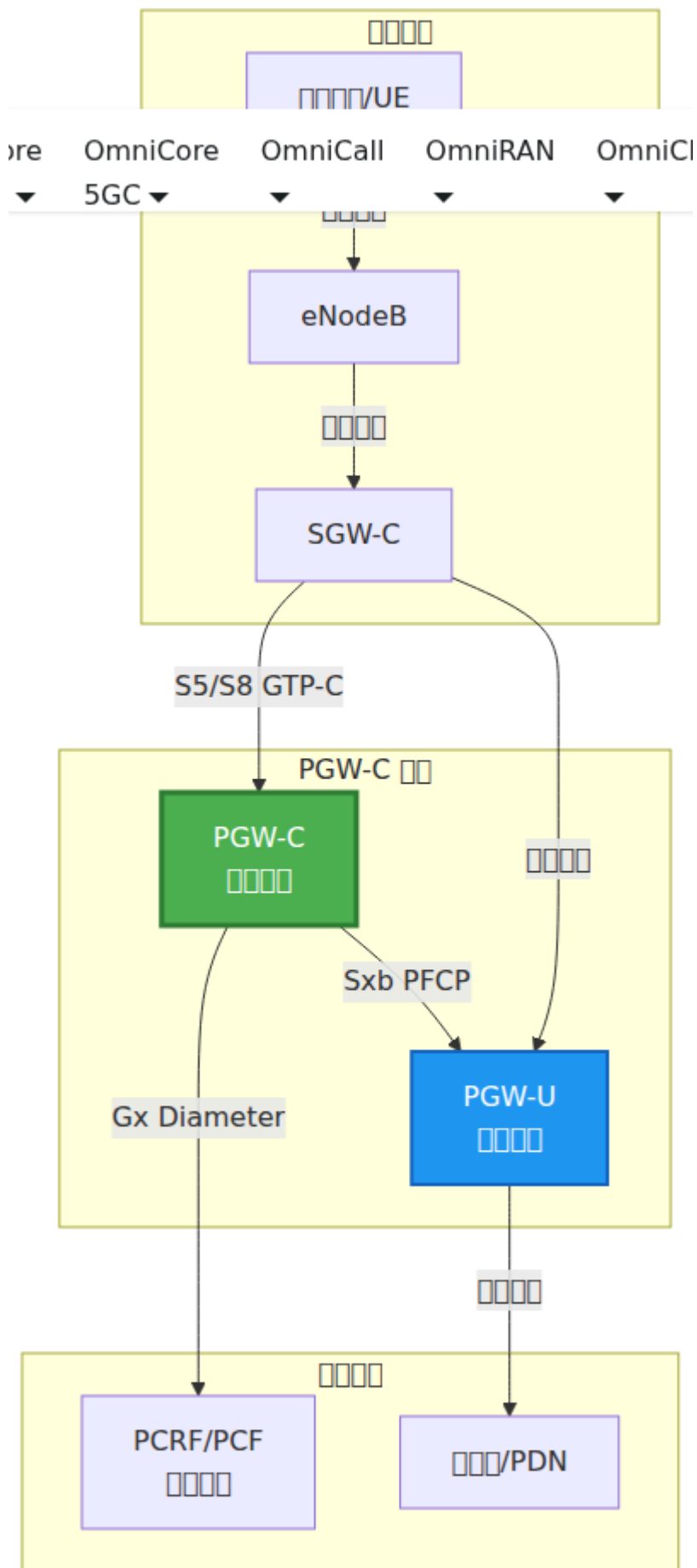
OmniPGW - (PGW-C)

Omnitouch

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
7. Web UI -
- 8.
- 9.
- 10.

OmniPGW (PGW-C) 3GPP LTE (EPC) Omnitouch

- - UE
- IP - IP
- - PCRF
- - PGW-U



PGW-C

- **SGW-C** S5/S8 (GTP-C)
 - **UE IP**
 - **Gx (Diameter) PCRF**
 - **Sxb (PFCP) PGW-U**
 - **QoS QoS**
 -
-

- [illegible]

PGW-C 3GPP

S5/S8 □□ (GTP-C v2)

SGW-C PGW-C

UDP → GTP-C → 2

□□□□□

- 000000/00
- 000000/00
- 000000/00
- 000000/00

□□□ □□□ S5/S8 □□

Sxb (PFCP)

PGW-C PGW-U

UDP PFCP

□ □ □ □ □

- □□□□□□/□□
- □□□□□□/□□

- 3GPP TS 29.261
- 3GPP TS 29.262
- 3GPP TS 29.263

3GPP TS 29.261 PFCP/Sxb 3GPP TS 29.262

Gx (Diameter)

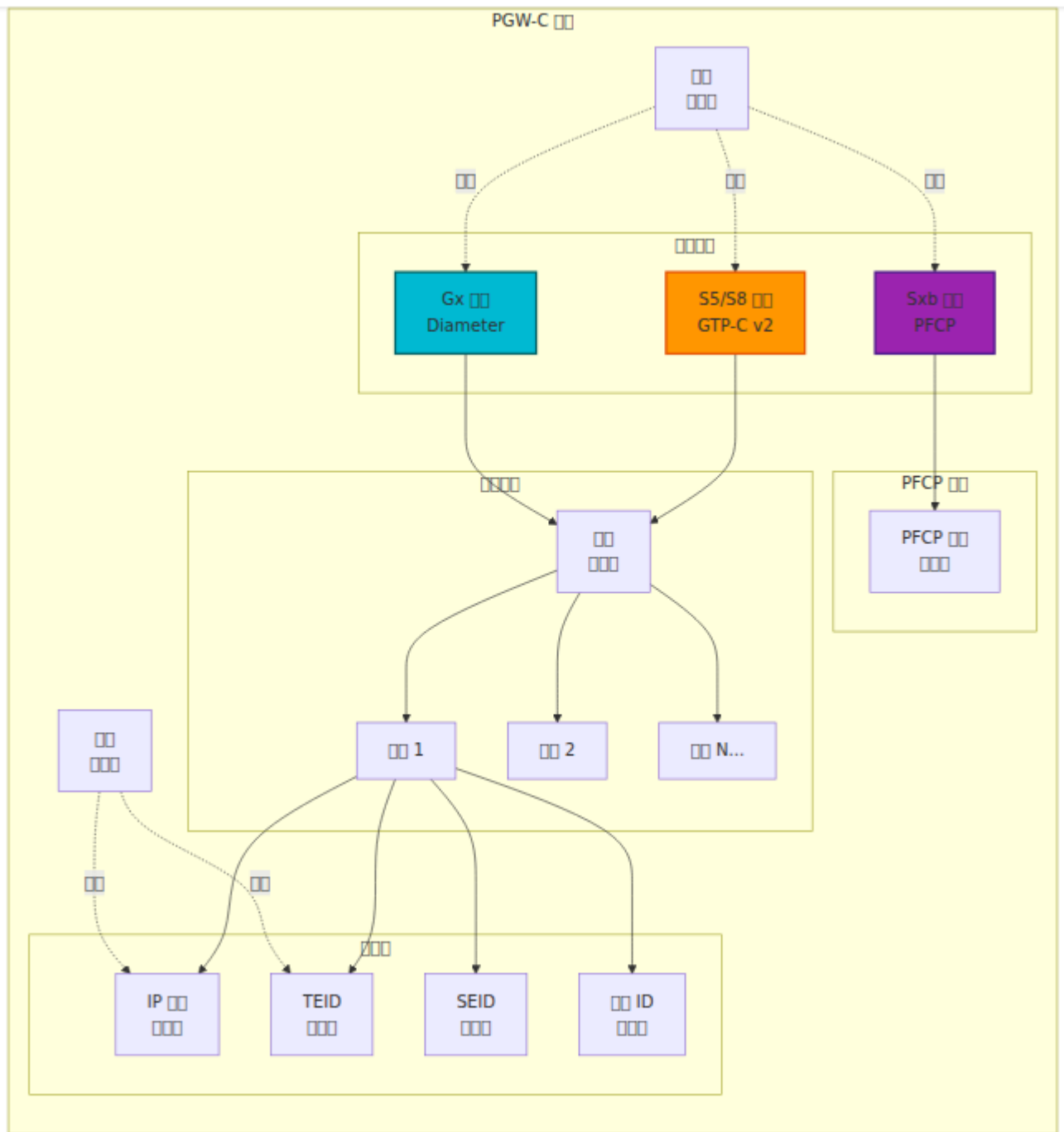
3GPP TS 29.262 (PCRF) 3GPP TS 29.263

3GPP TS 29.263 Diameter (IETF RFC 6733)

3GPP TS 29.263

- 3GPP TS 29.263 (CCR-I/CCA-I)
- 3GPP TS 29.263 (CCR-T/CCA-T)

3GPP TS 29.263 Diameter Gx 3GPP TS 29.263



PDN

PDN UE

- UE IP -

- **APN** - 網路位址
- - QoS
- **ID** -
- **TEID** ID - S5/S8
- **SEID** ID - Sxb

網路位址

網路位址 QoS

- - PDN
- - QoS
- **EBI** EPS ID -
- **QoS** - QCI ARP MBR GBR

PFCP

PGW-C PGW-U

- **PDR** -
- **FAR** -
- **QER** QoS -
- **BAR** -

PFCP

IP

UE IP

- **APN** - APN
- - IP
- - UE IP
- - IP

UE IP

□□

□□□□

- Elixir ~1.16
- Erlang/OTP 26+
- □ SGW-C□PGW-U □ PCRF □□□□
- □□ LTE EPC □□

□□ OmniPGW

1. □ `config/runtime.exs` □□□□□□□□
2. □□□□□□□□

```
mix deps.get
mix compile
```

3. □□□□□□□□

```
mix run --no-halt
```

□□□□

□□□□□□□□□□□□

```
[info] □□ OmniPGW...
[info] □ 127.0.0.42:42069 □□□□□□□□
[info] □ 127.0.0.10 □□ S5/S8 □□
[info] □ 127.0.0.20 □□ Sxb □□
[info] □□ Gx □□
[info] □□ PFCP □□□□□□
[info] OmniPGW □□□□
```

□□□□□□ `http://127.0.0.42:42069/metrics` □□□□□□□□

```

00000000 config/runtime.exs 00000000000000000000000000000000

```

[illegible]

Web UI

http://<omnipgw-ip>:<web-port>/

Table with 4 columns: Component, URL, Description, and Count.

Component	URL	Description	Count
UE	/ue_search	UE Search	1
PGW	/pgw_sessions	PGW Sessions	2
Session	/session_history	Session History	5
Topology	/topology	Network Topology	5
IP	/ip_pools	UE IP Pools	2
PFCP	/pfcpsessions	PGW-U PFCP Sessions	2
UPF	/upf_status	UPF PFCP Sessions	2
UPF	/upf_selection	UPF Selection P-CSCF	1
Diameter	/diameter	PCRF Diameter	1
P-CSCF	/pcscf_monitor	P-CSCF DNS Monitor	5
Gy	/gy_simulator	Gy/Ro Simulator	1
Cell	/cell_towers	OpenCellID	1
Logs	/logs	Logs	1

□□□□

□□□□

- □□□□□□□□□□□□□□□□
- □□ OmniPGW □□□□□□□□
- □□□□□□□□□□□□□□□□/□□□

□□□□□□

- □ IMSI□IP□MSISDN □ APN □□□□
- □□□□□□□□□□□□□□

□□□□□□□□

- □□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□

□□□□□□□□□□□□

- □□□□□□□□□□
- □□□□ NOC/□□□□□□□□
- □□□□□□ IP □□□□□□

□□□□□□□□

□□□□□□□□□□□□□□

1. 검색
2. UE 검색 (/ue_search)
3. IMSI/MSISDN IP 검색
4. 검색
 - a) 검색 - 검색
 - b) 검색 - TAC/Cell ID
 - c) 검색 -
 - QCI/MBR/GBR
 - APN-AMBR
 - d) 검색 - Gy ID
 - e) 검색 - Gx PCC
 - f) 검색 -
5. 검색 → Diameter PCRF
6. 검색 →

검색

1. 검색
2. PGW (/pgw_sessions)
3. IMSI
4.
 - UE IP
 - QoS
 -
5. 검색 → UE

검색

1. UPF → PGW-U “”
2. Diameter → PCRF “”
3. PGW →

검색

- PGW
- /
-
- APN

Web UI

Web UI

- UE
-
- PFCP Diameter
-
- IMSI/MSISDN/IP
-
- QoS MBR GBR QCI
-
-
- IP
-

Prometheus

-
-
-
-
-

- Web UI Prometheus

Web UI OmniPGW Prometheus

-

- teid_registry_count - S5/S8
 - seid_registry_count - PFCP
 - session_id_registry_count - Gx
 - address_registry_count - UE IP
 - charging_id_registry_count - ID
-
- s5s8_inbound_messages_total - GTP-C
 - sxb_inbound_messages_total - PFCP
 - gx_inbound_messages_total - Diameter
 -
-
- s5s8_inbound_errors_total - S5/S8
 - sxb_inbound_errors_total - PFCP
 - gx_inbound_errors_total - Diameter

□ □ □ □

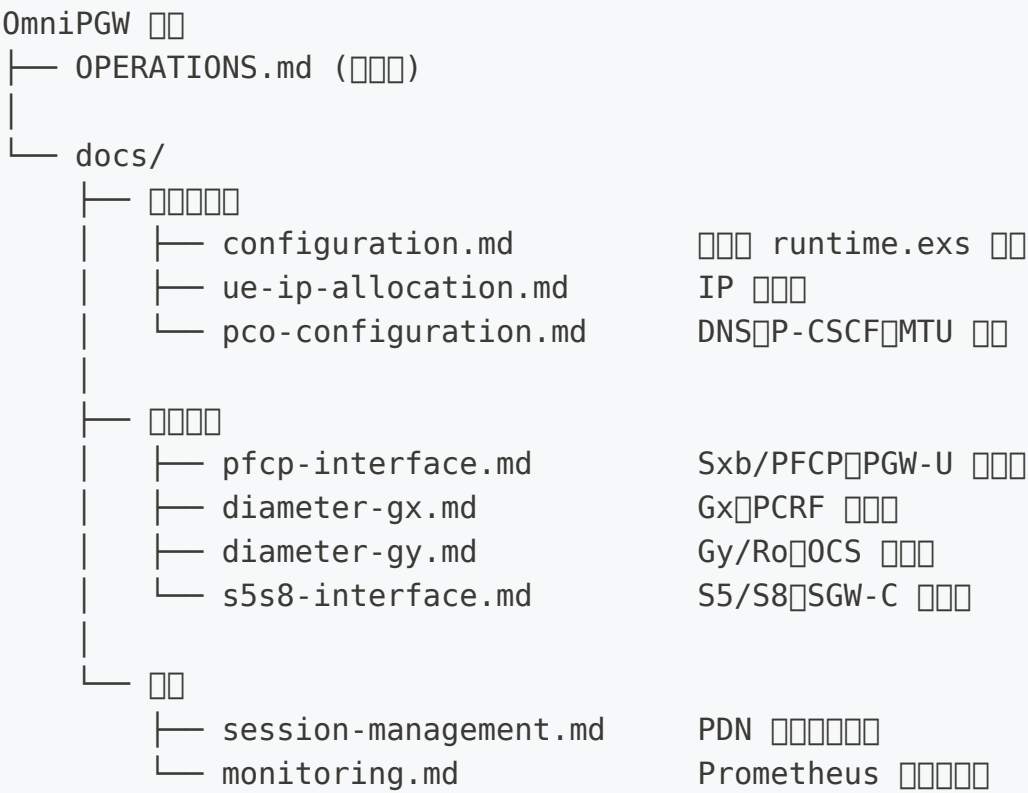
□□□□ HTTP □□□□□□□□□□

```
curl http://127.0.0.42:42069/metrics
```

□ □ □ □

OmniPGW

0000



0000000000

0 00

00	00	00
OPERATIONS.md	000000000000	00000000

⚙ 00

파일	설명	라인 수
configuration.md	runtime.exs 관련 설정	1,600+
ue-ip-allocation.md	UE IP 할당 관련	943
pco-configuration.md	DNS, P-CSCF, MTU 관련	344

이 문서들은

파일	설명	라인 수
pfcp-interface.md	PFCP/Sxb 관련 PGW-U	1,355
diameter-gx.md	Diameter Gx 관련 PCRF 관련	941
diameter-gy.md	Diameter Gy/Ro 관련 OCS 관련	1,100+
s5s8-interface.md	GTP-C S5/S8 관련 SGW-C	456

이 문서들은

파일	설명	라인 수
session-management.md	PDN 관련	435
monitoring.md	Prometheus, Grafana 관련	807
data-cdr-format.md	CDR 관련 URR 관련	847
qos-bearers.md	QoS 관련	448
troubleshooting.md	문제 해결 관련	687

이 문서들은

項目	項目	項目
pcscf-monitoring.md	P-CSCF 監視機能	894

目次

■ Mermaid 図

監視機能 Mermaid 図 監視機能

- 監視
- 監視機能
- 監視
- 監視

図 監視

監視機能

- 監視機能
- 監視機能
- 監視

図 監視

監視機能

- 監視機能
- 監視
- 監視

図 監視

監視機能

目次

概要

1. [OPERATIONS.md](#) - 運用
2. [configuration.md](#) - 設定
3. [monitoring.md](#) - 監視
4. [session-management.md](#) - セッション管理

ネットワーク

1. [OPERATIONS.md](#) - ネットワーク運用
2. [pfcg-interface.md](#) - pfcgインターフェース
3. [diameter-gx.md](#) - diameter-gx
4. [diameter-gy.md](#) - diameter-gy
5. [s5s8-interface.md](#) - s5s8インターフェース
6. [ue-ip-allocation.md](#) - IP 割り当て

セキュリティ

1. [configuration.md](#) - セキュリティ設定
2. [ue-ip-allocation.md](#) - IP 割り当て
3. [pco-configuration.md](#) - PCO設定
4. [monitoring.md](#) - セキュリティ監視

仕様

- 仕様書 14
- 仕様数 ~10,900+
- 仕様数 ~265 KB
- **Mermaid** 仕様 75+
- 仕様数 150+

開発環境

OS

- 消息/消息内容
- 消息 OTP/Elixir 消息
- 消息内容
- 消息 GenServer 消息

消息

- 消息 PFCP 消息内容
- 消息 GTP-C v2 消息 GPRS 消息内容
- 消息 Diameter 消息 RFC 6733 消息

3GPP 消息

- 消息 Sxb 消息 PGW-C ↔ PGW-U 消息
- 消息 Gx 消息 PGW-C ↔ PCRF 消息
- 消息 Gy/Ro 消息 PGW-C ↔ OCS 消息
- 消息 S5/S8 消息 SGW-C ↔ PGW-C 消息

消息

- 消息内容
 - 消息 IP 消息内容
 - 消息 QoS 消息
 - 消息内容
 - 消息内容
-

□□□□

3GPP □□

□□	□□
TS 29.274	GTP-C v2□S5/S8 □□□
TS 29.244	PFCP□Sxb □□□
TS 29.212	Diameter Gx □□□□□□□□
TS 32.299	Diameter □□□□□Gy/Ro□
TS 32.251	□□□□□□□□
TS 23.401	EPC □□

□□□□

- □□□□□ [config/runtime.exs](#)

OmniPGW

runtime.exe ██████████

□ Omnitouch □□□□□□

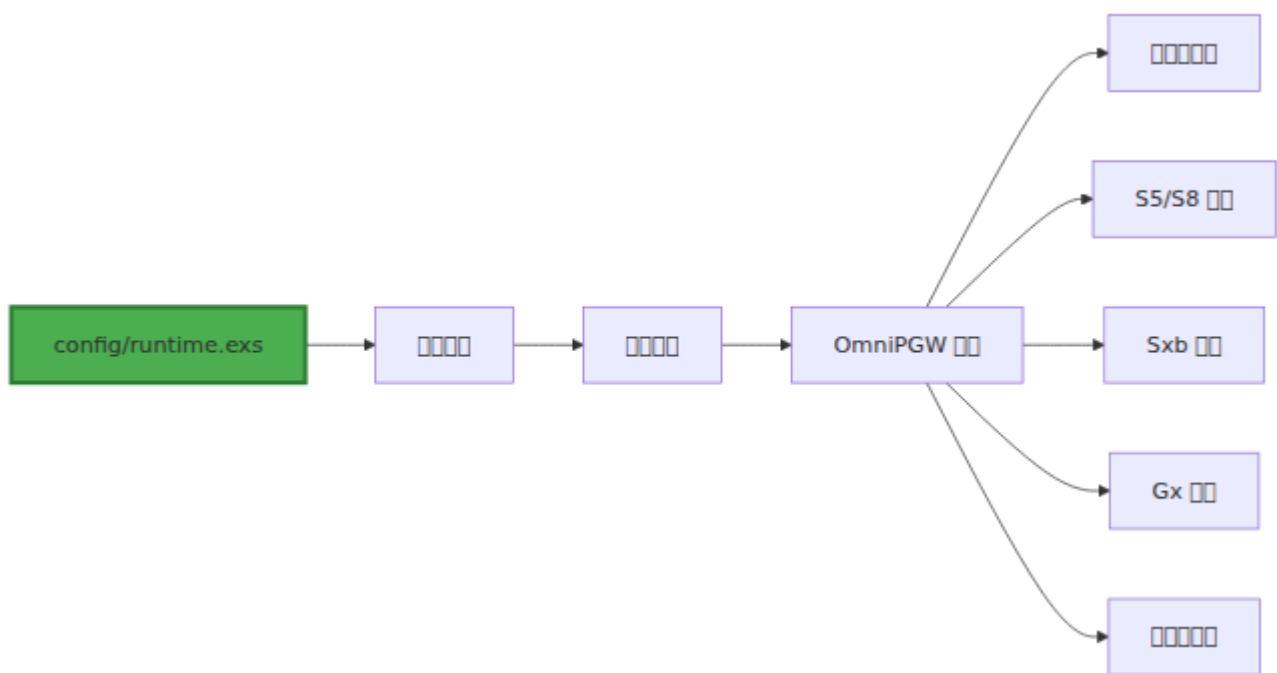
11

1. **IP**
2. **IPsec**
3. **IPsec**
4. **Diameter/Gx**
5. **S5/S8**
6. **Gn/Gp** (GGSN)
7. **Sxb/PFCP**
 - **UPF**
 - **IPsec**
 - **DNS**
 - **IPsec**
8. **UE IP**
9. **PCO**
10. **Web UI**
11. **IPsec**
12. **IPsec**

11

OmniPGW 的配置文件 `config/runtime.exs` 中配置了数据库连接信息。

□□□□



□□□□

- □□□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□
- □□□□□ - □□□□□□□□□□
- □□□□□ - □□□□□□□□□□□□

□□□□□□□

□□□□

```
pgw_c/
├─ config/
│   ├── config.exs      # □□□□□□□ runtime.exs□
│   ├── dev.exs         # □□□□□□□
│   ├── prod.exs        # □□□□□□□
│   └─ runtime.exs      # ← □□□□□□□
```

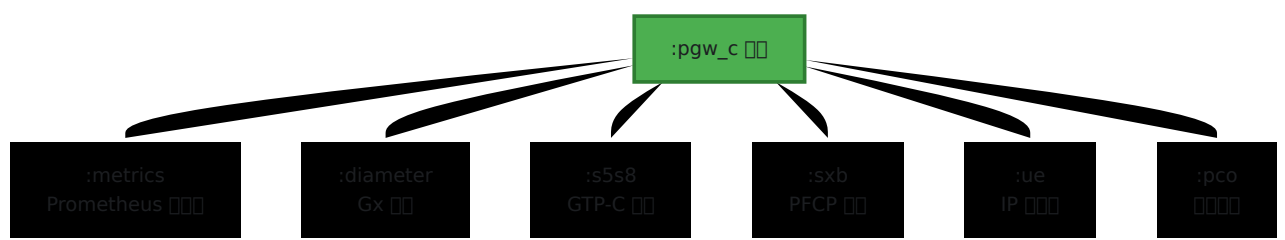
配置

```
# config/runtime.exs
import Config

config :logger, level: :info

config :pgw_c,
  metrics: %{...},
  diameter: %{...},
  s5s8: %{...},
  sxb: %{...},
  ue: %{...},
  pco: %{...}
```

架构图



部署

部署

部署 Prometheus 和 OmniPGW

配置

```
config :pgw_c,
  metrics: %{
    # 是否开启
    enabled: true,

    # 监听 HTTP 请求 IP 地址
    ip_address: "0.0.0.0",

    # 监听端口
    port: 9090,

    # 注册中心轮询周期
    registry_poll_period_ms: 10_000
  }
```

参数

参数名	类型	默认值	说明
enabled	布尔	true	是否开启
ip_address	字符串 IP地址	"0.0.0.0"	监听地址 0.0.0.0 = 所有地址
port	整数	9090	HTTP 请求端口 /metrics
registry_poll_period_ms	整数	10_000	注册中心轮询周期

注意

ip - 监听地址 IP

```
metrics: %{
  enabled: true,
  ip_address: "10.0.0.20", # 10.0.0.20
  port: 9090,
  registry_poll_period_ms: 5_000 # 5 000ms
}
```

00 - 00000000

```
metrics: %{
  enabled: true,
  ip_address: "127.0.0.1",
  port: 42069, # 42069
  registry_poll_period_ms: 10_000
}
```

000000

```
metrics: %{
  enabled: false
}
```

00000

```
# 0000
curl http://<ip_address>:<port>/metrics

# 00
curl http://10.0.0.20:9090/metrics
```

00 00000000 000000000000

Diameter/Gx 配置

配置

配置 Diameter 服务器 Gx 接口 PCRF 服务器

配置

```
config :pgw_c,  
  diameter: %{  
    # Diameter 服务器 IP 地址  
    listen_ip: "0.0.0.0",  
  
    # OmniPGW 与 Diameter 服务器 Origin-Host  
    host: "omnipgw.epc.mnc001.mcc001.3gppnetwork.org",  
  
    # OmniPGW 与 Diameter 服务器 Origin-Realm  
    realm: "epc.mnc001.mcc001.3gppnetwork.org",  
  
    # PCRF 服务器  
    peer_list: [  
      %{  
        # PCRF Diameter 服务器  
        host: "pcrf.epc.mnc001.mcc001.3gppnetwork.org",  
  
        # PCRF 服务器  
        realm: "epc.mnc001.mcc001.3gppnetwork.org",  
  
        # PCRF IP 地址  
        ip: "10.0.0.30",  
  
        # 是否 PCRF 服务器  
        initiate_connection: true  
      }  
    ]  
  }  
}
```

PCRF

項目	項目	項目	項目
listen_ip	PCRF IP	項目	Diameter 項目
host	PCRF FQDN	項目	OmniPGW 項目 Origin-Host項目 FQDN
realm	項目	項目	OmniPGW 項目 Origin-Realm
peer_list	項目	項目	PCRF 項目

PCRF 項目

項目	項目	項目	項目
host	PCRF FQDN	項目	PCRF Diameter 項目
realm	項目	項目	PCRF 項目
ip	PCRF IP	項目	PCRF IP 項目
initiate_connection	項目	項目	OmniPGW 項目 PCRF

FQDN 項目

Diameter 項目 FQDN

```
# 項目
host: "omnipgw.epc.mnc001.mcc001.3gppnetwork.org"

# 項目
host: "omnipgw"          # 項目 FQDN
host: "10.0.0.20"        # 項目 IP
```

3GPP 項目


```
<hostname>.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org
```

□□□

- omnigw.epc.mnc001.mcc001.3gppnetwork.org (MCC=001, MNC=001)
- pgw-c.epc.mnc260.mcc310.3gppnetwork.org (MCC=310, MNC=260 - □□ T-Mobile)

□□

□□ **PCRF**□

```
diameter: %{  
  listen_ip: "0.0.0.0",  
  host: "omnigw.epc.mnc001.mcc001.3gppnetwork.org",  
  realm: "epc.mnc001.mcc001.3gppnetwork.org",  
  peer_list: [  
    %{  
      host: "pcrf.epc.mnc001.mcc001.3gppnetwork.org",  
      realm: "epc.mnc001.mcc001.3gppnetwork.org",  
      ip: "10.0.0.30",  
      initiate_connection: true  
    }  
  ]  
}
```

□□ **PCRF**□□□□□

```
diameter: %{
  listen_ip: "0.0.0.0",
  host: "omnipgw.epc.mnc001.mcc001.3gppnetwork.org",
  realm: "epc.mnc001.mcc001.3gppnetwork.org",
  peer_list: [
    %{
      host: "pcrf-primary.epc.mnc001.mcc001.3gppnetwork.org",
      realm: "epc.mnc001.mcc001.3gppnetwork.org",
      ip: "10.0.1.30",
      initiate_connection: true
    },
    %{
      host: "pcrf-backup.epc.mnc001.mcc001.3gppnetwork.org",
      realm: "epc.mnc001.mcc001.3gppnetwork.org",
      ip: "10.0.2.30",
      initiate_connection: true
    }
  ]
}
```

PCRF 配置

```
diameter: %{
  listen_ip: "0.0.0.0",
  host: "omnipgw.epc.mnc001.mcc001.3gppnetwork.org",
  realm: "epc.mnc001.mcc001.3gppnetwork.org",
  peer_list: [
    %{
      host: "pcrf.epc.mnc001.mcc001.3gppnetwork.org",
      realm: "epc.mnc001.mcc001.3gppnetwork.org",
      ip: "10.0.0.30",
      initiate_connection: false # PCRF 不连接
    }
  ]
}
```

配置 Diameter Gx 接口

S5/S8

GTP-C SGW-C

```
config :pgw_c,  
  s5s8: %{  
    # S5/S8 IPv4  
    local_ipv4_address: "10.0.0.20",  
  
    # IPv6  
    local_ipv6_address: nil,  
  
    # GTP-C 2123  
    local_port: 2123,  
  
    # GTP-C 500ms  
    # GTP-C  
    request_timeout_ms: 500,  
  
    # GTP-C 3  
    # = request_timeout_ms * request_attempts  
    request_attempts: 3  
  }
```

이름

이름	타입	값	설명
local_ipv4_address	문자열IPv4	없음	S5/S8 인스턴스 IPv4 주소
local_ipv6_address	문자열IPv6	nil	S5/S8 인스턴스 IPv6 주소 (없음)
local_port	정수	2123	GTP-C 포트 UDP 포트번호 2123
request_timeout_ms	정수	500	요청 타임아웃 (밀리초)
request_attempts	정수	3	요청 시도 횟수

참고 사항

- 인스턴스 GTP-C 포트 2
- 인스턴스 UDP
- 포트번호 2123
- 인스턴스 SGW-C 인스턴스

이름

인스턴스 IPv4 주소

```
s5s8: %{\n  local_ipv4_address: "10.0.0.20"\n}
```

IPv4 + IPv6 인스턴스

```
s5s8: %{\n  local_ipv4_address: "10.0.0.20",\n  local_ipv6_address: "2001:db8::20"\n}
```

配置例

```
s5s8: %{\n  local_ipv4_address: "10.0.0.20",\n  local_port: 2124 # 送信ポート\n}
```

設定値

```
s5s8: %{\n  local_ipv4_address: "10.0.0.20",\n  request_timeout_ms: 1500, # 送信タイムアウト 1.5 秒\n  request_attempts: 3      # 送信回数 4.5 秒\n}
```

単位

S5/S8 送信待ち時間 GTP-C 送信待ち時間

計算式

送信待ち時間 = request_timeout_ms × request_attempts
500ms × 3 = 1.5 秒

単位換算

送信待ち時間	送信待ち時間	送信待ち時間
短時間 (<50ms)	200-300ms	600-900ms
中時間 (50-150ms)	500ms程度	1.5s
長時間 (>150ms)	1000-2000ms	3-6s
送信待ち時間	2000-3000ms	6-9s

Wireshark

- Wireshark “NetworkMiner” Wireshark
- Wireshark NetworkMiner
- Wireshark NetworkMiner

NetworkMiner

- NetworkMiner "NetworkMiner"
- PCRF Diameter 5012
- Charging-Rule-Remove

IP

IP

- IP/Port
- SGW-C
- VRRP/HSRP

iptables

```
# SGW-C GTP-C
iptables -A INPUT -p udp --dport 2123 -s <sgw_c_network> -j ACCEPT
```

Gn/Gp (GGSN)

IP

GTP-C v1 SGSN GGSN 2G/3G

配置

```
config :pgw_c,
  gn: %{
    # Gn 网元 IPv4 地址
    local_ipv4_address: "10.0.0.20",

    # 网元 IPv6 地址 2G/3G 网元 IPv4 地址
    local_ipv6_address: nil,

    # GTP-C 网元 S5/S8 网元
    local_port: 2123
  },

  # PC0 网元 DNS 网元 S5/S8 网元
  dns: %{
    primary_ipv4: {8, 8, 8, 8},
    secondary_ipv4: {8, 8, 4, 4}
  }
}
```

数据

属性	值	类型	描述
local_ipv4_address	10.0.0.20	IPv4	Gn 网元 IPv4 地址 GGSN 网元
local_ipv6_address	nil	IPv6	Gn 网元 IPv6 地址
local_port	2123	UDP	UDP 网元 GTP-C v1 网元 S5/S8 网元

DNS

IP	Port	IP	Port
primary_ipv4	53	{8, 8, 8, 8}	PCO 53 DNS 53
secondary_ipv4	53	{8, 8, 4, 4}	PCO 53 DNS 53

PCO

- GTP-C 1 (3GPP TS 29.060)
- UDP
- 2123 GTP-C v2
- SGSN

S5/S8

OmniPGW PGW 4G GGSN 2G/3G

- UDP 2123
- GTP
- IP 4G 2G/3G
- IP UPF/PFCP

Gn/Gp

Sxb/PFCP

PFCP PGW-U

111

```
config :pgw_c,
  sxb: %{
    # PFCP 本地 IP 地址
    local_ip_address: "10.0.0.20",

    # 本地 PFCP 端口
    local_port: 8805
  }
```

11

項目	項目	項目	項目
local_ip_address	送信元IP	ポート	PFCP 送信元
local_port	ポート	8805	PFCP UDP 送信元

103

- **UPF** **upf_selection** + **UPF**
- **UPF** **UPF**
 - **"UPF-<ip>:<port>"**
 - **PF** **UPF**
 - **5**
- **UPF** **upf_selection** **UPF**
- **UPF** **DNS** **UPF**

11

□ □ □ □ □

```
sxb: %{
  local_ip_address: "10.0.0.20"
}

# [] upf_selection [] UPF []
# - []`"UPF-10.0.0.21:8805"`
# - [] PCFP [] UPF []
# - 5 []
```

[] **PCFP** []

```
sxb: %{
  local_ip_address: "10.0.0.20",
  local_port: 8806 # [] PCFP []
}
```

[] **UPF** []

```
sxb: %{
  local_ip_address: "10.0.0.20"
},
upf_selection: %{
  rules: [
    %{
      name: "IMS ",
      priority: 10,
      match_field: :apn,
      match_regex: ~r/^ims$/,
      upf_pool: [
        %{remote_ip_address: "10.0.1.21", remote_port: 8805,
weight: 100},
        %{remote_ip_address: "10.0.1.22", remote_port: 8805,
weight: 100}
      ]
    }
  ],
  fallback_pool: [
    %{remote_ip_address: "10.0.2.21", remote_port: 8805, weight:
100}
  ]
}
# 3 UPF10.0.1.2110.0.1.2210.0.2.210000
```

DNS

```
sxb: %{
  local_ip_address: "10.0.0.20"
},
upf_selection: %{
  dns_enabled: true,
  dns_query_priority: [:ecgi, :tai],
  dns_suffix: "epc.3gppnetwork.org",
  fallback_pool: [
    %{remote_ip_address: "10.0.2.21", remote_port: 8805, weight:
100}
  ]
}
# DNS UPF
```

UPF ☐ ☐ ☐ ☐

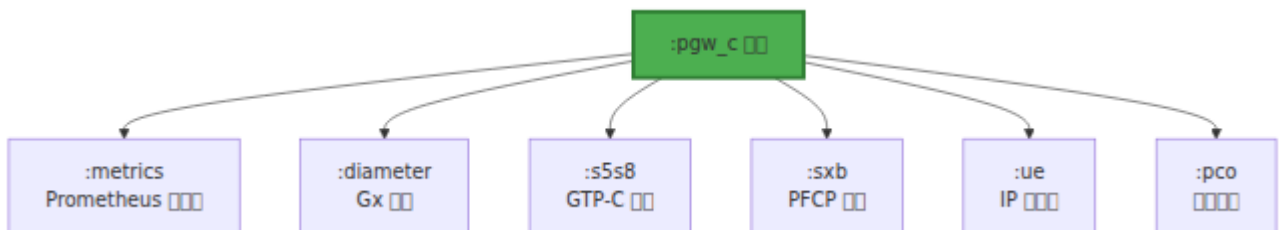
UPF 1000000000 UPF 1000000000 upf_selection 1000000000

UPF

1. **IP** - **IP** **Address**
2. **DNS** **Server** - **DNS** **Server** **UPF** **Server**
3. **Port** - **Port** **DNS** **Server**

1. 網際網路服務 - 網際網路服務
2. 網際網路服務 - 網際網路 UPF 網
3. 網際網路服務 - 網際網路 DNS 網

UPF □□□□□□



□ □ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

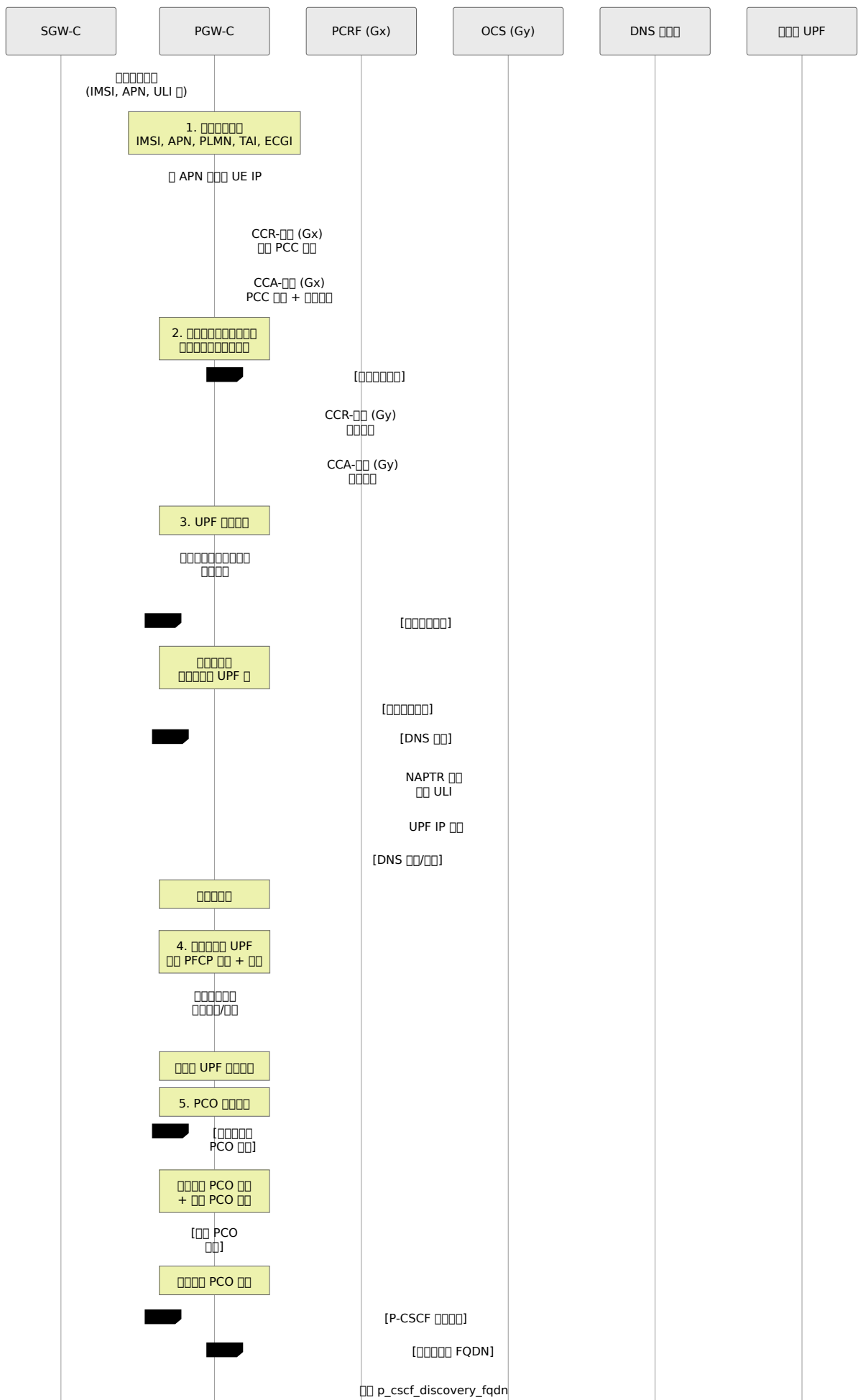
項目名	説明	形式
:imsi	IMSI	^313380.*
:apn	APN / DNN	^internet\. ^ims\.
:serving_network_plmn_id	PLMN ID	^313380\$
:sgw_ip_address	SGW IP	^10\.100\..*
:uli_tai_plmn_id	ULI PLMN ID	^313.*
:uli_ecgi_plmn_id	E-UTRAN PLMN ID	^313.*

設定項目

項目名	説明	形式	値
UPF	UPF	IPアドレス	UPF
APN	APN	IMS/	
IMSI	IMSI		
DNS	MEC/		DNS
		UPF	

□□□□□□□□

□□□□□□□□□□□□□□□□□□ UPF □□□ PCO □□□



- 網元 PCO 網元 網元 PCO 網元

3. P-CSCF 網元

- 網元 FQDN → 網元 DNS 網元 → 網元 PCO → 網元 PCO
- 網元 P-CSCF 網元 網元網元網元網元網元

4. 網元

- PCRF 網元網元網元網元網元 + 網元=1網元
- OCS 網元網元網元網元
- PGW-C 網元網元網元 CCR-網元網元
- 網元 Diameter Gx 網元 網元 Diameter Gy 網元 網元網元網元網元

網元網元網元

網元網元網元網元網元 UPF 網元網元網元網元

```

config :pgw_c,
  # PFCP [] - [] UPF [] upf_selection []
  sxb: %{
    local_ip_address: "127.0.0.20"
  },

  # UPF [] - [] UPF []
  upf_selection: %{
    # [] DNS []
    dns_enabled: false,
    dns_query_priority: [:ecgi, :tai, :rai, :sai, :cgi],
    dns_suffix: "epc.3gppnetwork.org",
    dns_timeout_ms: 5000,

    # []
    rules: [
      # [] 1[]IMS [] - []
      %{
        name: "IMS []",
        priority: 20,
        match_field: :apn,
        match_regex: "^ims",
        upf_pool: [
          weight: 80,
          %{remote_ip_address: "10.100.2.21", remote_port: 8805,
weight: 80},
          weight: 20,
          %{remote_ip_address: "10.100.2.22", remote_port: 8805,
weight: 20}
        ]
      },

      # [] 2[] APN
      %{
        name: "[]",
        priority: 15,
        match_field: :apn,
        match_regex: "^(enterprise|corporate)\\.apn",
        upf_pool: [
          weight: 100,
          %{remote_ip_address: "10.100.3.21", remote_port: 8805,
weight: 100}
        ]
      },

      # [] 3[] - []

```

```

    %{
      name: "互联网",
      priority: 5,
      match_field: :apn,
      match_regex: "^internet",
      upf_pool: [
        %{remote_ip_address: "10.100.1.21", remote_port: 8805,
weight: 33},
        %{remote_ip_address: "10.100.1.22", remote_port: 8805,
weight: 33},
        %{remote_ip_address: "10.100.1.23", remote_port: 8805,
weight: 34}
      ]
    },

    # 备用 - 本地 DNS 服务器
    fallback_pool: [
      %{remote_ip_address: "127.0.0.21", remote_port: 8805,
weight: 100}
    ]
  }

```

配置

配置

- 在 `upf_selection` 中配置 UPF 列表
- 在 `upf_selection` 中配置 UPF 权重
- 在 `upf_pool` 中配置 UPF 池
- 在 `fallback_pool` 中配置备用池
- 在 **DNS** 中配置 DNS 服务器
- 在 `DNS` 中配置 UPF 权重
- 在 `upf_selection` 中配置 UPF 权重 5 倍

配置

- 在 `sxb.peer_list` 中配置
- 在 `selection_list` 中配置

- 网络 UPF 选择策略 `upf_selection` 策略

UPF 选择策略

1. 网络选择策略 UPF 选择策略

- 网络 = PCF 网络 + 网络 3 网络
- 网络 UPF 选择策略
- 网络选择策略 UPF 选择策略 UPF 选择策略

2. 网络/网络选择策略 `weight: 0` 网络 UPF

- 网络 **UPF** 网络 > 0 网络
- 网络 **UPF** 网络 == 0 网络 UPF 网络
- 网络 UPF 网络选择策略 `weight: 1`

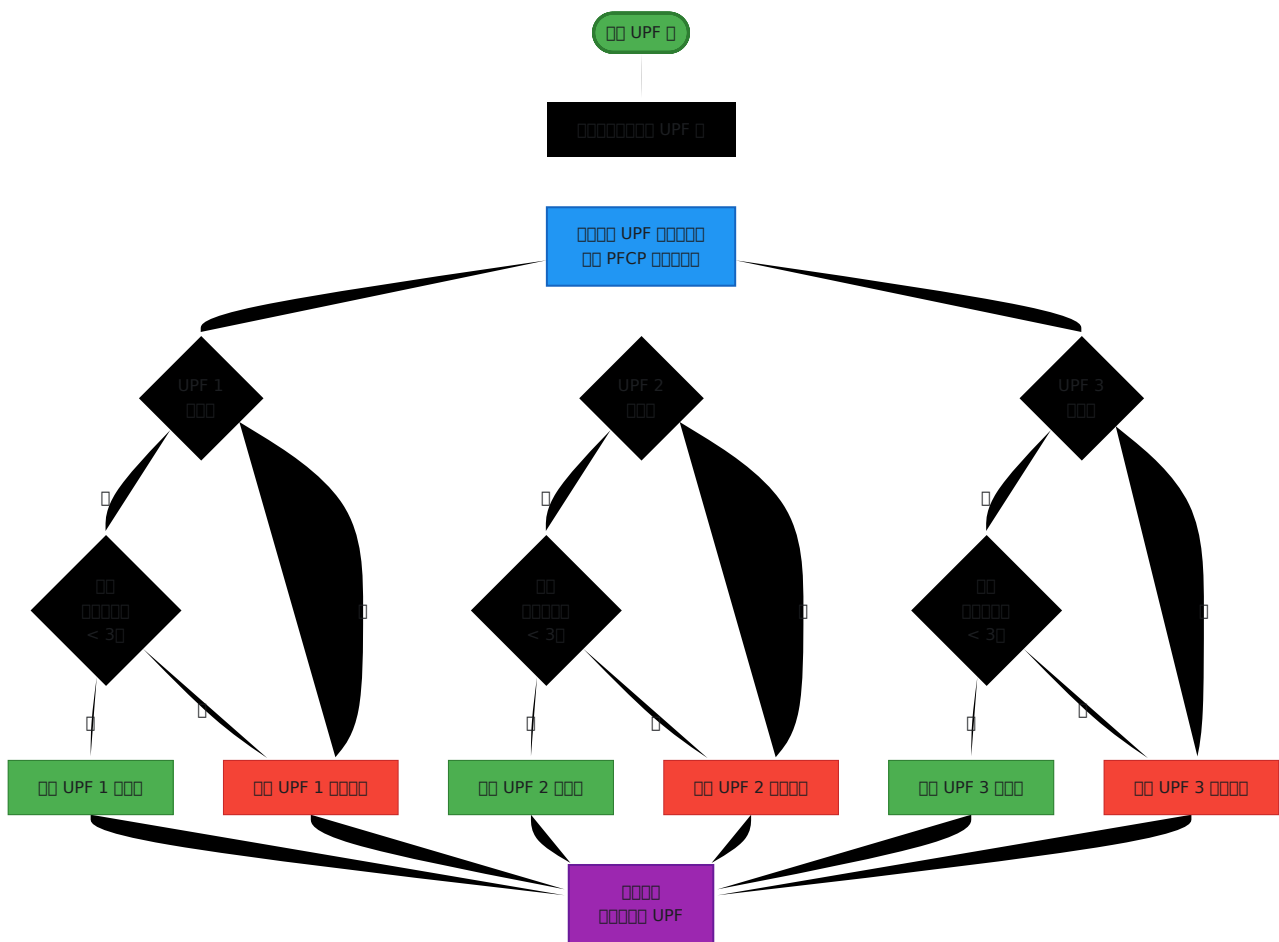
3. 网络选择策略网络选择策略网络 UPF

- 网络选择策略70% 网络 .21 20% 网络 .22 10% 网络 .23
- 网络选择策略 UPF 网络
- 网络 = 网络

4. 网络选择策略 UPF 网络选择策略

- 网络选择策略 `"UPF-<ip>:<port>"`
- 网络选择策略 PCF 网络5 网络
- 网络选择策略 UPF 网络选择策略

□□□□□□□□/□□




```
[[{"UPF-A": 100, "UPF-B": 0}]]
```

```
1. UPF-A 100  
→ [[{"UPF-A": 100}]]  
→ [[{"UPF-A": 100}]]
```

```
2. UPF-A 100, UPF-B 1  
→ [[{"UPF-A": 100, "UPF-B": 1}]]  
→ [[{"UPF-A": 100, "UPF-B": 1}]]  
→ [[{"UPF-A": 100, "UPF-B": 1}]]
```

```
3. UPF-A 100, UPF-B 0  
→ [[{"UPF-A": 100, "UPF-B": 0}]]  
→ [[{"UPF-A": 100, "UPF-B": 0}]]  
→ [[{"UPF-A": 100, "UPF-B": 0}]]
```

[[{"UPF-A": 100, "UPF-B": 0}]]


```

# 25%
upf_pool: [
    %{remote_ip_address: "10.0.1.1", remote_port: 8805, weight: 1},
    %{remote_ip_address: "10.0.1.2", remote_port: 8805, weight: 1},
    %{remote_ip_address: "10.0.1.3", remote_port: 8805, weight: 1},
    %{remote_ip_address: "10.0.1.4", remote_port: 8805, weight: 1}
]

# 90% / 10%
upf_pool: [
    %{remote_ip_address: "10.0.1.21", remote_port: 8805, weight: 90},
    %{remote_ip_address: "10.0.1.22", remote_port: 8805, weight: 10}
]

# 100% 0%
upf_pool: [
    %{remote_ip_address: "10.0.1.21", remote_port: 8805, weight: 100}, #
    %{remote_ip_address: "10.0.1.22", remote_port: 8805, weight: 0}
]

#
upf_pool: [
    %{remote_ip_address: "10.0.1.1", remote_port: 8805, weight: 100}, #
    %{remote_ip_address: "10.0.1.2", remote_port: 8805, weight: 0},
    # 1
    %{remote_ip_address: "10.0.1.3", remote_port: 8805, weight: 0}
    # 2
]
# 100% 50/50%

# A/B 50% / 50%
upf_pool: [
    %{remote_ip_address: "10.0.1.100", remote_port: 8805, weight: 50}, #
    %{remote_ip_address: "10.0.1.200", remote_port: 8805, weight: 50} #
]

```

111

- 0/1 loss weight: 0 000 UPF000 UPF 000
- 000000000 UPF 00 PFCP 0000000000000
- 00000000 UPF 000000000000
- 00000000000000 UPF 00
- 0000000000000000000095% 005% 00
- 00000000000000000000 UPF
- 00000000 UPF 000000

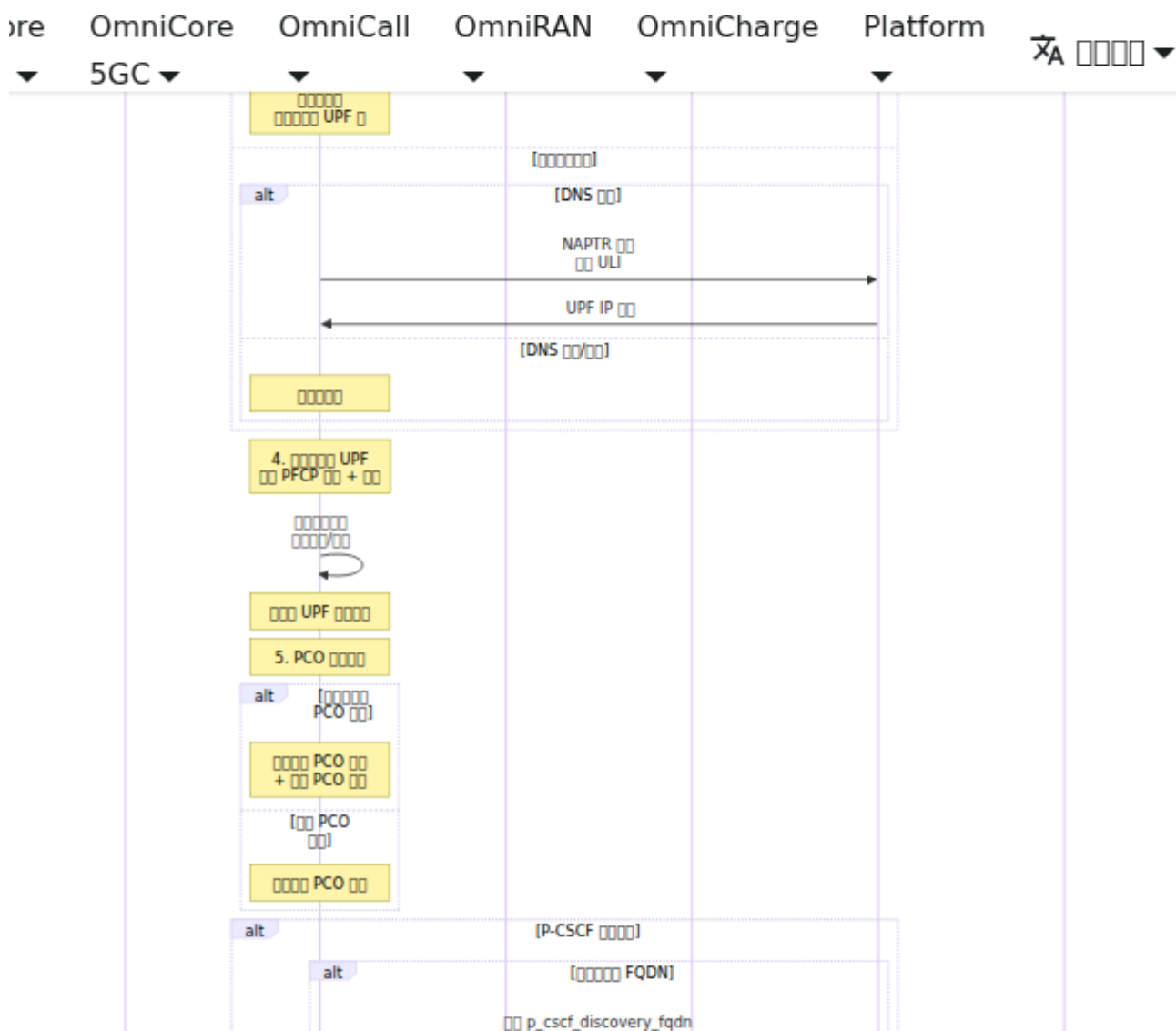
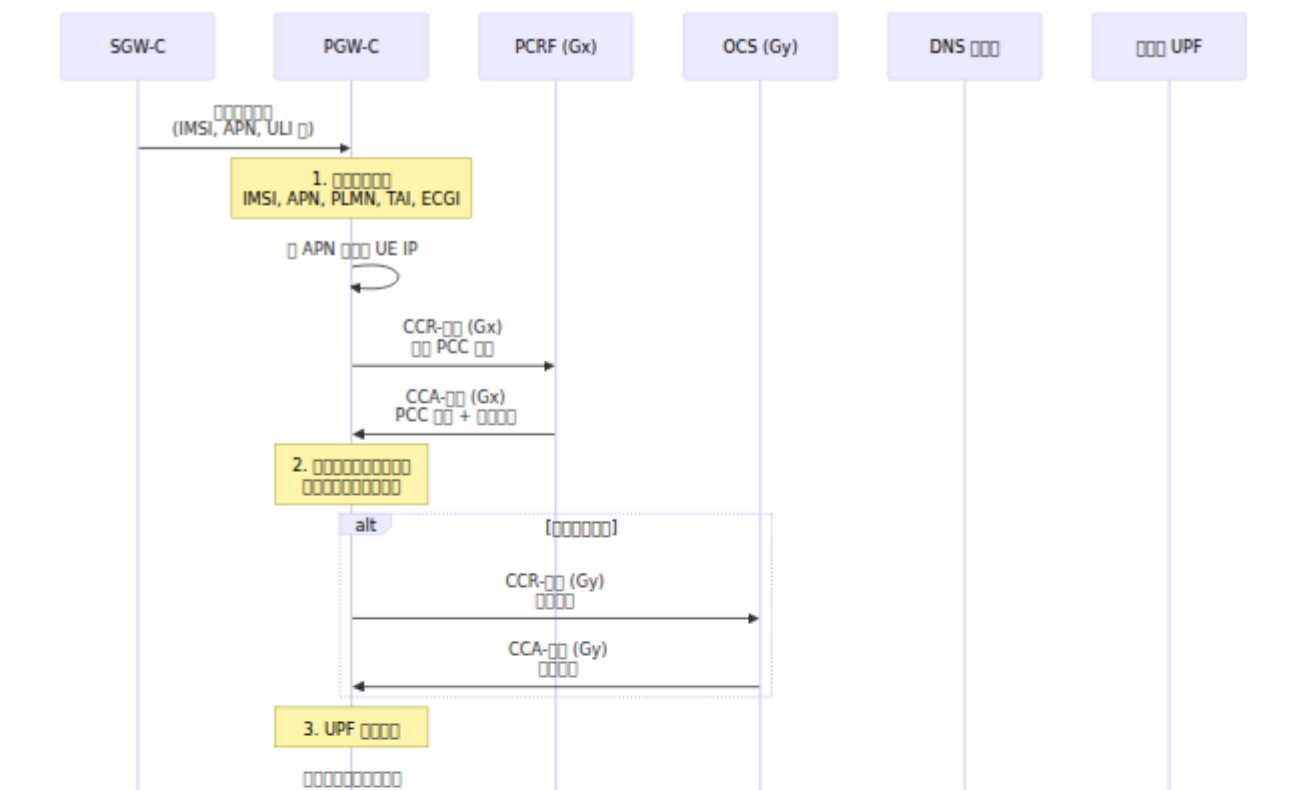
PCO□□□□□□□□□□

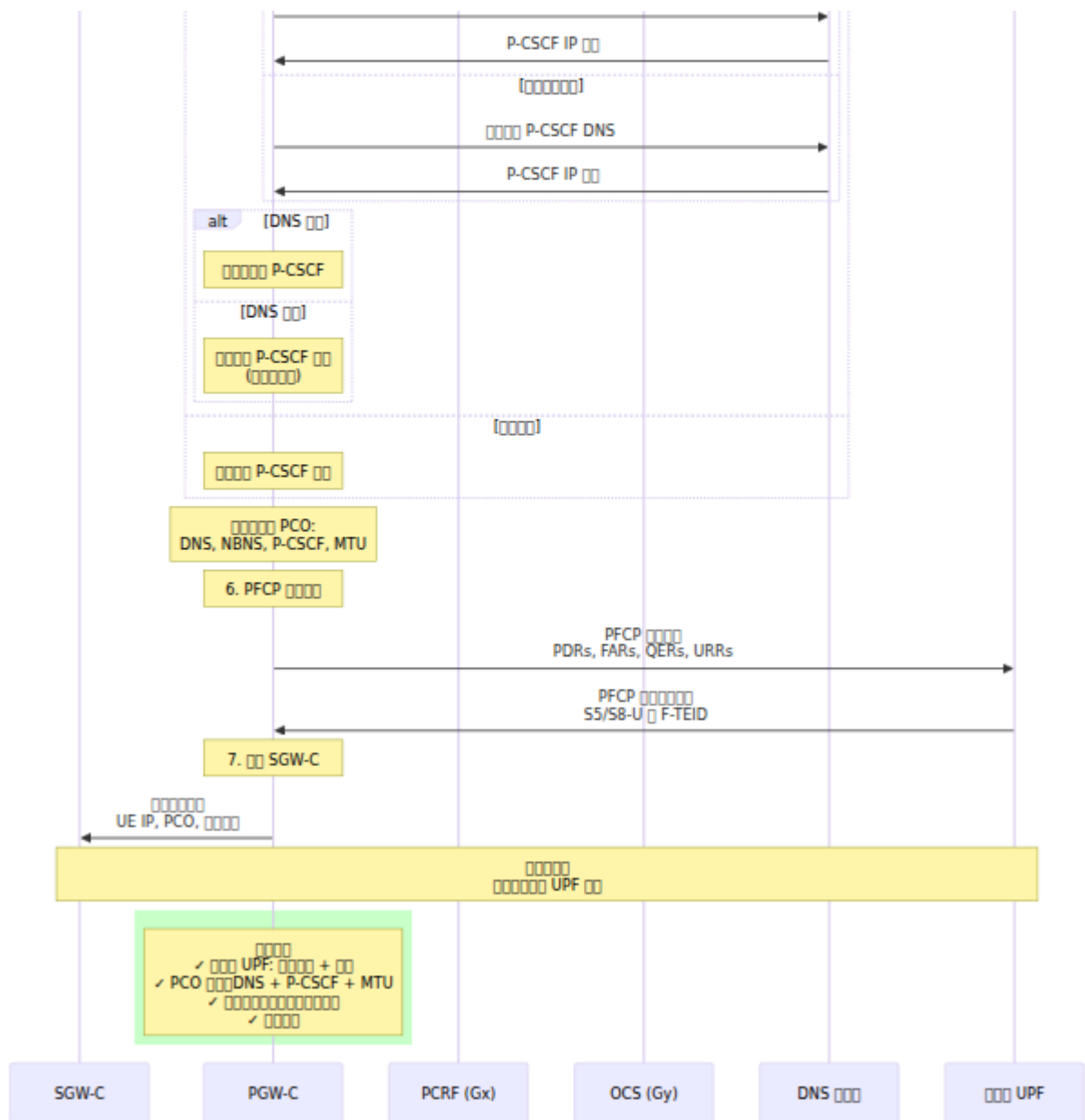
UPF 0000000000000000 PCO 0000000000000000 PCO 0000000000000000 APN 0000000000000000
0000

PCO □□□□□□□□

1. 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇 PCO 〇〇
2. 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇 PCO 〇〇〇〇〇〇
3. 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇 PCO 〇〇
4. 〇〇〇〇〇〇〇〇〇〇 PCO 〇〇〇〇〇〇 PCO

PCO □□□□□□





PCO establishment

1. PCO establishment
2. P-CSCF DNS establishment P-CSCF
3. PCO establishment /

IMS DNS establishment

IMS 参数“IMS 参数”参数

- └ DNS 参数参数参数参数参数
- └ P-CSCF参数 DNS 参数参数 p_cscf_discovery_fqdn
 - └ 参数 DNS 参数参数
- └ MTU参数参数参数参数

参数参数“参数”参数

- └ DNS 参数参数参数192.168.1.10参数192.168.1.11
- └ P-CSCF参数参数参数参数参数
- └ MTU参数参数1500

参数参数参数参数

- └ DNS 参数参数
- └ P-CSCF参数参数 DNS参数参数参数
- └ MTU参数

参数 **PCO** 参数

- primary_dns_server_address - 参数 DNS 参数参数 IP
- secondary_dns_server_address - 参数 DNS 参数 IP
- primary_nbns_server_address - 参数 WINS 参数 IP
- secondary_nbns_server_address - 参数 WINS 参数 IP
- p_cscf_ipv4_address_list - P-CSCF 参数 IP 参数参数 IMS - 参数 PCO 参数 参数 P-CSCF 参数 参数参数 P-CSCF 参数
- ipv4_link_mtu_size - MTU 参数参数

参数 **P-CSCF** 参数

PCO 通過UPF 與P-CSCF 通信

- `p_cscf_discovery_fqdn` - 通過DNS 獲 P-CSCF 的 FQDN
如 `"pcscf.mnc380.mcc313.3gppnetwork.org"`

過程如下

1. PGW-C 通過DNS 獲 FQDN
2. DNS 獲 P-CSCF IP
3. 通過 P-CSCF 與 PCO 通信 UE
4. 通過 DNS 獲 PCO 的 `p_cscf_ipv4_address_list` 獲 PCO 的 IP
5. 通過 **P-CSCF** 與 PCO 通信

過程如下

- **IMS APN** - 通過IMS 獲 P-CSCF 的 IP
- 通過 - 獲 P-CSCF 的 IP
- 通過 - DNS 獲 UE 獲 P-CSCF
- 通過 - DNS 獲 P-CSCF 的 IP

過程如下 **P-CSCF** 與 **IMS** 通信

```

rules: [
  %{
    name: "IMS ",
    priority: 20,
    match_field: :apn,
    match_regex: "^ims",
    upf_pool: [
      %{remote_ip_address: "10.100.2.21", remote_port: 8805,
weight: 80},
      %{remote_ip_address: "10.100.2.22", remote_port: 8805,
weight: 20}
    ],
    # P-CSCF  DNS  P-CSCF 
    # DNS  FQDN  P-CSCF IP
    p_cscf_discovery_fqdn: "pcscf.mnc380.mcc313.3gppnetwork.org",
    # IMS  P-CSCF  DNS 
    pco: %{
      p_cscf_ipv4_address_list: ["10.101.2.100", "10.101.2.101"]
      # DNSNBNSMTU  pco 
    }
  }
]

```

DNS


```

rules: [
  %{
    name: "企业",
    priority: 15,
    match_field: :apn,
    match_regex: "^(enterprise|corporate)\.apn",
    upf_pool: [
      %{remote_ip_address: "10.100.3.21", remote_port: 8805,
weight: 100}
    ],
    # 企业 DNS 企业 MTU
    pco: %{
      primary_dns_server_address: "192.168.1.10",
      secondary_dns_server_address: "192.168.1.11",
      ipv4_link_mtu_size: 1500
      # P-CSCF/NBNS 企业 pco 企业
    }
  }
]

```

企业 **PCO** 企业

```
rules: [
  %{
    name: "IoT APN - 物联网",
    priority: 10,
    match_field: :apn,
    match_regex: "^iot\\.m2m",
    upf_pool: [
      %{remote_ip_address: "10.100.5.21", remote_port: 8805,
weight: 100}
    ],
    # IoT 物联网 PC0
    pco: %{
      primary_dns_server_address: "8.8.8.8",
      secondary_dns_server_address: "8.8.4.4",
      primary_nbns_server_address: "10.0.0.100",
      secondary_nbns_server_address: "10.0.0.101",
      p_cscf_ipv4_address_list: [], # IoT 物联网 P-CSCF
      ipv4_link_mtu_size: 1280      # 物联网 MTU
    }
  }
]
```

物联网

- **IMS/VoLTE** 物联网 P-CSCF 物联网
- 物联网 **APN** 物联网 DNS 物联网
- **IoT/M2M** 物联网 DNS 物联网 MTU 物联网
- 物联网 DNS 物联网
- 物联网

物联网 **DNS** 物联网 **UPF** 物联网

物联网 ULI 物联网 UPF 物联网 DNS NAPTR 物联网 DNS 物联网 `upf_selection` 物联网

物联网 物联网 UPF 物联网 **PFCP** 物联网 物联网 UPF 物联网 PFCP 物联网 物联网 物联网

```
upf_selection: %{
  # DNS
  dns_enabled: true,

  #
  dns_query_priority: [:ecgi, :tai, :rai, :sai, :cgi],

  # 3GPP NAPTR DNS
  dns_suffix: "epc.3gppnetwork.org",

  # DNS
  dns_timeout_ms: 5000,

  # ...
}
```

DNS

1. DNS
2. UE DNS NAPTR
 - ECGI `eci-<hex>.ecgi.epc.mnc<MNC>.mcc<MCC>.epc.3gppnetwork.org`
 - TAI `tac-lb<hex>.tac-hb<hex>.tac.epc.mnc<MNC>.mcc<MCC>.epc.3gppnetwork.org`
 - RAI/SAI/CGI 3GPP TS 23.003
- 3.
4. DNS
- 5.

DNS

```
; NAPTR PLMN 313-380 TAC 100
tac-lb64.tac-hb00.tac.epc.mnc380.mcc313.epc.3gppnetwork.org IN
NAPTR 10 50 "a" "x-3gpp-upf:x-sxb" "" upf-edge-1.example.com.

; UPF A
upf-edge-1.example.com IN A 10.100.1.21
```

000

- 物理ネットワーク**MEC**ネットワークを構築する UPF
- 物理 **UPF** ネットワーク/物理 UPF ネットワーク PGW-C
- 物理ネットワーク UPF ネットワーク
- 物理ネットワークネットワーク UPF ネットワーク

UPF ☐ ☐ ☐ ☐

□□□□□□□□PGW-C □□□□□□ UPF □□□□□□□□□□□□□□ UPF□

□ □ □ □ □ □

UPF □□□□ □□□□□□□□□□□□□□□□

1. **PFCP** 3GPP TS 23.501 PFCP 3GPP TS 23.502
2. 3GPP TS 23.501 3GPP TS 23.502
3. 3GPP TS 23.501 PFCP 3GPP TS 23.502 GenServer 3GPP TS 23.502

□□□□□□□□□□ UPF □□□□ □□□□

- PFCP `associated: false`
- 3 `...`
- UPF `...`

□□□□

UPF `upf_selection`

- 8000000000000000
- PFCP 80000000
- 5 8000000000000000
- missed_heartbeats_consecutive 8000000000
- 8000000000000000 UPF 800000

DNS **UPF**

- [illegible]

- 000000000000
- 000000000000

0000

00/00000000 weight: 0 000

1. 000000 UPF
2. 000 000000 > 000 000000 == 00
3. 0000000000 UPF000000 UPF
4. 000000000000000000 UPF000000 10
5. 00000000 UPF0000000000

000000000000 > 000

1. 00000000 UPF
2. 0000 UPF 000000000000
3. 00000000 UPF0000000000

000000

```
[debug] 0000 UPF 002/3 00 UPF01 000
[info] 0000 UPF 00000000 UPF01 00 UPF0000 0 00 10
[warning] 00000000 UPF000 3 0000000000000000
```

00 **UPF** 00

000000

```
# 確認 UPF 状態
iex> PGW_C.PFCP_Node.is_peer_healthy?({10, 100, 1, 21})
true

# 詳細状態
iex> PGW_C.PFCP_Node.get_peer_health({10, 100, 1, 21})
%{
  associated: true,
  missed_heartbeats: 0,
  healthy: true,
  registered: true
}
```

Web UI

- 確認画面 `/upf_selection`
- 確認画面で UPF 状態を確認
- 確認画面 10-UP 10-100 10-100 100
- 確認画面 100 > 0 確認画面 == 0 確認画面 DNS 確認画面
- 10 UPF 確認画面

確認画面

1. `upf_selection` 画面 **UPF** 画面 UPF 画面

□□□□□□□□□□□□ DNS□□□□□□ UPF □□□□□□□□

1. **PFCP** **UPF**
2. **PFCP** **UPF** **PFCP**
3. **UPF**
4. 10
5. **UPF**

UPF

UPF

```
%{
    name: "Dynamic-UPF-<IP>",           # ""Dynamic-UPF-10-100-
1-21"
    remote_ip_address: <discovered_ip>, # "" DNS "" IP
    remote_port: 8805,                  # "" PCFP ""
    initiate_pfcp_association_setup: true, # PGW-C ""
    heartbeat_period_ms: 10_000         # 10 ""
}
```

UPF upf_selection

□□□DNS □□□□ UPF

```
# DNS resolve upf-edge-2.example.com -> 10.200.5.99
# Add UPF address upf_selection table

# Create table
# 1. Create table UPF 10.200.5.99
# 2. Add "UPF {10, 200, 5, 99} upf_selection..."
# 3. Add 10.200.5.99:8805 to PFCP table
# 4. Add UPF upf_selection table
# 5. Add UPF upf_selection table
```

11

□ □□□□□□□□□ DNS □ UPF □□□□□□□□□□
□ **

□ □ □ □ □ □ □ (CDR) □ □

PGW-C □□□□□

OmniPGW □ Omnitouch □□□□□□

11

1. 00
2. CDR 0000
3. CDR 00
4. CDR 00
5. 0000
6. 00
7. CDR 0000
8. 000000
9. 00
10. 00

11

CDR

SGW-C CDR EPC

□ □ □ □

- **CSV** - Comma Separated Values
- **JSON** - JavaScript Object Notation
- **XML** - Extensible Markup Language

- 時間 - 時刻
- 3GPP 4G - 3GPP TS 32.251 PS TS 32.298 CDR

時刻	時刻 CDR
時刻	時刻
時刻	時刻
時刻	時刻
時刻	時刻

CDR 時刻

時刻

<epoch_timestamp>

1726598022

時刻 Unix 時刻

時刻

時刻

- PGW-C: /var/log/pgw_c/cdrs/

cdr_directory #### config/runtime.exs

####

CDR

```
# ## CDR ##
# #####HH:MM:SS (unix_timestamp)
# #####HH:MM:SS (unix_timestamp)
# #####<gateway_name>
#
epoch,imsi,event,charging_id,msisdn,ue_imei,timezone_raw,plmn,tac,eci
```

####

- ##### - CDR ##### Unix #####
 - ##### - ##### Unix #####
 - ##### - PGW-C ##### pgw_name #####
 - ## - ##### CSV #####
-

CDR 表

表名

項目	項目名	型別	説明
0	epoch	整数	UNIX 時刻
1	imsi	文字列	IMSI
2	event	整数	CDR イベント番号“default_bearer_start”
3	charging_id	整数	充電 ID
4	msisdn	文字列	MSISDN
5	ue_imei	文字列	UE IMEI
6	timezone_raw	文字列	UE 時刻領域
7	plmn	文字列	PLMN
8	tac	整数	TAC
9	eci	整数	E-UTRAN セル ID
10	sgw_ip	文字列	SGW-C S5/S8 インターフェイス IP
11	ue_ip	文字列	UE IP アドレスIPv4 IPv6
12	pgw_ip	文字列	PGW-C S5/S8 インターフェイス IP
13	apn	文字列	APN
14	qci	整数	QoS

項目	フィールド名	型	説明
15	octets_in	uint32_t	受信バイト数
16	octets_out	uint32_t	送信バイト数

CDR 形式

形式

CDR 形式

フィールド名	型	型	説明
default_bearer_start	uint32_t	uint32_t	デフォルトベアラ開始
default_bearer_update	uint32_t	uint32_t	デフォルトベアラ更新
default_bearer_end	uint32_t	uint32_t	デフォルトベアラ終了

形式

- default - デフォルト PDN 形式
- dedicated - デフォルト PDN 形式

形式

- default_bearer_start - デフォルトベアラ開始
- default_bearer_update - デフォルトベアラ更新
- default_bearer_end - デフォルトベアラ終了
- dedicated_bearer_start - デフォルトベアラ開始
- dedicated_bearer_update - デフォルトベアラ更新
- dedicated_bearer_end - デフォルトベアラ終了

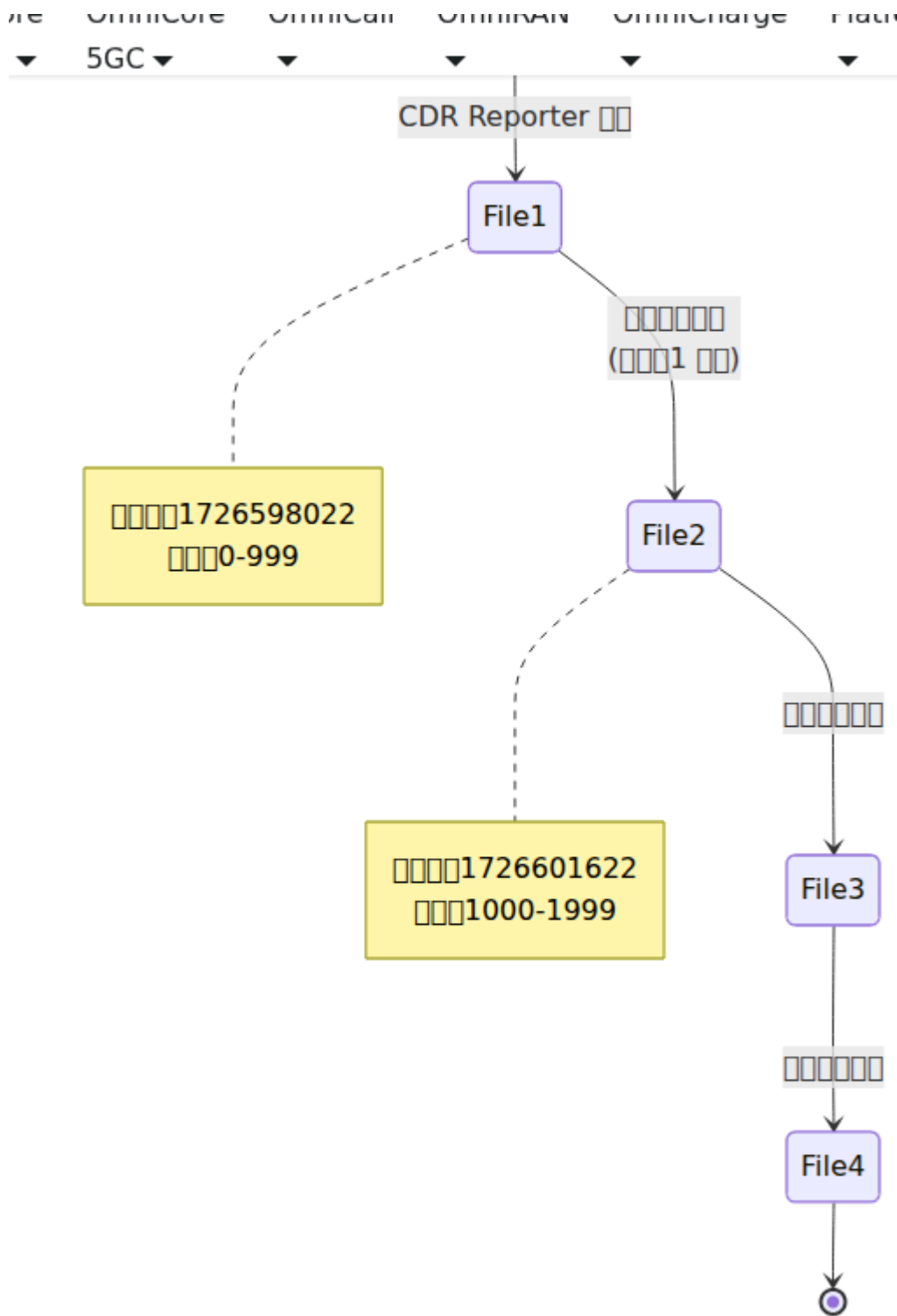
□□□□

□□ **CDR** □□

```
# □□ CDR □□□
# □□□□□□18:53:42 (1726598022)
# □□□□□□19:53:42 (1726601622)
# □□□□sgw-c-prod-01
# epoch,imsi,event,charging_id,msisdn,ue_imei,timezone_raw,plmn,tac,e
1726598022,310260123456789,default_bearer_start,12345,15551234567,123
1726598322,310260123456789,default_bearer_update,12345,15551234567,12
1726598622,310260123456789,default_bearer_update,12345,15551234567,12
1726598922,310260123456789,default_bearer_end,12345,15551234567,12345
```

□□□□

CDR □□□□□□□□□□□□□□□□



1. 1726598022

2. 0-999

3. 1726601622

4. 1000-1999

5. 1726598022

6. 0-999

□□

□□□□

PGW-C CDR □□□ `config/runtime.exs` □□□□

□□	□ □	□□	□□□	□□□
<code>pgw_name</code>	□ □ □	PGW □□□□ □□□□ □ CDR □□□	"omni- pgw01"	◆◆◆□□□□□□□ ID
<code>cdr_file_duration</code>	□ □	□□□□ □□□□ □□	3600000	3600000□1 □□□
<code>cdr_directory</code>	□ □ □	CDR □□□□ □□	"/tmp/pgw_c"	<code>/var/log/pgw_c/cdrs</code>
<code>usage_report_interval</code>	□ □	URR □□□□ □□□□ - PGW- U □□ □□□□ □□□	60000	60000□1 □□□

□□□□

□□□□ (**config/runtime.exs**):

```

config :pgw_c,
  # CDR
  pgw_name: "omni-pgw01",
  cdr_file_duration: 3_600_000,          # 1
  cdr_directory: "/var/log/pgw_c/cdrs",

  # URR PGW-U
  usage_report_interval: 60_000          # 60

```

CDR:

```

config :pgw_c,
  pgw_name: "pgw-c-prod-01",
  cdr_file_duration: 3_600_000,          # 1
  cdr_directory: "/var/log/pgw_c/cdrs",
  usage_report_interval: 60_000          # 1

```

CDR:

```

config :pgw_c,
  pgw_name: "pgw-c-dev",
  cdr_file_duration: 300_000,            # 5
  cdr_directory: "/tmp/pgw_c_cdrs",
  usage_report_interval: 30_000          # 30

```

CDR:

```

config :pgw_c,
  pgw_name: "pgw-c-prod-heavy",
  cdr_file_duration: 1_800_000,          # 30
  cdr_directory: "/mnt/fast-storage/cdrs",
  usage_report_interval: 300_000         # 5

```

URR

PGW-C **PFCEP URRs** PGW-U URR PGW-U CDR

URR の種類

1. `usage_report_interval` の値が 0 の URR は PFCP の URR
2. PGW-C の URR
3. PGW-U の URR
4. `bearer_update` CDR の URR
5. `bearer_end` CDR の URR

例 `usage_report_interval: 60_000` の URR

- PGW-U の 60 秒の URR
- 60 秒の CDR の URR
- `bearer_update` CDR の URR

URR の定義

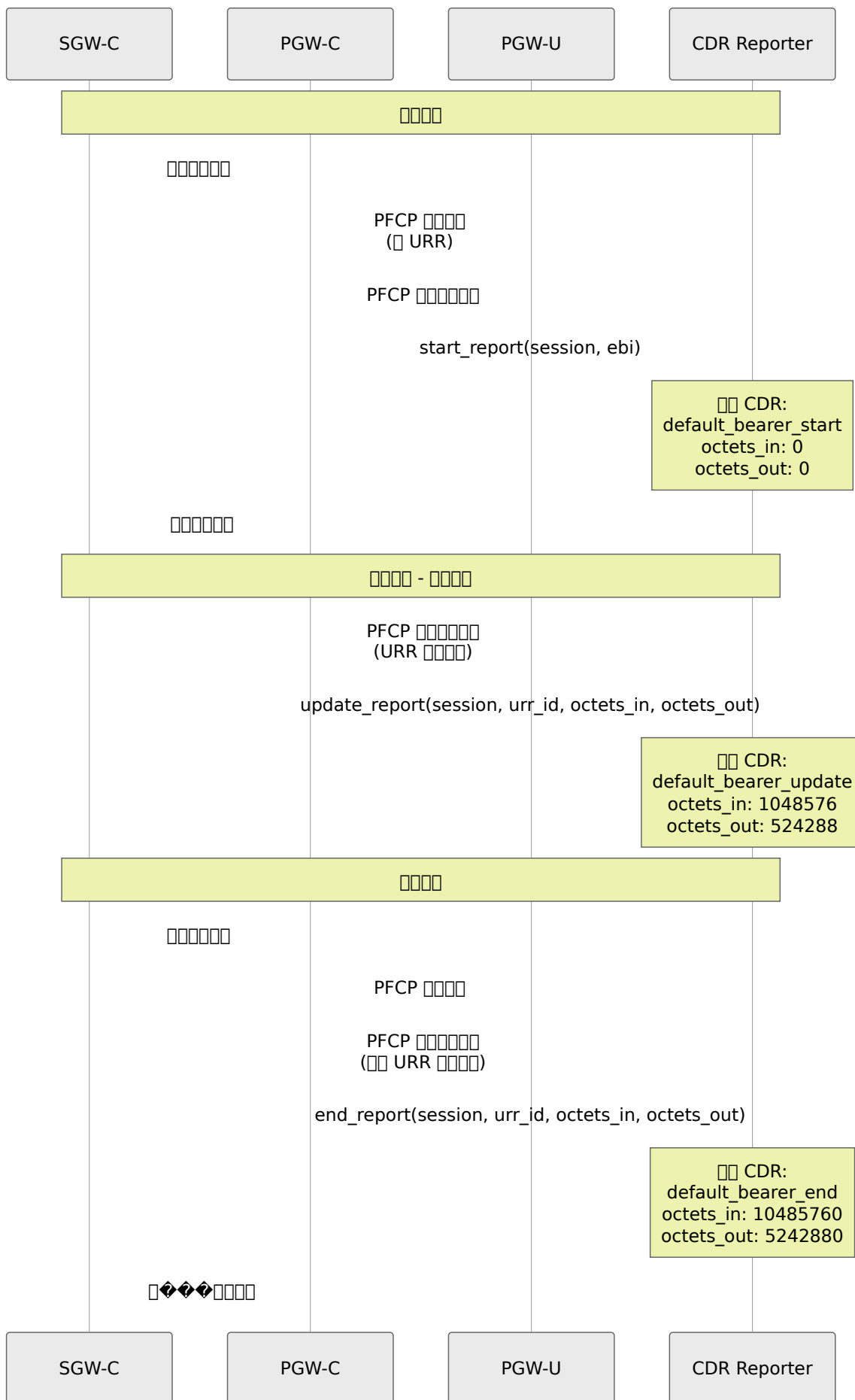
```
# lib/core/session/types.ex
defmodule PGW_C.Session.Types.URR do
  typedstruct do
    field :urr_id, non_neg_integer()
    field :measurement_method, :duration | nil
    field :reporting_triggers, :time_threshold | nil
    field :time_threshold, non_neg_integer() | nil # 0
  end
end
```

例 `PFCP` の URR は PFCP の `lib/core/session/impl/procedures.ex:468` の `URR` の定義

CDR の種類

例 `CDR` の種類

PGW-C CDR の種類



CDR 設定

1. 設定

- 設定 設定
- 設定 設定
- **octets_in** 0
- **octets_out** 0

2. 設定

- 設定 PGW-U 設定 PCF 設定 URR 設定
- 設定 設定
- **octets_in** 設定
- **octets_out** 設定
- 設定 URR 設定 `usage_report_interval` 設定

3. 設定

- 設定 PGW-U 設定 PCF 設定
- 設定 設定
- **octets_in** 設定
- **octets_out** 設定

設定

1. epoch 設定

設定 Unix 設定

設定 CDR 設定

設定

1726598022 → 2025-09-17 18:53:42 UTC

2. imsi

15

MCCMNC + MSIN

310260123456789

MCC MNC MSIN

(310)(260) (123456789)

UE

3. eventCDR

<bearer_type>_bearer_<event>

- default_bearer_start
- default_bearer_update
- default_bearer_end
- dedicated_bearer_start
- dedicated_bearer_update
- dedicated_bearer_end

- EBI EPS ID LBI ID default
- EBI LBI dedicated

EBI LBI

4. charging_id

32

12345

PGW-C

- SGW PGW
- Gy/Gz
-

5. msisdn

E.164

ISDN

+

15551234567



CC

(1) (5551234567)

UE MME HSS

6. ue_imei

15

TAC8+ SNR6+ Spare1

123456789012345



TAC

SNR S

UE MME

7. timezone_rawUE

/

UE

CSV

, ()

8. plmn

MCC: 505, MNC: 57

↓

"50557"

↓

"055570"

↓

0x055570 = 349552

349552 → MCC: 505, MNC: 57

MME UE

9. tac

16

UE

0 - 65535

1234

UE MME

-
-
- TAI

10. eciE-UTRAN

28

E-UTRAN UE

eNodeB ID20 + ID8

0 - 268,435,455

5678

MME UE

-
-
-

11. sgw_ipSGW IP

IPv4 IPv6

SGW-C S5/S8 IP F-TEID

IPv4IPv6

10.0.0.15	(IPv4)
2001:db8::15	(IPv6)

S5/S8

12. ue_ipUE IP

IPv4|IPv6

UE IP PDN

<ipv4>|<ipv6>

172.16.1.100	(IPv4)
2001:db8::1	(IPv6)
172.16.1.100 2001:db8::1	()

PGW-C PDN PAA

- IPv4IPv4
- IPv6IPv6
- PDN

13. pgw_ipPGW IP

IPv4 IPv6

PGW-C S5/S8 IP F-TEID

IPv4IPv6

10.0.0.20 (IPv4)
2001:db8::20 (IPv6)

PGW-C

14. apn

100

PDN

DNS

internet
ims
mms
enterprise.corporate

MME

-
-
- IP

15. qciQoS

8

QoS

1 - 9128-254

QCI

QCI					
1	GBR	2	100 ms	10^-2	
2	GBR	4	150 ms	10^-3	
3	GBR	3	50 ms	10^-3	
4	GBR	5	300 ms	10^-6	
5	Non-GBR	1	100 ms	10^-6	IMS
6	Non-GBR	6	300 ms	10^-6	
7	Non-GBR	7	100 ms	10^-3	
8	Non-GBR	8	300 ms	10^-6	
9	Non-GBR	9	300 ms	10^-6	

9 →

PGW-C QoS

16. octets_in

64

→ UE

1048576 → 1 MB

PGW-U PFCP URR

- update
- end
- start 0
- URR `usage_report_interval`

17. octets_out

64

UE →

524288 → 512 KB

PGW-U PFCP URR

- `update` 時間間隔
- `end` 時間間隔
- `start` 時間間隔 0
- URR 時間間隔 `usage_report_interval` 時間

例

例 1 時間間隔

例

1. 時間間隔
2. 5 時間間隔 10 MB 5 MB
3. 時間間隔

CDR 例

```
# 例 CDR 例
# 時間間隔10:00:00 (1726570800)
# 時間間隔11:00:00 (1726574400)
# 時間間隔pgw-c-01
# epoch,imsi,event,charging_id,msisdn,ue_imei,timezone_raw,plmn,tac,e
1726570800,310260111111111,default_bearer_start,10001,1555111111,111
1726571100,310260111111111,default_bearer_update,10001,1555111111,11
1726571400,310260111111111,default_bearer_end,10001,1555111111,1111
```

例 2 時間間隔

例

1. 時間間隔IPv4 + IPv6
2. 時間間隔
3. 時間間隔

CDR 000

1726570800,310260222222222,default_bearer_start,10002,15552222222,222
1726571100,310260222222222,default_bearer_update,10002,15552222222,22
1726571400,310260222222222,default_bearer_update,10002,15552222222,22
1726571700,310260222222222,default_bearer_update,10002,15552222222,22
1726572000,310260222222222,default_bearer_end,10002,15552222222,22222

00 3000000000000

0000

1. 00000000QCI 9
2. 000000000000QCI 6
3. 0000000000
4. 000000
5. 000000

CDR 000

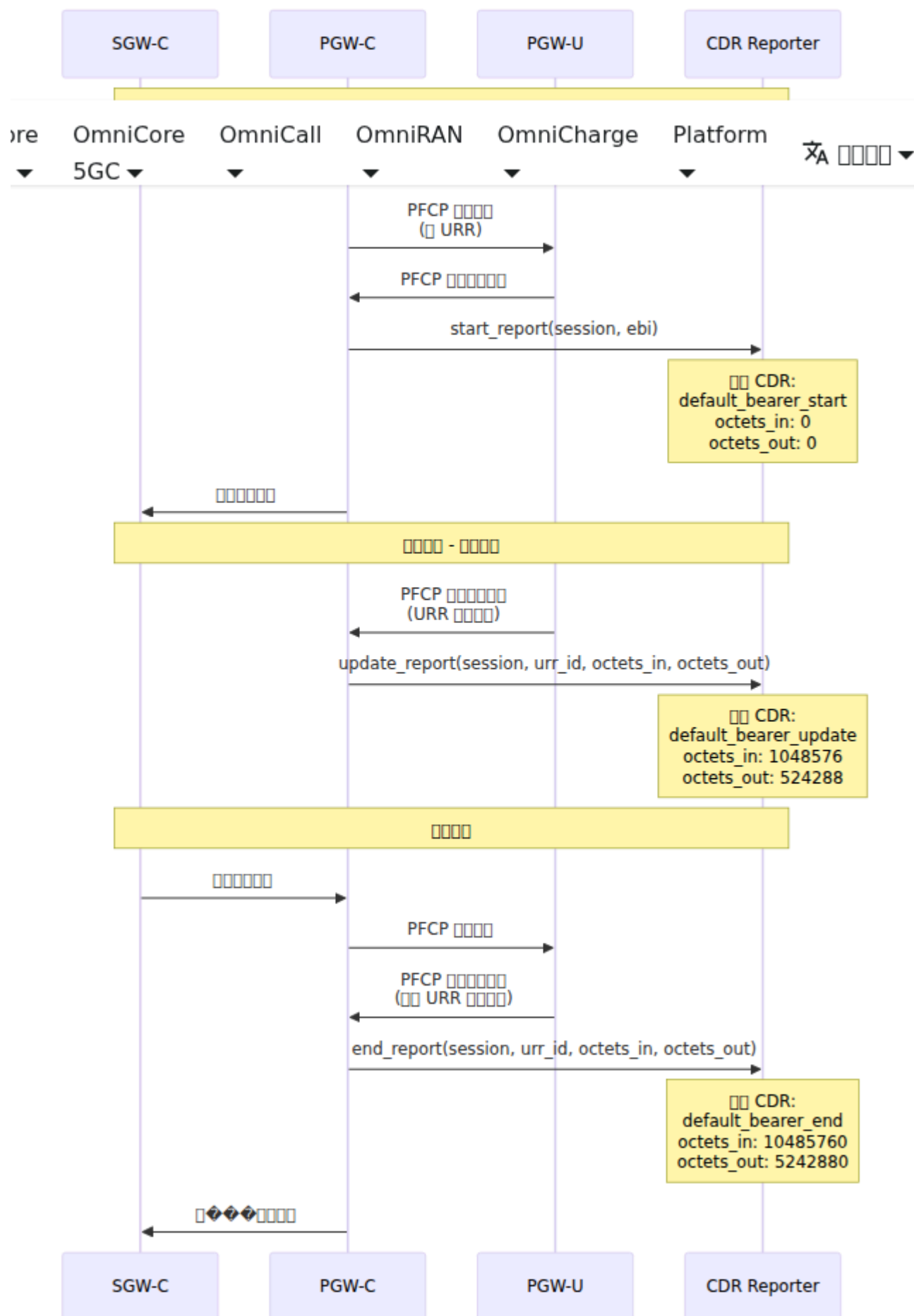
1726570800,310260333333333,default_bearer_start,10003,15553333333,333
1726571100,310260333333333,dedicated_bearer_start,10004,15553333333,3
1726571400,310260333333333,default_bearer_update,10003,15553333333,33
1726571400,310260333333333,dedicated_bearer_update,10004,15553333333,
1726571700,310260333333333,dedicated_bearer_end,10004,15553333333,333
1726572000,310260333333333,default_bearer_end,10003,15553333333,33333

000

- 00000100030000000000 10 MB000 4 MB
- 00000100040000000000 200 MB000 2 MB
- 000 QCI 009 6000000 QoS 00

□□

CDR □□□□



CDR 处理

1. 守护进程

```
# 守护 CDR 进程PGW-C
inotifywait -m /var/log/pgw_c/cdrs/ -e close_write | while read
path action file; do
    # 处理 CDR
    process_cdr "$path$file"
done
```

2. 日志

```
# 实时监控 CDR 日志
tail -F /var/log/pgw_c/cdrs/* | process_cdr_stream
```

CDR 类型

- **CDR** - 记录 CDR 信息
- **PFCP** - 记录 URRs 的 PGW-U 消息
- **CDR** - CDR 记录
- **CDR** - CDR 的 URR 记录
- **Gx** - CDR 的 QCI 记录
- **Gy** - 记录

3GPP

- TS 32.251 - PS 记录
- TS 29.274 - 3GPP EPS 的 GTP-C 消息
- TS 29.244 - CP 的 UP 消息 PFCP - **URR** 记录
- TS 32.298 - CDR 记录

CDR 00 - PGW-C 00000000

0 Omnitouch 000000

00000 1.0 00000 2025-12-28

Diameter Gx

Policy and Charging Rules Function (PCRF)

Overview

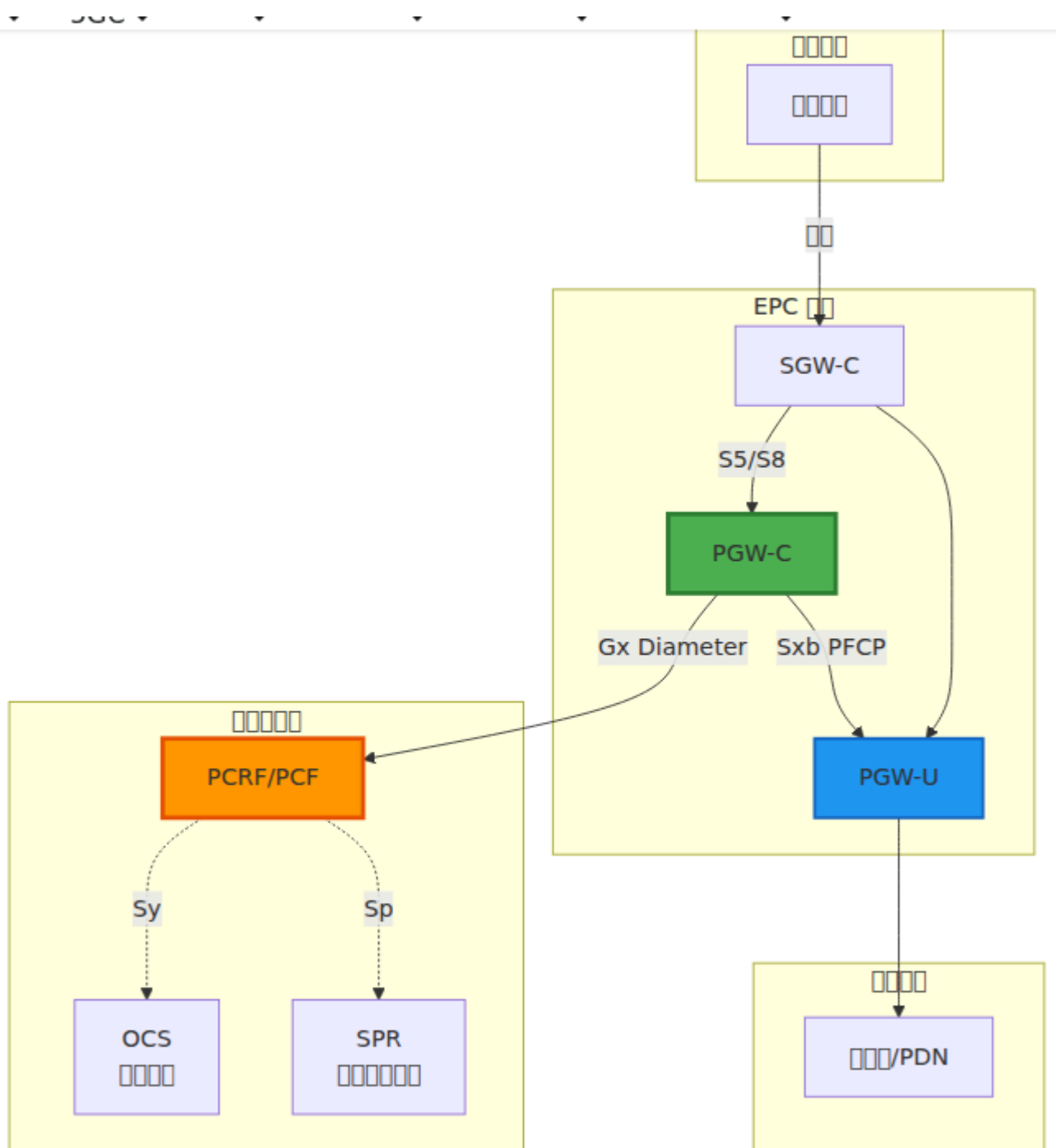
- Introduction
- Gx Interface
- Diameter Protocol
- Network Elements
- PCRF Functions
- QoS Management
- Charging Management
- Policy Management
- Security
- Conclusion

Architecture

Gx interface connects PGW-C to PCRF. PCRF is responsible for policy and charging control in 5G networks.

- Policy Management - QoS Management
- Charging Management - Charging Data Function (CDF)
- Policy Management - Policy Decision Function (PDF)
- Security - Security Function (SF)

Network Architecture Gx



1.

1.	2.
1.	PCRF 1. PCC 1.
QoS 1.	1. QoS 1.
1.	1./1.
1.	1./1.
1.	1.

Gx 1.

3GPP 1.

- 1. 3GPP TS 29.212
- 1. **Diameter** 1. **ID** 16777238 (Gx)
- 1. Diameter 1. (RFC 6733)

1.

1. UE PDN 1. **Gx** 1. **Session-ID** 1.

- 1. UE 1. (CCR-Initial)
- 1. 1. (CCR-Update) - 1.
- 1. UE 1. (CCR-Termination)

Session ID

Session-ID: <Origin-Host>;<high32>;<low32>[;<optional>]

Example: omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org;1234567890;98765

Fields

- **Origin-Host**: PGW-C or Diameter peer
- **high32**: high 32 bits
- **low32**: low 32 bits

Diameter

Fields

Diameter peer or PGW-C


```

Diameter Header (20 bytes)
├─ Version (1 byte) = 1
├─ Message Length (3 bytes)
├─ Flags (1 byte)
│   ├─ R: Request (1) / Answer (0)
│   ├─ P: Proxiabie
│   ├─ E: Error
│   └─ T: Potentially retransmitted
├─ Command Code (3 bytes)
├─ Application ID (4 bytes) = 16777238 (Gx)
├─ Hop-by-Hop ID (4 bytes)
└─ End-to-End ID (4 bytes)

```

```

AVPs (00-00)
├─ AVP Header
│   ├─ AVP Code
│   ├─ Flags (V, M, P)
│   └─ AVP Length
│   └─ Vendor ID (optional)
└─ AVP Data

```

00 Diameter 00

AVP0000-0000

- Diameter 00000000
- 00000000
- 00000000 AVP

000

- 00/000
- CCR00000000/ CCA00000000

00000

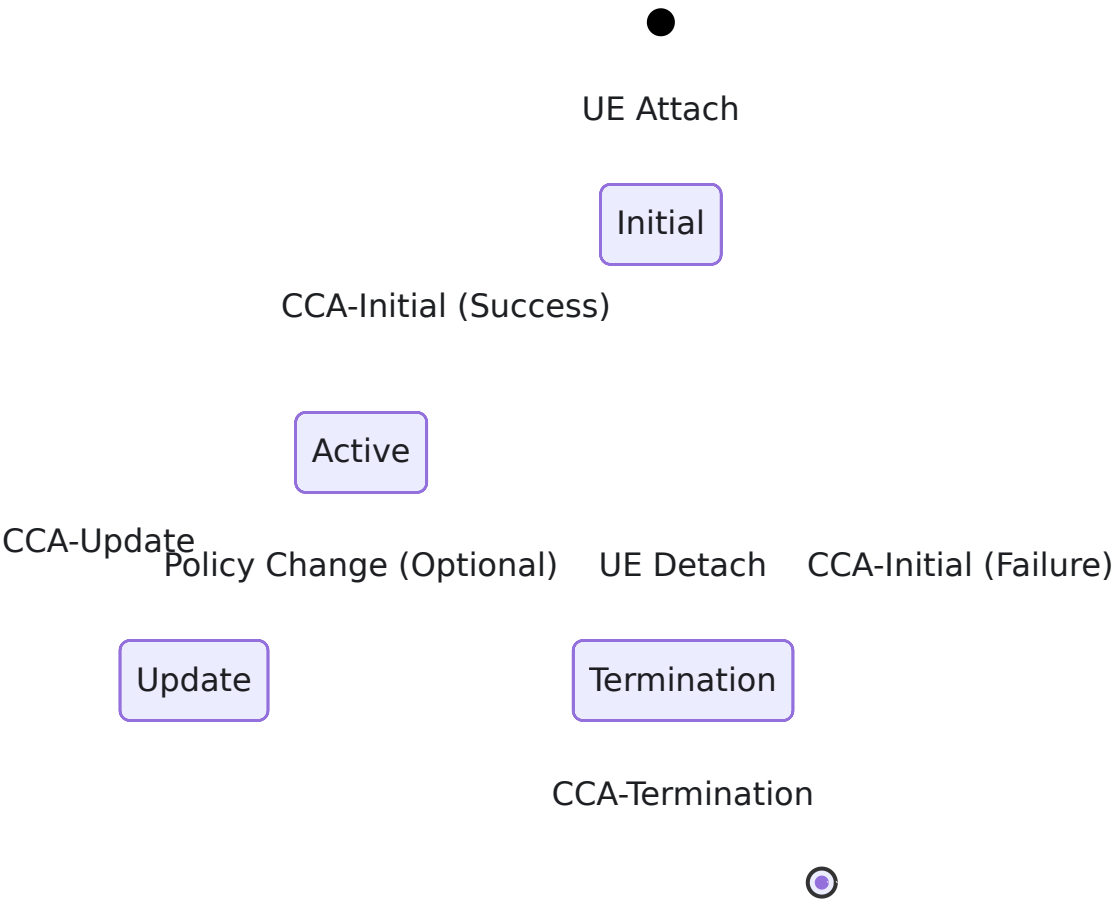
- 2001 - DIAMETER_SUCCESS
- 3xxx - 0000
- 4xxx - 0000

- 5xxx - 5G Core

5G Core

PGW-C is **Diameter** based (RFC 4006) Gx

5G Core



CCR-Initial - 5G Core

5G Core UE PDN

5G Core

- 5G Core
- PCRF 5G Core

- QoS

PGW-C AVP

AVP	AVP		
Session-Id	263	UTF8String	Gx
Auth-Application-Id	258	Unsigned32	16777238 (Gx)
Origin-Host	264	DiamIdent	PGW-C Diameter
Origin-Realm	296	DiamIdent	PGW-C Diameter
Destination-Realm	283	DiamIdent	PCRF
CC-Request-Type	416	Enumerated	1 = INITIAL_REQUEST
CC-Request-Number	415	Unsigned32	0
Subscription-Id	443	Grouped	UE IMSI/MSISDN
Called-Station-Id	30	UTF8String	APN
Framed-IP-Address	8	OctetString	UE IPv4
IP-CAN-Type	1027	Enumerated	5 = 3GPP-EPS
RAT-Type	1032	Enumerated	1004 = EUTRAN
QoS-Information	1016	Grouped	QoS (AMBR)
Network-Request-Support	1024	Enumerated	
Supported-Features	628	Grouped	Gx

CCR-I

```
CCR (Command Code: 272, Request)
├─ Session-Id: "pgw_c.example.com;123;456"
├─ Auth-Application-Id: 16777238
├─ Origin-Host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org"
├─ Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├─ Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├─ CC-Request-Type: INITIAL_REQUEST (1)
├─ CC-Request-Number: 0
├─ Subscription-Id (Grouped)
│   └─ Subscription-Id-Type: END_USER_IMSI (1)
│       └─ Subscription-Id-Data: "310260123456789"
├─ Called-Station-Id: "internet"
├─ Framed-IP-Address: 100.64.1.42
├─ IP-CAN-Type: 3GPP-EPS (5)
├─ RAT-Type: EUTRAN (1004)
├─ QoS-Information (Grouped)
│   └─ APN-Aggregate-Max-Bitrate-UL: 100000000 (100 Mbps)
│       └─ APN-Aggregate-Max-Bitrate-DL: 50000000 (50 Mbps)
├─ Network-Request-Support: 1
└─ Supported-Features: [...]
```

CCA-Initial -

PCRF CCR-I

-
- PCC
- QoS

PGW-C AVP

AVP 名称	AVP 代码	描述
Result-Code	268	结果 (2001) 成功
Experimental-Result	297	实验性结果
QoS-Information	1016	QoS 信息
Charging-Rule-Install	1001	安装 PCC 规则
Charging-Rule-Definition	1003	规则定义
Default-EPS-Bearer-QoS	1049	默认 EPS 承载 QoS

消息内容

```
CCA (Command Code: 272, Answer)
├─ Session-Id: "pgw_c.example.com;123;456"
├─ Result-Code: DIAMETER_SUCCESS (2001)
├─ Origin-Host: "pcrf.example.com"
├─ Origin-Realm: "example.com"
├─ Auth-Application-Id: 16777238
├─ CC-Request-Type: INITIAL_REQUEST (1)
├─ CC-Request-Number: 0
├─ QoS-Information (Grouped)
│   ├─ APN-Aggregate-Max-Bitrate-UL: 50000000 (50 Mbps - reduced)
│   └─ APN-Aggregate-Max-Bitrate-DL: 100000000 (100 Mbps -
increased)
├─ Charging-Rule-Install (Grouped)
│   ├─ Charging-Rule-Name: "default_internet_rule"
│   └─ Charging-Rule-Name: "video_streaming_rule"
└─ Charging-Rule-Definition (Grouped)
    ├─ Charging-Rule-Name: "default_internet_rule"
    ├─ QoS-Information: {...}
    └─ Precedence: 1000
```

CCR-Termination消息 - 消息

消息 UE 消息 PDN 消息

PCRF

- PCRF
- /

CCR-I

- CC-Request-Type: TERMINATION_REQUEST (3)
-
- AVP

CCR-T

```
CCR (Command Code: 272, Request)
├─ Session-Id: "pgw_c.example.com;123;456"
├─ Auth-Application-Id: 16777238
├─ Origin-Host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org"
├─ Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├─ Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├─ CC-Request-Type: TERMINATION_REQUEST (3)
├─ CC-Request-Number: 1
└─ Termination-Cause: DIAMETER_LOGOUT (1)
```

CCA-Termination

PCRF CCR-T

-
-

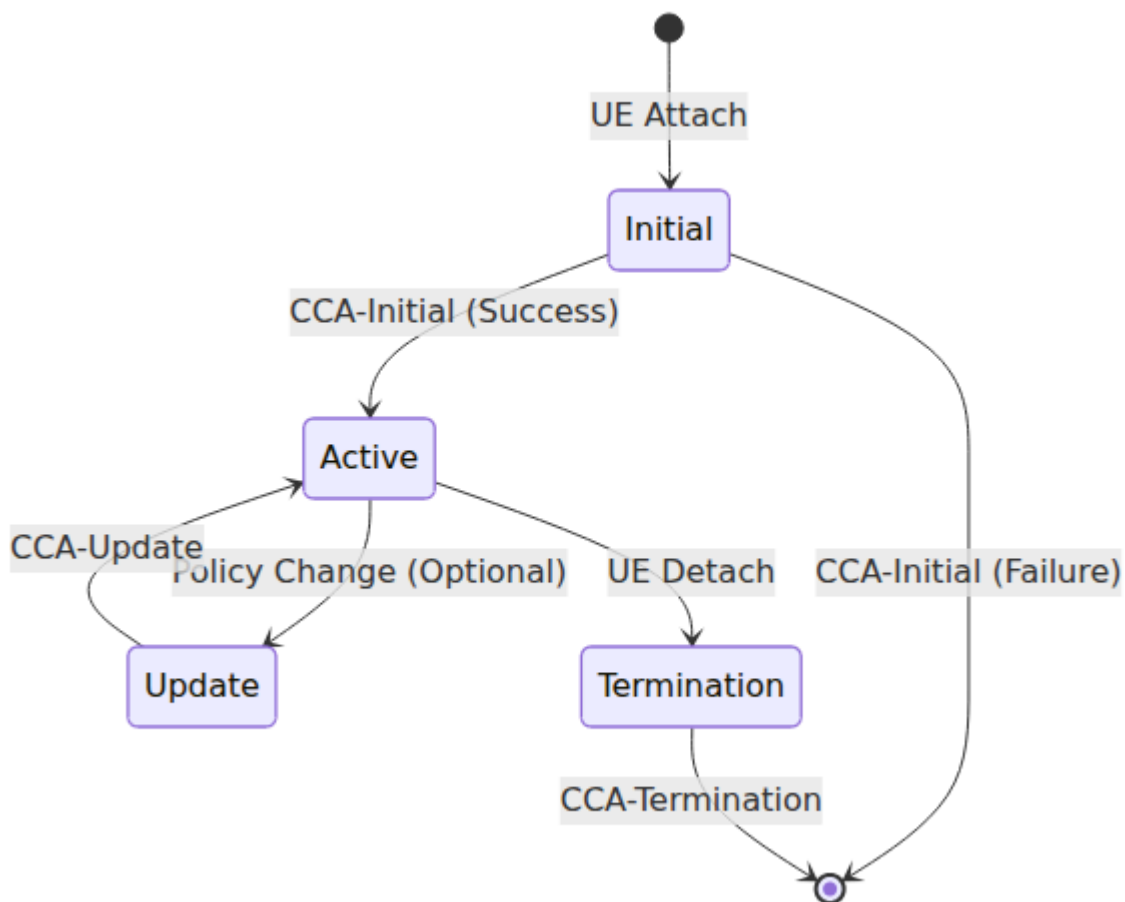
CCA-T

```
CCA (Command Code: 272, Answer)
├─ Session-Id: "pgw_c.example.com;123;456"
├─ Result-Code: DIAMETER_SUCCESS (2001)
├─ Origin-Host: "pcrf.example.com"
├─ Origin-Realm: "example.com"
├─ Auth-Application-Id: 16777238
├─ CC-Request-Type: TERMINATION_REQUEST (3)
└─ CC-Request-Number: 1
```

□□□□□□□□

PCC □□□□

PCC□□□□□□□□□□ □□□□□□□□□□□□



配置

1. 配置

- 配置
- 配置 "video_streaming_rule"

2. 配置

- 配置 = 配置
- 配置 0-65535
- 配置

3. 配置TFT - 配置

- 配置
- 配置
 - IP 5 配置/配置 IP/配置
 - "permit out ip from any to 8.8.8.8 80"

4. QoS 配置

- **QCI** **QoS** 配置 1-9 配置 128-254 配置
 - QCI 1 配置
 - QCI 5 IMS 配置
 - QCI 9 配置
- **ARP** 配置 配置
- **MBR/GBR** 配置/配置

5. 配置

- 配置 配置 OCS 配置 - 配置 Diameter Gy 配置
- 配置 配置
- 配置/配置 OCS 配置 Diameter Gy 配置 CDR 配置 - 配置 CDR 配置

6. 配置

- 配置 配置

- 配置 配置

配置配置

PCRF 配置配置配置配置

1. 配置配置配置配置

```
Charging-Rule-Install (Grouped)
├─ Charging-Rule-Name: "gold_subscriber_internet"
└─ Charging-Rule-Name: "video_qos_boost"
```

2. 配置配置配置配置

```
Charging-Rule-Definition (Grouped)
├─ Charging-Rule-Name: "dynamic_rule_123"
├─ Precedence: 100
├─ Flow-Information (Grouped)
│   ├─ Flow-Description: "permit out ip from any to 192.0.2.0/24"
│   └─ Flow-Direction: DOWNLINK
├─ QoS-Information (Grouped)
│   ├─ QoS-Class-Identifier: 5
│   ├─ Max-Requested-Bandwidth-UL: 100000000
│   └─ Max-Requested-Bandwidth-DL: 500000000
└─ Rating-Group: 1000
```

QoS 配置 AVP

APN-AMBR配置配置配置配置

配置配置 APN 配置配置 GBR 配置

```
QoS-Information (Grouped)
├─ APN-Aggregate-Max-Bitrate-UL: 100000000 # 100 Mbps
└─ APN-Aggregate-Max-Bitrate-DL: 200000000 # 200 Mbps
```

PGW-C 配置

- AMBR
 - PGW-U QER
-

Gx

config/runtime.exs

```

config :pgw_c,
  diameter: %{
    # Diameter Server IP
    listen_ip: "0.0.0.0",

    # PGW-C Diameter Server (Origin-Host)
    host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org",

    # PGW-C Diameter Server (Origin-Realm)
    realm: "epc.mnc999.mcc999.3gppnetwork.org",

    # PCRF Server
    peer_list: [
      %{
        # PCRF Diameter Server
        host: "pcrf.epc.mnc999.mcc999.3gppnetwork.org",

        # PCRF Realm PGW-C Realm
        realm: "epc.mnc999.mcc999.3gppnetwork.org",

        # PCRF IP
        ip: "10.0.0.30",

        # PGW-C Connects to PCRF
        # true = PGW-C connects to PCRF
        # false = PCRF connects to PGW-C
        initiate_connection: true
      }
    ]
  }
}

```

PCRF Server

Configuration

```

config :pgw_c,
  diameter: %{
    listen_ip: "0.0.0.0",
    host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org",
    realm: "epc.mnc999.mcc999.3gppnetwork.org",
    peer_list: [
      %{
        host: "pcrf-primary.example.com",
        realm: "epc.mnc999.mcc999.3gppnetwork.org",
        ip: "10.0.1.30",
        initiate_connection: true
      },
      %{
        host: "pcrf-backup.example.com",
        realm: "epc.mnc999.mcc999.3gppnetwork.org",
        ip: "10.0.2.30",
        initiate_connection: true
      }
    ]
  }
}

```

□□□□□

- Diameter □□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□

□□□□□□

Diameter □□□□□ **FQDN**□□□□□□□□□□

```

# □□ - FQDN □□
host: "pgw_c.epc.mnc999.mcc999.3gppnetwork.org"

# □□□ - □□□□□ Diameter □□
host: "pgw_c"
host: "10.0.0.20" # □□□ IP □□

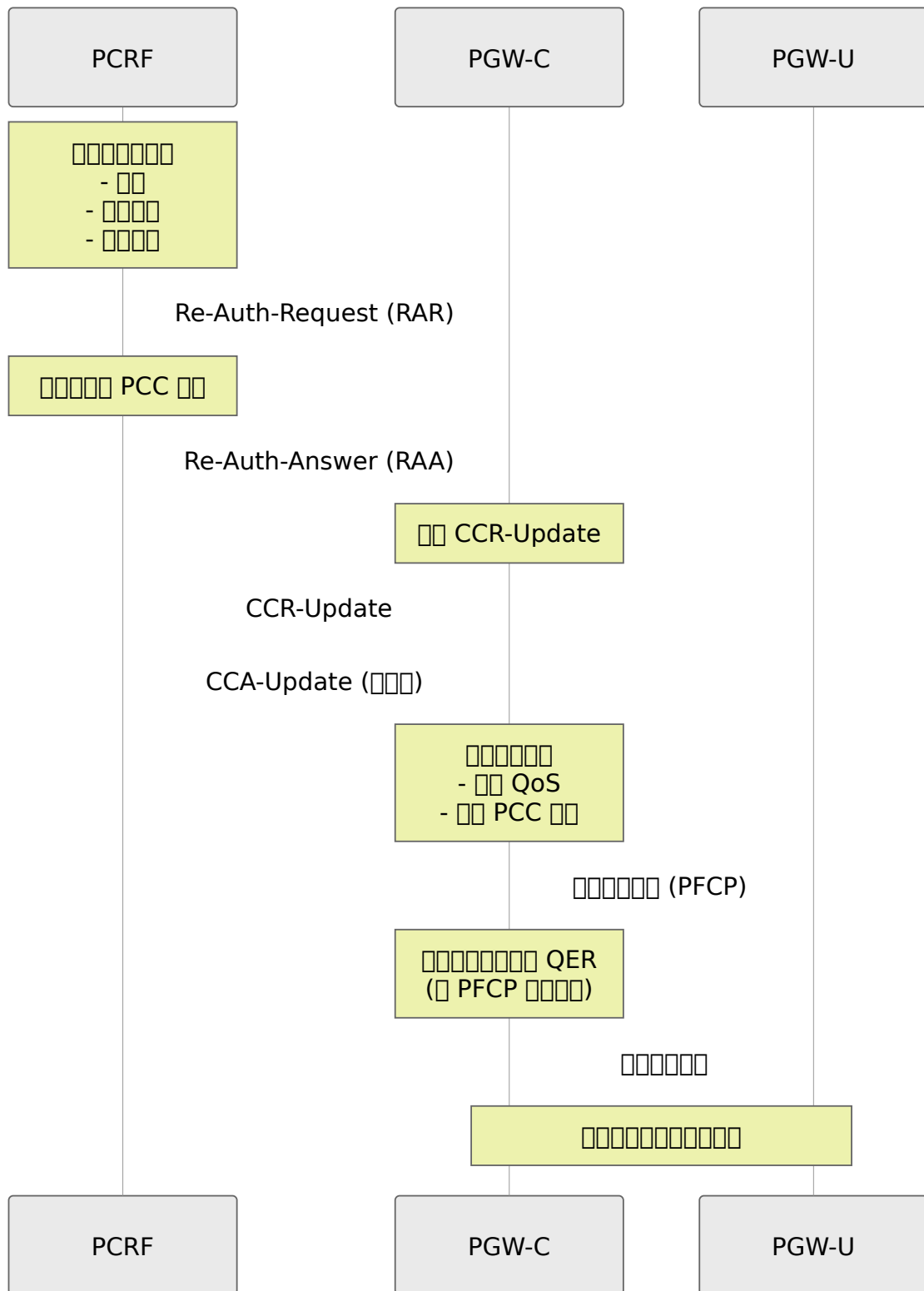
```

□□□□□

- `0000000000`
 - `3GPP PLMN 000000 epc.mncXXX.mccYYY.3gppnetwork.org`
-



□□□□□□□□□□



```

sequenceDiagram
    participant 5GC
    participant OmniCore
    participant OmniCall
    participant OmniRAN
    participant OmniCharge
    participant Platform

    5GC->>OmniCore: GTP-C (1. UE IP, 2. Session-ID, 3. Charging-ID)
    OmniCore->>OmniCall: CCR-Initial (Diameter Gx)
    OmniCall->>OmniCharge: CCA-Initial (IMSI, APN, UE IP, QoS)
    OmniCharge->>OmniRAN: CCA-Initial (Result-Code: 2001, QoS-Information, PCC)
    OmniRAN->>OmniCore: CCA-Initial (AMBR, PCC)
    OmniCore->>Platform: PCF (PCRF, QoS, QER, PCF)
    Platform->>OmniCore: PCF
    OmniCore->>5GC: GTP-C
    
```

The diagram illustrates the QoS management process across various network components. The 5GC initiates the process by sending a GTP-C message to the OmniCore, which then sends a CCR-Initial (Diameter Gx) message to the OmniCall. The OmniCall sends a CCA-Initial message to the OmniCharge, which in turn sends a CCA-Initial message to the OmniRAN. The OmniRAN sends a CCA-Initial message to the OmniCore, which then sends a PCF message to the Platform. The Platform sends a PCF message back to the OmniCore, which then sends a GTP-C message back to the 5GC.

PGW-C

PGW-C

PGW-C CCA Diameter

PGW-C

PGW-C	PGW-C	PGW-C
2001	DIAMETER_SUCCESS	PGW-C

PGW-C (5xxx)

PGW-C	PGW-C	PGW-C
5002	DIAMETER_UNKNOWN_SESSION_ID	PGW-C
5030	DIAMETER_USER_UNKNOWN	PGW-C
5140	DIAMETER_ERROR_INITIAL_PARAMETERS	PGW-C
5003	DIAMETER_AUTHORIZATION_REJECTED	PGW-C

PGW-C (4xxx)

PGW-C	PGW-C	PGW-C
4001	DIAMETER_AUTHENTICATION_REJECTED	PGW-C
4010	DIAMETER_TOO_BUSY	PGW-C
4012	DIAMETER_UNABLE_TO_COMPLY	PGW-C

实验结果

实验结果

Experimental-Result (Grouped)
└─ Vendor-Id: 10415 (3GPP)
└─ Experimental-Result-Code: <vendor-specific code>

3GPP 实验结果

错误码	错误名称	描述
5065	IP_CAN_SESSION_NOT_AVAILABLE	PCRF 不可用
5143	INVALID_SERVICE_INFORMATION	无效的服务信息

实验结果

CCR-I 实验

PCRF 实验 CCR-Initial

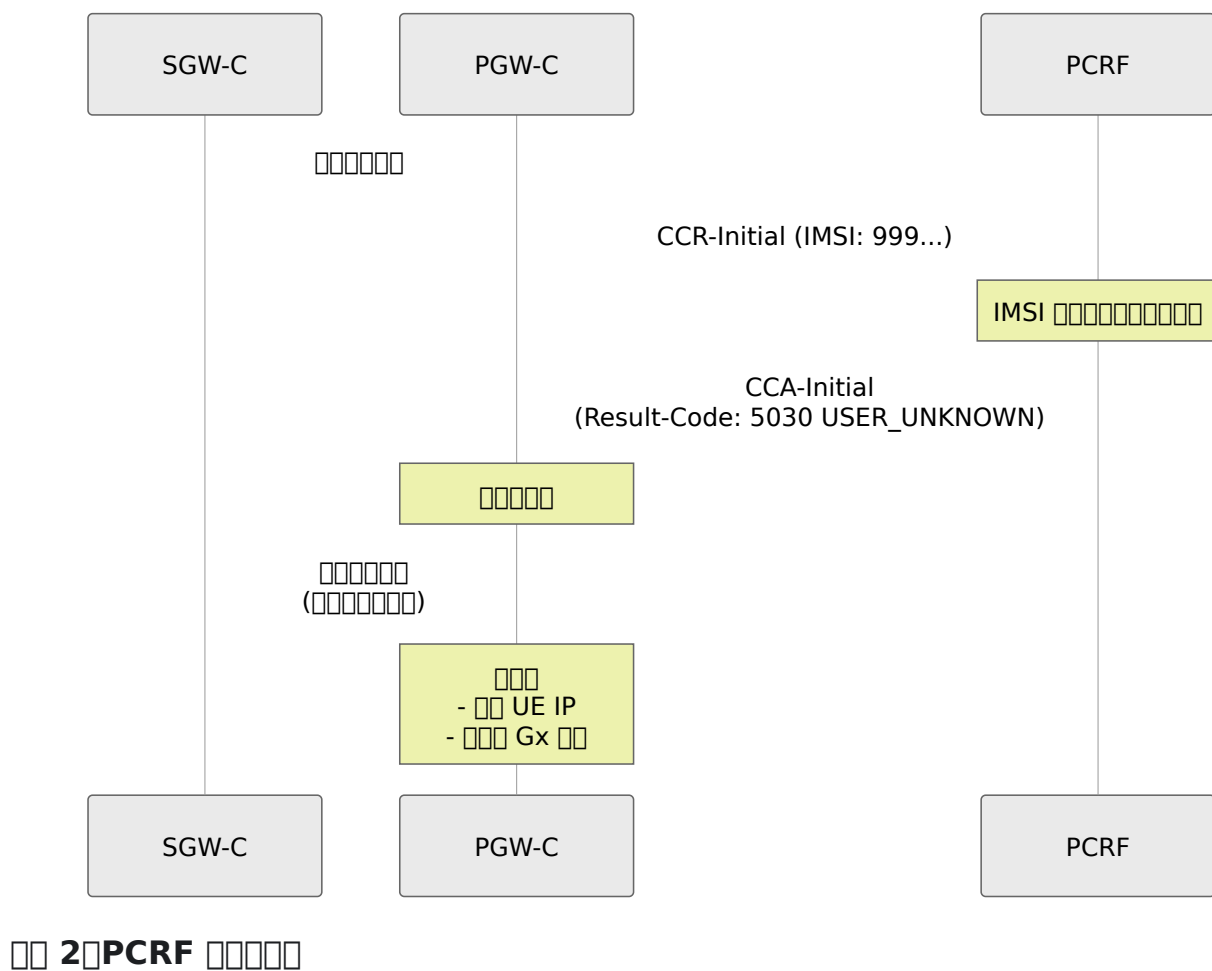
- PGW-C 实验 5
- 实验 CCA
 - “Session-ID CCR-Initial ...”
 - SGW-C 实验
 - 实验
- SGW-C 实验

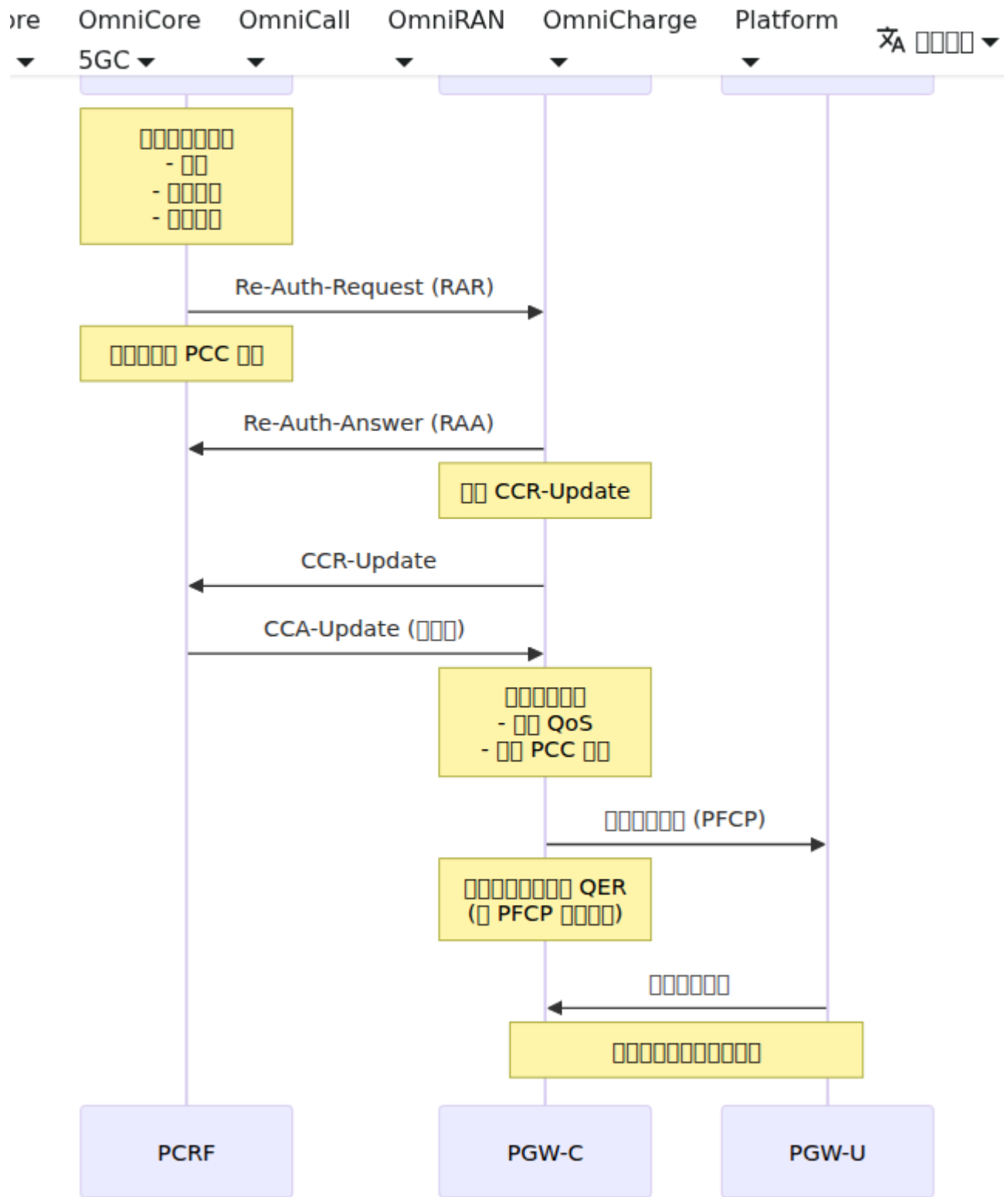
SGW-C 实验

CCR-Initial PGW-C SGW-C 实验
:remote_peer_not_responding

实验结果

1 PCRF 实验





□□□□

□□□□

1. Diameter □□□□□□

□□□

- □□□“Diameter □□□□□□”
- □□□ CCR-Initial

□□□□□

- PCRF □□□□
- □□□ PCRF IP □□□
- □□□□□ Diameter □□□3868□
- Diameter □□□□□/□□□□□□

□□□□□

```
# □□□□□□  
ping <pcrf_ip>  
  
# □□ Diameter □□□TCP 3868□  
telnet <pcrf_ip> 3868  
  
# □□ Diameter □□□□  
# □□□□□□□□ FQDN□□□□ IP
```

□□□□□

```

config :pgw_c,
  diameter: %{
    # FQDN or IP
    host: "pgw_c.epc.mnc999.mcc999.3gppnetwork.org",
    realm: "epc.mnc999.mcc999.3gppnetwork.org",
    peer_list: [
      %{
        host: "pcrf.epc.mnc999.mcc999.3gppnetwork.org",
        ip: "10.0.0.30"
      }
    ]
  }
}

```

2. CCR-Initial

-
- “CCR-Initial”

- PCRF
-
- PCRF Session-ID

1. PCRF
2. PCRF
3. ping <pcrf_ip>
- 4.

3. PCRF

- CCA-Initial Result-Code != 2001
-

□□□□□□

□□□□	□□□□	□□□□
5030	IMSI □□□□□□□□	□ HSS/SPR □□□□□
5003	□□□□□	□□□□□□
4010	PCRF □□□□	□□□□□ PCRF □□

□□□□□

```
# PGW-C □□□□□  
[error] Diameter Gx □□□□□□□ 5030 (DIAMETER_USER_UNKNOWN)  
[error] IMSI 3102609999999999 □ PCRF □□
```

4. QoS □□□

□□□

- □□□□□□ QoS □□
- □□□□□□□□□□

□□□□□

1. □□ CCA-Initial□

- □□ QoS-Information AVP □□□□
- □□ APN-Aggregate-Max-Bitrate-UL/DL □

2. □□ PFCP □□□□□

- □□ QER □□□□□□□ MBR □□□
- □□ PGW-U □□□□□ QER □□

3. □□ PCRF □□□

- □□ PCRF □□

- 配置 QoS

5. Diameter 配置

配置

- Diameter 配置 PCRF
- 配置“配置”

配置

- 配置

配置

配置

```
# 配置
config :pgw_c,
  diameter: %{
    realm: "epc.mnc999.mcc999.3gppnetwork.org", # PGW-C 配置
    peer_list: [
      %{
        realm: "epc.mnc999.mcc999.3gppnetwork.org" # PCRF 配置
      }
    ]
  }
}
```

CCR-Initial 配置

```
Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
```

配置 Gx 配置

配置


```

# Gx 消息
rate(gx_inbound_messages_total{message_type="gx_CCA"}[5m])
rate(gx_outbound_messages_total{message_type="gx_CCR"}[5m])

# Gx 错误
rate(gx_inbound_errors_total[5m])

# Gx 成功率
sum(rate(gx_outbound_responses_total{result_code_class="2xxx"}[5m])) /
sum(rate(gx_outbound_responses_total[5m])) * 100

# 每个 PCRF 的 Gx 成功率
rate(gx_outbound_responses_total{result_code_class!="2xxx"}[5m])
by (diameter_host)

# Gx 会话
session_id_registry_count

# Gx 处理时长
histogram_quantile(0.95,
rate(gx_inbound_handling_duration_bucket[5m]))

```

数据源和聚合函数

`gx_outbound_responses_total` 聚合函数 `rate` 聚合函数 `sum` 聚合函数 `by` 聚合函数 `histogram_quantile` 聚合函数

- `message_type` 聚合函数 `gx_RAA` `gx_CCA`
- `result_code_class` 聚合函数 `2xxx` `3xxx` `4xxx` `5xxx`
- `diameter_host` 聚合函数 `PCRF` 聚合函数

聚合函数

- **2001** `DIAMETER_SUCCESS` - 成功
- **3001** `DIAMETER_COMMAND_UNSUPPORTED` - 不支持命令
- **5012** `DIAMETER_UNABLE_TO_COMPLY` - 无法遵守
- **5030** `DIAMETER_USER_UNKNOWN` - 用户未知

聚合函数

```

# Gx 错误率
- alert: GxErrorRateHigh
  expr: rate(gx_inbound_errors_total[5m]) > 0.1
  for: 5m
  annotations:
    summary: "Gx 错误率"

# Gx 响应失败率
- alert: GxResponseFailureRate
  expr: |

sum(rate(gx_outbound_responses_total{result_code_class!="2xxx"}
[5m])) /
  sum(rate(gx_outbound_responses_total[5m])) > 0.1
  for: 5m
  annotations:
    summary: "Gx 响应失败率"
    description: "Gx 10% 响应失败率"

# PCRF 失败率
- alert: GxPCRFFailures
  expr:
rate(gx_outbound_responses_total{result_code_class=~"4xxx|5xxx"}
[5m]) by (diameter_host) > 0.05
  for: 3m
  annotations:
    summary: "PCRF {{ $labels.diameter_host }} 失败率"
    description: "PCRF 失败率"

# 会话拒绝率
- alert: GxSessionRejection
  expr: rate(gx_inbound_errors_total{result_code="5030"}[5m]) >
0.01
  for: 5m
  annotations:
    summary: "PCRF USER_UNKNOWN"

```

错误率

响应失败率 Diameter 失败率

```
# config/runtime.exs
config :logger, level: :debug

# 配置日志
iex> Logger.configure(level: :debug)
```

配置日志

- [debug] 配置 CCR-InitialSession-ID: ...
 - [debug] 配置 CCA-InitialResult-Code 2001
 - [error] Diameter 配置...
-

Web UI - Diameter 配置

OmniPGW 配置 Web UI 配置 Diameter 配置

Diameter 配置

配置 `http://<omnipgw-ip>:<web-port>/diameter`

PCRF Diameter Gx PCRF

1. PCRF

- PCRF - PCRF
- PCRF - PCRF
- 1 PCRF

2. PCRF PCRF

- PCRF - Diameter PCRF (Origin-Host)
- **IP** PCRF - PCRF IP
- PCRF - Diameter PCRF 3868
- PCRF - PCRF/PCRF
- PCRF - TCP PCRF SCTP
- PCRF - PCRF PGW PCRF
- PCRF - Diameter PCRF
- PCRF - PCRF PCRF
- PCRF **ID** - PCRF Diameter PCRF Gx = 16777238

3. PCRF PCRF

- PCRF
- PCRF CER/CEA PCRF
- PCRF
- PCRF

PCRF

1. 設定する Diameter 関
2. 設定する PCRF 関“設定”
3. 設定する関
4. 設定する ID 関 Gx16777238

設定する関**Gx** 関

1. 設定する関“PCRF 関”
2. 設定する Diameter 関
3. 設定する関
 - 設定
 - 設定
 - 設定 PCRF 関
 - 設定 TCP 3868 関
 - 設定
 - 設定
 - PCRF 関

設定 **Diameter** 関

1. 設定する PCRF 関
2. 設定する Diameter 関
3. 設定する関
4. 設定する関“設定”
5. 設定する関
 - 設定
 - 設定 ID 関 Gx
 - 設定する PCRF 関

設定

- 設定する PCRF 関
1. Diameter 関“設定”
 2. 設定する PCRF 関“設定”
 3. 設定する関
 4. 設定する関“設定”

設定 **Diameter** 関

- 配置“名称”参数
- ID 为 Gx16777238
- 配置 PCRF 参数

配置参数

```
Web UI 参数
配置“名称”
参数
initiate_connection: true
```

- 配置
- OmniPGW 参数
 - PCRF 参数
 - 配置参数

配置

- 配置 - 1 参数
- 配置 - 参数/参数
- 配置 **Diameter** 参数 - 配置 Diameter CLI 参数
- 配置 - 配置参数
- 配置 - 配置 Diameter 参数
- 配置 - 配置 Diameter 参数

配置

配置 Web UI 配置 Prometheus 参数

- Gx 参数
- CCR/CCA 参数
- 参数

Web UI = “配置参数” 参数 = “配置参数”

PCRF

PCRF 功能

- **PCRF** - Diameter 协议 PCRF 功能
- **PFCP** 协议 - 用于 PCC 策略 QER 和 QoS 控制
- **PCRF** - 策略控制功能
- **QoS** 策略 - 用于 QoS 控制

PCRF 接口

- **Diameter Gy** 接口 - 用于 PCC 策略控制
- **CDR** 接口 - 用于计费数据记录
- **PCO** 接口 - IMS 网络 P-CSCF 接口

PCRF 部署

- **PCRF** - Gx 接口 PCRF 部署
- **S5/S8** 接口 - 用于策略控制

PCRF 部署

5G/LTE Gy/Ro

5G/LTE OCS

1

1. 5G
2. 3GPP
3. Gy/Ro
4. 5G
5. 5G
6. 5G
7. 5G
8. 5G
9. 5G
10. 5G
11. 5Gx
12. 5G

2

Gy IMS Ro PGW-C** OCS**

- 5G - 5G
- 5G - 5G
- 5G - 5G
- 5G - 5G
- 5G - 5G

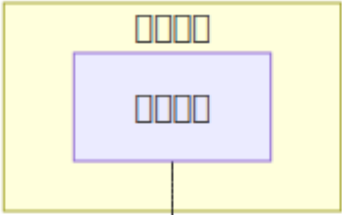
□□□□□□□

□□	□□□□□Gy/Ro□	□□□□□Gz/Rf□
□□	□□□□□□□	□□□□□
□□	□□□□□	□□□□□
□□□□	□□□□□□□□□	□□□□□□□□
□□	OCS□□□□□□□□	CGF/CDF□□□□□□□□
□□	□□□□□	□□□□□□□□□
□□□	□□□□□□□	□□□□□□□
□□□□	□□□□□□□□□□□	□□□□□□

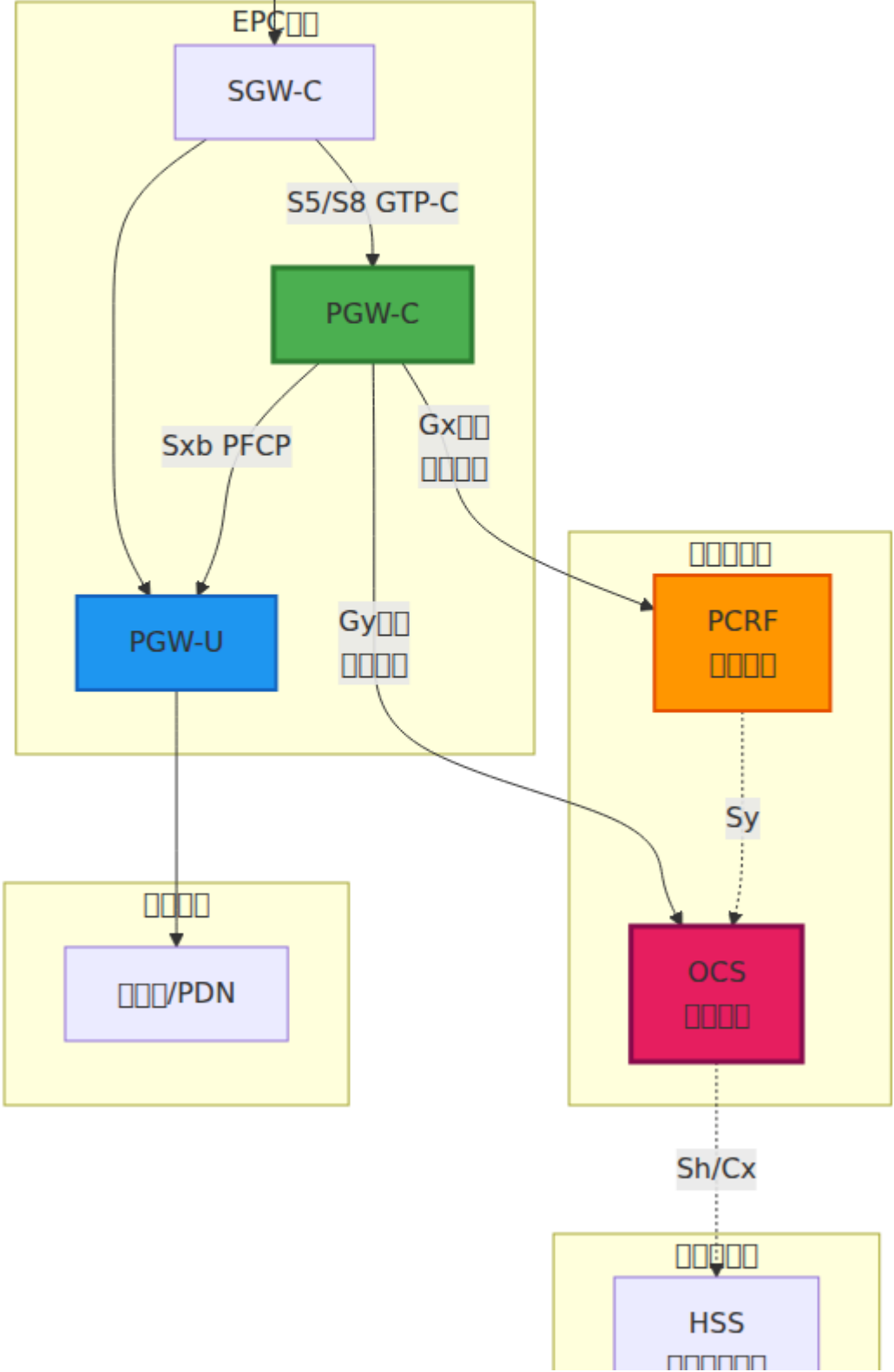
□□□ □□CDR□□ □□□□□□□□□□□□□□

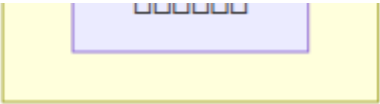
□□□ □□□□ □□□□□PDN□□□□□□□□□□□□

□□□□□Gy



ore OmniCore OmniCall OmniRAN OmniCharge Platform
▼ 5GC ▼ ▼ ▼ ▼

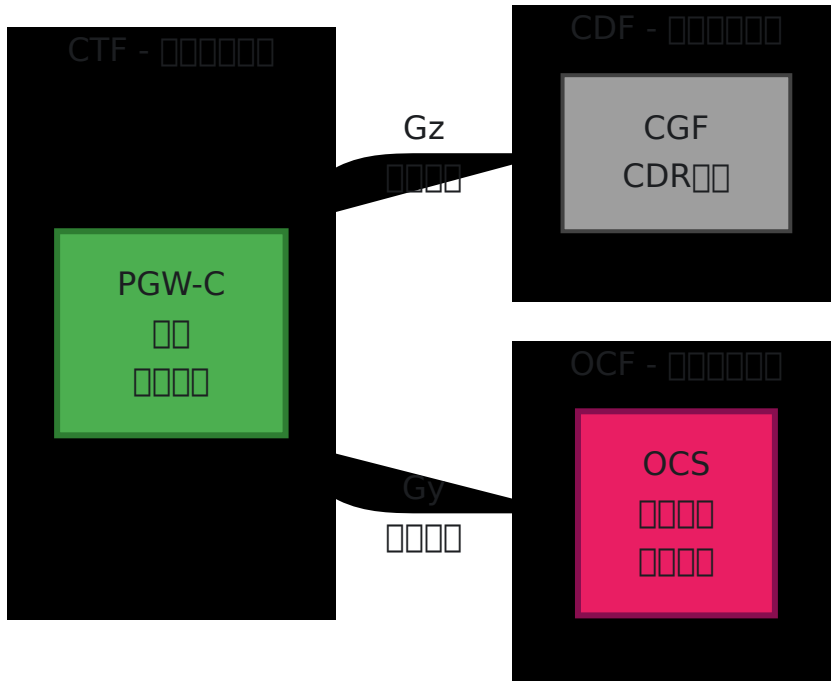




0000

00	00
0000	0000000000OCS000
0000	000000000000000000
000000	000000
0000	00000000000000
0000	0000000000
0000	00000000000000

3GPP



CTF

PGW-C CTF

1. PGW-C - PGW-C
2. PGW-C - PGW-C
3. PGW-C - PGW-C
4. PGW-C - PGW-C
5. PGW-C - PGW-C

OCF

OCS OCF

1. OCS - OCS
2. OCS - OCS MB
3. OCS - OCS

4. Gy/Ro - 3GPP TS 32.299
5. Gy/Ro - 3GPP TS 32.251

Gy/Ro

3GPP

- 3GPP TS 32.299
- 3GPP TS 32.251
- 4Gy/Ro - 3GPP TS 32.299
- RFC 4006

UE PDN Session-ID Gy/Ro

- CCR-Initial
- CCR-Update
- CCR-Termination

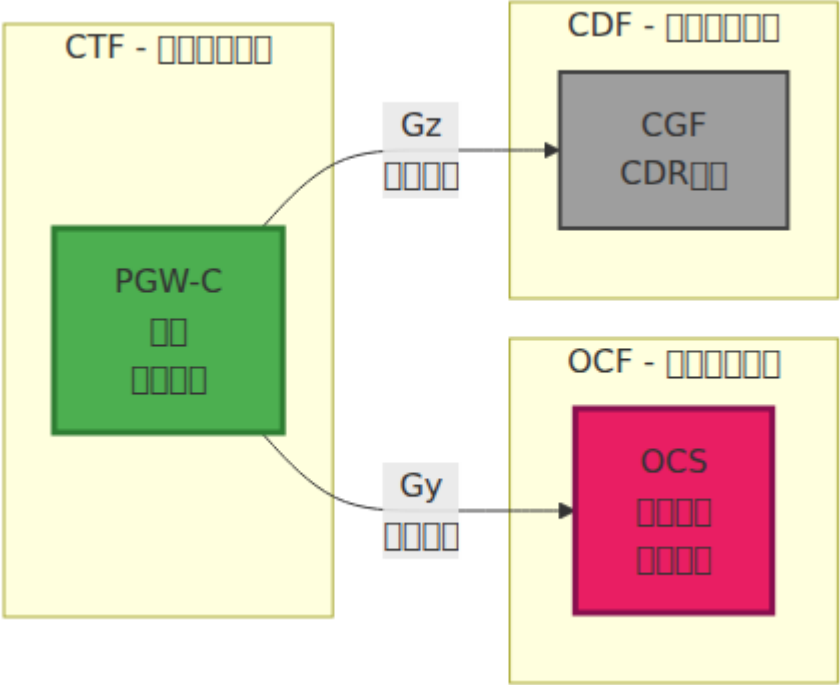
ID

Session-ID: <Origin-Host>;<high32>;<low32>[;<optional>]
omni-
pgw_c.epc.mnc999.mcc999.3gppnetwork.org;9876543210;12345;gy

- Origin-Host:** PGW-C
- high32:** 32
- low32:** 32
- optional:** "gy" Gx

□□□□□□

□□□□



CCR-Initial□□□□□□□ - □□□

□□□ UE□□PDN□□□□□□□□□□□□□□

□□□

- □□OCS□□□□□□□□
- □□□□□□□□□□
- □□Gy/Ro□□

PGW-C□□□□□□**AVP**□

AVP名	AVP番号	型	注
Session-Id	263	UTF8String	10Gy
Auth-Application-Id	258	Unsigned32	4
Origin-Host	264	DiamIdent	PGW-C
Origin-Realm	296	DiamIdent	PGW-C
Destination-Realm	283	DiamIdent	OCS
CC-Request-Type	416	Enumerated	1 = INITIAL_REQUEST
CC-Request-Number	415	Unsigned32	0
Subscription-Id	443	Grouped	UE IMS/MSISDN
Service-Context-Id	461	UTF8String	
Multiple-Services-Credit-Control	456	Grouped	
Requested-Service-Unit	437	Grouped	
Used-Service-Unit	446	Grouped	0
Service-Identifier	439	Unsigned32	
Rating-Group	432	Unsigned32	

CCR-I


```
CCR (消息: 272, 消息)
├─ Session-Id: "pgw_c.example.com;123;456;gy"
├─ Auth-Application-Id: 4
├─ Origin-Host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org"
├─ Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├─ Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├─ CC-Request-Type: INITIAL_REQUEST (1)
├─ CC-Request-Number: 0
├─ Subscription-Id (Grouped)
│   ├─ Subscription-Id-Type: END_USER_IMSI (1)
│   └─ Subscription-Id-Data: "310260123456789"
├─ Subscription-Id (Grouped)
│   ├─ Subscription-Id-Type: END_USER_E164 (0)
│   └─ Subscription-Id-Data: "15551234567"
├─ Service-Context-Id: "32251@3gpp.org"
├─ Multiple-Services-Credit-Control (Grouped)
│   ├─ Service-Identifier: 1
│   ├─ Rating-Group: 100
│   └─ Requested-Service-Unit (Grouped)
│       └─ CC-Total-Octets: 10000000 (10 MB)
└─ Used-Service-Unit (Grouped)
    └─ CC-Total-Octets: 0 (0)
```

CCA-Initial消息 - 消息

消息 OCS消息CCR-I

消息

- 消息消息
- 消息消息
- 消息消息

PGW-C消息AVP

AVP名	AVP番号	説明
Result-Code	268	結果コード (2001: 成功)
Multiple-Services-Credit-Control	456	複数サービスのクレジット制御
Granted-Service-Unit	431	許可されたサービス単位
Validity-Time	448	有効時間
Result-Code	268	結果コード (2001: 成功)
Final-Unit-Indication	430	最終単位指示
Volume-Quota-Threshold	-	体積クォータ閾値

例

```
CCA (Sequence: 272, Type: CCR)
├─ Session-Id: "pgw_c.example.com;123;456;gy"
├─ Result-Code: DIAMETER_SUCCESS (2001)
├─ Origin-Host: "ocs.example.com"
├─ Origin-Realm: "example.com"
├─ Auth-Application-Id: 4
├─ CC-Request-Type: INITIAL_REQUEST (1)
├─ CC-Request-Number: 0
├─ Multiple-Services-Credit-Control (Grouped)
│   ├── Result-Code: DIAMETER_SUCCESS (2001)
│   ├── Service-Identifier: 1
│   ├── Rating-Group: 100
│   ├── Granted-Service-Unit (Grouped)
│   │   └─ CC-Total-Octets: 10000000 (10 MB)
│   ├── Validity-Time: 3600 (1時間)
│   └─ Volume-Quota-Threshold: 8000000 (8 MB使用率80%)
```

CCR-Updateメッセージ - 例

例

- 80%
-
-
-

-
-
-

CCR-I

- CC-Request-Type: UPDATE_REQUEST (2)
- CC-Request-Number
- Used-Service-Unit
- Requested-Service-Unit

CCR-U

```
CCR (seq: 272, )
├─ Session-Id: "pgw_c.example.com;123;456;gy"
├─ Auth-Application-Id: 4
├─ Origin-Host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org"
├─ Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├─ Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├─ CC-Request-Type: UPDATE_REQUEST (2)
├─ CC-Request-Number: 1
└─ Multiple-Services-Credit-Control (Grouped)
    ├─ Service-Identifier: 1
    ├─ Rating-Group: 100
    ├─ Used-Service-Unit (Grouped)
    │   └─ CC-Total-Octets: 8000000 (8 MB)
    └─ Requested-Service-Unit (Grouped)
        └─ CC-Total-Octets: 10000000 (10 MB)
```

CCA-Update

OCSCCR-U

-
-
-

1.

```
CCA ( )
└─ Multiple-Services-Credit-Control
    └─ Result-Code: DIAMETER_SUCCESS (2001)
    └─ Granted-Service-Unit
        └─ CC-Total-Octets: 10000000 (10 MB)
    └─ Validity-Time: 3600
```

2.

```
CCA ( )
└─ Multiple-Services-Credit-Control
    └─ Result-Code: DIAMETER_SUCCESS (2001)
    └─ Granted-Service-Unit
        └─ CC-Total-Octets: 1000000 (1 MB)
    └─ Final-Unit-Indication
        └─ Final-Unit-Action: TERMINATE (0)
```

3.

CCA ()

```
└─ Result-Code: DIAMETER_CREDIT_LIMIT_REACHED (4012)
└─ Multiple-Services-Credit-Control
    └─ Result-Code: DIAMETER_CREDIT_LIMIT_REACHED (4012)
        └─ Final-Unit-Indication
            └─ Final-Unit-Action: TERMINATE (0)
```

CCR-Termination -

- UE
- PDN
-

-
- Gy/Ro
-

- CC-Request-Type: TERMINATION_REQUEST (3)
- Used-Service-Unit
- Requested-Service-Unit
- Termination-Cause

CCR-T

```
CCR (消息ID: 272, 方向)
├── Session-Id: "pgw_c.example.com;123;456;gy"
├── Auth-Application-Id: 4
├── Origin-Host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org"
├── Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── CC-Request-Type: TERMINATION_REQUEST (3)
├── CC-Request-Number: 5
├── Termination-Cause: DIAMETER_LOGOUT (1)
└── Multiple-Services-Credit-Control (Grouped)
    ├── Service-Identifier: 1
    ├── Rating-Group: 100
    └── Used-Service-Unit (Grouped)
        └── CC-Total-Octets: 18500000 (约18.5 MB)
```

CCA-Termination消息 - 方向

消息 OCS消息CCR-T

消息

- 消息ID
- 消息
- 消息ID

消息CCA-T

```
CCA (消息ID: 272, 方向)
├── Session-Id: "pgw_c.example.com;123;456;gy"
├── Result-Code: DIAMETER_SUCCESS (2001)
├── Origin-Host: "ocs.example.com"
├── Origin-Realm: "example.com"
├── Auth-Application-Id: 4
├── CC-Request-Type: TERMINATION_REQUEST (3)
└── CC-Request-Number: 5
```

计费策略

计费策略

OCS计费策略

计费策略	AVP	计费策略	计费策略
计费策略	CC-Time	计费策略	计费策略
计费策略	CC-Total-Octets	计费策略+计费策略	计费策略
计费策略	CC-Input-Octets, CC-Output-Octets	计费策略	计费策略
计费策略	CC-Service-Specific-Units	计费策略	计费策略API
计费策略	-	计费策略	计费策略

计费策略

计费策略 PGW-C计费策略

计费策略 OCS计费策略**Volume-Quota-Threshold**计费策略**Time-Quota-Threshold**计费策略PGW-C计费策略PGW-U计费策略PFCP计费策略**PFCP**计费策略

计费策略

1. OCS[]10 MB[]80%[]8 MB[]
2. PGW-C[]PGW-U[]PFCP[]
3. []8 MB[]
→ PGW-C[]CCR-Update
→ []
4. OCS[]
5. []CCR-Update[]
→ PGW-C[]

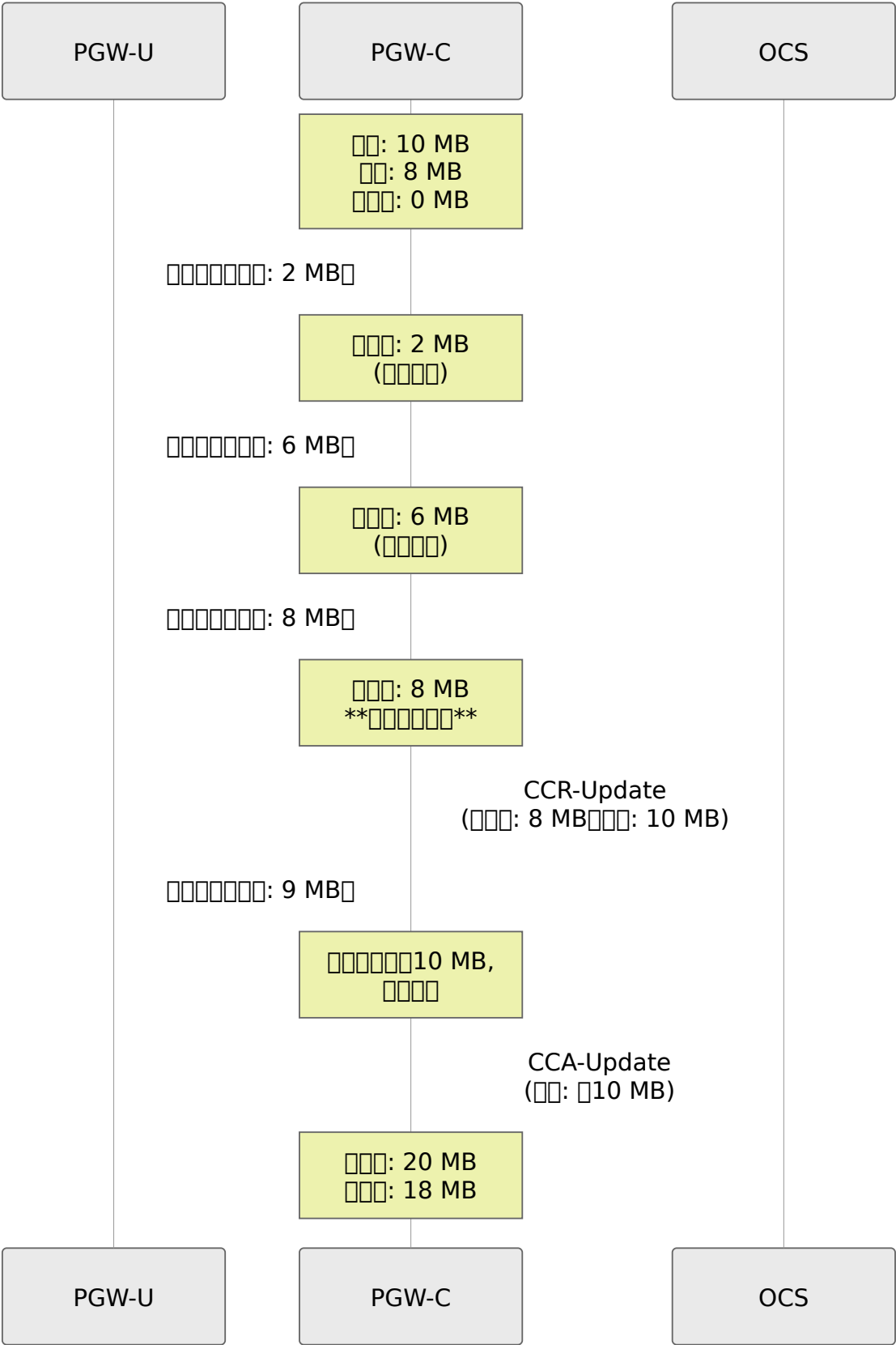
[]

Granted-Service-Unit: 10000000[]10 MB[]
Volume-Quota-Threshold: 80000000[]8 MB[]

[]8 MB[] → 00[]CCR-Update
[]2 MB[]OCS[]

PGW-C[]

PGW-C[]PGW-U[]**PFCP**[]



10 MB

8 MB

OCS CCA Final-Unit-Indication AVP

Final-Unit-Action		PGW-C
TERMINATE	0	
REDIRECT	1	
RESTRICT_ACCESS	2	

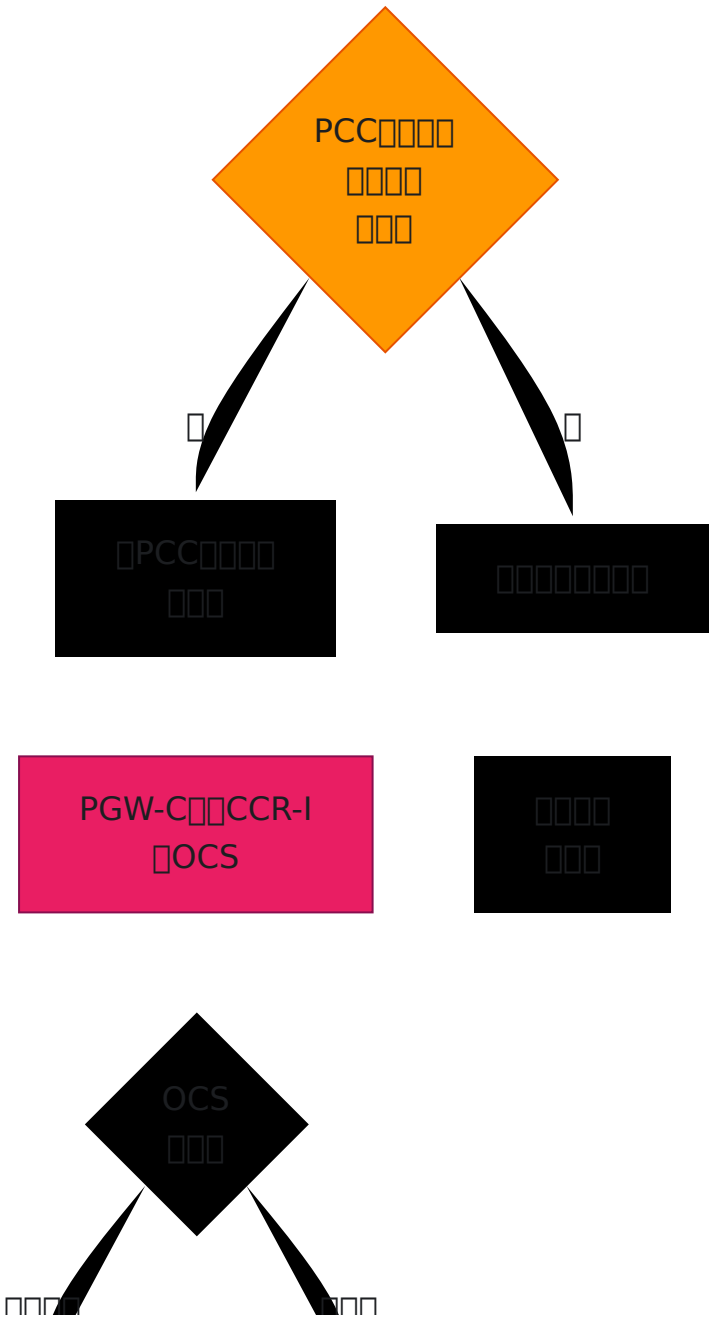
```
CCA ( )
├─ Multiple-Services-Credit-Control
│   ├── Result-Code: DIAMETER_SUCCESS (2001)
│   ├── Granted-Service-Unit
│   │   └─ CC-Total-Octets: 1000000 (1 MB)
│   └─ Final-Unit-Indication
│       ├── Final-Unit-Action: REDIRECT (1)
│       └─ Redirect-Server (Grouped)
│           ├── Redirect-Address-Type: URL (2)
│           └─ Redirect-Server-Address:
│               "http://topup.example.com"
```

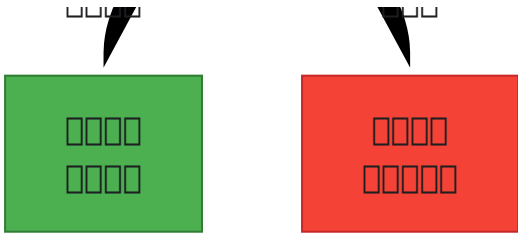
PGW-C

- 1. **TERMINATE:** CCR-T
- 2. **REDIRECT:** PFCP HTTP URL
- 3. **RESTRICT_ACCESS:** PFCP IP

□□□□□PCR□□Gx□□□□□**PCC**□□□□□□□□□□PCC□□□□□□□□□□Gx□□□

□□□□□□□





PGW-C PCC

PCRF Gx CCA-I

```
CCA (Gx)
└─ Charging-Rule-Definition (Grouped)
    ├── Charging-Rule-Name: "prepaid_data_rule"
    ├── Rating-Group: 100
    ├── Online: 1 ( )
    ├── Offline: 0 ( )
    ├── Metering-Method: VOLUME (1)
    ├── Precedence: 100
    ├── Flow-Information: [...]
    └─ QoS-Information: [...]
```

PCC AVP

AVP	AVP		
Rating-Group	432	Unsigned32	OCSE
Online	1009	0=, 1=	Gy
Offline	1008	0=, 1=	Gz
Metering-Method	1007	0=, 1=, 2=	
Reporting-Level	1011	0=, 1=	

		Rating-Group	
1	0		
0	1		
1	1		
0	0	-	

PDN

0000000000

└─ Rating-Group: 1000000000
└─ Online: 1

00001000000

└─ Rating-Group: 2000000000
└─ Online: 1

000020IMS000

└─ Rating-Group: 3000000
└─ Online: 1

PGW-C Gy000

- 00CCR-I00000MSCC00000000000000

CCR-Initial

└─ Session-Id: "..."
└─ Multiple-Services-Credit-Control
 └─ [Rating-Group: 100] → 0000
 └─ [Rating-Group: 200] → 0000
 └─ [Rating-Group: 300] → 00

OCS000

CCA-Initial

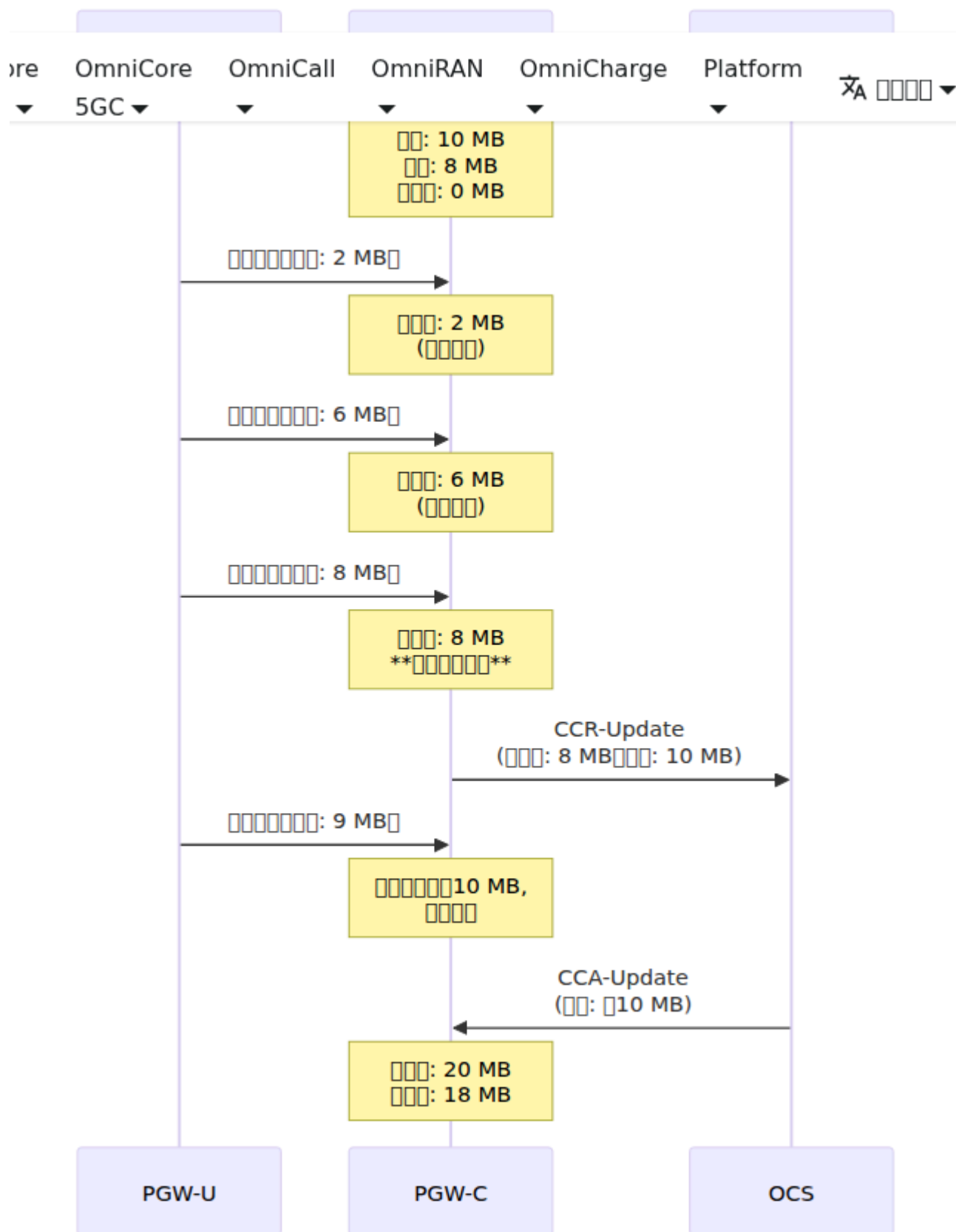
└─ Multiple-Services-Credit-Control
 └─ [Rating-Group: 100] → 00: 10 MB
 └─ [Rating-Group: 200] → 00: 5 MB000000
 └─ [Rating-Group: 300] → 00: 600

0000000000

PGW-C00000000000000

```
# []  
state.charging_quotas = %  
  100 => %{granted: 10_000_000, used: 0, threshold: 8_000_000},  
  200 => %{granted: 5_000_000, used: 0, threshold: 4_000_000},  
  300 => %{granted: 60_000, used: 0, threshold: 48_000} # []  
}
```

□□□□□□□□□□



□□□□□□□□

MSCC□□□□□□□□AVP

□□□ □□□□□/□□□□□□□□□□

◆◆□□

```
Multiple-Services-Credit-Control (Grouped, AVP 456)
├─ Service-Identifier (Unsigned32, AVP 439)
├─ Rating-Group (Unsigned32, AVP 432)
├─ Requested-Service-Unit (Grouped, AVP 437)
│   └─ CC-Time (Unsigned32, AVP 420)
│   └─ CC-Total-Octets (Unsigned64, AVP 421)
│   └─ CC-Input-Octets (Unsigned64, AVP 412)
│   └─ CC-Output-Octets (Unsigned64, AVP 414)
├─ Used-Service-Unit (Grouped, AVP 446)
│   └─ [□Requested-Service-Unit□□□□□]
├─ Granted-Service-Unit (Grouped, AVP 431)
│   └─ [□Requested-Service-Unit□□□□□]
├─ Validity-Time (Unsigned32, AVP 448)
├─ Result-Code (Unsigned32, AVP 268)
└─ Final-Unit-Indication (Grouped, AVP 430)
    └─ Final-Unit-Action (Enumerated, AVP 449)
```

Table 1: Example of a simple table

Item	Description	Price
Item 1	Description 1	Price 1
Item 2	1=Item 2=Item 3=Item	100=Item 200=Item
Item 3	Description 3	Price 3
Item 4	Item	Description 4
Item 5	Item 5 RG	OCS Item 5

Table 2

Service-Identifier: 1Item 1

- Rating-Group: 100Item 1 - \$0.01/MB
- Rating-Group: 200Item 1 - \$0.05/MB

Service-Identifier: 2Item 2

- Rating-Group: 300Item 2 - \$0.10/Item

Table 3

Table 4 Gy

Table 5 config/runtime.exs

```

config :pgw_c,
  gy: %{
    # 计费功能
    enabled: true,

    # OCS鉴权
    timeout_ms: 5000,

    # PCRF鉴权
    default_requested_quota: 10_000_000, # 10 MB

    # 鉴权
    # (0.8 = 80%CCR-Update)
    quota_threshold_percentage: 0.8,

    # OCS鉴权
    # :block, :allow
    timeout_action: :block,

    # OCS鉴权
    # :terminate, :redirect
    no_credit_action: :terminate,

    # 鉴权URL
    no_credit_action: :redirect
    topup_redirect_url: "http://topup.example.com"
  },
  diameter: %{
    listen_ip: "0.0.0.0",
    host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org",
    realm: "epc.mnc999.mcc999.3gppnetwork.org",

    # OCS鉴权
    peer_list: [
      # PCRF鉴权
      %{
        host: "pcrf.epc.mnc999.mcc999.3gppnetwork.org",
        realm: "epc.mnc999.mcc999.3gppnetwork.org",
        ip: "10.0.0.30",
        initiate_connection: true
      },
      # OCS鉴权
      %{
        host: "ocs.epc.mnc999.mcc999.3gppnetwork.org",

```

```
    realm: "epc.mnc999.mcc999.3gppnetwork.org",
    ip: "10.0.0.40",
    initiate_connection: true
  }
]
```

配置参数

enabled

- `true` 启用CCR和OCS
- `false` 禁用CCR和Gy

timeout_ms

- OCS CCA超时
- 3000-5000 ms

default_requested_quota

- PCRF请求的默认配额
- 1-100 MB

quota_threshold_percentage

- CCR-Update配额阈值
- 0.75-0.85 75%-85%
- `0` = 禁用
- `1` = 启用

timeout_action

- `:block` - OCS请求超时后阻塞
- `:allow` - OCS请求超时后允许

no_credit_action

- `:terminate` - 终止会话

- **:redirect** - 重定向

配置示例

配置示例

```
config :pgw_c,  
  gy: %{  
    enabled: true,  
    timeout_action: :block,  
    no_credit_action: :terminate,  
    quota_threshold_percentage: 0.8  
  }
```

配置示例

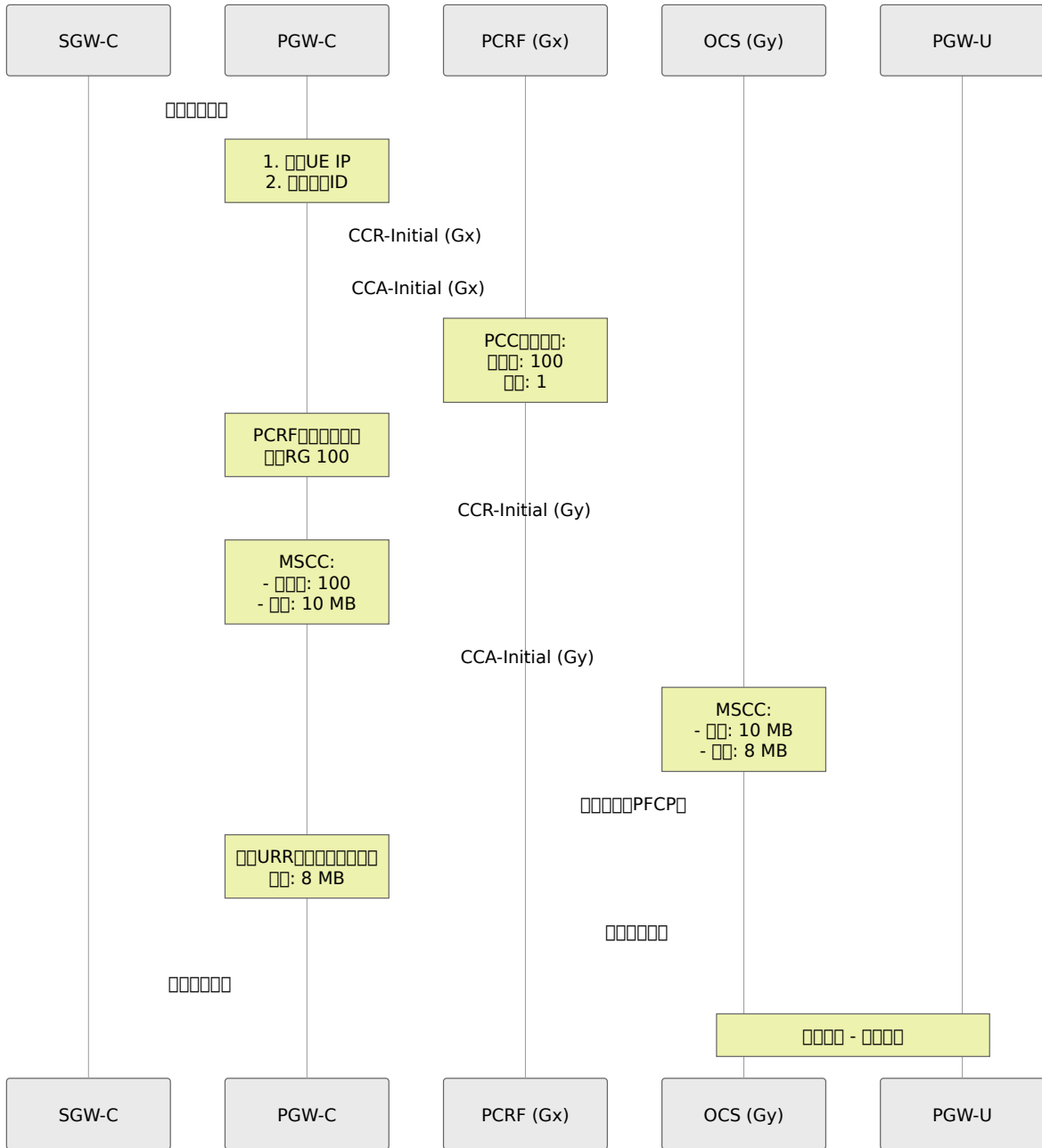
```
config :pgw_c,  
  gy: %{  
    enabled: false # 禁用  
  }
```

配置示例

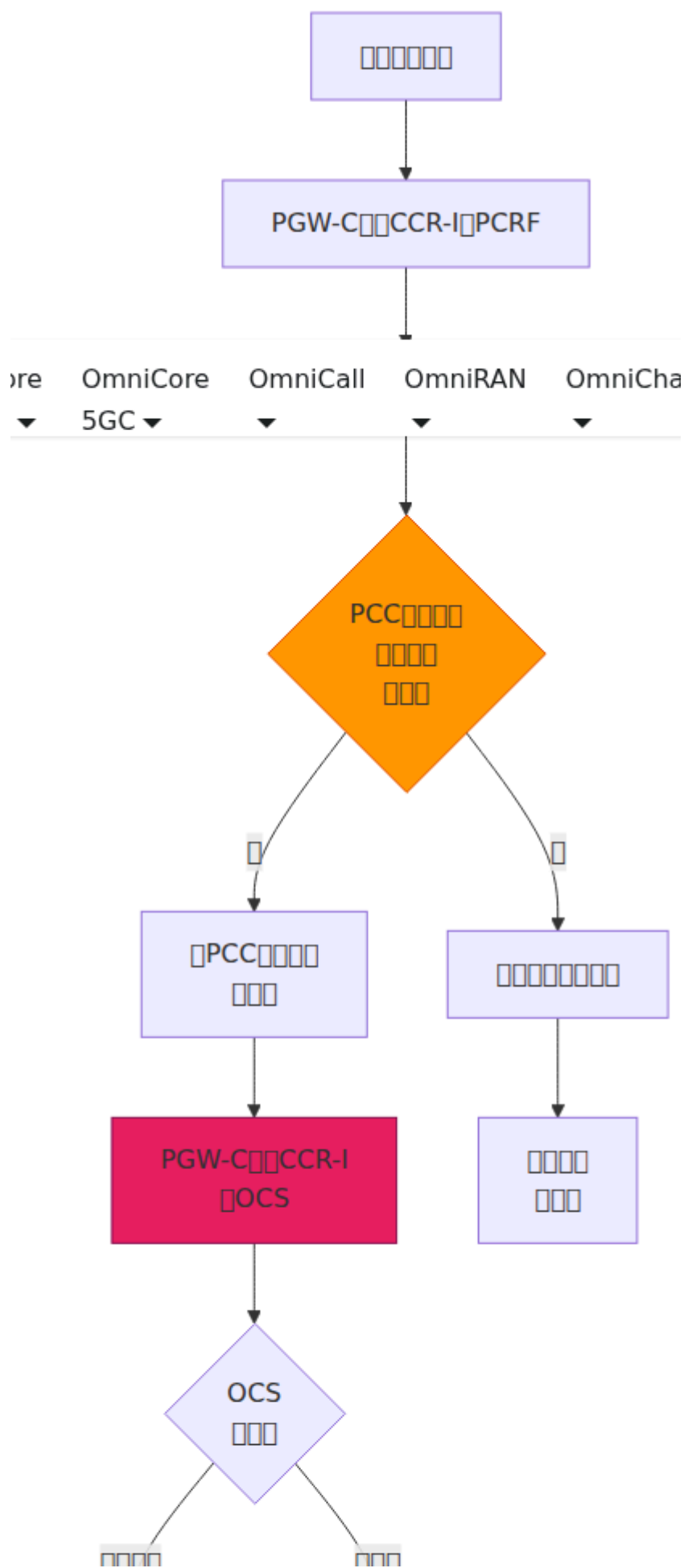
```
config :pgw_c,  
  gy: %{  
    enabled: true, # PCRF  
    timeout_action: :allow, # OCS  
    no_credit_action: :terminate  
  }
```

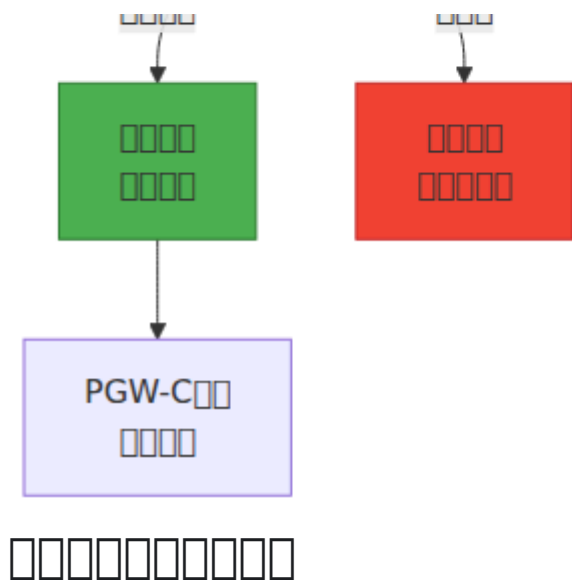
□□□□

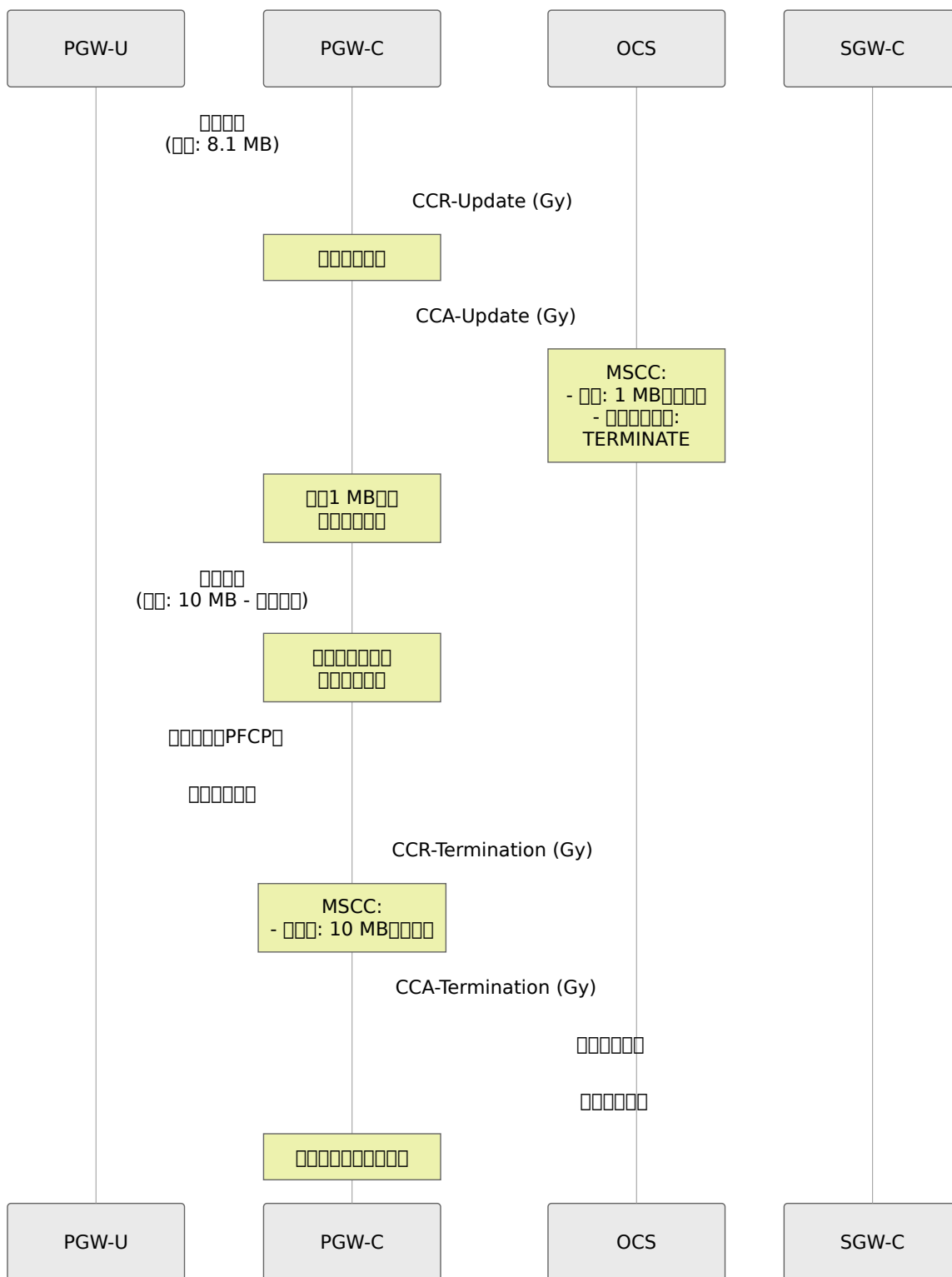
□□□□□□□□



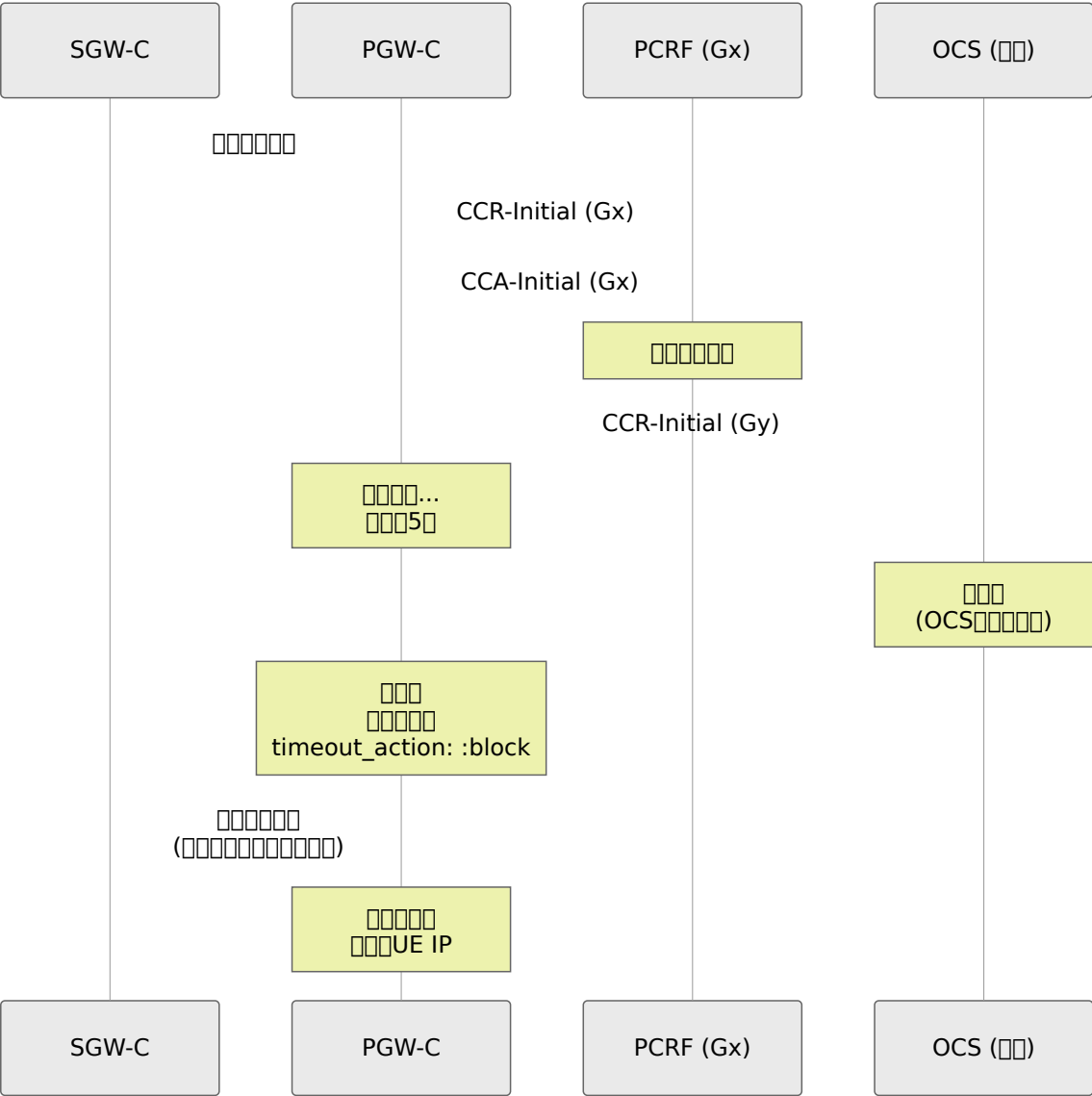
□□□□□□**CCR-Update**□







OCS



Blocked

Blocked

Blocked

PGW-C	PGW-C	PGW-C
2001	DIAMETER_SUCCESS	PGW-C

PGW-C 4xxx

PGW-C	PGW-C	PGW-C
4010	DIAMETER_TOO_BUSY	PGW-C
4011	DIAMETER_UNABLE_TO_COMPLY	PGW-C
4012	DIAMETER_CREDIT_LIMIT_REACHED	PGW-C

PGW-C 5xxx

PGW-C	PGW-C	PGW-C
5003	DIAMETER_AUTHORIZATION_REJECTED	PGW-C
5031	DIAMETER_USER_UNKNOWN	PGW-C

PGW-C

PGW-C Result-Code

1. PGW-C - PGW-C
2. **MSCC** - PGW-C

PGW-C

CCA-Initial

└─ Result-Code: DIAMETER_SUCCESS (2001) ← OK

└─ Multiple-Services-Credit-Control

└─ [Rating-Group: 100]

└─ Result-Code: DIAMETER_SUCCESS (2001) ← RG 100 OK

└─ [Rating-Group: 200]

└─ Result-Code: DIAMETER_CREDIT_LIMIT_REACHED (4012) ←
RG 200

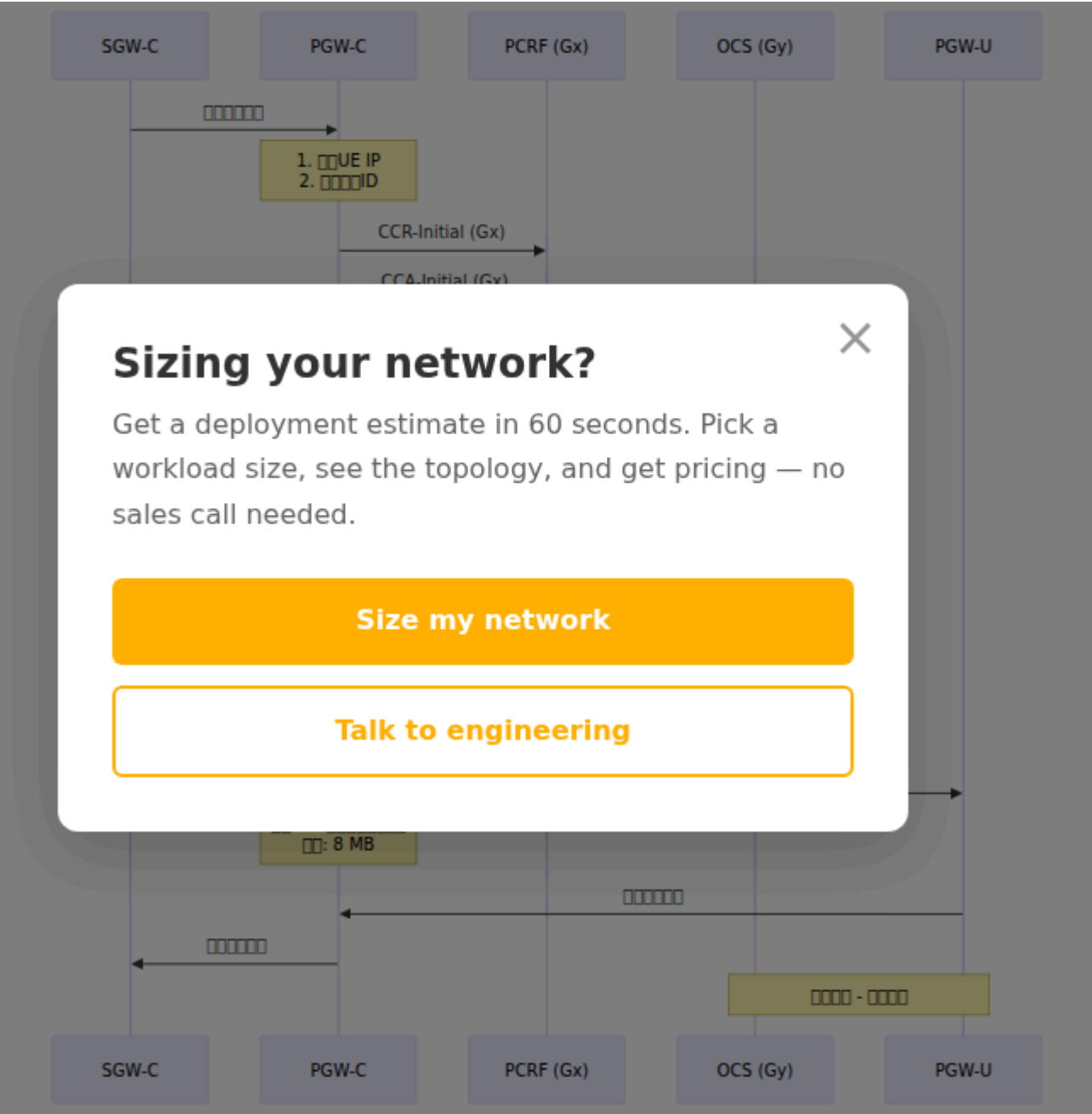
PGW-C

- 100
- 200

Gx

Gx PCRF  Gy 

GxGy



1.

PGW-C

↓

CCR-I PCRF Gx

↓

CCA-I PCC

↓

PCC

-

- =1

↓

CCR-I OCS Gy

↓

CCA-I

↓

2. PCRF RAR

PCRF Gx RAR

↓

PCC =1 =200

↓

PGW-C CCR-U OCS Gy

- 200 MSCC

↓

OCS

↓

□□□□

□□□□

1. CCR-Initial□OCS□□

□□□

- □□□□□□“OCS□□”
- □□□“CCR-Initial□Gy□□□”

□□□□□

- OCS□□□
- □□□□OCS IP□□□
- □□□□□□□□□3868□
- OCS□□

□□□□□

```
# □□□□□□
ping <ocs_ip>

# □□□□□□□TCP 3868□
telnet <ocs_ip> 3868

# □□□□
# □□□peer_list□□□□□OCS□□□
```

2. □□□OCS□□

□□□

- CCA-I□Result-Code != 2001
- □□□□□□□□

□□□□□□□

消息	消息	消息
4012	消息	消息
5003	消息	消息
5031	消息	消息OCS消息

消息

1. 消息OCS消息
2. 消息OCS消息
3. 消息CCR-I消息IMSI/MSISDN消息

3. 消息

消息

- 消息
- 消息CCR-Update

消息

- PGW-U消息URR消息
- 消息
- 消息PFCP消息

消息

1. 消息PFCP消息URR

```

消息URR
├─ URR-ID: 1
├─ Measurement-Method: VOLUME
├─ Volume-Threshold: 80000008 MB
└─ Reporting-Triggers: VOLUME_THRESHOLD

```

2. 消息PGW-U消息

3. 配置 quota_threshold_percentage

4. 配置

配置

- OCS配置“配置”
- 配置

配置

- CCR-I配置OCS配置
- PCRF配置

配置

1. 配置PCRF配置PCC配置
 2. 配置OCS配置
 3. 配置PCC配置OCS配置
-

□□

□□□□

```
# Gy□□□□
rate(gy_inbound_messages_total{message_type="cca"}[5m])
rate(gy_outbound_messages_total{message_type="ccr"}[5m])

# Gy□□□□
rate(gy_inbound_errors_total[5m])

# □□□□□□
rate(gy_quota_exhausted_total[5m])

# OCS□□□
rate(gy_timeout_total[5m])

# Gy□□□□□□□□
histogram_quantile(0.95,
rate(gy_inbound_handling_duration_bucket[5m]))
```

11

```
# Gy
- alert: GyErrorRateHigh
  expr: rate(gy_inbound_errors_total[5m]) > 0.1
  for: 5m
  annotations:
    summary: "Gy"

# OCS
- alert: OcsTimeout
  expr: rate(gy_timeout_total[5m]) > 0.05
  for: 2m
  annotations:
    summary: "OCS"

# CreditExhaustionSpike
- alert: CreditExhaustionSpike
  expr: rate(gy_quota_exhausted_total[5m]) > 10
  for: 5m
  annotations:
    summary: "CreditExhaustionSpike"
```

Web UI - Gy○○○○○○○

OmniPGW Gy/Ro OCS

```
□□□ http://<omnipgw-ip>:<web-port>/gy_simulator
```

0000 00000000000000000000

0000

1. 0000

- **IMSI** - 0000000000“310170123456789”0
- **MSISDN** - 000000000“14155551234”0
- 0000 - 0000000000000000
- 00**ID** - 00000000
- 000 - 0000

2. CCR-I00

- 00CCR-Initial0000000000
- 0000000000000000
- 00OCS0000000000

3. 00

- 0000 - 00000000Gy00
- **OCS**00 - 00OCS000000
- 0000 - 0000000000

- 网络 - 网络地址
- 网络 - 网络地址

网络

1. 网络地址IMSI/MSISDN
2. 网络地址1000000001 MB
3. 网络ID
4. 网络“CCR-I”
5. 网络OCS

网络

- 网络OCS
- 网络
- 网络
- 网络

网络

网络

- **Gx** - PCRF网络地址PCC
- **CDR** - 网络地址
- **网络** - 网络地址

网络

- **网络** - PDN网络地址
- **PFCP** - 网络URR/PGW-U网络地址
- **S5/S8** - GTP-C网络地址

□□

- □□□□ - Gy□□□□□□□□OCS□□□□
 - **UE IP**□□ - □□□□□□IP□□□□
-

□□□□□□

Gn/Gp 介面

SGSN 與 GTP-C 透過 2G/3G 網路

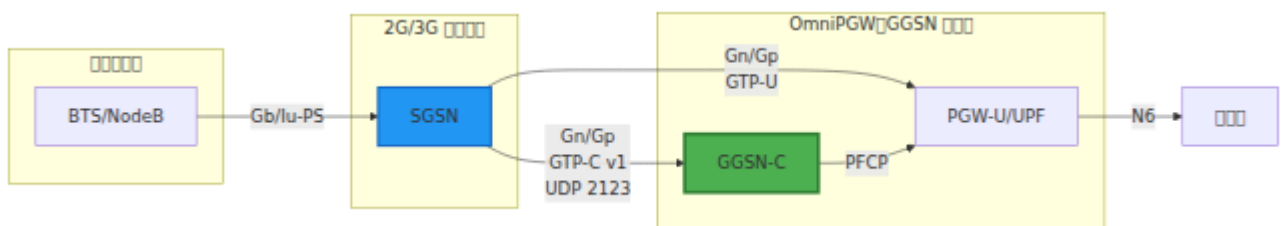
OmniPGW 與 Omnitouch 網路架構

網路

Gn/Gp 介面 與 OmniPGW 透過 2G/3G 網路 GGSN 透過 GPRS 網路與 GTP-C v1 介面
透過 GGSN 與 SGSN 透過 PDP 網路與 GTP-C v1 介面

- Gn 網路與 PLMN 網路 SGSN
- Gp 網路與 PLMN 網路 SGSN

網路與 GPRS 網路 - 網路與 GPRS 網路



PGW S5/S8

	GGSN (Gn/Gp)	PGW (S5/S8)
	GTP-C v1	GTP-C v2
3GPP	TS 29.060	TS 29.274
	PDP	PDN +
	NSAPI (0-15)	EBI (5-15)
		Gx (PCRF)
QoS	R99 QoS	EPS QoS (QCI)
	2123	2123

GTP-C 1

- GTP-C v1 (3GPP TS 29.060)
- UDP
- 2123 S5/S8 GTP-C v2
-

TEID

PDP TEID

- GGSN TEID OmniPGW
- GGSN TEID UPF

- **SGSN** 的 **TEID** 的 SGSN 的
- **SGSN** 的 **TEID** 的 SGSN 的

的

SGSN → GGSN: 的 TEID = GGSN 的 TEID

GGSN → SGSN: 的 TEID = SGSN 的 TEID

CUPS 的

OmniPGW 的 2G/3G 的 CUPS 的

- 的 **GSN** 的 OmniPGW 的 IP 的 `gn.local_ipv4_address` 的
- 的 **GSN** 的 UPF 的 IP 的 PFCP 的

的 GGSN-C 的 PGW-U/UPF 的 IP 的 CUPS 的

NSAPI 的

NSAPI 的 PDP 的

- 的 5-15 的 0-4 的
- 的 PDP 的
- 的 LTE/EPC 的 EBI

□□

□□□□

```
# config/runtime.exs
config :pgw_c,
  # □□□ gn □□□□Gn/Gp □□□□□□
  gn: %{
    # Gn □□□□□ IPv4 □□
    local_ipv4_address: "10.0.0.20",

    # □□□□□ IPv6 □□
    local_ipv6_address: nil,

    # GTP-C v1 □□□□□ S5/S8 □□□
    local_port: 2123
  },

  # □ PCO □□□□□ DNS □□□□□ PGW □□□
  dns: %{
    primary_ipv4: {8, 8, 8, 8},
    secondary_ipv4: {8, 8, 4, 4}
  }
```

Gn □□□□

□□	□□	□ □	□□	□□
local_ipv4_address	□□ □	□	-	□□ Gn □□□□□ IPv4 □□□□□□ SGSN □ □□ GGSN □□□
local_ipv6_address	□□ □	□	nil	□□□ IPv6 □□□□□□ 2G/3G □□□□□ IPv4□
local_port	□□	□	2123	GTP-C □ UDP □□□□□ 3GPP TS 29.060 □□□□□□□ S5/S8 □□□

DNS

IP	IP	IP	IP	IP
primary_ipv4	IP	IP	{8, 8, 8, 8}	IP PCO IP DNS IP IPv4 IP
secondary_ipv4	IP	IP	{8, 8, 4, 4}	IP PCO IP DNS IP IPv4 IP
primary_ipv6	IP	IP	nil	IP DNS IP IPv6 IP
secondary_ipv6	IP	IP	nil	IP DNS IP IPv6 IP

IP

IP

```
# IP SGSN IP GTP-C
iptables -A INPUT -p udp --dport 2123 -s <sgsn_network>/24 -j ACCEPT

# IP SGSN IP GTP-C
iptables -A OUTPUT -p udp --dport 2123 -d <sgsn_network>/24 -j ACCEPT
```

IP

PDP

IP PDP IP

UE SGSN → GGSN

UE PDP Context

Information Elements

IE	Length	Description
IMSI	13	IMSI (13 octets)
NSAPI	5-15	NSAPI (5-15 octets)
TEID I	4	SGSN TEID (4 octets)
TEID S	4	SGSN TEID (4 octets)
PDN	4	PDN IP (4 octets)
APN	16	APN (16 octets)
GSN I	4	SGSN IP (4 octets)
GSN S	4	SGSN IP (4 octets)
QoS	4	QoS (4 octets)
MSISDN	16	MSISDN (16 octets)
APN	16	APN (16 octets)
...	...	...

...

UE PDP Context

- └ IMSI: 310260123456789
- └ NSAPI: 5
- └ TEID MME I: 0x12345678
- └ TEID S-GW: 0x87654321
- └ Address: IPv4Address
- └ APN: internet
- └ GSN MME: 10.1.1.100
- └ GSN S-GW: 10.1.1.100
- └ QoS Class: [R99 QoS Class]
- └ PCO: [DNS Class]

UE **PDP** Context

UE GGSN → SGSN

UE PDP Context

UE Context

IE 名称	类型	说明
名称	名称	名称名称
名称名称	名称	名称名称名称
名称	名称	GGSN 名称
TEID 名称 I	名称	GGSN 名称名称名称
TEID 名称	名称	GGSN 名称名称名称
NSAPI	名称	名称 NSAPI
名称 ID	名称	名称 ID
名称名称	名称	名称 UE IP 名称
GSN 名称名称	IP	GGSN 名称名称
GSN 名称名称	IP	GGSN 名称名称
QoS 名称	名称	名称 QoS 名称
名称名称	名称	DNS 名称名称

名称名称

PDP Context

- Length: 128
- TEID I: 0xAABBCCDD
- TEID S: 0xDDCCBBAA
- NSAPI: 5
- IP ID: 0x11223344
- IPv4: 100.64.1.42
- GSN MNC: 10.0.0.20
- GSN MCC: 10.0.0.20
- QoS Class: [Default QoS]
- PCO
 - DNS: 8.8.8.8
 - DNS: 8.8.4.4

PDP Context

SGSN → GGSN

PDP Context SGSN QoS Class

Context

- SGSN Context
- QoS Class
- SGSN Context

Context

IE	
NSAPI	PDP Context
TEID I	SGSN TEID
GSN MNC	SGSN
QoS Class	QoS Class

IE **PDP** 詳細

IE GGSN → SGSN

IE PDP 詳細

IE

IE	
TEID	GGSN TEID
GSN	GGSN
QoS	QoS

IE **PDP** 詳細

IE SGSN → GGSN

IE PDP 詳細

IE

IE	
NSAPI	PDP

IE **PDP** 詳細

IE GGSN → SGSN

IE PDP 詳細

IE

IE 00	00
00	00000000

0000

0000 / 00

00000

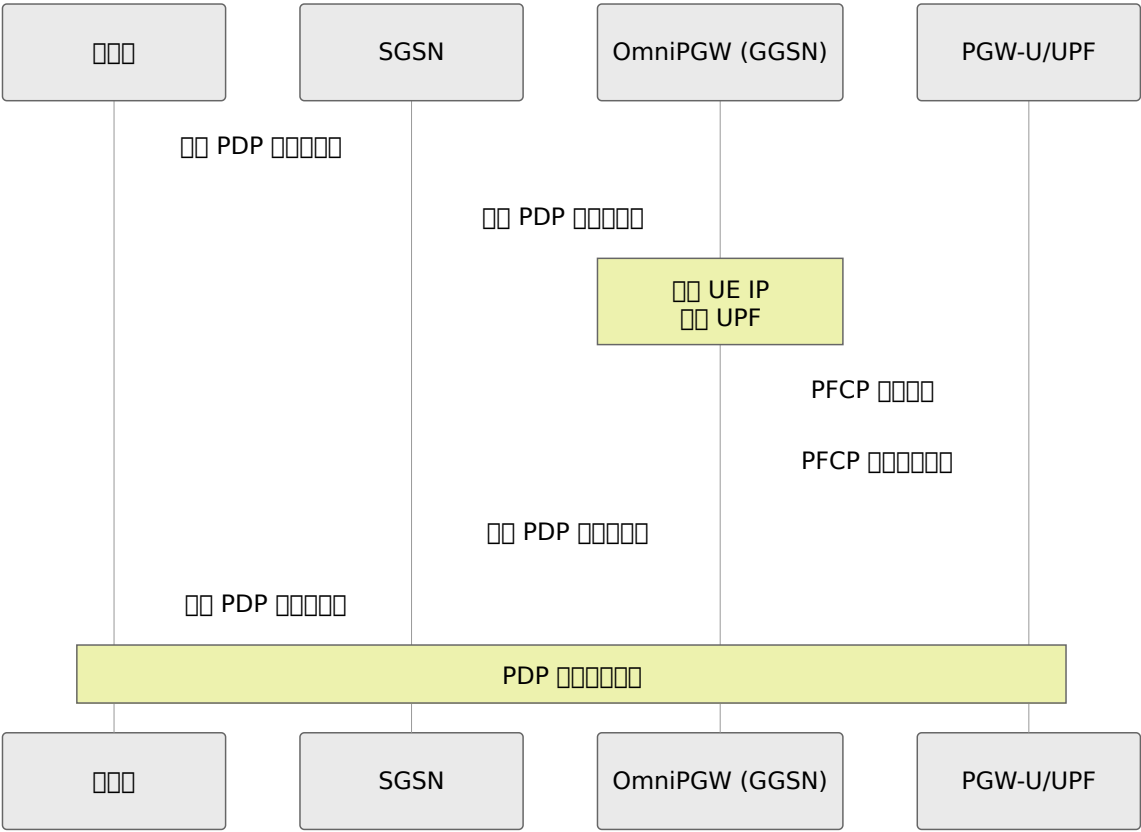
0000000000000000

0000000

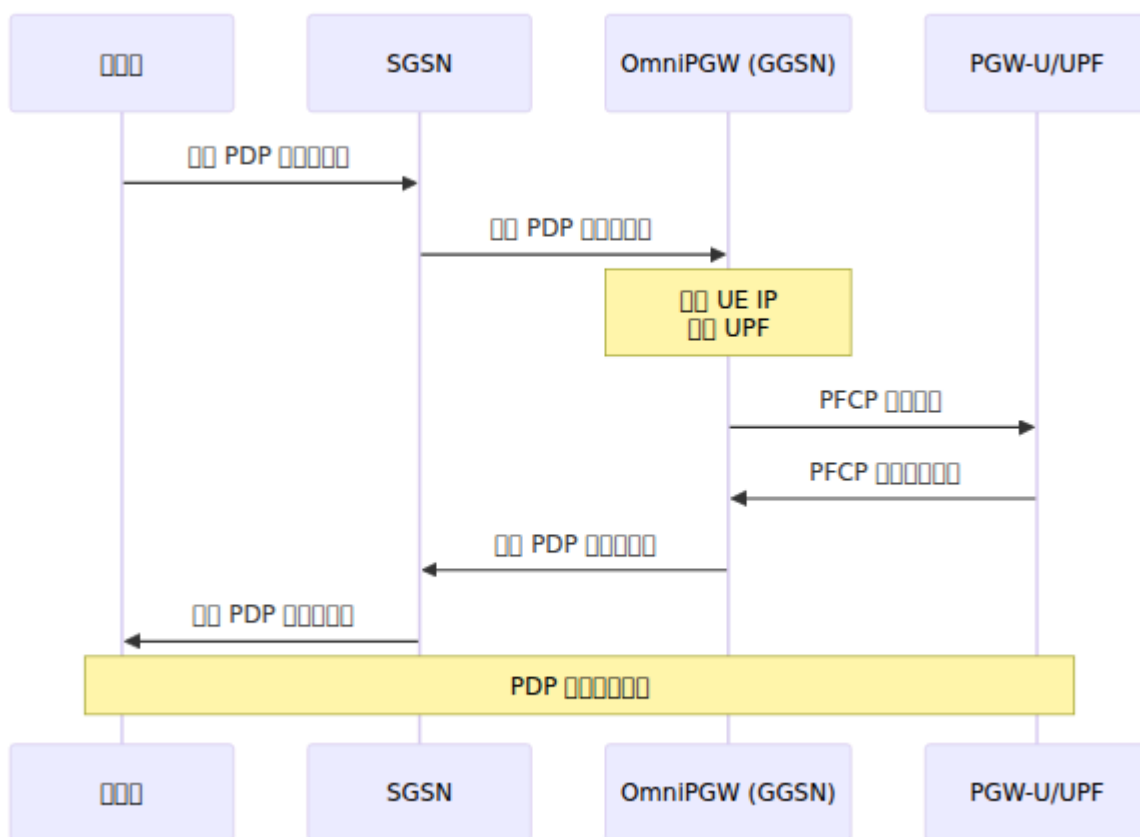
IE 00	00
00	0000000000000000

□□□

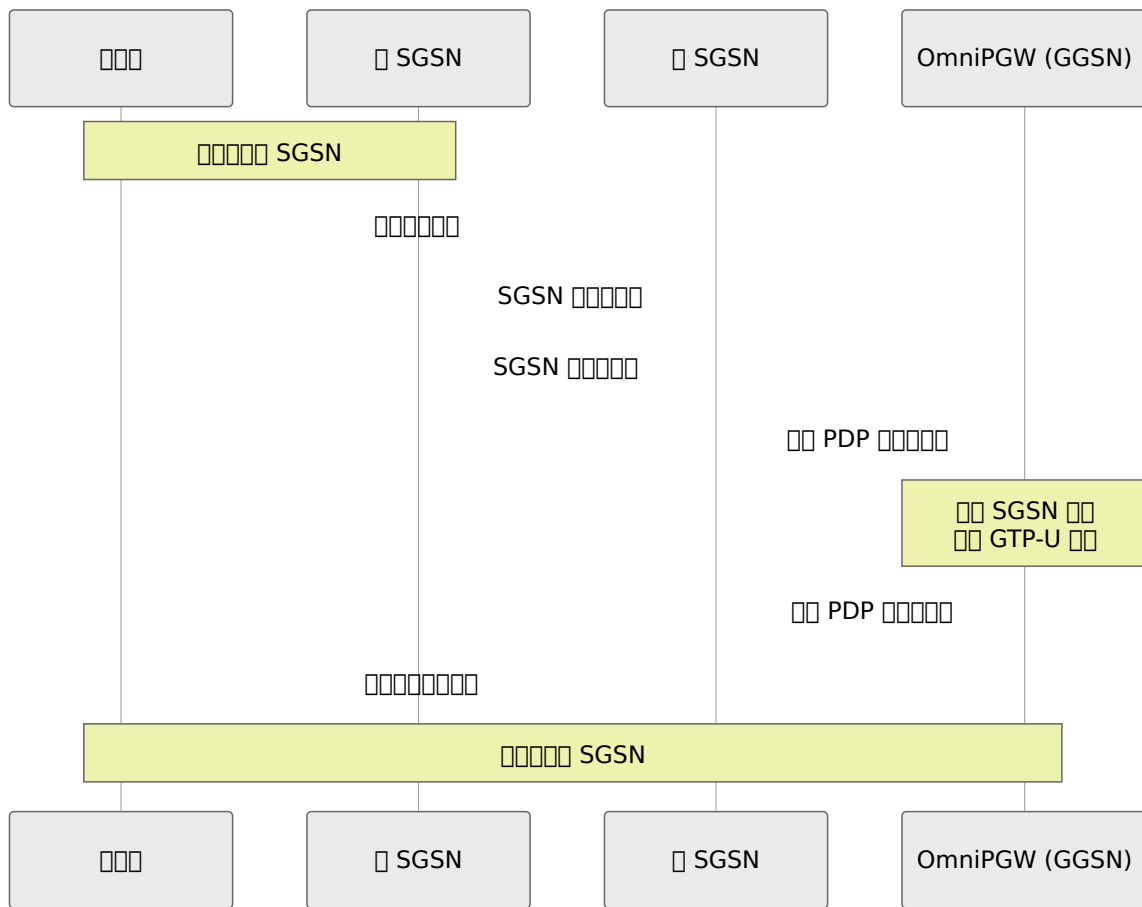
PDP □□□□□



PDP 激活



SGSN



OmniPGW □ GGSN □□□ PGW □□□□□□□□□□

UE IP

PGW AddressRegistry UE IP

UPF/PFCP ☐☐☐☐

□□□ PGW □□□ PFCP □□□□□

- PDP → PDR/FAR/QER →
- upf_selection → UPF

- 3GPP TS 23.060 PFCP 4.10

3GPP TS 23.060 Gy

3GPP TS 23.060 Diameter Gy 4.10

- 3GPP TS 23.060
- PGW 3GPP TS 23.060 Gy 4.10
- 3GPP TS 23.060 Diameter Gy 4.10

CDR 4.10

PGW 3GPP TS 23.060 CDR 4.10 3GPP TS 23.060 CDR 4.10

3GPP TS 23.060

3GPP

3GPP	3GPP	3GPP
128	3GPP TS 23.060	3GPP TS 23.060

0000

00	00	0000
192	000000	000 PDP 000
193	0000000	GTP 000000
194	IMSI 00	000000
195	MS 0 GPRS 00	MS 000
196	MS 000 GPRS	MS 0000
197	MS 00	MS 0000
198	0000000	GTP 000000
199	000000	0000
200	0000000	000000
201	00 IE 00	IE 0000
202	00 IE 00	000000 IE
203	00 IE 00	00 IE 00
204	0000	0000
205	0000	000000
206	P-TMSI 000000	0000
207	GPRS 0000	0000
208	0000000	0000000

項目	単位	説明
209	メッセージ/秒	メッセージ/秒

##

Gn メトリクス

```
# メトリクス
gn_inbound_messages_total{message_type="create_pdp_context_request"}
gn_inbound_messages_total{message_type="update_pdp_context_request"}
gn_inbound_messages_total{message_type="delete_pdp_context_request"}

gn_outbound_messages_total{message_type="create_pdp_context_response"}
gn_outbound_messages_total{message_type="update_pdp_context_response"}
gn_outbound_messages_total{message_type="delete_pdp_context_response"}

# 0000
gn_outbound_transaction_duration_bucket
```

メトリクス

PDP メトリクス

```
rate(gn_inbound_messages_total{message_type="create_pdp_context_request"}
[5m])
```

計算

```
sum(rate(gn_outbound_messages_total{message_type="create_pdp_context_response"}
[5m]))
/
sum(rate(gn_inbound_messages_total{message_type="create_pdp_context_request"}
[5m]))
```

📡 **PDP** 📡📡

```
gn_inbound_messages_total{message_type="create_pdp_context_request"}  
-  
gn_inbound_messages_total{message_type="delete_pdp_context_request"}
```

📡📡📡📡

📡📡📡📡 **PDP** 📡📡📡📡📡

📡📡

- SGSN 📡📡 PDP 📡📡📡
- 📡📡📡📡
- SGSN 📡📡

📡📡📡📡

1. 📡📡📡📡
2. OmniPGW 📡📡📡 IP 📡📡
3. 📡📡📡📡 UDP 2123
4. UPF/PFCP 📡📡📡

📡📡

```
# [] OmniPGW []
netstat -ulnp | grep 2123

# []
tcpdump -i any -n port 2123

# []
grep "local_ipv4_address" config/runtime.exs

# [] PFCP []
curl http://pgw:9090/metrics | grep pfc
```

[] **PDP** []

[]

- [] PDP []
- PDP []

[]???

```
[] 199[]
→ IP []
→ [] curl http://pgw:9090/metrics | grep address_registry_count
→ [] ue.subnet_map [] IP []

>[] 202[] IE []
→ SGSN [] IE
→ [] SGSN []
→ [] tcpdump []

>[] 204[]
→ UPF/PFCP []
→ [] PFCP []
→ [] OmniPGW []
```

[] **PDP** []

[]

- PDP Context
- Context

Context

- PDP Context TEID
- NSAPI
- Context

Context

```
# Context
curl http://pgw:9090/metrics | grep session_count

# Context
journalctl -u omnipgw | grep "PDP Context"
```

Context

Context

1. S5/S8 Context

- Gn S5/S8 UDP 2123
- OmniPGW Context GTP
- IP Context 2G/3G 4G

2. SGSN Context

- Context SGSN Context GGSN IP
- Gn Context VLAN
- Context Gp

3. IP Context

- IP Context PGW GGSN

- 2G/3G APN 設定

概要

1. 概要

- session_count
- 2G/3G
- 2G/3G

2. UDP

- S5/S8
- 4-8 MB

設定

GSN OmniPGW

1. APN

- APN GSN
- PCO DNS

2. IP

- IP
- APN

3. QoS

- R99 QoS
 - 2G/3G QCI
-

□□□□

□□□□

- □□□□ - □□□□□□
- □□□□ - □□□□□□□
- **UE IP** □□ - IP □□□□
- **PCO** □□ - □□□□□□

□□□□

- **S5/S8** □□ - 4G/LTE □□□□
- **PFCP** □□ - □□□□□□
- **Diameter Gy** □□ - □□□□

□□

- □□□□ - □□□□□□
- □□□□ - □□□□
- □□ **CDR** □□ - CDR □□

□□□□

- **3GPP TS 29.060** - GPRS □□□□GTP□□□□ Gn □ Gp □□
- **3GPP TS 23.060** - □□□□□□□□GPRS□□□□□□
- **3GPP TS 24.008** - □□□□□□□ 3 □□□□QoS □□□□□□

□□□□□□

OmniPGW Gn/Gp □□ - □ *Omni*touch □□□□□□

OmniPGW 部署架构图

Prometheus 部署图

OmniTouch 部署图

部署

1. 部署
2. 部署
3. 部署
4. Prometheus 部署
5. Grafana 部署
6. 部署
7. 部署
8. 部署

部署

OmniPGW 部署架构图

1. 部署架构图

- 部署
- PFCP 部署
- Diameter 部署
- 部署

2. Prometheus 部署架构图

- 部署
- 部署

- 設定
- 設定

設定 Prometheus 監視システム

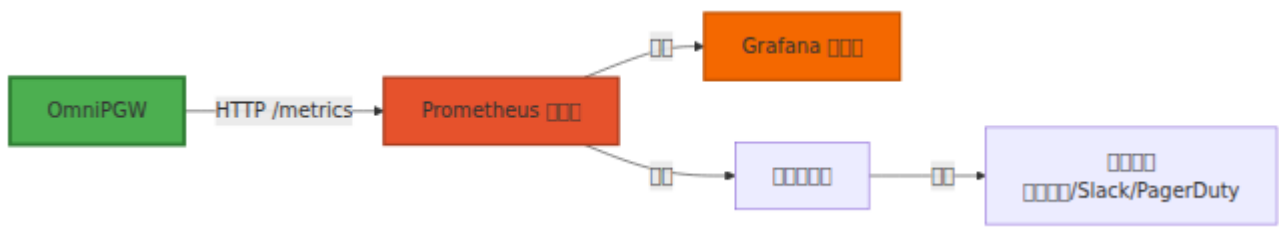
- 設定 - 設定
- PFCP 設定 - 設定
- Diameter Gx - 設定

Prometheus 設定

OmniPGW 設定 Prometheus 監視システム

- 設定 - 設定
- 設定 - 設定
- 設定 - 設定
- 設定 - 設定
- 設定 - 設定

設定



設定

設定

config/runtime.exs 設定


```
config :pgw_c,
  metrics: %{
    enabled: true,
    ip_address: "0.0.0.0", # 0.0.0.0
    port: 9090, # HTTP 9090
    registry_poll_period_ms: 5_000 # 5000
  }
```

0000

HTTP 000

```
http://<omnipgw_ip>:<port>/metrics
```

000

```
curl http://10.0.0.20:9090/metrics
```

0000

000 **Prometheus** 0000 000

```
# HELP teid_registry_count 000000 TEID 00
# TYPE teid_registry_count gauge
teid_registry_count 150

# HELP address_registry_count 0000000000
# TYPE address_registry_count gauge
address_registry_count 150

# HELP s5s8_inbound_messages_total  S5/S8 0000000000
# TYPE s5s8_inbound_messages_total counter
s5s8_inbound_messages_total{message_type="create_session_request"}
1523
s5s8_inbound_messages_total{message_type="delete_session_request"}
1487
```

Overview

OmniPGW Overview

Summary

Key Metrics

Metric Name	Type	Description
teid_registry_count	Gauge	Number of S5/S8 TEIDs registered
seid_registry_count	Gauge	Number of PFCP SEIDs registered
session_id_registry_count	Gauge	Number of Gx Diameter session IDs registered
session_registry_count	Gauge	Number of IMSI/EBI sessions registered
address_registry_count	Gauge	Number of UE IP addresses registered
charging_id_registry_count	Gauge	Number of charging IDs registered (CDR related)
sxb_sequence_number_registry_count	Gauge	Number of PFCP sequence numbers registered
s5s8_sequence_number_registry_count	Gauge	Number of S5/S8 sequence numbers registered
sxb_peer_registry_count	Gauge	Number of PFCP peers registered

Details

```
# Counter
teid_registry_count

# Histogram
rate(teid_registry_count[5m])

# Histogram
max_over_time(teid_registry_count[1h])
```

Queries

S5/S8 (GTP-C) Queries

Counter	Unit	Field	Description
s5s8_inbound_messages_total	Counter	message_type	S5/S8 inbound messages
s5s8_outbound_messages_total	Counter	message_type	S5/S8 outbound messages
s5s8_inbound_errors_total	Counter	message_type	S5/S8 inbound errors

Fields

- create_session_request
- create_session_response
- delete_session_request
- delete_session_response
- create_bearer_request
- delete_bearer_request

Sxb (PFCP) Queries

计数器	类型	消息	描述
sxb_inbound_messages_total	Counter	message_type	所有 PFCP 消息
sxb_outbound_messages_total	Counter	message_type	所有 PFCP 消息
sxb_inbound_errors_total	Counter	message_type	PFCP 消息错误
sxb_outbound_errors_total	Counter	message_type	PFCP 消息错误

消息类型

- association_setup_request
- association_setup_response
- heartbeat_request
- heartbeat_response
- session_establishment_request
- session_establishment_response
- session_modification_request
- session_deletion_request

Gx (Diameter) 消息

項目名	単位	属性	説明
gx_inbound_messages_total	Counter	message_type	受信メッセージの総数 Diameter メッセージ
gx_outbound_messages_total	Counter	message_type	送信メッセージの総数 Diameter メッセージ
gx_inbound_errors_total	Counter	message_type	受信エラーの総数 Diameter メッセージ
gx_outbound_errors_total	Counter	message_type	送信エラーの総数 Diameter メッセージ
gx_outbound_responses_total	Counter	message_type, result_code_class, diameter_host	送信レスポンスの総数 Diameter メッセージ 結果コード クラス ホスト

メッセージタイプ

- gx_CCA (Credit-Control-Answer)
- gx_CCR (Credit-Control-Request)
- gx_RAA (Re-Auth-Answer)
- gx_RAR (Re-Auth-Request)

結果コードクラス gx_outbound_responses_total

- 2xxx - 2001 DIAMETER_SUCCESS
- 3xxx - 3001 DIAMETER_COMMAND_UNSUPPORTED
- 4xxx - 4001 DIAMETER_AUTHENTICATION_REJECTED
- 5xxx - 5012 DIAMETER_UNABLE_TO_COMPLY

ホスト

```
# Gx
sum(rate(gx_outbound_responses_total{result_code_class="2xxx"}[5m]))
sum(rate(gx_outbound_responses_total[5m])) * 100

# PCRF
rate(gx_outbound_responses_total{result_code_class!="2xxx"}[5m]) by (

# Re-Auth-Answer
gx_outbound_responses_total{message_type="gx_RAA",result_code_class='

# PCRF
rate(gx_outbound_responses_total{result_code_class=~"4xxx|5xxx",diame
[5m]) > 0.1
```

rescues_total	Counter	module, function	

項目名	種類	単位	説明
s5s8_inbound_handling_duration	Histogram	request_message_type	5番台5番台宛先への着信処理時間
sxb_inbound_handling_duration	Histogram	request_message_type	5番台5番台宛先への着信処理時間
gx_inbound_handling_duration	Histogram	request_message_type	5番台5番台宛先への着信処理時間

項目名

項目名	種類	単位	説明
s5s8_outbound_transaction_duration	Histogram	request_message_type	5番台5番台宛先への着信処理時間
sxb_outbound_transaction_duration	Histogram	request_message_type	5番台5番台宛先への着信処理時間
gx_outbound_transaction_duration	Histogram	request_message_type	5番台5番台宛先への着信処理時間

項目名

- 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 5.0
- 100µs, 500µs, 1ms, 5ms, 10ms, 50ms, 100ms, 500ms, 1s, 5s

例

```
# 95th S5/S8
histogram_quantile(0.95,
  rate(s5s8_inbound_handling_duration_bucket[5m])
)

# PFCP
rate(sxb_inbound_handling_duration_sum[5m]) /
rate(sxb_inbound_handling_duration_count[5m])
```

UPF

UPF

名前	タイプ	ラベル	説明
upf_peers_total	Gauge	-	UPF のピアの総数
upf_peers_healthy	Gauge	-	UPF の健全なピアの数 + 欠落したピアの数
upf_peers_unhealthy	Gauge	-	UPF の不健全なピアの数
upf_peers_associated	Gauge	-	PFCP と UPF が関連しているピアの数
upf_peers_unassociated	Gauge	-	PFCP と UPF が関連していないピアの数
upf_peer_healthy	Gauge	peer_ip	UPF の健全なピア 1 = 1, 0 = 0
upf_peer_missed_heartbeats	Gauge	peer_ip	UPF の欠落したハートビートの数

000

```
# 00 UPF 000
upf_peers_healthy / upf_peers_total

# 00000 UPF 0000
upf_peers_unhealthy > 0

# 0000 UPF 00
upf_peer_healthy{peer_ip="10.98.0.20"}

# 00000000 UPF
upf_peer_missed_heartbeats > 2
```

00000

```

# [] UPF []
- alert: UPF_Peer_Down
  expr: upf_peer_healthy == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "UPF {{ $labels.peer_ip }} []"
    description: "UPF [] PFCP []"

# [] UPF []
- alert: UPF_Pool_Degraded
  expr: (upf_peers_healthy / upf_peers_total) < 0.5
  for: 2m
  labels:
    severity: critical
  annotations:
    summary: "UPF []"
    description: "[] {{ $value | humanizePercentage }} % UPF []"

# []
- alert: UPF_Heartbeat_Issues
  expr: upf_peer_missed_heartbeats > 2
  for: 30s
  labels:
    severity: warning
  annotations:
    summary: "UPF {{ $labels.peer_ip }} []"
    description: "{{ $value }} []"

```

P-CSCF []

P-CSCF []

項目名	タイプ	単位	説明
pcscf_fqdns_total	Gauge	-	登録済みの P-CSCF FQDN の数
pcscf_fqdns_resolved	Gauge	-	DNS 解決済みの P-CSCF FQDN の数
pcscf_fqdns_failed	Gauge	-	DNS 解決失敗の P-CSCF FQDN の数
pcscf_servers_total	Gauge	-	登録済みの P-CSCF サーバの数
pcscf_servers_healthy	Gauge	fqdn	健康な FQDN の P-CSCF サーバの数
pcscf_servers_unhealthy	Gauge	fqdn	不健康な FQDN の P-CSCF サーバの数

この P-CSCF 情報は IMS からも提供

監視項目

監視項目

項目名	タイプ	単位
license_status	Gauge	0 = 許可 / 1 = 拒否

設定

```
# 許可
license_status == 1

# 拒否
license_status == 0
```

監視項目

```
- alert: PGW_C_License_Invalid
  expr: license_status == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "PGW-C 许可证过期"
    description: "许可证过期 - 许可证过期"
```

许可证过期

许可证过期是指PGW-C的许可证过期，GTP-C 许可证 "许可证" (73)许可证过期
许可证

Wireshark 许可证过期许可证 "许可证" 许可证

许可证

- 许可证过期许可证 omnipgwc
- 许可证 URL 许可证 config/runtime.exs 许可证 :license_client 许可证
- 许可证 license_status == 0 许可证 GTP-C 许可证 73许可证
- 许可证许可证许可证许可证许可证
- Diameter许可证GTP-C 许可证 PFCP 许可证许可证
- 许可证许可证 - 许可证许可证

監視

Erlang VM 監視

メトリック名	タイプ	説明
vm_memory_total	Gauge	VM の総メモリ使用量
vm_memory_processes	Gauge	プロセスのメモリ使用量
vm_memory_system	Gauge	システムのメモリ使用量
vm_system_process_count	Gauge	Erlang のプロセス数
vm_system_port_count	Gauge	ポートの数

Prometheus

監視

OmniPGW 監視 Prometheus `prometheus.yml`

```
# prometheus.yml
global:
  scrape_interval: 15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: 'omnipgw'
    static_configs:
      - targets: ['10.0.0.20:9090']
        labels:
          instance: 'omnipgw-01'
          environment: 'production'
          site: 'datacenter-1'
```

OmniPGW

```
scrape_configs:
  - job_name: 'omnipgw'
    static_configs:
      - targets:
          - '10.0.0.20:9090'
          - '10.0.0.21:9090'
          - '10.0.0.22:9090'
    labels:
      environment: 'production'
```


Kubernetes

```
scrape_configs:
  - job_name: 'omnipgw'
    kubernetes_sd_configs:
      - role: pod
    relabel_configs:
      - source_labels: [__meta_kubernetes_pod_label_app]
        action: keep
        regex: omnipgw
      - source_labels: [__meta_kubernetes_pod_ip]
        target_label: __address__
        replacement: '${1}:9090'
```



```
# Prometheus
curl http://prometheus:9090/api/v1/targets

# 
curl 'http://prometheus:9090/api/v1/query?
query=teid_registry_count'
```

Grafana 安装

安装步骤

1. 安装 Prometheus 服务

安装 → 配置 → 启动 Prometheus
URL: <http://prometheus:9090>

2. 配置 Grafana

配置 Grafana 数据源 JSON 模板

配置项

配置 1 数据源

```
# 名称  
teid_registry_count  
  
# 数据类型Gauge  
# 单位  
# 范围 < 5000  
# 范围 5000-8000  
# 范围 > 8000
```

配置 2 数据源

```
# 名称  
rate(s5s8_inbound_messages_total{message_type="create_session_request"  
[5m])  
  
# 数据类型  
# 单位/秒
```

例 3 IP 割合

```
# 10.0.0.0 /24 の IP 数
(address_registry_count / 254) * 100

# 10.0.0.0 Gauge
# 10.0.0.0 0-100%
# 10.0.0.0
# 10.0.0.0 < 70%
# 10.0.0.0 70-85%
# 10.0.0.0 > 85%
```

例 4 95th 百分位

```
# 95th 百分位
histogram_quantile(0.95,

rate(s5s8_inbound_handling_duration_bucket{request_message_type="create",
[5m])
)

# 95th 百分位
# 95th 百分位
```

例 5 500 エラー

```
# 500 エラー
rate(s5s8_inbound_errors_total[5m])

# 500 エラー
# 500 エラー/秒
# 500 エラー > 0.1
```

例 6 Gx 割合


```
# 計算Gx の割合
sum(rate(gx_outbound_responses_total{result_code_class="2xxx"}
[5m])) /
sum(rate(gx_outbound_responses_total[5m])) * 100

# Gauge
# 0-100
#
# > 95%
# 90-95%
# < 90%
```

図 1 - 計算結果

```
# 計算結果
sum(rate(gx_outbound_responses_total[5m])) by (result_code_class)

# 計算結果
# {{ result_code_class }}
```

図 2 - **PCRf** の計算

```
# PCRf の計算
sum(rate(gx_outbound_responses_total[5m])) by (diameter_host,
result_code_class)

# 計算結果
# {{ diameter_host }} - {{ result_code_class }}
```

図 3 - **UPF** の計算

```
# 仪表盘显示
(upf_peers_healthy / upf_peers_total) * 100

# 仪表盘Gauge
# 仪表盘范围0-100
# 单位
# 100%
# 50-99%
# < 50%
```

UPF - 仪表盘

```
# 仪表盘 UPF 仪表盘
upf_peer_healthy

# 仪表盘
# 单位
# 1 = "UP"仪表盘
# 0 = "DOWN"仪表盘
```

ダッシュボード

```
{
  "dashboard": {
    "title": "OmniPGW - ダッシュボード",
    "panels": [
      {
        "title": "登録数",
        "targets": [
          {
            "expr": "teid_registry_count",
            "legendFormat": "登録数"
          }
        ],
        "type": "graph"
      },
      {
        "title": "セッション数",
        "targets": [
          {
            "expr":
"rate(s5s8_inbound_messages_total{message_type=\"create_session_request\"}[5m])",
            "legendFormat": "セッション数"
          }
        ],
        "type": "graph"
      },
      {
        "title": "IP 利用率",
        "targets": [
          {
            "expr": "(address_registry_count / 254) * 100",
            "legendFormat": "利用率 %"
          }
        ],
        "type": "gauge"
      },
      {
        "title": "セッション数p95",
        "targets": [
          {
            "expr": "histogram_quantile(0.95,
```

```
rate(s5s8_inbound_handling_duration_bucket[5m]))",
    "legendFormat": "S5/S8 p95"
  },
  {
    "expr": "histogram_quantile(0.95,
rate(sxb_inbound_handling_duration_bucket[5m]))",
    "legendFormat": "PFCP p95"
  }
],
"type": "graph"
}
]
```

□□

□□□□

□□ `omnipgw_alerts.yml` □

```

groups:
- name: omnipgw
  interval: 30s
  rules:
    # 会话数
    - alert: OmniPGW_HighSessionCount
      expr: teid_registry_count > 8000
      for: 5m
      labels:
        severity: warning
      annotations:
        summary: "OmniPGW 会话数高"
        description: "{{ $value }}"
    - alert: OmniPGW_SessionCountCritical
      expr: teid_registry_count > 9500
      for: 2m
      labels:
        severity: critical
      annotations:
        summary: "OmniPGW 会话数临界"
        description: "{{ $value }}"

    # IP 地址
    - alert: OmniPGW_IPPoolUtilizationHigh
      expr: (address_registry_count / 254) * 100 > 80
      for: 10m
      labels:
        severity: warning
      annotations:
        summary: "OmniPGW IP 地址池利用率高"
        description: "IP 地址池 {{ $value }}% 利用率"
    - alert: OmniPGW_IPPoolExhausted
      expr: address_registry_count >= 254
      for: 1m
      labels:
        severity: critical
      annotations:
        summary: "OmniPGW IP 地址池耗尽"
        description: "IP 地址池耗尽"

    # 异常

```

```

- alert: OmniPGW_HighErrorRate
  expr: rate(s5s8_inbound_errors_total[5m]) > 0.1
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "OmniPGW S5/S8 S5/S8"
    description: "{{ $value }}"

- alert: OmniPGW_GxErrorRate
  expr: rate(gx_inbound_errors_total[5m]) > 0.05
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "OmniPGW Gx S5/S8"
    description: "{{ $value }}" Diameter S5/S8"

# Gx S5/S8
- alert: OmniPGW_GxResponseFailureRate
  expr: |

sum(rate(gx_outbound_responses_total{result_code_class!="2xxx"}
[5m])) /
    sum(rate(gx_outbound_responses_total[5m])) > 0.1
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "OmniPGW Gx S5/S8"
    description: "{{ $value | humanizePercentage }}" Gx S5/S8
    "2xxx S5/S8"

- alert: OmniPGW_GxPCRFFailures
  expr:
rate(gx_outbound_responses_total{result_code_class=~"4xxx|5xxx"}
[5m]) by (diameter_host) > 0.05
  for: 3m
  labels:
    severity: warning
  annotations:
    summary: "PCRF {{ $labels.diameter_host }}"
    description: "{{ $value }}" PCRF {{
$labels.diameter_host }}"

```

```

# UPF 告警
- alert: OmniPGW_UPF_PeerDown
  expr: upf_peer_healthy == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "UPF 告警 {{ $labels.peer_ip }} 告警"
    description: "UPF 告警 PFCP 告警"

- alert: OmniPGW_UPF_PoolDegraded
  expr: (upf_peers_healthy / upf_peers_total) < 0.5
  for: 2m
  labels:
    severity: critical
  annotations:
    summary: "UPF 告警"
    description: "{{ $value | humanizePercentage }} % UPF 告警
告警< 50%"

- alert: OmniPGW_UPF_HeartbeatFailures
  expr: upf_peer_missed_heartbeats > 2
  for: 30s
  labels:
    severity: warning
  annotations:
    summary: "UPF {{ $labels.peer_ip }} 告警"
    description: "{{ $value }} 告警"

- alert: OmniPGW_UPF_AllDown
  expr: upf_peers_healthy == 0 and upf_peers_total > 0
  for: 30s
  labels:
    severity: critical
  annotations:
    summary: "告警 UPF 告警"
    description: "告警 UPF 告警"

# 告警
- alert: OmniPGW_HighLatency
  expr: |
    histogram_quantile(0.95,
      rate(s5s8_inbound_handling_duration_bucket[5m])

```

```

    ) > 100000
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "OmniPGW 高延迟"
    description: "p95 延迟 {{ $value }}μs (> 100ms)"

# 内存使用
- alert: OmniPGW_HighMemoryUsage
  expr: vm_memory_total > 2000000000
  for: 10m
  labels:
    severity: warning
  annotations:
    summary: "OmniPGW 高内存使用"
    description: "VM 内存 {{ $value | humanize }}B 使用"

- alert: OmniPGW_HighProcessCount
  expr: vm_system_process_count > 100000
  for: 10m
  labels:
    severity: warning
  annotations:
    summary: "OmniPGW 高进程数"
    description: "{{ $value }} Erlang 进程数"

```


alertmanager.yml

```
# alertmanager.yml
global:
  resolve_timeout: 5m

route:
  receiver: 'ops-team'
  group_by: ['alertname', 'instance']
  group_wait: 10s
  group_interval: 10s
  repeat_interval: 12h

  routes:
    - match:
        severity: critical
      receiver: 'pagerduty'

    - match:
        severity: warning
      receiver: 'slack'

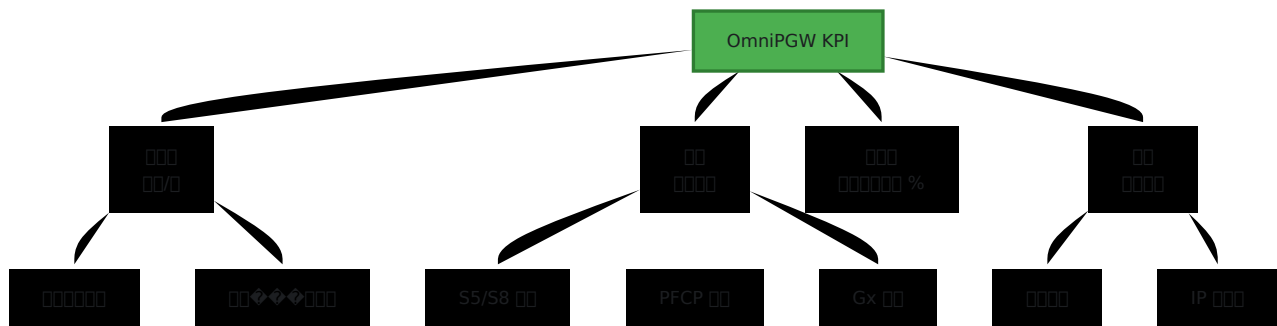
receivers:
  - name: 'ops-team'
    email_configs:
      - to: 'ops@example.com'

  - name: 'slack'
    slack_configs:
      - api_url:
          'https://hooks.slack.com/services/YOUR/SLACK/WEBHOOK'
        channel: '#omnipgw-alerts'
        title: 'OmniPGW {{ .GroupLabels.alertname }}'
        text: '{{ range .Alerts }}{{ .Annotations.description }}{{
end }}'

  - name: 'pagerduty'
    pagerduty_configs:
      - service_key: 'YOUR_PAGERDUTY_KEY'
```

□□□□

□□□□□□ (KPI)



□□□□□

□□□□□□

```
rate(s5s8_inbound_messages_total{message_type="create_session_request"
[5m] )
```

□□□□□□

```
rate(s5s8_inbound_messages_total{message_type="delete_session_request"
[5m] )
```

□□□□□

```
rate(s5s8_inbound_messages_total{message_type="create_session_request"
[5m] ) -
rate(s5s8_inbound_messages_total{message_type="delete_session_request"
[5m] )
```

□□□□

□□□□□□□□□□□□

```
# p50
histogram_quantile(0.50,
  rate(s5s8_inbound_handling_duration_bucket[5m])
)

# p95
histogram_quantile(0.95,
  rate(s5s8_inbound_handling_duration_bucket[5m])
)

# p99
histogram_quantile(0.99,
  rate(s5s8_inbound_handling_duration_bucket[5m])
)
```

スケーリング

```
histogram_quantile(0.95,
  rate(s5s8_inbound_handling_duration_bucket[5m])
) by (request_message_type)
```

スケーリング

スケーリング **24**時間

```
teid_registry_count -
teid_registry_count offset 24h
```

スケーリング

```
# スケーリング 10,000 回
10000 - teid_registry_count
```

スケーリング

```
# 00 1 0000000000000000  
(10000 - teid_registry_count) /  
(rate(teid_registry_count[1h]) * 86400)
```


□□□□

103

```
rate(s5s8_inbound_errors_total[5m]) by (message_type)
```

10

- 5G
- 4G PCRF 4G Gx 4G
- 4G IP 4G

103

```
histogram_quantile(0.95,  
  
rate(s5s8_inbound_handling_duration_bucket{request_message_type="crea  
[5m])  
)
```

```
rate(s5s8_inbound_handling_duration_bucket{request_message_type="create",
[5m])
)
```

103

- Gx PCRF
- PFCP PGW-U

- 配置参数

配置**PCRF** 参数

配置

```
# 配置 Gx 参数
sum(rate(gx_outbound_responses_total{result_code_class!="2xxx"}
[5m])) /
sum(rate(gx_outbound_responses_total[5m])) * 100

# 配置 PCRF 参数
sum(rate(gx_outbound_responses_total[5m])) by (diameter_host,
result_code_class)

# 配置参数
rate(gx_outbound_responses_total{result_code_class="5xxx"}[5m]) by
(diameter_host)
```

配置

- 配置 PCRF 参数
- 配置 PCRF 参数5xxx 参数
- 配置 Diameter 参数
- 配置 PCRF 参数
- 配置 5012DIAMETER_UNABLE_TO_COMPLY Re-Auth-Request 参数

配置参数

配置

```
# 配置参数
rate(vm_memory_total[1h])

# 配置参数
rate(vm_memory_processes[1h])

# 配置参数
rate(vm_system_process_count[1h])
```

□□□

- □□□□□□
- □□□□□□
- □□□□□□□□□□

□□□□

□□□□□□□□□□

```
max_over_time(teid_registry_count[24h])
```

□□□□□□□□

```
teid_registry_count /  
avg_over_time(teid_registry_count[7d])
```

□□□□□

```
abs(  
    teid_registry_count -  
    avg_over_time(teid_registry_count[1h])  
) > 100
```

□□□□

□□□□

1. □□□□□ 15-30 □□□□□□□□□□
2. □□□ 15 □□□□□□□□□□
3. □□□ □□□□□□□□□□□□□□□□□□□

Network

1. Network - NOC KPI
2. Network - Performance
3. Network - Security

Cloud

1. Cloud - Infrastructure
2. Cloud - Platform
3. Cloud - Application

Security

Network

- Network - Prometheus
- Network - Grafana

Cloud

- PFCP - PFCP
- Diameter Gx - Gx
- Diameter Gy - Gy
- S5/S8 - GTP-C

Security

- P-CSCF - P-CSCF
- Network - Security
- UE IP - IP

Summary

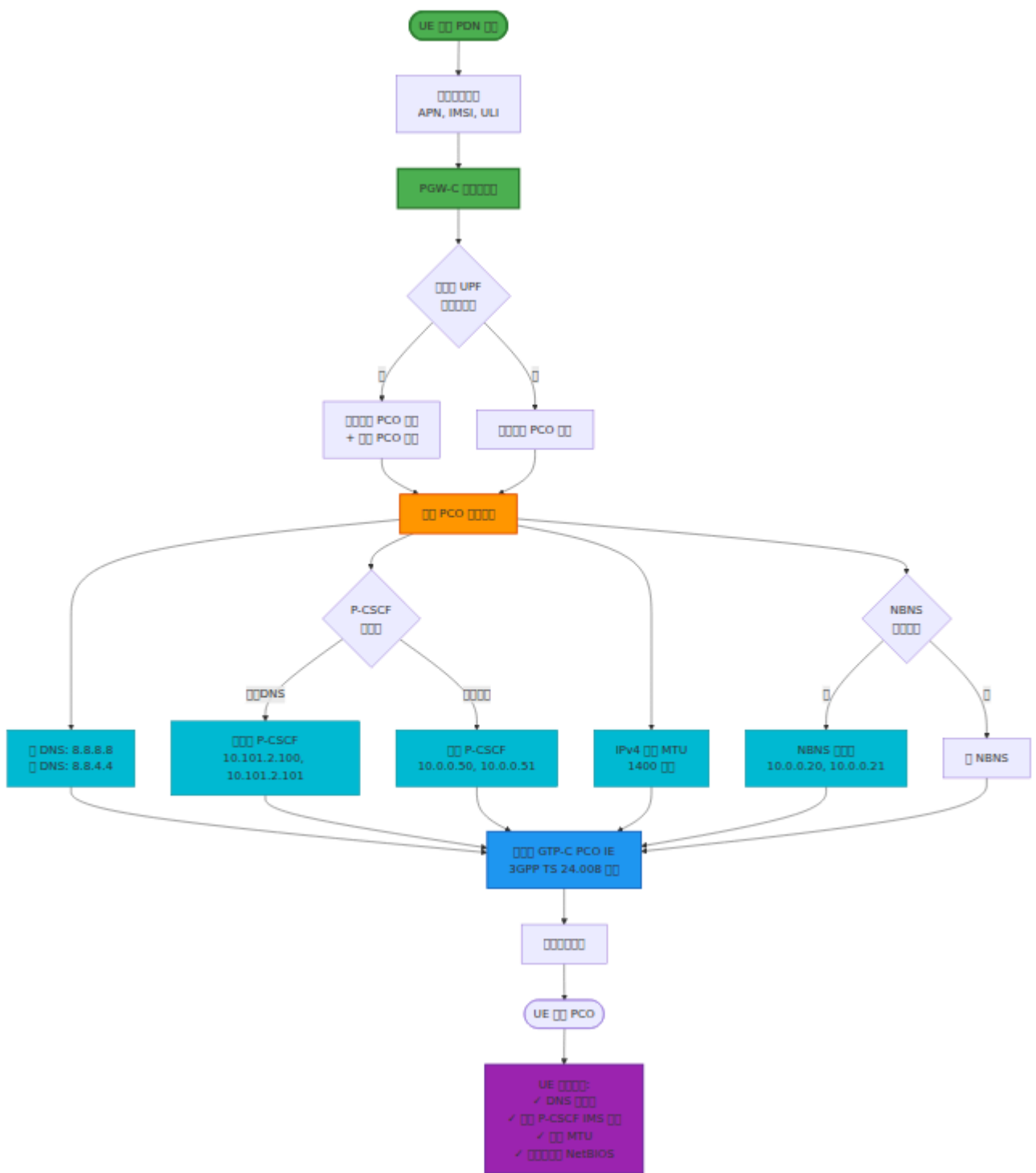
OmniPGW □□□□ - □ *Omnitouch* □□□□□□

PCO (PCO)

UE UE

OmniPGW OmniTouch

PCO () PDN UE () UE DNS
IMS



PCO

IE 名称	选项 ID	选项	备注
DNS 选项 IPv4 地址	0x000D	IP 地址 DNS	IP 地址
DNS 选项 IPv4 地址	0x000D	IP 地址 DNS	IP 地址
P-CSCF IPv4 地址	0x000C	IMS 域 P-CSCF	IP 地址 (IMS)
IPv4 选项 MTU	0x0010	MTU 大小	MTU 大小
NBNS 选项 IPv4 地址	0x0011	NetBIOS 名称	IP 地址

□□

□□□□

```
# config/runtime.exs
config :pgw_c,
  pco: %{
    # DNS □□□ (□□)
    primary_dns_server_address: "8.8.8.8",
    secondary_dns_server_address: "8.8.4.4",

    # NBNS □□□ (□□□□□□ Windows □□)
    primary_nbns_server_address: nil,
    secondary_nbns_server_address: nil,

    # IMS/VoLTE □ P-CSCF □□ (□□)
    p_cscf_ipv4_address_list: [],

    # P-CSCF □□□□ (□□)
    p_cscf_discovery_enabled: false,
    p_cscf_discovery_dns_server: nil,
    p_cscf_discovery_timeout_ms: 5000,

    # IPv4 MTU □□ (□□)
    ipv4_link_mtu_size: 1400
  }
```

PCO □□

DNS □□□□□

□ **DNS** □□ **DNS**□

```
pco: %{\n  primary_dns_server_address: "8.8.8.8",\n  secondary_dns_server_address: "8.8.4.4"\n}
```

DNS

	DNS	DNS
Google	8.8.8.8	8.8.4.4
Cloudflare	1.1.1.1	1.0.0.1
Quad9	9.9.9.9	149.112.112.112
OpenDNS	208.67.222.222	208.67.220.220

DNS

```
pco: %{\n  primary_dns_server_address: "10.0.0.10",\n  secondary_dns_server_address: "10.0.0.11"\n}
```

P-CSCF (IMS)

IMS/VoLTE

```
pco: %{\n  p_cscf_ipv4_address_list: [\n    "10.0.0.50", # P-CSCF\n    "10.0.0.51"  # P-CSCF\n  ]\n}
```

P-CSCF ()

- IMS 認證
- VoLTE/VoWiFi/RCS 認證
- UE 認證 SIP

P-CSCF 認證

透過 DNS 找 P-CSCF 認證

OmniPGW 透過 DNS 認證 P-CSCF 認證 3GPP TS 23.003 及 TS 24.229 認證
PGW-C 透過 DNS 找 P-CSCF 認證

```
pco: %{
  # 透過 P-CSCF 認證
  p_cscf_discovery_enabled: true,

  # P-CSCF 透過 DNS 認證 (認證)
  p_cscf_discovery_dns_server: {10, 179, 2, 177},

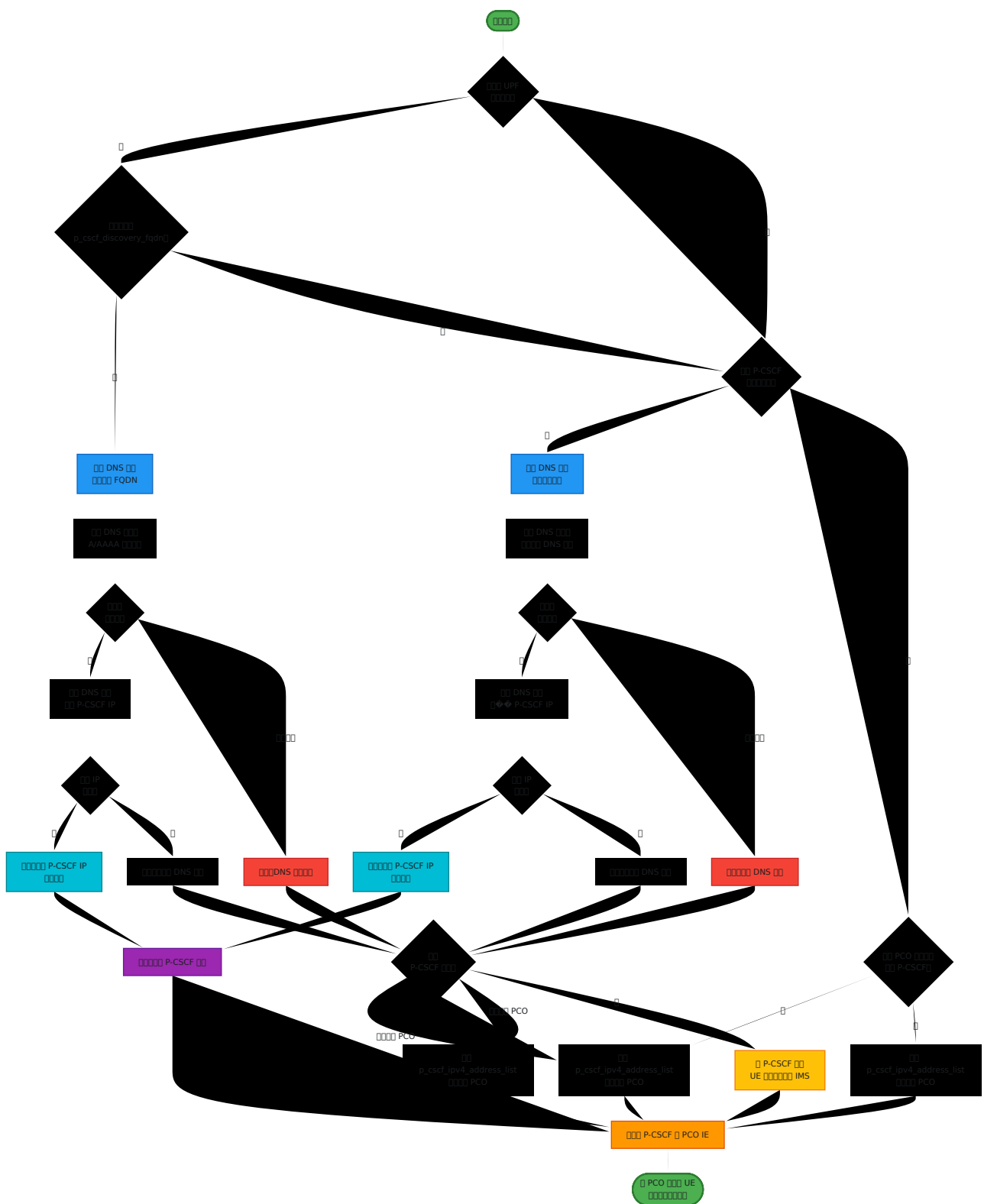
  # DNS 認證 (認證)
  p_cscf_discovery_timeout_ms: 5000,

  # 透過 P-CSCF 認證 (透過 DNS 認證)
  p_cscf_ipv4_address_list: ["10.0.0.50"]
}
```

認證

1. 透過 `p_cscf_discovery_enabled: true` PGW-C 透過 DNS 認證 P-CSCF 認證
2. DNS 認證 `p_cscf_discovery_dns_server`
3. 透過 DNS 認證 P-CSCF 認證 PCO 認證 UE
4. 透過 DNS 認證 `p_cscf_ipv4_address_list`
5. 認證 P-CSCF 認證

P-CSCF ☐☐☐☐



□ □ □ □ □ □

1. **FQDN** (FQDN) - `p_cscf_discovery_fqdn` UPF FQDN
2. **DNS** (DNS) - `p_cscf_discovery_enabled: true` PCO DNS

3. 在 **PCO** 配置 - `p_cscf_ipv4_address_list` 配置 PCO 配置
4. 在 **PCO** 配置 (IP) - `p_cscf_ipv4_address_list` 配置 PCO 配置

配置

在 P-CSCF 配置配置配置配置

- DNS 配置/配置
- 配置
- 配置配置
- 配置配置配置

配置配置配置配置 **P-CSCF** 配置

配置配置

配置	配置	配置	配置
<code>p_cscf_discovery_enabled</code>	配置	<code>false</code>	配置 DNS 配置 P-CSCF 配置
<code>p_cscf_discovery_dns_server</code>	配置 (IP)	<code>nil</code>	DNS 配置 IP 配置 4 配置 (配置 {10, 179, 2, 177})
<code>p_cscf_discovery_timeout_ms</code>	配置	<code>5000</code>	DNS 配置配置配置

配置

- 在 **IMS** 配置 - P-CSCF 配置 DNS 配置
- 配置配置 - DNS 配置配置 P-CSCF 配置
- 配置配置 - DNS 配置配置配置 P-CSCF 配置
- 配置配置 - 配置配置配置配置 P-CSCF 配置

配置配置 **DNS** 配置配置 **IMS**


```
pco: %{
  primary_dns_server_address: "10.0.0.10",
  secondary_dns_server_address: "10.0.0.11",

  # P-CSCF
  p_cscf_discovery_enabled: true,
  p_cscf_discovery_dns_server: {10, 179, 2, 177}, # IMS DNS
  p_cscf_discovery_timeout_ms: 3000,

  # P-CSCF (DNS)
  p_cscf_ipv4_address_list: [
    "10.0.0.50", #
    "10.0.0.51" #
  ],

  ipv4_link_mtu_size: 1400
}
```

P-CSCF

P-CSCF UPF APN DNS P-CSCF

```
# upf_selection
rules: [
  %{
    name: "IMS",
    priority: 20,
    match_field: :apn,
    match_regex: "^ims",
    upf_pool: [...],

    # P-CSCF
    p_cscf_discovery_fqdn: "pcscf.mnc380.mcc313.3gppnetwork.org"
  }
]
```

P-CSCF UPF

P-CSCF P-CSCF

NBNS 配置 (NetBIOS)

Windows 配置

```
pco: %{\n  primary_nbns_server_address: "10.0.0.20",\n  secondary_nbns_server_address: "10.0.0.21"\n}
```

配置

- Windows 配置
-
- NetBIOS 配置

MTU 配置

配置

```
pco: %{\n  ipv4_link_mtu_size: 1400 # \n}
```

MTU 配置

MTU	
1500	
1400	GTP
1420	
1280	IPv6 MTU
1360	VPN/

□□□ □□ LTE □□ **1400** □□□ GTP-U □□□

□□□□

Internet APN

```
pco: %{  
  primary_dns_server_address: "8.8.8.8",  
  secondary_dns_server_address: "8.8.4.4",  
  ipv4_link_mtu_size: 1400  
}
```

IMS APN

```
pco: %{  
  primary_dns_server_address: "10.0.0.10",  
  secondary_dns_server_address: "10.0.0.11",  
  p_cscf_ipv4_address_list: [  
    "10.0.0.50",  
    "10.0.0.51"  
  ],  
  ipv4_link_mtu_size: 1400  
}
```

□□ **P-CSCF** □□ □□□ IMS □□□□□□ P-CSCF □□□□

APN

```
pco: %{
  primary_dns_server_address: "10.100.0.10",
  secondary_dns_server_address: "10.100.0.11",
  primary_nbns_server_address: "10.100.0.20",
  secondary_nbns_server_address: "10.100.0.21",
  ipv4_link_mtu_size: 1400
}
```

PCO ☐ **GTP-C** ☐☐☐☐☐☐

OmniPGW 〇 〇〇〇〇〇〇 〇〇〇〇〇 PCO〇

```

| | | | |
| — IP: | | | | |
| — UE IP | | : 100.64.1.42
| — PC0 (| | | | |)
|   | — DNS | | IPv4 | | : 8.8.8.8
|   | — DNS | | IPv4 | | : 8.8.4.4
|   | — P-CSCF IPv4 | | : 10.0.0.50
|   | — P-CSCF IPv4 | | : 10.0.0.51
|   └─ IPv4 | | MTU: 1400

```

UE ☐ ☐

UE □□ PCO □□

1. DNS 設定
2. P-CSCF 設定 IMS 設定
3. MTU 設定

UE

UE DNS

UE

- UE IP
- DNS

PCO

1. PCO DNS
2. UE IP DNS
3. DNS

PCO

```
# DNS
ping 8.8.8.8

# UE DNS
nslookup google.com 8.8.8.8

# PCO
grep "primary_dns_server_address" config/runtime.exs
```

IMS

IMS

- VoLTE
- UE IMS

IMS

1. P-CSCF
2. P-CSCF IP
3. P-CSCF

配置

```
# P-CSCF 配置
pco: %{
  p_cscf_ipv4_address_list: ["10.0.0.50"] # 配置
}
```

配置MTU

配置

- 配置
- 配置
- 配置

配置

- MTU 配置
- MTU 配置

配置

```
# GTP 配置 1400
pco: %{
  ipv4_link_mtu_size: 1400
}

# 配置
pco: %{
  ipv4_link_mtu_size: 1360
}
```

□□□□

DNS □□

1. □□□□ **DNS** □□
 - □□□Google (8.8.8.8)□Cloudflare (1.1.1.1)
 - □□□□□□ DNS
2. □□□□□ **DNS**
 - □□□□
 - □□□□□
3. □□ **DNS** □□□
 - □□ DNSSEC □□□□
 - □□ DNS □□□□□□□□

IMS □□

1. □□□□ **P-CSCF**
 - □□ 2 □□□□□□
 - □□□□□□□□□□
2. □□□□□
 - P-CSCF □□□□ UE IP □□□
 - □□ SIP □□□

MTU □□

1. □□□□
 - GTP-U: 36 □□ (IPv4)
 - IPsec: □□ (50-100 □□)

2. **LTE MTU**

- **1400**
-

3.

- **MTU**
-

- **runtime.exs** - **UPF** **PCO**
- **UE IP** - IP **APN**
- **P-CSCF** - P-CSCF

- **PDN** - **PDN**
- **S5/S8** - GTP-C **PCO**
- **PFCP** -

IMS **VoLTE**

- **Diameter Gx** - IMS
- - **PCO**

OmniPGW PCO - *Omnitouch*

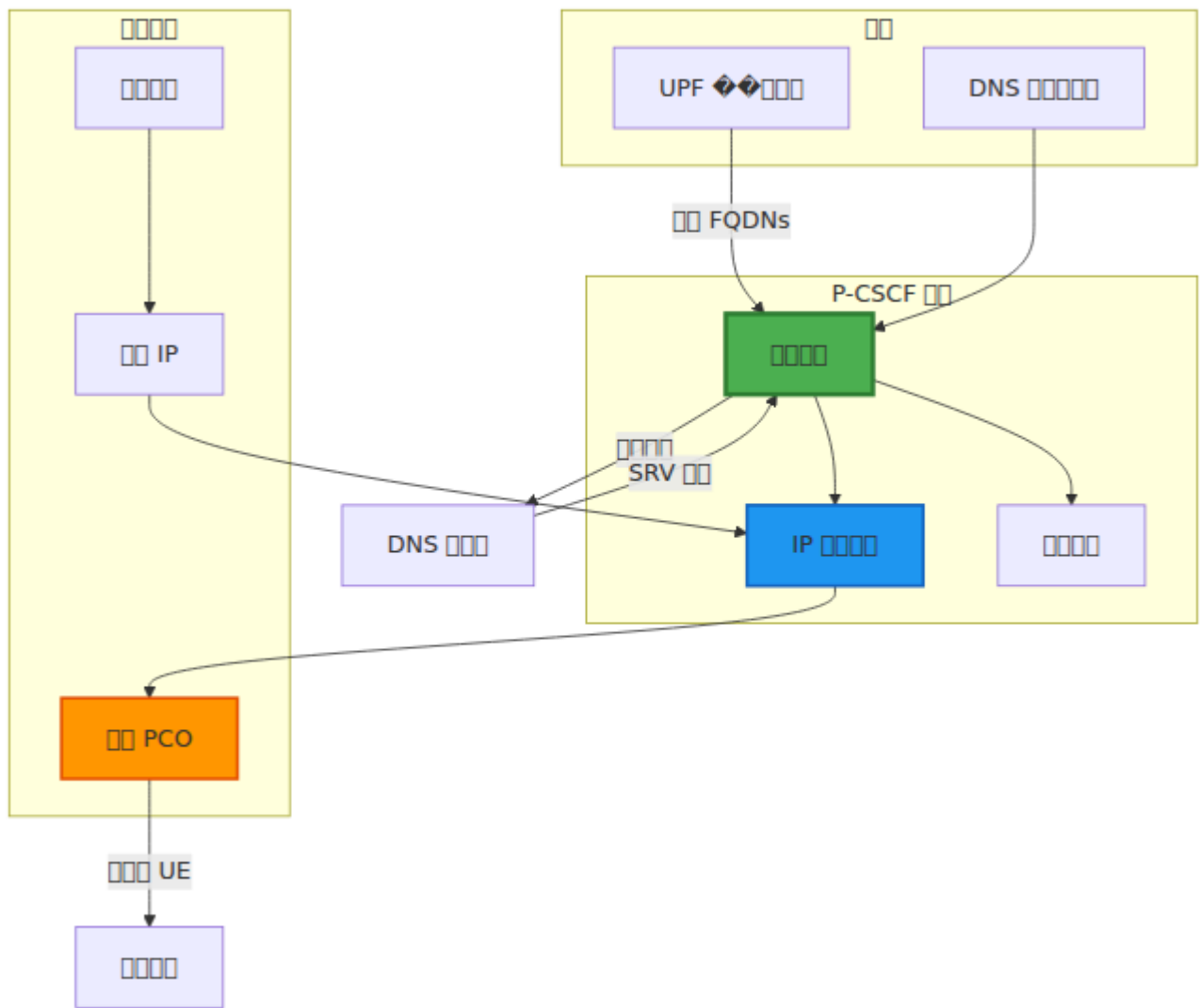
P-CSCF

P-CSCF

OmniPGW / Omnitouch

P-CSCF DNS SRV IMS P-CSCF SIP OPTIONS

- **P-CSCF** P-CSCF
- DNS 60
- **SIP OPTIONS** SIP OPTIONS ping P-CSCF
 - **TCP** TCP SIP OPTIONS
 - **UDP** TCP UDP
 - :up :down
- Web UI IP
-
- **Prometheus** Prometheus



□□

1. □□□□
2. □□
3. □□□□
4. Web UI □□
5. □□□□□□□□
6. □□□□
7. DNS □□
8. □□□□
9. □□□□

□□□□

□□□□

```
# config/runtime.exs

# □□ PCO □□□□ P-CSCF □□ DNS □□□□
config :pgw_c,
  pco: %{
    p_cscf_discovery_dns_server: "10.179.2.177",
    p_cscf_discovery_enabled: true,
    p_cscf_discovery_timeout_ms: 5000
  },

  upf_selection: %{
    rules: [
      # IMS □□ - □□ P-CSCF □□
      %{
        name: "IMS □□",
        priority: 20,
        match_field: :apn,
        match_regex: "^ims",
        upf_pool: [
          weight: 80}
          %{remote_ip_address: "10.100.2.21", remote_port: 8805,
        ],
        # P-CSCF □□ FQDN□□□□□ UPF □□□□□□□□□□□□
        p_cscf_discovery_fqdn:
        "pcscf.mnc380.mcc313.3gppnetwork.org",
        # □□□□□□□□ PCO □□□□□
        pco: %{
          p_cscf_ipv4_address_list: ["10.101.2.100",
          "10.101.2.101"]
        }
      }
    ]
  }
}
```

□□□ □□□□ □□□□□□ UPF □□□□□□□□□□ PCO □□ □□□□□ P-CSCF □□□□□

準備

1. 確認 OmniPGW
 2. 確認 **Web UI → P-CSCF** 欄 (https://localhost:8086/pcscf_monitor)
 3. 確認 P-CSCF の IP
-

確認

確認 P-CSCF の設定

確認 PCO の P-CSCF 設定 DNS 設定

```
pco: %{  
  # P-CSCF 設定 DNS 設定 UE の DNS 設定  
  p_cscf_discovery_dns_server: "10.179.2.177",  
  
  # P-CSCF DNS 設定  
  p_cscf_discovery_enabled: true,  
  
  # DNS SRV 設定  
  p_cscf_discovery_timeout_ms: 5000,  
  
  # P-CSCF 設定  
  p_cscf_ipv4_address_list: ["10.101.2.146"]  
}
```

確認 P-CSCF FQDNs

確認 UPF の P-CSCF 設定 FQDN

```

upf_selection: %{
  rules: [
    # IMS 00 - IMS 000 P-CSCF
    %{
      name: "IMS 00",
      match_field: :apn,
      match_regex: "^ims",
      upf_pool: [...],
      p_cscf_discovery_fqdn:
"pcscf.ims.mnc380.mcc313.3gppnetwork.org",
      pco: %{
        p_cscf_ipv4_address_list: ["10.101.2.100"] # 00
      }
    },

    # 00 - 000000 P-CSCF
    %{
      name: "00000",
      match_field: :apn,
      match_regex: "^enterprise",
      upf_pool: [...],
      p_cscf_discovery_fqdn: "pcscf.enterprise.example.com",
      pco: %{
        p_cscf_ipv4_address_list: ["192.168.1.50"] # 00
      }
    },

    # 000 - 0 P-CSCF 000000000000
    %{
      name: "00000",
      match_field: :apn,
      match_regex: "^internet",
      upf_pool: [...]
      # 0 p_cscf_discovery_fqdn - 0000 PCO 00
    }
  ]
}

```

目標

前提

1. 確認

- P-CSCF 側 GenServer 確認
- 確認する UPF 確認する P-CSCF FQDNs

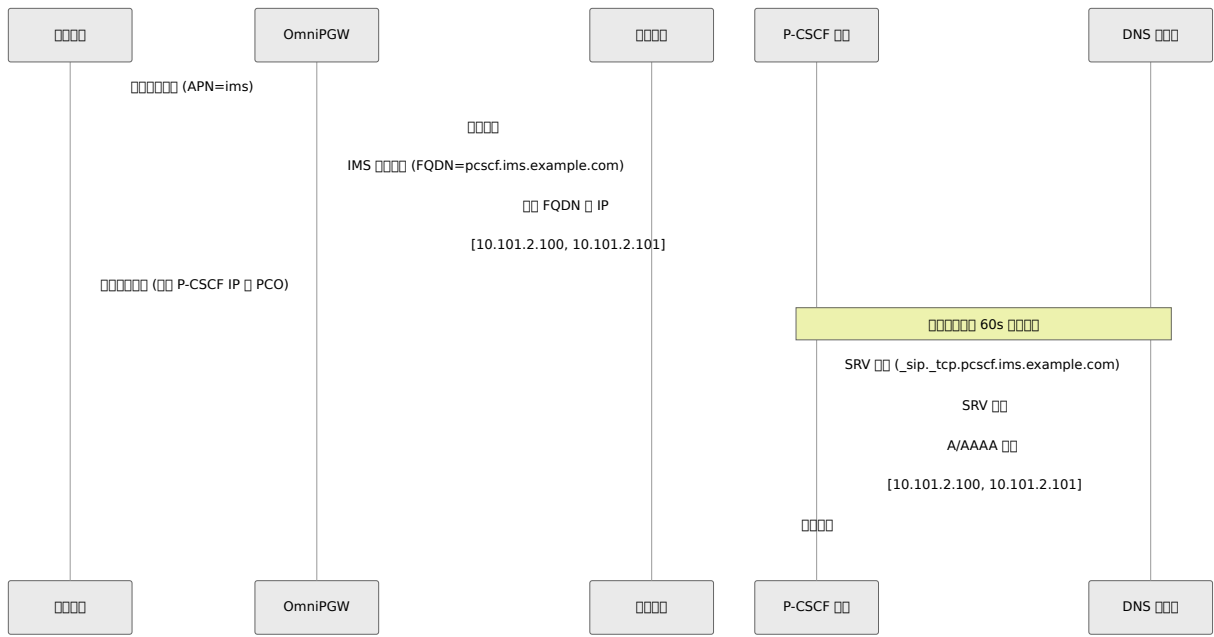
2. FQDN 設定

- 確認する FQDN 確認する
- 確認する FQDN 確認 DNS SRV 設定
- **SIP OPTIONS** 確認する
 - 確認 TCP SIP/2.0/TCP 5060 確認
 - 確認 TCP 確認 UDP SIP/2.0/UDP 5060 確認
 - 確認 :up 確認 :down 確認/確認
- 確認 IP 確認

3. 確認する 60 確認

- 確認する FQDNs
- DNS 確認する
- 確認する
 - 確認 TCP 確認 SIP OPTIONS 5 確認
 - 確認 TCP 確認 UDP 5 確認
 - 確認
- 確認する DNS 確認

网络架构图



DNS 配置

配置 DNS SRV 记录指向 P-CSCF 地址

1. **SRV** 记录指向 `_sip._tcp.{fqdn}` 的 SRV 记录
2. 配置 DNS 记录
3. 配置 DNS SRV 记录
4. 配置 DNS 记录指向 IP 地址 A/AAAA 记录
5. 配置 DNS 记录指向 IP 地址

P-CSCF 配置

配置 DNS FQDN 记录指向 PCO 的 FQDN 记录

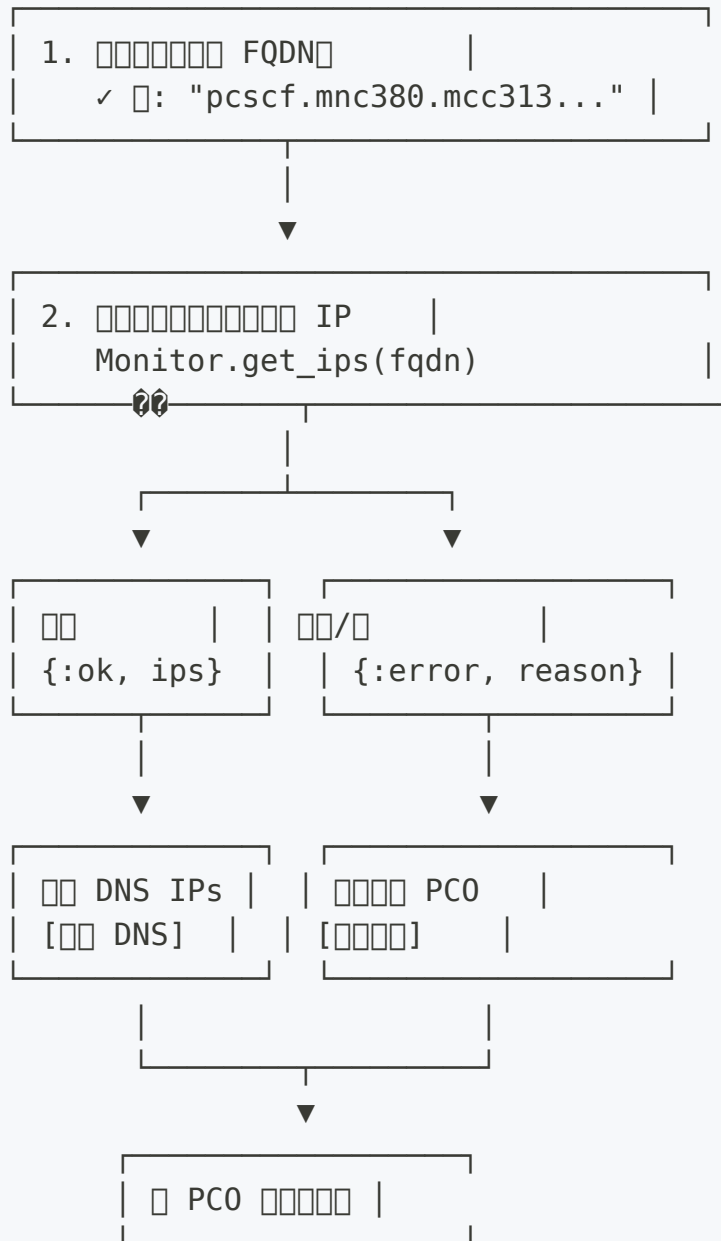
```
%{
  name: "IMS  ",
  p_cscf_discovery_fqdn: "pcscf.mnc380.mcc313.3gppnetwork.org",  #
←  ""
  pco: %{
    p_cscf_ipv4_address_list: ["10.101.2.100", "10.101.2.101"]  #
←  ""
  }
}
```

Table 1

Field	P-CSCF Field	IP	Value
FQDN Field	DNS Field	DNS IP	"FQDN pcscf.example.com P-CSCF IP"
FQDN Field	PCO Field	PCO pco.p_cscf_ipv4_address_list IP	"FQDN P-CSCF IPs..., "
FQDN Field	PCO Field	PCO pco.p_cscf_ipv4_address_list IP	
Field	PCO Field	PCO pco.p_cscf_ipv4_address_list IP	
Field FQDN	PCO Field	IP	

Table 2

IMS 鉴权流程



鉴权

1. DNS 鉴权

```
DNS [[10.101.2.150, 10.101.2.151]
UE [[10.101.2.150, 10.101.2.151] ← [[ DNS
[[: [[ DNS [[[[[[ PC0 [[[[
```

□□□□□□□□□□ FQDN □□□ PCO□

```
# ✓ 00: 000000
%{
  p_cscf_discovery_fqdn: "pcscf.ims.example.com", # 00
  pco: %{
    p_cscf_ipv4_address_list: ["10.101.2.100"] # 000
  }
}

# △ 00: 0000000000 PC00
%{
  p_cscf_discovery_fqdn: "pcscf.ims.example.com"
  # 0000000000
}

# ✓ 00: 000000 DNS 000
%{
  pco: %{
    p_cscf_ipv4_address_list: ["192.168.1.50"]
  }
}
```

Web UI 00

P-CSCF 0000

00000000 https://localhost:8086/pcscf_monitor

111

- [illegible]

- 設定

- 設定可能な FQDNs
- **FQDN** 設定可能な FQDN
- 設定可能な 5 設定

- 設定

- **FQDNs** 設定可能な FQDN 設定
- 設定可能な DNS 設定可能な FQDNs
- **DNS** 設定可能な FQDNs
- **P-CSCF** 設定可能な FQDNs 設定可能な
- ✓ **SIP OPTIONS** 設定可能な SIP OPTIONS 設定可能な
- X **SIP OPTIONS** 設定可能な SIP OPTIONS 設定可能な
- **DNS** 設定可能な DNS 設定可能な
- 設定可能な SIP OPTIONS 設定可能な 60s 5s 設定可能な

設定可能な DNS 設定可能な P-CSCF 設定可能な SIP OPTIONS

UPF 設定

UPF 設定 (/upf_selection) 設定可能な P-CSCF 設定可能な

```
IMS (20)
  APN: ^ims
  UPF-IMS-Primary (10.100.2.21:8805)

P-CSCF
  FQDN: pcscf.mnc380.mcc313.3gppnetwork.org
  : ✓ (2 IP)
  IPs: 10.101.2.100, 10.101.2.101

PCO
  DNS: 10.103.2.195
  P-CSCF: 10.101.2.100, 10.101.2.101
```

Prometheus

P-CSCF Prometheus 42069

Gauge

```
# FQDN
pcscf_fqdns_total          # FQDN
pcscf_fqdns_resolved      # FQDNs DNS
pcscf_fqdns_failed        # FQDNs DNS

# 
pcscf_servers_total       # DNS SRV P-CSCF
pcscf_servers_healthy     # SIP OPTIONS
pcscf_servers_unhealthy   # SIP OPTIONS

# FQDN
pcscf_servers_healthy{fqdn="..."} # FQDN
pcscf_servers_unhealthy{fqdn="..."} # FQDN
```

- **healthy**: 成功した SIP OPTIONS ping (TCP 或 UDP)
- **unhealthy**: 失敗した SIP OPTIONS (5s 未満)

例

DNS 監視

```
# 監視対象の FQDNs
pcscf_fqdns_resolved

# DNS 成功率
(pcscf_fqdns_resolved / pcscf_fqdns_total) * 100

# 監視対象の総数
pcscf_servers_total
```

SIP OPTIONS 監視

```
# SIP OPTIONS 成功した FQDNs
pcscf_servers_healthy

# SIP OPTIONS 失敗した FQDNs
pcscf_servers_unhealthy

# SIP OPTIONS 成功率
(pcscf_servers_healthy / pcscf_servers_total) * 100

# 特定の FQDN 監視
pcscf_servers_healthy{fqdn="pcscf.mnc380.mcc313.3gppnetwork.org"}

# 監視対象の総数
pcscf_servers_healthy == 0 AND pcscf_servers_total > 0
```

Prometheus 監視

```

# [] P-CSCF []
- alert: AllPCSCFServersDown
  expr: pcscf_servers_healthy == 0 AND pcscf_servers_total > 0
  for: 5m
  labels:
    severity: critical
  annotations:
    summary: "[] P-CSCF []"
    description: "{{ $value }} [] (0) - [] SIP OPTIONS []"

# [] 50% []
- alert: MajorityPCSCFServersDown
  expr: (pcscf_servers_healthy / pcscf_servers_total) < 0.5
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "[] P-CSCF []"
    description: "[] {{ $value }}% [] SIP OPTIONS"

# [] DNS []
- alert: PCSCFDNSResolutionFailed
  expr: pcscf_fqdns_failed > 0
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "P-CSCF DNS []"
    description: "{{ $value }} [] FQDN []"

```

[][][]

[][][][][]


```
[info] P-CSCF [] []
[info] [] 2 [] [] P-CSCF FQDNs [] []: ["pcscf.ims.example.com",
"pcscf.enterprise.example.com"]
[info] P-CSCF []: [] FQDN pcscf.ims.example.com
[debug] P-CSCF []: [] pcscf.ims.example.com [] 2 [] IP
[warning] P-CSCF []: [] pcscf.enterprise.example.com: :nxdomain
[debug] [] FQDN pcscf.ims.example.com [] P-CSCF []: [{10, 101,
2, 100}, {10, 101, 2, 101}]
```

配置

配置 DNS 和 PCO 配置

1 DNS 配置

```
p_cscf_discovery_fqdn: "pcscf.ims.example.com"
```

- 配置 DNS 配置 IP
- 配置 DNS 配置 IP
- 配置 DNS 配置 IP

2 配置 PCO

```
pco: %{
  p_cscf_ipv4_address_list: ["10.101.2.100", "10.101.2.101"]
}
```

- 配置 DNS 配置 IP
- 配置 DNS 配置 IP
- 配置 DNS 配置 IP

3. PC0 設定

```
# PC0 設定
pco: %{
  p_cscf_ipv4_address_list: ["10.101.2.146"]
}
```

- DNS 設定
- P-CSCF 設定
- 接続設定

DNS 設定

設定 "IMS 設定" の DNS

1. DNS 設定 "pcscf.ims.example.com"
 - └ IP → IP [10.101.2.100, 10.101.2.101] ✓
 - └ IP → IP
2. PC0 設定
 - └ IP → IP [10.101.2.100, 10.101.2.101] ✓
 - └ IP → IP
3. PC0 設定
 - └ IP [10.101.2.146] ✓ (設定)

DNS 設定

DNS 設定

DNS 設定 SRV 型 A/AAAA 型 P-CSCF 設定

```

; P-CSCF SRV _sip._tcp
_sip._tcp.pcscf.mnc380.mcc313.3gppnetwork.org. IN SRV 10 50 5060
pcscf1.example.com.
_sip._tcp.pcscf.mnc380.mcc313.3gppnetwork.org. IN SRV 20 50 5060
pcscf2.example.com.

; A
pcscf1.example.com. IN A 10.101.2.100
pcscf2.example.com. IN A 10.101.2.101

```

OmniPGW FQDN `_sip._tcp.` `p_cscf_discovery_fqdn:`
`"pcscf.mnc380.mcc313.3gppnetwork.org"`
`_sip._tcp.pcscf.mnc380.mcc313.3gppnetwork.org`

SRV

SRV

```

_service._proto.domain. IN SRV priority weight port target.

```

- 10 ~ 20
- =
- SIP TCP 5060 UDP 5060
- IP

❏❏ DNS ❏❏

```
# ❏❏ SRV ❏❏❏❏❏ _sip._tcp ❏❏❏
dig SRV _sip._tcp.pcscf.mnc380.mcc313.3gppnetwork.org
@10.179.2.177

# ❏❏❏❏❏
# _sip._tcp.pcscf.mnc380.mcc313.3gppnetwork.org. 300 IN SRV 10 50
5060 pcscf1.example.com.

# ❏❏❏ P-CSCF ❏❏❏❏❏ IP
dig A pcscf1.example.com @10.179.2.177

# ❏❏❏❏❏
# pcscf1.example.com. 300 IN A 10.101.2.100
```

❏❏❏❏❏

❏❏❏FQDN ❏❏“❏❏”❏❏

❏❏❏

- Web UI ❏❏ X ❏❏❏❏
- ❏❏❏ :nxdomain❏:timeout❏ :no_naptr_records

❏❏❏❏❏

1. DNS ❏❏❏❏❏❏
2. FQDN ❏ DNS ❏❏❏❏
3. ❏❏❏ NAPTR ❏❏
4. DNS ❏❏❏❏❏

❏❏❏❏❏

```
# 1. Ping DNS Server
ping 10.179.2.177

# 2. Verify NAPTR Record
dig NAPTR pcscf.mnc380.mcc313.3gppnetwork.org @10.179.2.177

# 3. Check OmniPGW Log
grep "P-CSCF" /var/log/pgw_c.log

# 4. Check Configuration
grep "p_cscf_discovery_dns_server" config/runtime.exs

# 5. Check Web UI Configuration
# Verify FQDN is "0.0.0.0"
```

Verify IP

Check

- Web UI "0.0.0.0 IP"
- Configuration ✓ IP is X

Check

1. NAPTR Record FQDN is
2. Verify IMS/SIP
3. Check A/AAAA

Check

```
# Verify NAPTR Record
dig NAPTR pcscf.example.com @10.179.2.177

# Verify "SIP" and "IMS":
# SIP: "SIP+D2U", "x-3gpp-ims:sip"
# SIP: "HTTP", "FTP"

# Check A/AAAA Record
dig pcscf1.example.com A @10.179.2.177
```

🔍🔍🔍🔍🔍🔍 P-CSCF

🔍🔍

- UE 📶📶📶 P-CSCF 📶
- 📶📶📶📶📶📶 IP

🔍🔍🔍🔍

1. DNS 📶📶📶📶📶📶
2. 📶📶📶📶
3. FQDN 📶📶

🔍🔍🔍🔍

```
# 1. 📶 P-CSCF 📶📶  
# 📶 FQDN 📶📶📶📶  
  
# 2. 📶📶📶  
grep "📶📶 FQDN 📶 P-CSCF 📶" /var/log/pgw_c.log  
  
# 3. 📶 UPF 📶📶  
# 📶📶📶📶📶📶 FQDN 📶📶  
  
# 4. 📶📶📶  
# 📶📶 APN 📶📶📶📶📶📶
```

🔍🔍🔍 DNS 📶📶🔍

🔍🔍

- 📶📶📶📶
- 📶📶📶 `pcscf_discovery_query_duration_seconds`

🔍🔍🔍🔍

1. DNS 📶📶📶📶
2. 📶 DNS 📶📶📶📶
3. 📶📶📶📶

配置

```
# 配置
pco: %{
  p_cscf_discovery_timeout_ms: 2000 # 5000ms
}

# 配置 DNS
pco: %{
  p_cscf_discovery_dns_server: "10.0.0.10" # DNS
}
```

配置

1. DNS 配置

配置 DNS 配置

```
pco: %{
  # 配置 P-CSCF DNS UE DNS
  p_cscf_discovery_dns_server: "10.179.2.177",

  # UE DNS 配置
  primary_dns_server_address: "8.8.8.8",
  secondary_dns_server_address: "8.8.4.4"
}
```

配置

- 配置 UE DNS 配置 IMS DNS
- 配置 DNS 配置
- 配置 DNS 配置

2. 配置 P-CSCF

```
%{
  p_cscf_discovery_fqdn: "pcscf.ims.example.com", # P-CSCF
  pco: %{
    p_cscf_ipv4_address_list: ["10.101.2.100"] # P-CSCF IP
  }
}
```

配置

- 配置 DNS 记录
- 配置
- 配置 SLA

3. 配置 P-CSCF FQDN

```
rules: [
  # IMS
  %{
    name: "IMS",
    match_regex: "^ims",
    p_cscf_discovery_fqdn:
"pcscf.ims.mnc380.mcc313.3gppnetwork.org"
  },

  # P-CSCF
  %{
    name: "P-CSCF",
    match_regex: "^enterprise",
    p_cscf_discovery_fqdn: "pcscf.enterprise.example.com"
  }
]
```

配置

- 配置 P-CSCF
- 配置

- 配置

4. 配置 DNS 告警

```
# 配置 P-CSCF 告警
alert: HighPCSCFQueryLatency
expr: histogram_quantile(0.95,
pcscf_discovery_query_duration_seconds_bucket) > 2
for: 5m
labels:
  severity: warning
annotations:
  summary: "P-CSCF DNS 异常 (p95 > 2s)"
```

5. 配置 DNS 告警

- **Web UI** 配置 P-CSCF 告警
- 配置 `pcscf_monitor_fqdns_failed` 告警
- 配置 DNS 告警
- 配置 DNS 告警

6. 配置告警

```
# 配置告警
pco: %{
  p_cscf_discovery_timeout_ms: 5000 # 5 秒
}

# 配置告警
pco: %{
  p_cscf_discovery_timeout_ms: 2000 # 2 秒
}
```

7. 配置 DNS 告警

配置 DNS 告警

```
# 1 P-CSCF DNS
pcscf.mnc380.mcc313.3gppnetwork.org. IN NAPTR 10 50 "s" "SIP+D2U"
"" _sip._udp.pcscf1.example.com.

# 2 P-CSCF DNS
pcscf.mnc380.mcc313.3gppnetwork.org. IN NAPTR 20 50 "s" "SIP+D2U"
"" _sip._udp.pcscf2.example.com.
```

□□□□

- **PCO** □□ - □□□□□□□□ DNS □ P-CSCF □□
 - □□□□ - □□□ OmniPGW □□□□
 - □□ - □□□□□□□□□□
 - □□□□ - □□□□□□□□ PCO □□
 - **PFCP** □□ - □□□□□□□□
-

□□□□□

OmniPGW P-CSCF □□ - *Omnitouch* □□□□□□

PFCP/Sxb

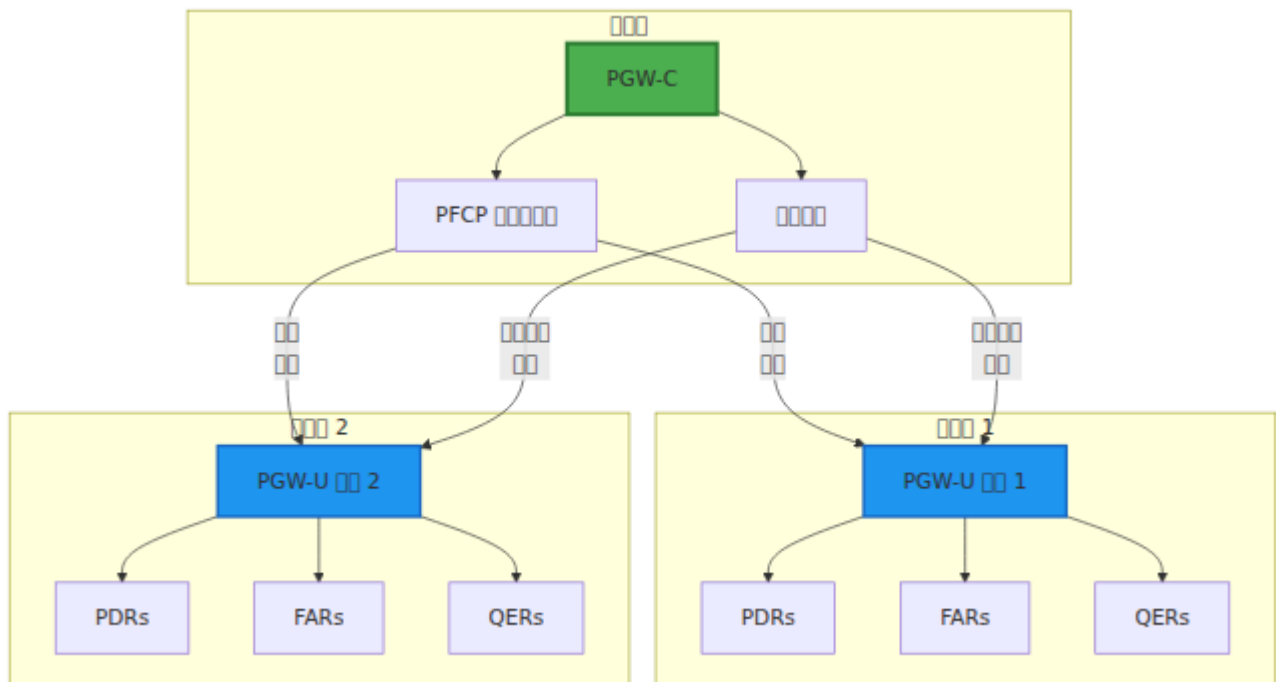
PGW-C - PGW-U

-
-
- PFCP
- PFCP
-
-
- DNS UPF
-
-
- Web UI - PFCP
-

Sxb PFCP PGW-C PGW-U

- PGW-C
- PGW-U

PFCP



PGW-U

PFCP

PGW-C and PFCP are defined in 3GPP TS 29.244

PGW-U

- User Plane
- Port 8805
- PFCP and User Plane

PGW ID

PGW ID is defined in

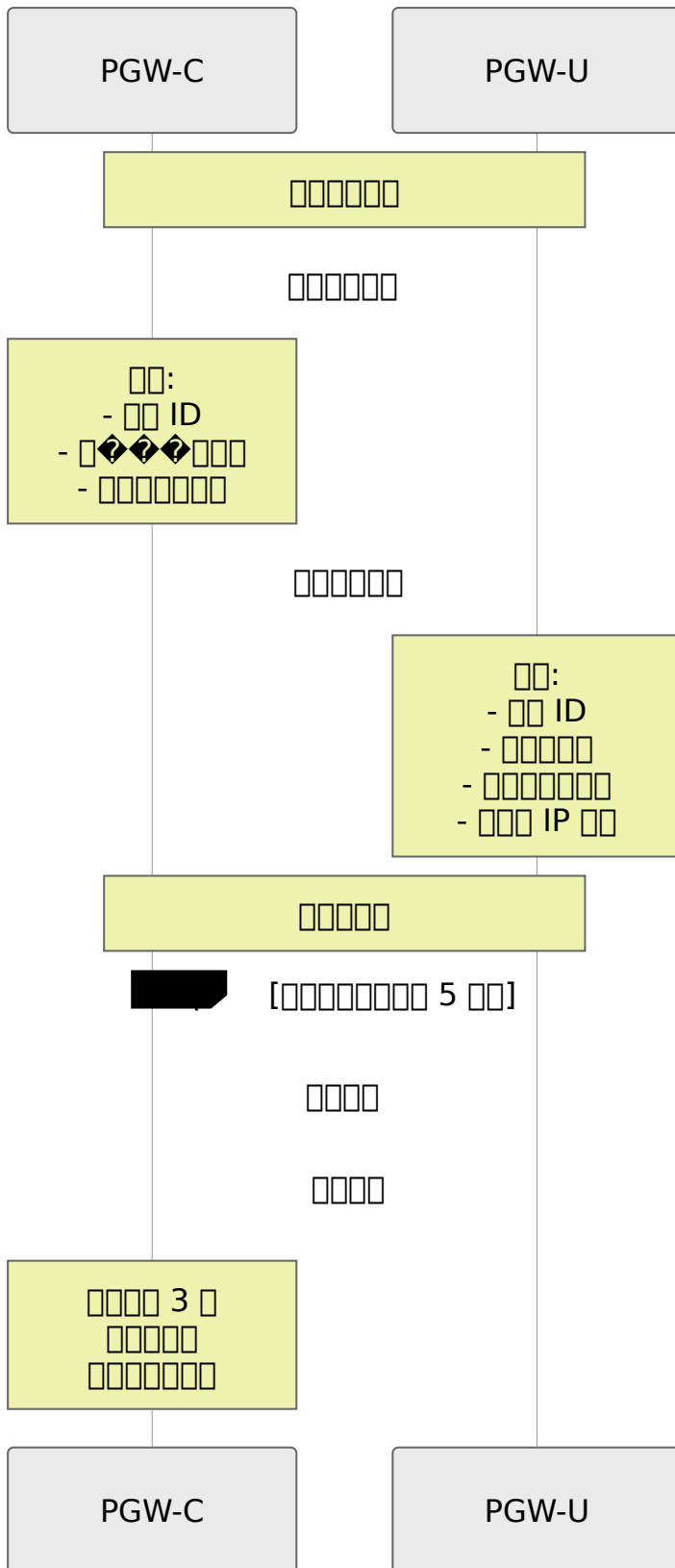
- IPv4 - Port
- IPv6

- FQDN□□□□□□□□

PFCP □□□□

□□□□□□□□□□ PGW-C □ PGW-U □□□□ PFCP □□□

□□□□□□



节点信息

节点 PFCP 配置信息

属性	说明
is_associated	是否关联
remote_node_id	远端节点 ID/IP 或 FQDN
remote_ip_address	远端节点 IP 地址
remote_port	UDP 远端节点 8805 端口
heartbeat_period_ms	心跳周期
missed_heartbeats_consecutive	连续丢失心跳次数
up_function_features	UP 功能特性
up_recovery_time_stamp	UP 恢复时间戳

节点 ID

节点 ID 配置信息

节点 ID

```
# config/runtime.exs
sxb: %{
  local_ip_address: "10.0.0.20"
},
upf_selection: %{
  fallback_pool: [
    %{remote_ip_address: "10.0.0.21", remote_port: 8805, weight:
100}
  ]
}
# UPF 5
```

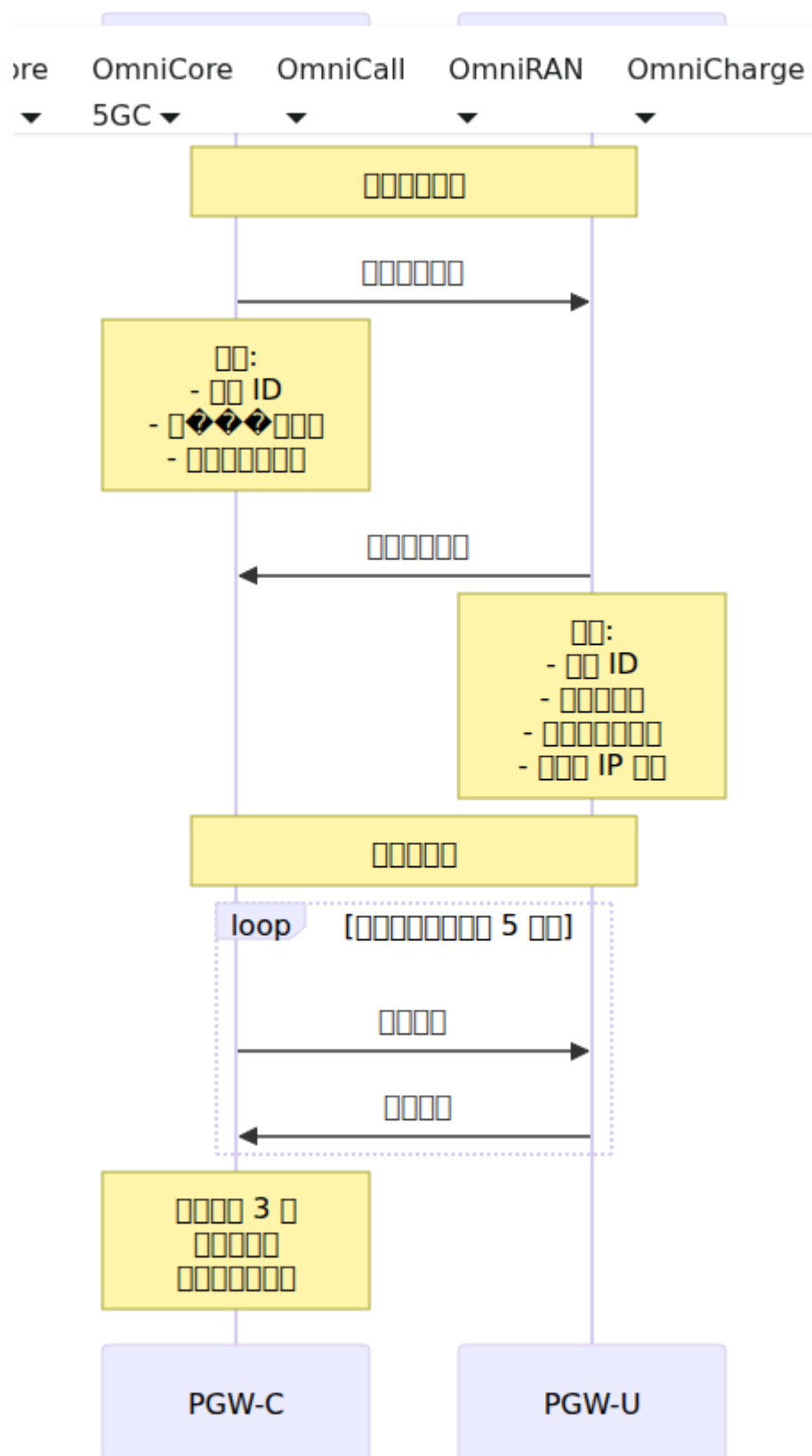
•

- missed_heartbeats_consecutive
- 3
-

PFCP

PFCP UE PDN

□□□□□□



PGW-C

UE PDN

PGW-C PGW-U

PGW-C

- **SEID** - ID -
- **ID** - PGW-C ID
- **F-SEID** - SEID IP + SEID
- **PDRs** - 2 +
- **FARs** - 2 +
- **QERs** - QoS
- **BAR** -

PGW-U

PGW-U

- -
- **F-SEID** - PGW-U
- **PDRs** -
- **F-TEID** - TEID S5/S8

QoS

QoS

QoS

- PDRsFARsQERs
-
-

UE PDN

111

1. PGW-C 0000000000 SEID
2. PGW-U 0000000000
3. PGW-U 00000000

F-TEID ☐

F-TEID□□□□□□□□□□ □□□□□□□□ GTP-U □□□□□□□□ PFCP □□□□□□□□ F-TEID□□□
□ UPF □□□□□□□□□□□□□□□□□□□□

□□ **F-TEID** □□

XXXXXXXXXX F-TEID XXXXXXXXXX

- **TEID**□□□□□□□□ - 32 □□□□□□□□
- **IP** □□ - □□□□ GTP-U □□□□UPF □ IP □□□□

TEID

1UPF

- PGW-C “TEID” CHOOSE
- UPF TEID

2 PGW-C

- PGW-C 通过 TEID 通知 UPF“源网络 TEID”
- UPF 通过 TEID 通知源网络

UPF 000000 - 0000

111

```
sxb: %{  
    allocate_uplink_f_teid: false # []  
}
```

[][]

1. PGW-C [] [] F-TEID CHOOSE [] [] PFCP [] [] [] []
2. UPF [] [] [] [] [] [] [] [] TEID
3. UPF [] [] [] [] F-TEID[]TEID + IP [] []
4. PGW-C [] [] [] [] [] [] [] [] F-TEID

[] [] [] [] [] [] [] []

[] [] [] [] []

- UPF [] [] [] [] = UPF [] [] [] [] [] [] [] []
- PGW-C [] [] [] [] UPF [] [] [] TEID
- [] [] [] [] [] [] [] [] [] []

[] [] **PGW-C** [] [] [] []

- [] [] PGW-C [] [] [] [] [] [] UPF [] [] [] [] [] [] [] []
- [] [] PGW-C [] [] [] [] [] TEID [] [] [] [] []
- UPF [] [] [] [] [] [] [] [] [] [] [] [] [] []

[] [] **3GPP** [] []

- CHOOSE [] [] [] 3GPP TS 29.244 [] [] []
- [] [] UPF [] [] [] [] [] [] [] []
- [] [] “[] [] [] [] [] []” [] [] [] []

[] [] [] [] [] [] [] []

- [] [] PGW-C [] [] [] [] [] UPF [] [] [] [] TEID [] [] [] []
- [] [] [] [] TEID [] [] [] [] []
- UPF [] [] [] [] [] [] [] [] TEID

[] [] [] [] []

- 1 个 UPF 地址
- 1 个 UPF 地址 PGW-C 地址
- 1 个地址
- 1 个地址 3GPP PCF 地址

地址

- △ 1 个 UPF 地址 CHOOSE 地址
- △ 1 个地址“地址 IE”地址 UPF 地址 CHOOSE

PGW-C 地址

地址

```
sxb: %{
  allocate_uplink_f_teid: true
}
```

地址

1. PGW-C 地址 TEID
2. PGW-C 地址 TEID 地址 PCF 地址
3. UPF 地址 TEID 地址
4. PGW-C 地址 UPF 地址 TEID 地址

地址

1 UPF 地址 CHOOSE

- 1 UPF 地址/地址
- UPF 地址 PCF 地址 TEID
- 地址

1 TEID 地址

- 地址 PGW-C 地址 TEID
- 地址 PGW-C 地址 TEID 地址
- 地址 TEID 地址

□ □□□□□

- □□□□□□□□ TEID □□□□
- □□□□□□□□□□□ TEID □□

□□□

△ □ **PGW-C** □□□□

- □□ UPF □□□ PGW-C □□□□□□ TEID □□
- □□□
 - □□ PGW-C □□□ TEID □□□□□□□□
 - □□ TEID □□□□□□□□□□□□
 - □□□□□□□□□□□□□□□□

△ □□□□

- PGW-C □□□□□□□□ TEID □□□□□□
- PGW-C □□□□□□ TEID □□□□□□□□□□□□□□
- □□□□□□□□□□

△ □□□□□

- □□□□□□ PFCP □□□□□
- □□□□□□□□ CHOOSE □ UPF □□□□□□

□□□□□

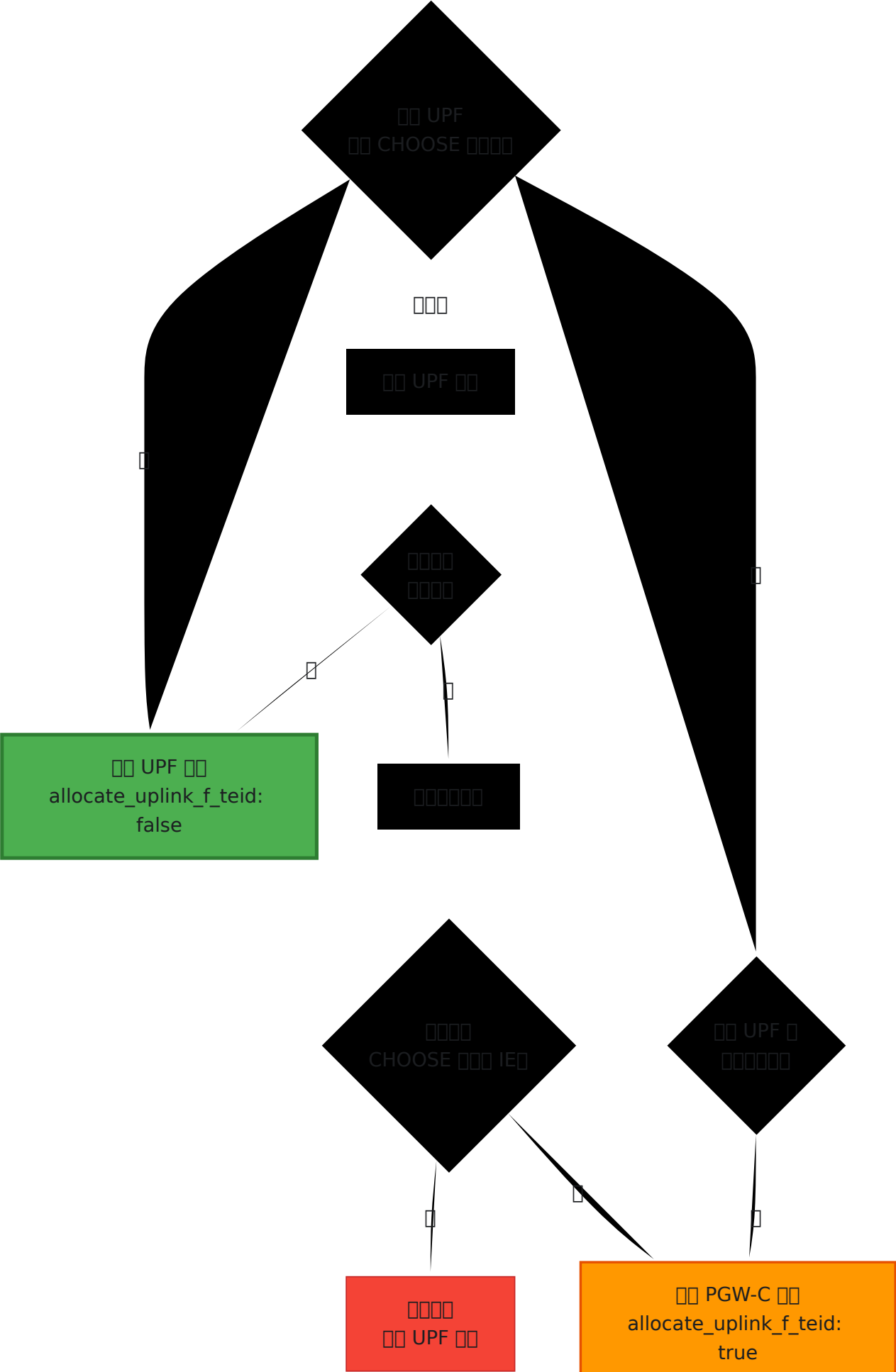
- △ □□ **UPF** □□□ **CHOOSE** □□□
- △ □□ UPF □□□□□□□□□□□□□□
- △ □□□□□□□□
- △ □□ PGW-C TEID □□□□□□□□

TEID □□□□□ PGW-C □□□□□□□□□□□□□□

- TEID □□□1 □ 0xFFFFFFFF□42 □□□□□
- □□□□□□□ 100 □□□□□□□□ 0.023%
- □□□□□□□□□□□□□□□□

- 000000000000 TEID

0000



□□□□

□□□□□□□□□□

□□ PFCP □□□

```
# □□□ CHOOSE □□□□□
grep -i "choose\|mandatory.*missing" /var/log/pgw_c.log

# □□ PFCP □□□□□□□□□□
grep "Session Establishment Response" /var/log/pgw_c.log
```

□□ **UPF** □□ **CHOOSE** □□□

- □□□□□□“□□□□ IE”□“□□ IE”
- UPF □□□□ F-TEID□□□□□ CHOOSE
- □□□□□ □□ `allocate_uplink_f_teid: true`

□□ **PGW-C** □□□□□□□

- □□□□ - TEID □□□□□40 □□□□
- □□ TEID □□□□□□□□□□□□□□□□□□□□

```
# □□□□□□□
grep "registered_teid_count" /var/log/pgw_c.log
```

□□□□□□□

```
# □□ config/runtime.exs
sxb: %{
  local_ip_address: "10.0.0.20",
  allocate_uplink_f_teid: false # □□ UPF □□□ CHOOSE□□□□□ true
}
```

□□□□ PGW-C□

```
systemctl restart pgw_c
```

確認する必要があるのは PFCP かどうか

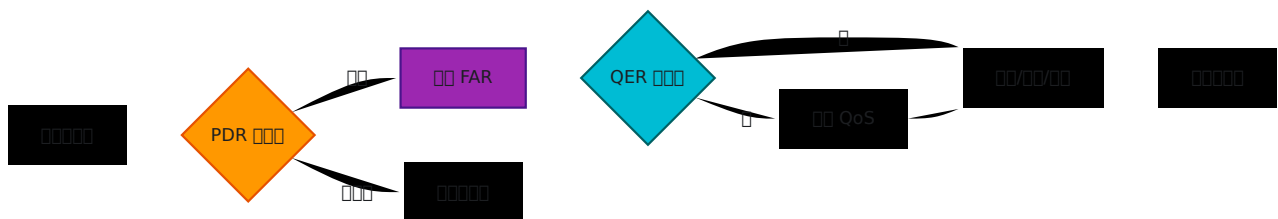
```
# PFCP 確認
tcpdump -i any -n port 8805 -w pfcp.pcap

# Wireshark で確認する
# F-TEID が "CH00SE" になるまで UPF 確認
# F-TEID が TEID が PGW-C になるまで
```

確認する

PFCP 確認する必要がある

確認



PDR確認する

確認 確認する必要がある

PGW-C 確認

PDR #1 - 確認

FAR ID: 1□□□□□□□□□□

QER ID: 1QoS

FAR #1 - 000000 → UE00

- ACCESS GW
- GTP-U/UDP/IPv4
- F-TEID: <SGW S5/S8 >

- 内核CORE
- 用户空间 - IP

- **FAR ID** - 00000000
- 0000 - 000000000000
- 00000
 - 00000ACCESS/CORE0
 - 0000000000 GTP-U 0000
 - 00000VRF/0000

□□ QER□

QER ID: 1

□□□□□□

□□□□□□

- □□□100 Mbps

- □□□50 Mbps

□□□□□□□□□□ GBR □□□

- □□□10 Mbps

- □□□10 Mbps

□□ **QER** □□□

- **QER ID** - □□□□□□□
- □□□ - □□□□□□□□□□□□□□
- **MBR** - □□□□□□□□/□□□
- **GBR** - □□□□□□□□□□□□□□
- **QCI** - QoS □□□□□□□□□□□□

BAR□□□□□□□□

□□□ □□ UE □□□□□□□□□□□□

□□ **BAR**□

BAR ID: 1

□□□□□□□□□□100ms

□□□□□□□□□□10

□□□ □□□□ DRX□□□□□□□□□□

□□

□□ **Sxb** □□

□□ `config/runtime.exs`□

```

config :pgw_c,
  sxb: %{
    # PFCP IP
    local_ip_address: "10.0.0.20",

    # 8805
    local_port: 8805,

    # PFCP 500ms
    # UPF
    # UPF
    request_timeout_ms: 500,

    # PFCP 3
    # = request_timeout_ms * request_attempts
    request_attempts: 3,

    # F-TEID
    # false UPF F-TEID CHOOSE
    # true PGW-C F-TEID
    # UPF CHOOSE
    allocate_uplink_f_teid: false
  },

  # UPF - UPF
  upf_selection: %{
    fallback_pool: [
      %{
        # PGW-U IP
        remote_ip_address: "10.0.0.21",

        # PFCP 8805
        remote_port: 8805,

        # 100 = 0 =
        weight: 100
      }
    ]
  }
}

```

1. 背景

PGW-C と UPF の間での PFCP プロトコルの動作

2. 概要

パラメータ	単位	値	説明
request_timeout_ms	ms	500	PGW-C が UPF からの応答を待つタイムアウト時間
request_attempts	回	3	タイムアウトした場合の再試行回数

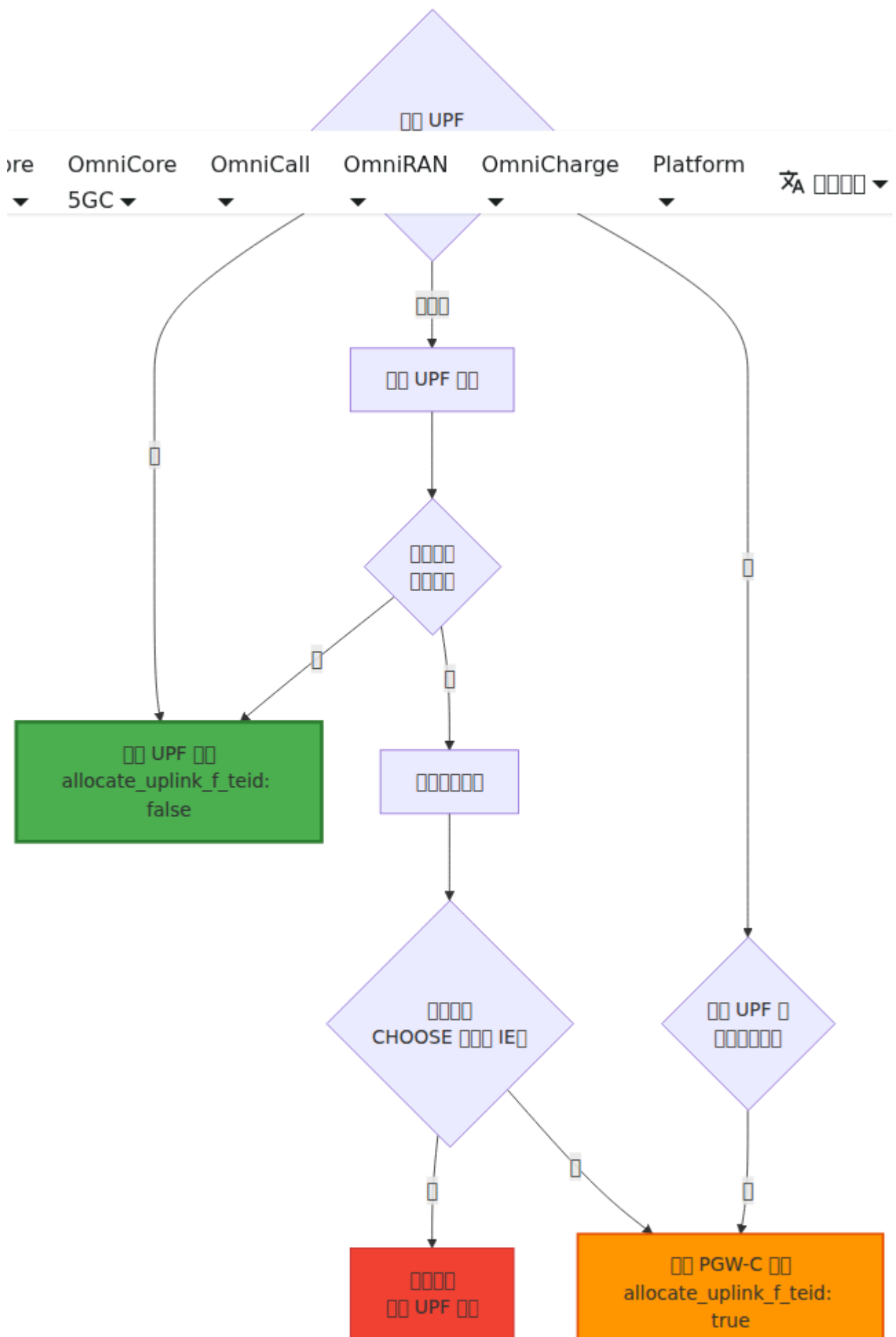
総待ち時間は $\text{request\_timeout\_ms} \times \text{request\_attempts}$ で計算される。

つまり $500\text{ms} \times 3 = 1.5$ 秒かかる。

3. 詳細な手順

PGW-C は request_timeout_ms を設定し、UPF に対して PFCP プロトコルを送信する。

- PGW-C が UPF に対して PFCP プロトコルを送信する。
- UPF が PGW-C からの PFCP プロトコルを受信する。
- PGW-C が UPF からの PFCP プロトコルを受信する。
- UPF が PGW-C に対して PFCP プロトコルを送信する。
- PGW-C が UPF からの PFCP プロトコルを受信し、ID を確認する。
- PGW-C が UPF に対して PFCP プロトコルを送信する。



1111

UPF 级别	请求 request_timeout_ms	超时时间
一级 <100ms	200-300ms	600-900ms 3 倍
二级 100-300ms	500ms	1.5s 3 倍
三级 300-500ms	750-1000ms	2.25-3s 3 倍
四级 >500ms	1500-2000ms	4.5-6s 3 倍

```
request_timeout_ms 2 UPF
```

□□ - □ **UPF**

```
sxb: %{
    local_ip_address: "10.0.0.20",
    request_timeout_ms: 1000, # 1000ms 1s
    request_attempts: 3 # 3 attempts
}
```

□□□□□

□ □ □ □ □ □ □ □

- UPF 可以 PFCP 和网元交互
- 网元向 UPF 注册
- 网元向“网元管理器”注册

□□□□□

1. UPF API 호출하면 UPF 응답
2. PGW-C 호출하면 PGW-C
3. UPF 호출하면 UPF

□□□□□

1. `request_timeout_ms` `UPF`
2. `PGW-C`
3. `UPF`

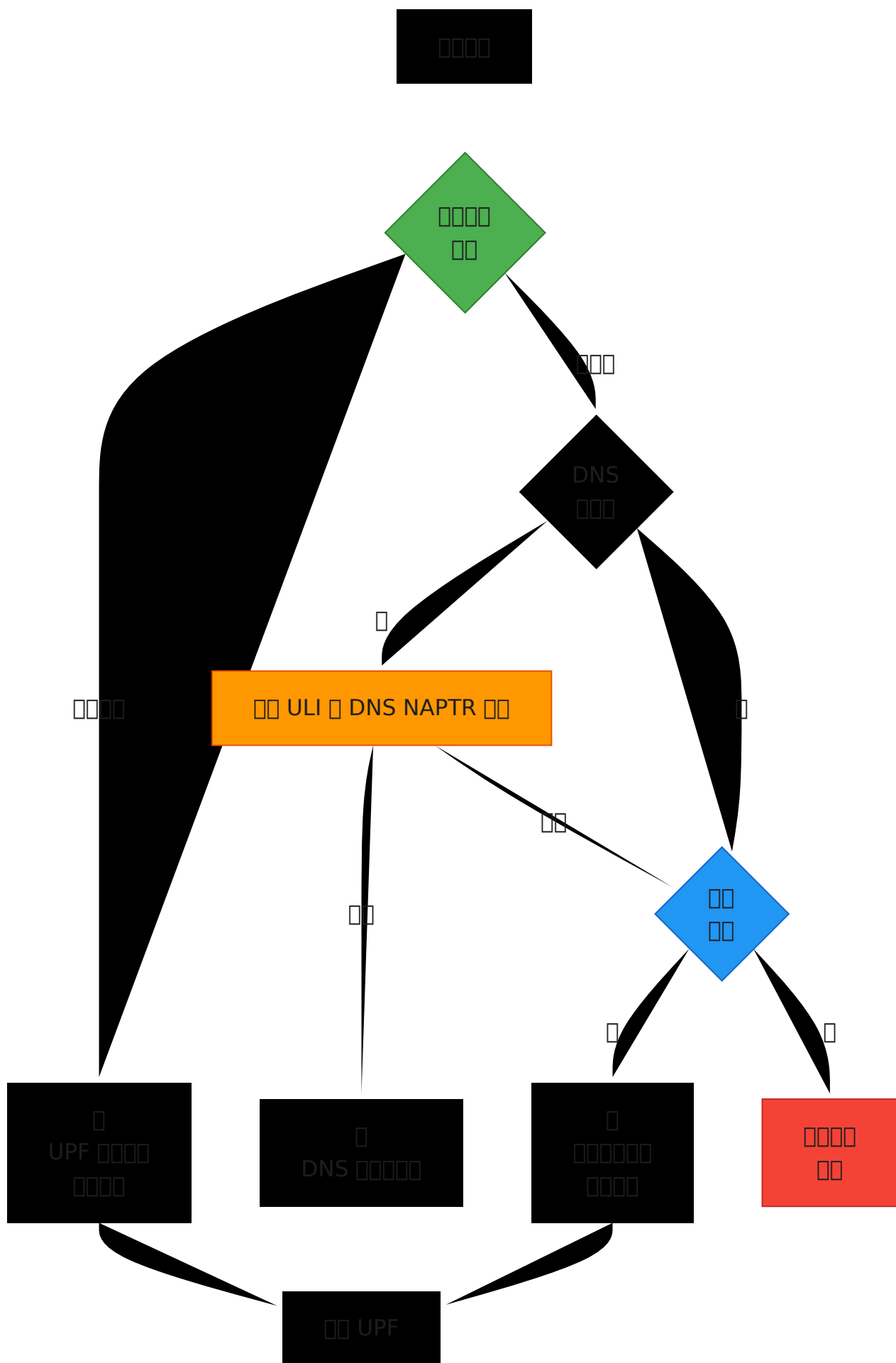
`PGW-U`

```
config :pgw_c,  
  sxb: %{  
    local_ip_address: "10.0.0.20"  
  },  
  upf_selection: %{  
    fallback_pool: [  
      %{remote_ip_address: "10.0.1.21", remote_port: 8805, weight:  
50}, # 50%  
      %{remote_ip_address: "10.0.2.21", remote_port: 8805, weight:  
50} # 50%  
    ]  
  }  
# UPF 5
```

UPF 四四四

PGW-C 四四 **UPF** 四四四四四四四四四四

1. 四四四四四四四四 - 四四四四四四四四
2. 四四 **DNS** 四四四四四四四四 - 四四 DNS NAPTR 四四四四四四四四
3. 四四四四四四四四 - 四四四四四四四四四四 UPF 四





000 **UPF** 0000

```

config :pgw_c,
  # PFCP
  sxb: %{
    local_ip_address: "10.0.0.20"
  },

  # UPF
  upf_selection: %{
    #
=====

    # DNS
    #

=====

    # ULI DNS
    # UPF
    dns_enabled: false,
    dns_query_priority: [:ecgi, :tai, :rai, :sai, :cgi],
    dns_suffix: "epc.3gppnetwork.org",
    dns_timeout_ms: 5000,

    #
=====

    #
    #
=====

    #
    # UPF
    rules: [
      # 1 IMS -
      %{
        name: "IMS",
        priority: 20,
        match_field: :apn,
        match_regex: "^ims",
        upf_pool: [
          %{remote_ip_address: "10.100.2.21", remote_port: 8805,
weight: 80},
          %{remote_ip_address: "10.100.2.22", remote_port: 8805,
weight: 20}
        ],
        # PCO
        pco: %{
          p_cscf_ipv4_address_list: ["10.101.2.100", "10.101.2.101"]

```

```

    }
},

# 2 APN - 
%{
    name: "",
    priority: 15,
    match_field: :apn,
    match_regex: "^(enterprise|corporate)\.apn",
    upf_pool: [
        weight: 100}
        {%remote_ip_address: "10.100.3.21", remote_port: 8805,
    ],
    pco: {%
        primary_dns_server_address: "192.168.1.10",
        secondary_dns_server_address: "192.168.1.11",
        ipv4_link_mtu_size: 1500
    }
},

# 3 - 
%{
    name: "",
    priority: 10,
    match_field: :serving_network_plmn_id,
    match_regex: "^(310|311|312|313)", # 
    upf_pool: [
        weight: 100}
        {%remote_ip_address: "10.100.4.21", remote_port: 8805,
    ],
},

# 4 - 
%{
    name: "",
    priority: 5,
    match_field: :apn,
    match_regex: "^internet",
    upf_pool: [
        weight: 33},
        {%remote_ip_address: "10.100.1.21", remote_port: 8805,
        weight: 33},
        {%remote_ip_address: "10.100.1.22", remote_port: 8805,
        weight: 33},
        {%remote_ip_address: "10.100.1.23", remote_port: 8805,

```

```
weight: 34}
    ]
  }
],

#
=====

# 
#
=====

#  DNS 
fallback_pool: [
  %{remote_ip_address: "127.0.0.21", remote_port: 8805, weight:
100}
]
}
```

Parameters

Parameter	Unit	Value
:imsi	IMSI	"310260123456789"
:apn	APN	"internet" or "ims"
:serving_network_plmn_id	PLMN ID (MCC+MNC)	"310260" or "310260.01"
:sgw_ip_address	SGW IP	"10.0.1.50"
:uli_tai_plmn_id	TAI PLMN ID	"310260"
:uli_ecgi_plmn_id	E-UTRAN PLMN ID	"310260"

UPF Parameters

UPF Parameters


```
upf_pool: [  
  {%remote_ip_address: "10.100.1.21", remote_port: 8805, weight:  
50},  
  {%remote_ip_address: "10.100.1.22", remote_port: 8805, weight:  
30},  
  {%remote_ip_address: "10.100.1.23", remote_port: 8805, weight:  
20}  
]
```

□□□□□□□□□□

1. □□□□□□ $50 + 30 + 20 = 100$
2. □□□□□□ $0.0 \sim 100.0$
3. □□□□□□□□□□ UPF□
 - 0-50□UPF-1□50% □□□
 - 50-80□UPF-2□30% □□□
 - 80-100□UPF-3□20% □□□

□□□

- □□□□□ □□□□□□□33, 33, 34□
- □/□□ □□□□□80□□□□□□□20□
- □□□□□ □□□ UPF □□□□□

PCO □□

□□□□□□ PCO□□□□□□□□□□

```
%{
  name: "IMS ",
  match_field: :apn,
  match_regex: "^ims",
  upf_pool: [...],
  pco: %{
    # 
    p_cscf_ipv4_address_list: ["10.101.2.100", "10.101.2.101"],
    #  PCO 
  }
}
```

PCO

- primary_dns_server_address
- secondary_dns_server_address
- primary_nbns_server_address
- secondary_nbns_server_address
- p_cscf_ipv4_address_list
- ipv4_link_mtu_size

DNS

PGW-C DNS NAPTR

```
upf_selection: %{
  dns_enabled: true,
  dns_query_priority: [:ecgi, :tai, :rai, :sai, :cgi],
  dns_suffix: "epc.3gppnetwork.org",
  dns_timeout_ms: 5000
}
```


1. **ECGI**E-UTRAN -
2. **TAI** -
3. **RAI** - 3G/2G
4. **SAI** - 3G

5. CGI - 2G

DNS

```
# ECGI
eci-1a2b3c.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org

# TAI
tac-lb64.tac-hb00.tac.epc.mnc999.mcc999.epc.3gppnetwork.org
```

DNS

- ECGI TAI
- DNS
- UPF
- DNS DNS

DNS UPF

DNS UPF

DNS UPF UE  ULI DNS NAPTR

3GPP TS 23.003 - UPF DNS

- UPF
-
-
- UPF

ECI - LTE ECI

ECI

```
eci-<HEX-ECI>.ecgi.epc.mnc<MNC>.mcc<MCC>.<dns_suffix>
```

ECI

```
# ECI ID: 0x1A2B3C1,715,004 ECI
# PLMN: MCC=999, MNC=999
eci-1a2b3c.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org
```

ECI LTE 4G ECI

2. TAI

ECI - TAI

ECI

```
tac-lb<LB>.tac-hb<HB>.tac.epc.mnc<MNC>.mcc<MCC>.<dns_suffix>
```

ECI

```
# TAC: 0x0064100 TAC
# ECI: 0x641000: 0x00
tac-lb64.tac-hb00.tac.epc.mnc999.mcc999.epc.3gppnetwork.org
```

ECI LTE 4G ECI

3. RAI

3G/2G ECI

ECI

```
rac<RAC>.lac-lb<LB>.lac-hb<HB>.lac.raimnc<MNC>.mcc<MCC>.  
<dns_suffix>
```

□□□

```
# RAC: 0x0A10 □□□□  
# LAC: 0x12344660 □□□□  
rac0a.lac-lb34.lac-hb12.lac.raimnc999.mcc999.epc.3gppnetwork.org
```

□□□□ 3G/2G UMTS/GPRS □□

4. SAI□□□□□□□□

3G □□□□

□□□

```
sac<SAC>.lac-lb<LB>.lac-hb<HB>.lac.saimnc<MNC>.mcc<MCC>.  
<dns_suffix>
```

□□□

```
# SAC: 0x0001  
# LAC: 0x1234  
sac0001.lac-lb34.lac-hb12.lac.saimnc999.mcc999.epc.3gppnetwork.org
```

□□□□ 3G UMTS □□□□

5. CGI□□□□□□□□

2G □□□

□□□

```
ci<CI>.lac-lb<LB>.lac-hb<HB>.lac.cgi.mnc<MNC>.mcc<MCC>.  
<dns_suffix>
```

例

```
# CI: 0x5678  
# LAC: 0x1234  
ci5678.lac-lb34.lac-hb12.lac.cgi.mnc999.mcc999.epc.3gppnetwork.org
```

例 2G GSM

DNS

NAPTR

DNS UPF IP NAPTR

```
eci-1a2b3c.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org.  
IN NAPTR 10 50 "a" "x-3gpp-upf:x-s5-gtp:x-s8-gtp" ""  
upf1.epc.mnc999.mcc999.3gppnetwork.org.  
  
upf1.epc.mnc999.mcc999.3gppnetwork.org.  
IN A 10.100.1.21
```

PGW-C

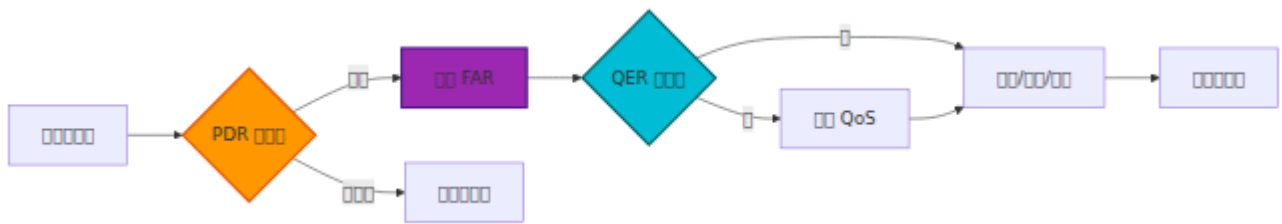
1. NAPTR UPF IP
2. DNS
- 3.

例

DNS [10.100.1.21, 10.100.5.99]

10.100.1.21
upf_selection

□□□□□□□□



□□

1. □□□□□□

□□□ □□□□□□□□□□ UPF

DNS □□□

```
# □□□□□□
eci-aaa.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org → UPF-□□□
□10.1.1.21□

# □□□□□□
eci-bbb.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org → UPF-□□
□10.2.1.21□

# □□□□□□
eci-ccc.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org → UPF-□□□
□10.3.1.21□
```

□□□ □□□□□□□□□□ UPF□□□□□□□□□□

2. □□□□□

□□□ □□□□□□□□ MEC□□□□□□□□□□UPF

DNS □□□

```
# □□□□□□□□□□ UPF
eci-*.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org → □□□□ UPF
```


DNS 配置

3. DNS 配置

DNS 配置在 UPF 中完成

DNS 配置在 PGW-C 中完成

DNS 配置

DNS 配置

配置

- 配置“DNS UPF 配置:nxdomain”
- 配置

配置

1. DNS 配置
2. DNS 配置 ID
3. GTP-C 配置 ULI

配置

```
# 配置 DNS 配置
dig eci-1a2b3c.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org NAPTR

# 配置 PGW-C 配置 DNS 配置
grep "DNS UPF selection: querying" /var/log/pgw_c.log

# 配置 ULI
# 配置 "uli" 配置
```

DNS 配置 UPF

配置

- DNS 配置 `upf_selection` 配置 UPF
- 配置

- DNS PFCP 10.99.1.50 UPF
- DNS PFCP 10.100.1.21, 10.100.1.22

10.99.1.50

```
DNS 10.99.1.50
upf_selection: [10.100.1.21, 10.100.1.22]
```

10.99.1.50

- DNS PFCP
- 10.100.1.21
- 10.100.1.22

10.99.1.50

1. upf_selection 10.100.1.21, 10.100.1.22

```
upf_selection: %{
  fallback_pool: [
    %{remote_ip_address: "10.99.1.50", remote_port: 8805, weight:
100}
  ]
}
```

2. DNS 10.99.1.50 UPF IP

3. 10.100.1.21, 10.100.1.22 MEC/10.99.1.50

10.99.1.50

10.99.1.50

- “DNS UPF 10.99.1.50”
- 10.100.1.21

10.99.1.50

```
upf_selection: %{\n  dns_timeout_ms: 10000 # DNS timeout 10 s\n}
```

[[DNS]]

[[[

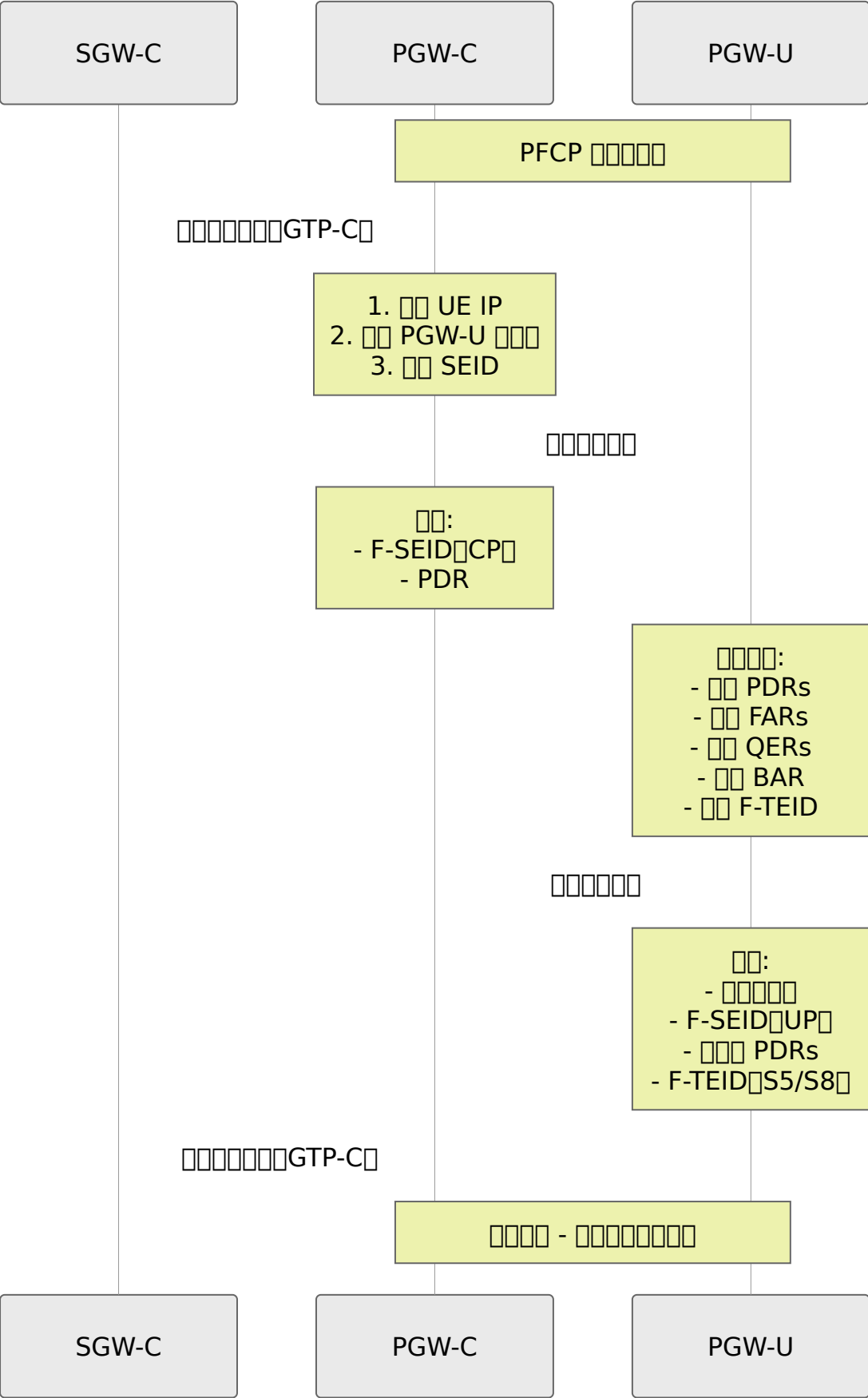
```
# DNS success rate\nrate(upf_selection_dns_success_total[5m]) /\nrate(upf_selection_dns_attempts_total[5m])\n\n# DNS duration\nhistogram_quantile(0.95,\nrate(upf_selection_dns_duration_seconds_bucket[5m]))\n\n# DNS fallback rate\nrate(upf_selection_fallback_used_total[5m])
```

[[[

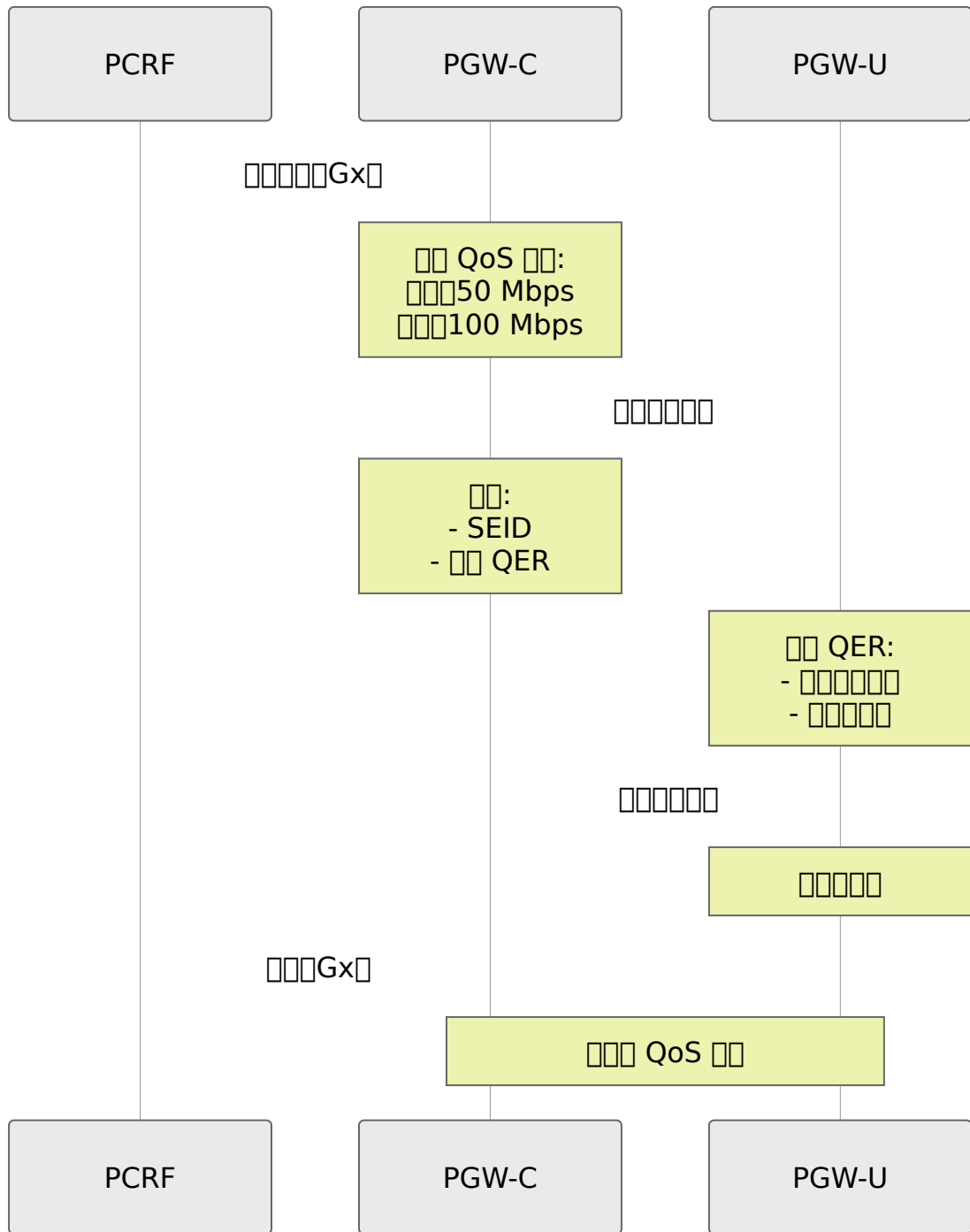
```
[debug] DNS UPF selection: querying eci-1a2b3c.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org\n[debug] DNS UPF selection: got 2 candidates from DNS\n[info] DNS UPF selection: selected 10.100.1.21
```

□□□

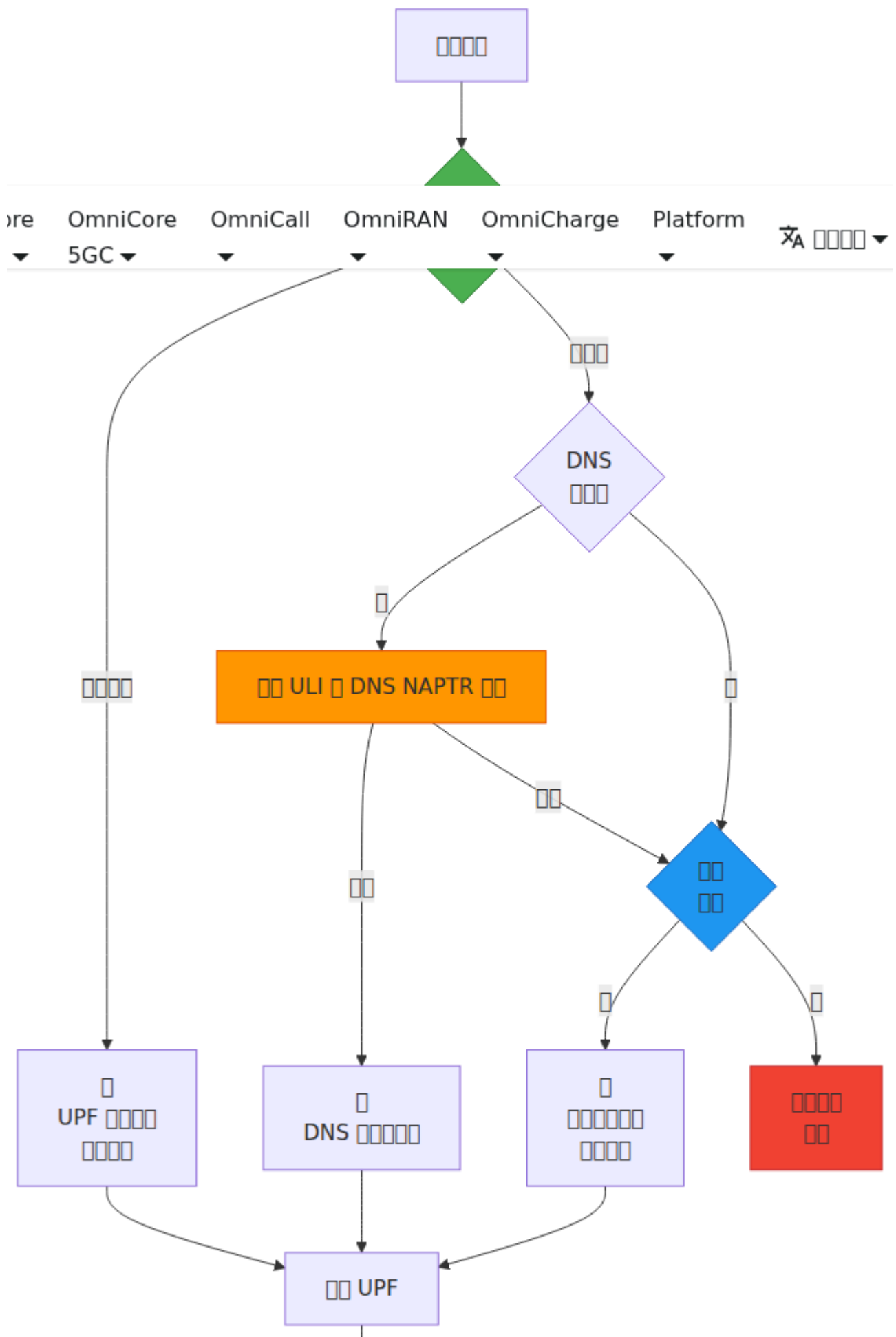
□□□□□□□

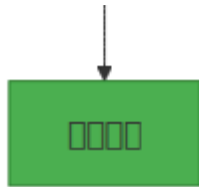


□□□□□



□□□□□□





PGW-U

PGW-U

1. PGW-U

PGW-U

- PGW-U “PCFIP PGW-U”
- PGW-U PGW-U

PGW-U

- PGW-U PGW-U
- PGW-U PGW-U
- PGW-U UDP PGW-U 8805
- PGW-U `remote_ip_address` PGW-U

PGW-U

```
# PGW-U
ping <pgw_u_ip_address>

# PGW-U UDP PGW-U
nc -u -v <pgw_u_ip_address> 8805

# PGW-U
iptables -L -n | grep 8805
```

2. PGW-U

PGW-U

- SEID 註冊
- PGW-U 與 PGW-C 間

配置

- 配置 PFCP 相關參數
- PGW-C 與 PGW-U 間

配置 PFCP

配置

```
# PFCP 配置
pfcpeer_associated{peer="PGW-U Primary"} 1

# PFCP 配置
seid_registry_count 150

# PFCP 配置
rate(sxb_inbound_messages_total[5m])

# PFCP 配置
rate(sxb_inbound_errors_total[5m])

# 配置
pfcpeer_consecutive_heartbeat_failures{peer="PGW-U Primary"} 0
```

配置

```
# PFCPPeerDown
- alert: PFCPPeerAssociationDown
  expr: pfcpeer_associated == 0
  for: 1m
  annotations:
    summary: "PFCP {{ $labels.peer }} 断连"

# PFCPSessionEstablishmentFailureHigh
- alert: PFCPSessionEstablishmentFailureHigh
  expr:
rate(sxb_inbound_errors_total{message_type="session_establishment_res
[5m]) > 0.1
  for: 5m
  annotations:
    summary: "PFCP 会话建立失败率高"
```

Web UI - PFCP 配置

OmniPGW 配置 Web UI 配置 PFCP/Sxb 配置

UPF/PFCP 配置

访问 `http://<omnipgw-ip>:<web-port>/upf_status`

PGW-U PFCP

1.

- PFCP
-
- 2

2.

-
- **IP** - PGW-U IP
-
- **ID** - PFCP
-
-
-
- **UP** - PGW-U

3.

- 配置参数
- UP 配置
- 配置
- 配置

PFCP 配置

配置 `http://<omnipgw-ip>:<web-port>/pfcpsessions`

配置 OmniPGW 的 PGW-U 配置 PFCP 配置

配置

1. 配置

- 配置 PFCP 配置
- 配置

2. 配置 PFCP 配置

- 配置 - 配置
- 配置 ID - 配置

- **IMSI** - IMSI - IMSI
- IMSI - IMSI

3. IMSI - IMSI

- IMSI PFCP IMSI
- PDRs/FARs/QERs/BARs/IMSI
- F-SEIDs/IMSI
- PGW-U IMSI

IMSI

IMSI **PFCP** IMSI

1. IMSI UPF IMSI
2. IMSI IMSI "IMSI"
3. IMSI IMSI = 0
4. IMSI IMSI "IMSI"
 - IMSI IP IMSI
 - IMSI IMSI
 - IMSI IMSI UDP 8805

IMSI IMSI

1. IMSI IMSI
2. IMSI PGW IMSI - IMSI
3. IMSI PFCP IMSI - PFCP IMSI
4. IMSI PFCP IMSI
 - IMSI UPF IMSI - IMSI
 - IMSI PFCP IMSI
5. IMSI PFCP IMSI
 - IMSI PDRs/FARs IMSI
 - IMSI PGW-U IMSI

IMSI IMSI

1. 配置 PGW-U 节点
2. 配置 PCF 节点
3. 配置策略和计费控制
4. 配置策略和计费控制

配置策略和计费控制

- 配置 UPF 节点
- 配置“策略和计费控制”节点
- 配置策略和计费控制
- 配置 Web UI 策略和计费控制

配置

- 配置 - 配置 SSH
- 配置 - 配置策略和计费控制/节点
- 配置策略和计费控制 - 配置策略和计费控制
- 配置策略和计费控制 - 配置策略和计费控制 PDRs/FARs/QERs
- 配置策略和计费控制 - 配置策略和计费控制

配置策略和计费控制

配置

- 配置 - UPF 配置策略和计费控制 PCF 节点
- 配置 - PDN 配置策略和计费控制

配置策略和计费控制

- **Diameter Gx** 节点 - 配置 PCF QoS 节点 PCC 节点
- **Diameter Gy** 节点 - 配置 URR 配置策略和计费控制
- **配置 CDR** 节点 - 配置 PCF 配置策略和计费控制 CDR
- 配置策略和计费控制 - PCF 配置策略和计费控制 UPF 节点

□□□□

- **S5/S8** □□ - □□□□□□
 - **UE IP** □□ - □□ PFCP □□ UE □□□□
-

□□□□□□

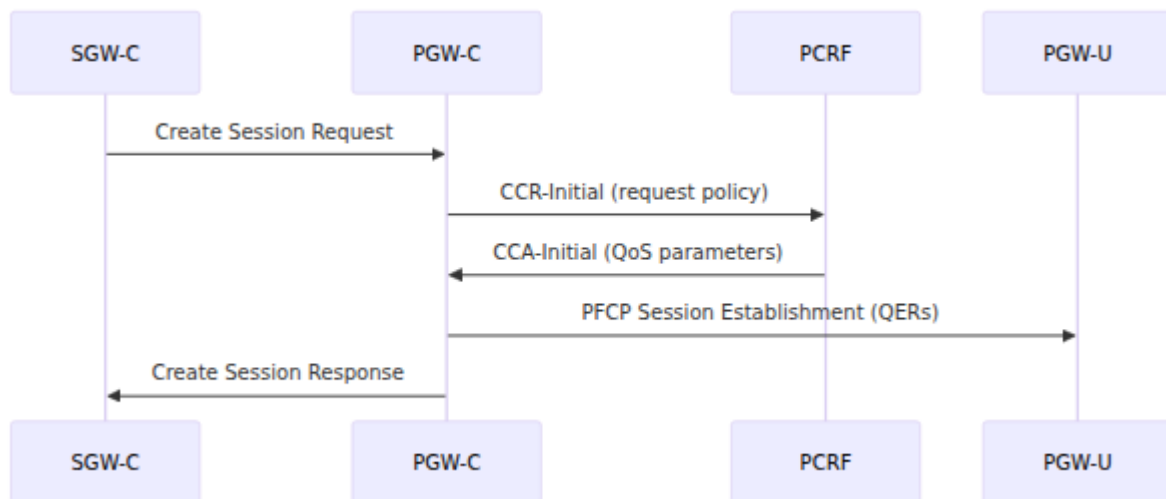
QoS

QoS

PGW-C QoS

- **Gx (Diameter)** - PCRF QoS
- **S5/S8 (GTP-C)** - SGW-C
- **Sxb (PFCP)** - QoS PGW-U

QoS



QoS

- UE: UE PDR/FAR/QER/BAR AMBR
- EBI: EBI (EPS ID) PDR/FAR QER
- **QER (QoS)**: MBR/GBR
- PDN: PDN
- PCRF: PCRF QoS

□□

□□□□: □□ **QoS** □□

□□ QoS □□□□ Diameter Gx □□□□□□□□ PCRF□□□□ PCRF □□□□□□□□□□□□□□ OmniHSS□□

□□□□ `config/runtime.exs` □□□□ **PCRF** □□□

```
config :pgw_c,
  diameter: %{
    listen_ip: "0.0.0.0",
    host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org",
    realm: "epc.mnc999.mcc999.3gppnetwork.org",
    peer_list: [
      %{
        host: "pcrf.epc.mnc999.mcc999.3gppnetwork.org",
        realm: "epc.mnc999.mcc999.3gppnetwork.org",
        ip: "192.168.1.100",
        initiate_connection: true
      }
    ]
  }
}
```

QoS □□□□□□□□□□□□□□ **PCRF** □□□□□□□□□□ **PGW-C** □□□□□□

□□□□□□□□

□□□□□□□□

□□□□□□ PDN □□□□□□□□□□



Create Session Request

AllocateIP

UE IP assigned

RequestPolicy

CCR-Initial sent to PCRF

CreateBearer

CCA-Initial received
with QoS

ProgramUPF

PFCP Session
Establishment

Active

Delete Session Request



□□□□:

1. SGW-C □□ Create Session Request
2. PGW-C □□□□□□□□ UE IP □□
3. PGW-C □□ CCR-Initial □ PCRF□□□ IMSI□APN□IP □□
4. PCRF □□ CCA-Initial□□□ QoS □□□
 - Default-EPS-Bearer-QoS (QCI, ARP)
 - QoS-Information (AMBR □□)

5. PGW-C 消息交互

- 消息 ID: 消息 PDR=1 消息 PDR=2 消息 FAR=1 消息 FAR=2 消息 QER=1 消息 BAR=1
- QER 消息 QoS 消息 MBR

6. PGW-C 向 PGW-U 发送 PFCP Session Establishment Request

7. PGW-C 向 SGW-C 发送 Create Session Response

消息内容:

- 本地 PDN 地址
- 本地 QCI 5 本地 QCI 9 本地 GBR
- EBI 地址
- 本地地址

消息内容

消息内容 PCRF 消息内容

消息: 消息 PCRF 向 Re-Auth Request (RAR) 消息 Charging-Rule-Install

消息:

1. PCRF 向 RAR 消息 Charging-Rule-Definition 消息

- Charging-Rule-Name (消息)
- Flow-Information (消息)
- QoS-Information (QCI, MBR, GBR, ARP)
- Precedence (消息)

2. PGW-C 向 PGW-U 发送 PFCP 消息

- 本地 Flow-Information 消息 → 本地 PDR 消息 SDF 消息
- QoS-Information → 本地 QER 消息 MBR/GBR 消息
- Flow-Description → IP 5-消息

3. PGW-C 向 PGW-U 发送 PFCP Session Modification Request 消息 PDRs/FARs/QERs

4. PGW-C 向 SGW-C 发送 Create Bearer Request

5. SGW-C 向 SGW-U 发送 Create Bearer Response 消息

消息 **Charging-Rule-Definition:**

```
Charging-Rule-Name: "video_streaming"
Flow-Information:
  - Flow-Description: "permit in ip from any to 10.0.0.1 5000-6000"
    Flow-Direction: 1 (downlink)
QoS-Information:
  QoS-Class-Identifier: 7
  Max-Requested-Bandwidth-UL: 5000000 (5 Mbps)
  Max-Requested-Bandwidth-DL: 10000000 (10 Mbps)
  Guaranteed-Bitrate-UL: 1000000 (1 Mbps)
  Guaranteed-Bitrate-DL: 2000000 (2 Mbps)
Precedence: 100
Flow-Status: 2 (ENABLED)
```

□□□□

□□ QoS □□□□□□□□□□

- **Gx RAR**□□□□□ Charging-Rule-Definition
- **PFCP Session Modification**□□□□□□ QERs□□□□□□□□ FARs□□□□□□□ PDRs□□□□□□□□□□

□□□□

□□:

- **Delete Session Request**□SGW □□□ - □□□□□□□□□□□□
- **Re-Auth Request with Charging-Rule-Remove**□PCRF □□□ - □□□□□□□

□□□□:

1. □□□□□□□□□□□□
2. □□□□□ PDRs/FARs/QERs
3. □ SGW-C □□ Delete Bearer Request□□□□□ PCRF □□□
4. □□ PFCP Session Modification□□□□□□□□ Session Deletion□□□□□□□□□□

QoS

QCI (QoS Class Identifier)

PCRF → Gx → **QoS-Class-Identifier** → AVP

QCI:

- **QCI 1:** VoLTE (GBR, 100ms delay)
- **QCI 2:** VoWiFi (GBR, 150ms delay)
- **QCI 3:** RTT (GBR, 50ms delay)
- **QCI 4:** VoWiFi (GBR, 300ms delay)
- **QCI 5:** IMS Signaling (GBR, 100ms delay) - Guaranteed Bit Rate
- **QCI 6:** TCP (GBR, 300ms delay)
- **QCI 7:** VoWiFi (GBR, 100ms delay)
- **QCI 8:** TCP (GBR, 300ms delay) YouTube (GBR, 300ms delay)
- **QCI 9:** VoWiFi (GBR, 300ms delay)

QCI:

- QCI → PCRF → QoS IE → SGW-C →
- PGW-C → QCI → QERs → MBR/GBR →
- QCI →
- QCI →

ARP (Allocation-Retention-Priority)

PCRF → **Allocation-Retention-Priority** → AVP

ARP:

- ARP: 1 to 15
- ARP: 0 = , 1 =
- **?** ARP:

- 0 = 00000000
- 1 = 11111111

000:

- 00000: 1
- 0000: 00 (0)
- 00000: 00 (1)

00000:

- ARP → SGW-C → 00000000 eNodeB
- **PGW-C** 00000 - 00000 eNodeB 0000000000
- 000000000000000000000000
- 00000000000000 10000000000000

MBR (000000)

00: PCRF 00 Max-Requested-Bandwidth-UL 0 Max-Requested-Bandwidth-DL
AVPs

00: 000000000000 kbps: bytes / 1000

0000: 000000000000

0000:

- PGW-C 00 QER000 mbr: %Bitrate{ul: kbps_ul, dl: kbps_dl}
- QER 00 PFCP 000 PGW-U
- **PGW-U** 0000000000000000
- 00 MBR 00000000

00:

Max-Requested-Bandwidth-UL: 5000000 (5 Mbps)
Max-Requested-Bandwidth-DL: 10000000 (10 Mbps)

- QER `qer` mbr: {ul: 5000, dl: 10000} kbps
- PGW-U `pgw-u` 5 Mbps `pgw-u`
- PGW-U `pgw-u` 10 Mbps `pgw-u`

GBR (Guaranteed Bit Rate)

`qer`: PCRF `qer` `Guaranteed-Bitrate-UL` `Guaranteed-Bitrate-DL` AVPs

`qer`: `qer` kbps

`qer`: `qer` GBR `qer`

`qer`:

- `qer` Charging-Rule-Definition `qer` GBR `qer` **GBR** `qer`
- PGW-U `qer` QER `qer`
- `qer` eNodeB `qer`
- GBR `qer` - `qer`

`qer`:

Guaranteed-Bitrate-UL: 1000000 (1 Mbps)
Guaranteed-Bitrate-DL: 2000000 (2 Mbps)

- QER `qer` gbr: {ul: 1000, dl: 2000} kbps
- `qer` 1 Mbps `qer` 2 Mbps `qer`
- `qer` VoIP `qer`

`qer`:

- GBR `qer`
- `qer` GBR `qer`
- `qer` GBR `qer`

AMBR (Aggregate-Max-Bitrate-Rate)

PCRF sends **APN-Aggregate-Max-Bitrate-UL** and **APN-Aggregate-Max-Bitrate-DL** AVPs

PCRF sends **GBR** and **APN** AVPs

PCRF:

- AMBR is calculated GBR
- Create Session Response from SGW-C
- eNodeB/SGW
- PGW-C sends AMBR to SGW-C

PCRF:

APN-Aggregate-Max-Bitrate-UL: 50000000 (50 Mbps)
APN-Aggregate-Max-Bitrate-DL: 100000000 (100 Mbps)

- GBR is calculated 50 Mbps / 100 Mbps
- MBR
- AMBR is UE/APN

PCRF:

- HSS/PCRF
- 10 Mbps / 100 Mbps
- GBR

(Gx) **(PFCP)**

PCRF sends **Flow-Status** AVP and **Charging-Rule-Definition**

Flow-Status (Gx)	Gate-Status (PFCP QER)	備考
0 = ENABLED-UPLINK	ul: OPEN, dl: CLOSED	アップリンクのみ許可
1 = ENABLED-DOWNLINK	ul: CLOSED, dl: OPEN	ダウンリンクのみ許可
2 = ENABLED	ul: OPEN, dl: OPEN	アップリンク・ダウンリンク両方許可
3 = DISABLED	ul: CLOSED, dl: CLOSED	アップリンク・ダウンリンク両方禁止
4 = REMOVED	ul: CLOSED, dl: CLOSED	アップリンク・ダウンリンク両方禁止

備考:

- **DISABLED:** アップリンク・ダウンリンク両方禁止
- **ENABLED-UPLINK:** アップリンクのみ許可
- **ENABLED-DOWNLINK:** ダウンリンクのみ許可
- **ENABLED:** アップリンク・ダウンリンク両方許可

セッション登録

Prometheus 監視

◆◆◆監視:

```
session_registry_count      # セッション (IMSI, EBI 別)
address_registry_count      # ユーザ IP
charging_id_registry_count  # 課金ID
```

Gx 監視:

```

gx_inbound_messages_total{message_type="gx_RAR"}      # [] PCRF []
[]
gx_outbound_messages_total{message_type="gx_CCR"}     # [] PCRF []
[]
gx_outbound_transaction_duration_bucket              # [] PCRF []

```

PCFP :

```

sxb_outbound_messages_total{message_type="pfcg_session_establishment_
sxb_outbound_messages_total{message_type="pfcg_session_modification_r
sxb_outbound_transaction_duration_bucket

```

:

```

s5s8_inbound_messages_total{message_type="create_session_request"}
# []
s5s8_outbound_messages_total{message_type="create_bearer_request"}
# []

```

Web UI

PGW (/pgw_sessions):

- IMSI/IP []MSISDN [] APN []
- []
- [] QoS [] (QCI, MBR, GBR, AMBR)
- [] (2)

Diameter (/diameter):

- PCRF []
- Gx []
- []/[]

(/logs):

- []

- "Credit Control" CCR/CCA
- "Re-Auth" RAR
- "PFCP"

```
[debug] Sending Credit Control Request: ... # CCR  PCRF
[debug] Handling Credit Control Answer: ... # CCA  PCRF
      QoS
[debug] Handling Re-Auth Request # RAR  PCRF
      QoS
[debug] Sending Session Establishment Request # PFCP  PGW-
      QERs
[debug] Sending Session Modification Request # PFCP  PGW-
      QERs
```

QoS

1. Web UI → **PGW**
2. IMSI 999000123456789
- 3.
4. **qer_map**

```
qer_id: 1
gate_status: {ul: OPEN, dl: OPEN}
mbr: {ul: 50000, dl: 100000} # kbps
gbr: {ul: 10000, dl: 20000} # kbps nil GBR
```

5. PCRF

QoS

: QoS

11

1. **PCRf** **:**

- Web UI → **Diameter**
- PCRF state = "connected"
- Diameter

2. **CCR/CCA** **□□**:

- Web UI → **Logs**
- "Credit Control Answer"
- CCA **QoS-Information** AVP
- CCA **Result-Code** 2001 = SUCCESS

3. **PFCP** **□□**:

- "PCF Session Establishment Request"
- QER
- PGW-U PCF

4. ☐ **PCRf** ☐

- 网络 PCRF 鉴权
- 网络 APN 鉴权
- 网络 PCRF 鉴权

Prometheus ☐☐:

```
# 5S8 inbound messages / min
rate(s5s8_inbound_messages_total{message_type="create_session_request"}[5m])

# 5S8 outbound messages / min
rate(s5s8_outbound_messages_total{message_type="create_bearer_request"}[5m])

# PCRF RAR messages / min
rate(gx_inbound_messages_total{message_type="gx_RAR"}[5m])
```

Queries

Queries:

```
# UE IP address registry count
(address_registry_count / <configured_pool_size>) * 100

# Session registry count
session_registry_count

# PCRF P95 transaction duration
histogram_quantile(0.95, gx_outbound_transaction_duration_bucket)
```

Queries:

- `config/runtime.exs` `ue.subnet_map`
- TEID 32 40
-

Queries:

- IP - 80%
- PCRF -
-

□□□□

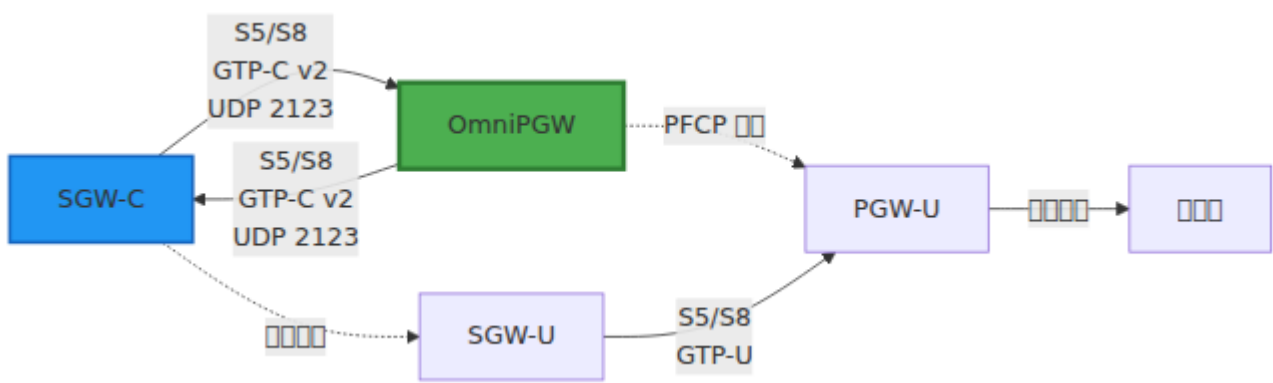
- □□□□ - PDN □□□□□□
- Diameter Gx □□ - PCRF □□□□□□□□
- PFCP □□ - □□□□□□
- □□□□ - □□□□
- □□□□ - □□□□□□□□

S5/S8

SGW-C GTP-C

OmniPGW Omnitouch

S5/S8 GTP-C v2 GPRS - OmniPGW SGW-C



GTP-C 2

- GTP-C v2 (3GPP TS 29.274)
- UDP
- 2123
-

TEID

TEID

- **TEID** - OmniPGW
- **TEID** - SGW-C

SGW-C → OmniPGW: TEID = OmniPGW TEID

OmniPGW → SGW-C: TEID = SGW-C TEID

```
# config/runtime.exs
config :pgw_c,
  s5s8: %{
    # S5/S8 IPv4
    local_ipv4_address: "10.0.0.20",

    # IPv6
    local_ipv6_address: nil,

    #
    local_port: 2123,

    # GTP-C 500
    # GTP-C
    request_timeout_ms: 500,

    # GTP-C 3
    # = request_timeout_ms * request_attempts
    # 500 * 3 = 15001.5
    request_attempts: 3
  }
```

S5/S8 GTP-C

□□□

- `request_timeout_ms` - □□□□□□□□□□□□□□□□500□□□
- `request_attempts` - □□□□□□□□□□□□3□

□□□□□□ `request_timeout_ms × request_attempts`

□□□□□ 500□□ × 3 □□□ = **1.5** □□□□□□

□□□□□

□□□□	□□ <code>request_timeout_ms</code>	□□□□□
□□□□<50□□□	200-300□□	600-900□□□3 □□□□
□□□50-150□□□	500□□□□□□	1.5□□3 □□□□
□□□□>150□□□	1000-2000□□	3-6□□3 □□□□
□□□/□□	2000-3000□□	6-9□□3 □□□□

□□ - □□□□□□

```
s5s8: %{  
  local_ipv4_address: "10.0.0.20",  
  request_timeout_ms: 1500, # □□□□ 1.5 □  
  request_attempts: 3      # □□□□□ 4.5 □  
}
```

□□□□□□□

- OmniPGW □□□□□ "□□□□□□□□"
- □□□□□ PCRF□□□□□□□□5012 UNABLE_TO_COMPLY□
- □□□□□□□□□□□□□□□□ Charging-Rule-Remove □□

□□□□□

□□□□□□

```
# SGW-C GTP-C
iptables -A INPUT -p udp --dport 2123 -s <sgw_network>/24 -j
ACCEPT

# SGW-C GTP-C
iptables -A OUTPUT -p udp --dport 2123 -d <sgw_network>/24 -j
ACCEPT
```

```
# SGW-C
ip route add <sgw_network>/24 via <gateway_ip> dev eth0
```

S5/S8 PDN GTP-C

SGW-C → OmniPGW

PDN

IE

IE 名称	类型	值
IMSI	字符串	310260123456789
MSISDN	字符串	14155551234
APN	字符串	internet
RAT 类型	枚举	EUTRAN
QoS 参数	字符串	QCI 9, ARP, S5/S8 F-TEID (SGW-U FQDN)
UE 标识	字符串	UE ID
ULI	字符串	TAI, ECGI
PLMN	字符串	MCC/MNC

数据

```
UE 标识
├─ IMSI: 310260123456789
├─ MSISDN: 14155551234
├─ APN: internet
├─ RAT 类型: EUTRAN (6)
├─ QoS 参数
│   ├─ EBI: 5
│   └─ QoS (QCI 9, ARP, S5/S8 F-TEID (SGW-U FQDN))
└─ ULI
    ├─ TAI: MCC 310, MNC 260, TAC 12345
    └─ ECGI: MCC 310, MNC 260, ECI 67890
```

网络名称

OmniPGW → SGW-C

网络名称

Parameters

IE	Field	Value
Protocol	Field	Protocol
Protocol	Field	Protocol
PDN	IP	UE IP Address UE IP Address
APN	Field	APN
PCO	Field	PCO

Parameters

Parameters

- Field: (16)
- PDN
 - IPv4: 100.64.1.42
- Field
 - EBI: 5
 - Field
 - S5/S8 F-TEID (PGW-U PFCP)
- APN: Public-1 (1)
- PCO
 - DNS: 8.8.8.8
 - DNS: 8.8.4.4
 - MTU: 1400

Parameters

SGW-C → OmniPGW

PDN

Parameters

IE 項目	項目
EBI	ネットワーク EPS 項目 ID
項目 EBI	ネットワーク項目項目

ネットワーク項目

項目 OmniPGW → SGW-C

項目 項目項目項目

ネットワーク項目

IE 項目	項目
項目	ネットワーク項目項目

ネットワーク項目

ネットワーク項目

項目 OmniPGW → SGW-C

項目 項目項目項目項目 PCRF 項目項目

ネットワーク項目

- PCRF 項目項目 PCC 項目項目項目項目
- OmniPGW 項目 SGW-C 項目項目

ネットワーク項目

項目 OmniPGW → SGW-C 項目 SGW-C → OmniPGW

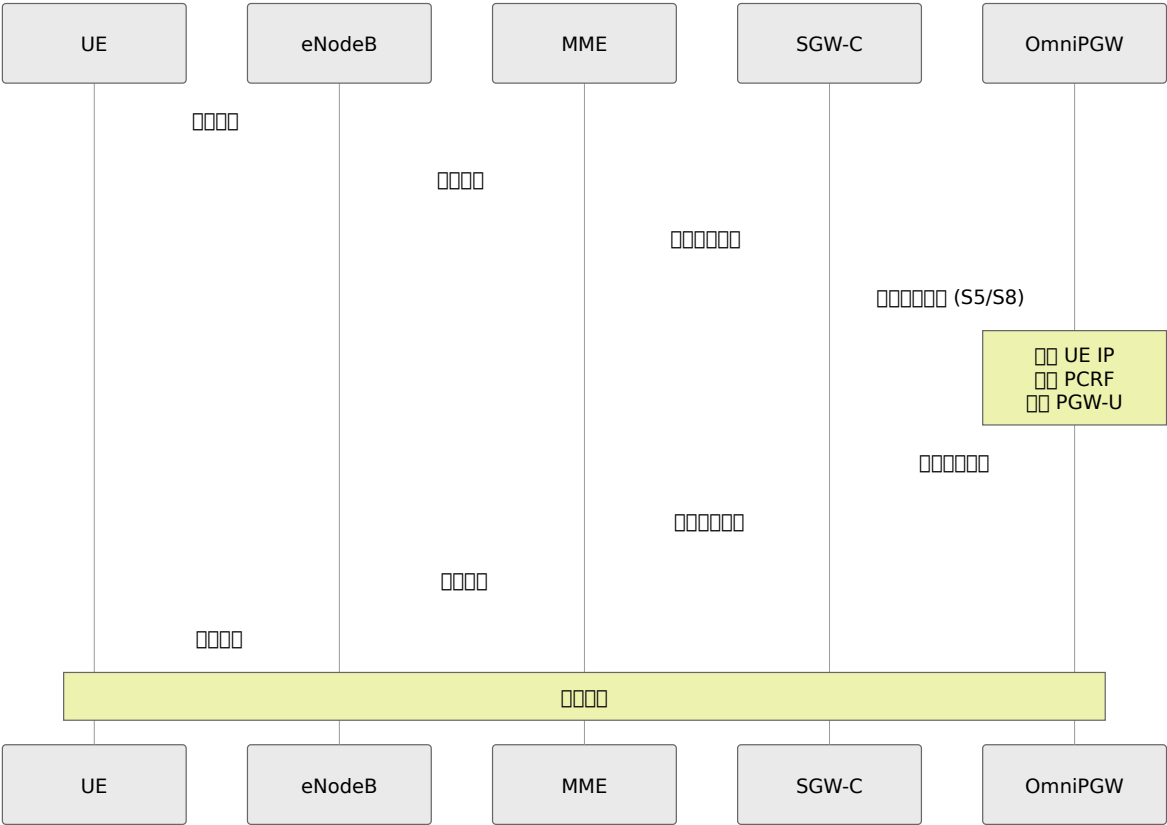
項目 項目項目項目

項目項目

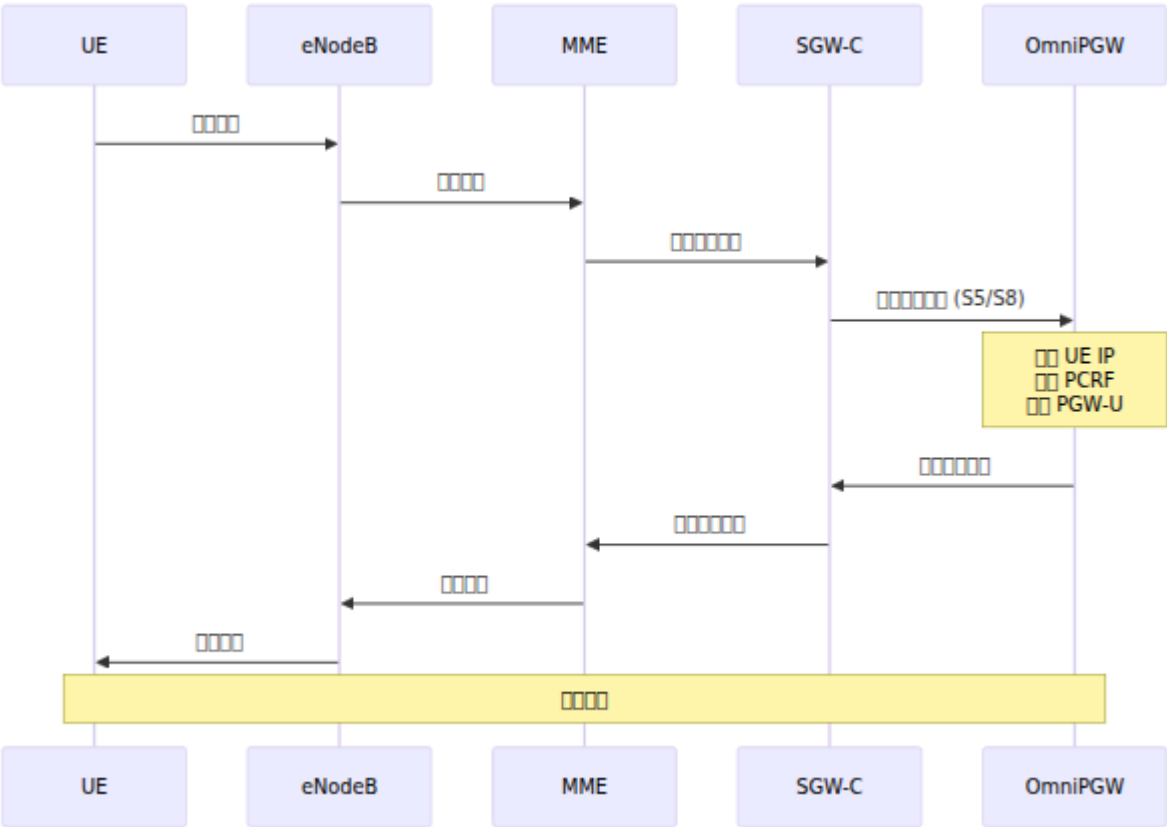
- **PGW** 通过 PCRF 实现策略控制
- **SGW** 实现路由转发

网络

网络



□□□□



□□□□

□□

□□	□□	□□
16	□□□□□□	□□□□

□□□□□□□□

□□	□□	□□□□
65	□□□□	PCRF □□□IMSI □□□□
66	□□□□□	IP □□□
93	□□□□□□	□□□ APN
94	TFT □□□□□□	□□□□□□□□

□□□□□□□□

□□	□□	□□□□
72	□□□□□□□□	PCRF/PGW-U □□
73	□□□□□□□□□□	□□□□

□□

S5/S8 □□

```
# □□□□□
s5s8_inbound_messages_total{message_type="create_session_request"}
s5s8_inbound_messages_total{message_type="delete_session_request"}

# □□□□□
s5s8_inbound_errors_total

# □□□□□□
s5s8_inbound_handling_duration_bucket

# □□ TEID
teid_registry_count
```

□□□□□

□□□□□

```
rate(s5s8_inbound_messages_total{message_type="create_session_request"}[5m])
```

□□□□

```
rate(s5s8_inbound_errors_total[5m])
```

□□□**p95**□□

```
histogram_quantile(0.95,

rate(s5s8_inbound_handling_duration_bucket{request_message_type="create_session_request"}[5m])
)
```

環境構築

環境構築 OmniPGW 構築

前提

- SGW-C 環境構築済み
- 環境構築済み
- SGW-C あり

手順

1. 環境構築
2. OmniPGW 環境構築 IP あり
3. 環境構築 UDP 2123
4. 環境構築 TEID あり

コマンド

```
# 環境構築 OmniPGW 環境構築
netstat -ulnp | grep 2123

# 環境構築
tcpdump -i any -n port 2123

# 環境構築
grep "local_ipv4_address" config/runtime.exs

# 環境構築
iptables -L -n | grep 2123
```

環境構築環境構築

前提

- 環境構築環境構築
- 環境構築

□□□□□

□□ 65□□□□□□□□

- PCRF □□□□□
- □□ HSS/SPR □□ IMSI

□□ 66□□□□□□

- IP □□□
- □□□curl http://pgw:9090/metrics | grep address_registry_count
- □□ IP □

□□ 72□□□□□□□□□□□□

- PCRF □□□ PGW-U □□
- □□ Gx □□
- □□ PFCP □□

□□□TEID □□

□□□

- □□□□□□□□□□
- □□□□

□□□

- TEID □□□□□□□□
- TEID □□□□□□

□□□□□

-   □□□□ TEID □□
 - □□ TEID □□□□□□□□
-

□□□□

□□□□

1. □□□□□□

- □ S5/S8 □□□□ VLAN
- □□□□□□□

2. **MTU** □□

- □□ MTU □□ GTP □
- □□ MTU□1500 □□□1464 □□ + 36 GTP□

3. □□

- □□ OmniPGW □□
- □ SGW-C □□□□ DNS □□□□□

□□

1. **UDP** □□□□□

- □□□□□□□□□□□□
- □□□□□□□□ 4-8 MB

2. □□□□

- □□□□□□□□□□
- □□ TEID □□□□□

□□

1. **IP** □□

- □□□□□□□ SGW-C IP □ GTP-C
- □□ iptables □□□ ACL

2. 詳細

- OmniPGW 詳細
 - 詳細 GTP-C 詳細
-

詳細

詳細

- 詳細 - S5/S8 詳細 IP 詳細
- 詳細 - PDN 詳細
- **UE IP** 詳細 - 詳細 IP 詳細
- **PCO** 詳細 - GTP-C 詳細 PCO 詳細

詳細

- **Gn/Gp** 詳細 - 詳細 2G/3G GGSN 詳細 GTP-C v1
- **PFCP** 詳細 - 詳細 S5/S8 詳細
- 詳細 **Gx** 詳細 - 詳細
- 詳細 **Gy** 詳細 - 詳細

詳細

- 詳細 - S5/S8 GTP-C 詳細
 - 詳細 **CDR** 詳細 - 詳細 GTP-C 詳細 CDR
-

詳細

OmniPGW S5/S8 詳細 - 詳細 *Omni*touch 詳細



11

```
graph TD; Start(( )) -- "Initial State" --> Creating[Creating]; Creating -- "State Vector 1" --> Active[Active]; Active -- "State Vector 2" --> Modifying[Modifying]; Modifying -- "State Vector 3" --> Active; Active -- "State Vector 4" --> Terminating[Terminating]; Terminating -- "State Vector 5" --> End((( ))) Creating -- "State Vector 6" --> End;
```

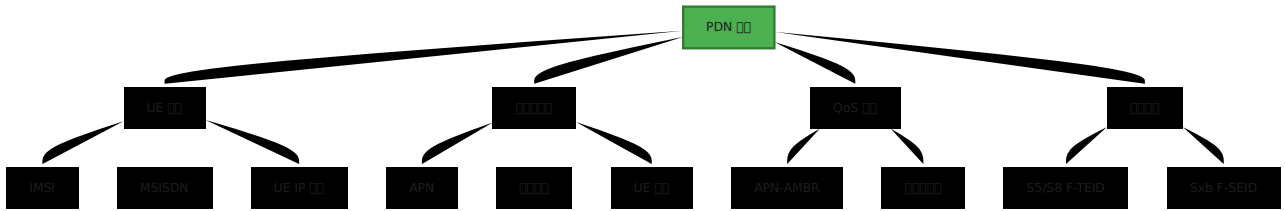

□□□□

□□□□□

□□□□□□□□□□□□□□□□

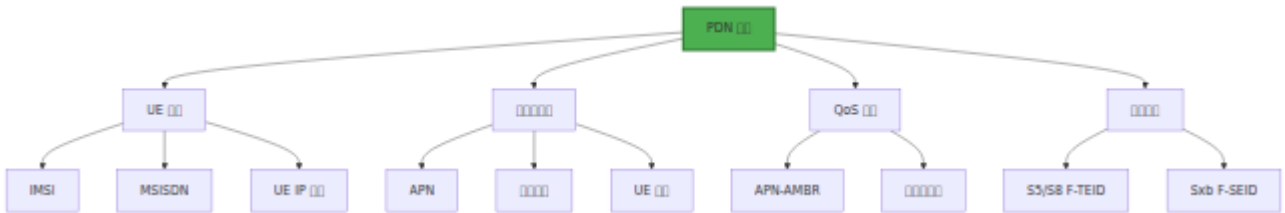
□□□	□□	□□
TEID	S5/S8 (GTP-C)	SGW-C □□□□□□□ ID
SEID	Sxb (PFCP)	PGW-U □□□□□□□ ID
Session-ID	Gx (Diameter)	PCRF □□□ Diameter □□
Charging-ID	□□	□□□□□/□□ ID

□□□□



□□□□

□□□□



□□

1. □□□□□□ (S5/S8)

□□□□□ S5/S8 □□□ GTP-C □□□□□□□□ GTP-C □□□□□□□□□□ S5/S8 □□□

□□□

- IMSI, MSISDN, IMEI
- APN (□□□“internet”)
- RAT □□ (EUTRAN)
- UE □□ (TAI, ECGI)
- □□□□ (QoS, F-TEID)

2. □□□□

- □ APN □□□□ UE IP
- □□□□ ID
- □□ Gx □□-ID
- □□ S5/S8 TEID
- □□ PGW-U □□□

3. □□□□ (Gx)

□ PCRF □□□□□

- □□ CCR-□□
- □□□□ QoS □ PCC □□□ CCA-□□

4. □□□□□□ (PFCP)

□□□□□□□□ PGW-U□

- □□□□□□□□
- □□ PDRs, FARs, QERs, BAR
- □□ S5/S8 □□□ F-TEID

5. □□ SGW-C

QoS Parameters

- UE IP
 - S5/S8 F-TEID (PGW-U)
 - PCO (DNS, P-CSCF, MTU)
 - QoS Class
-

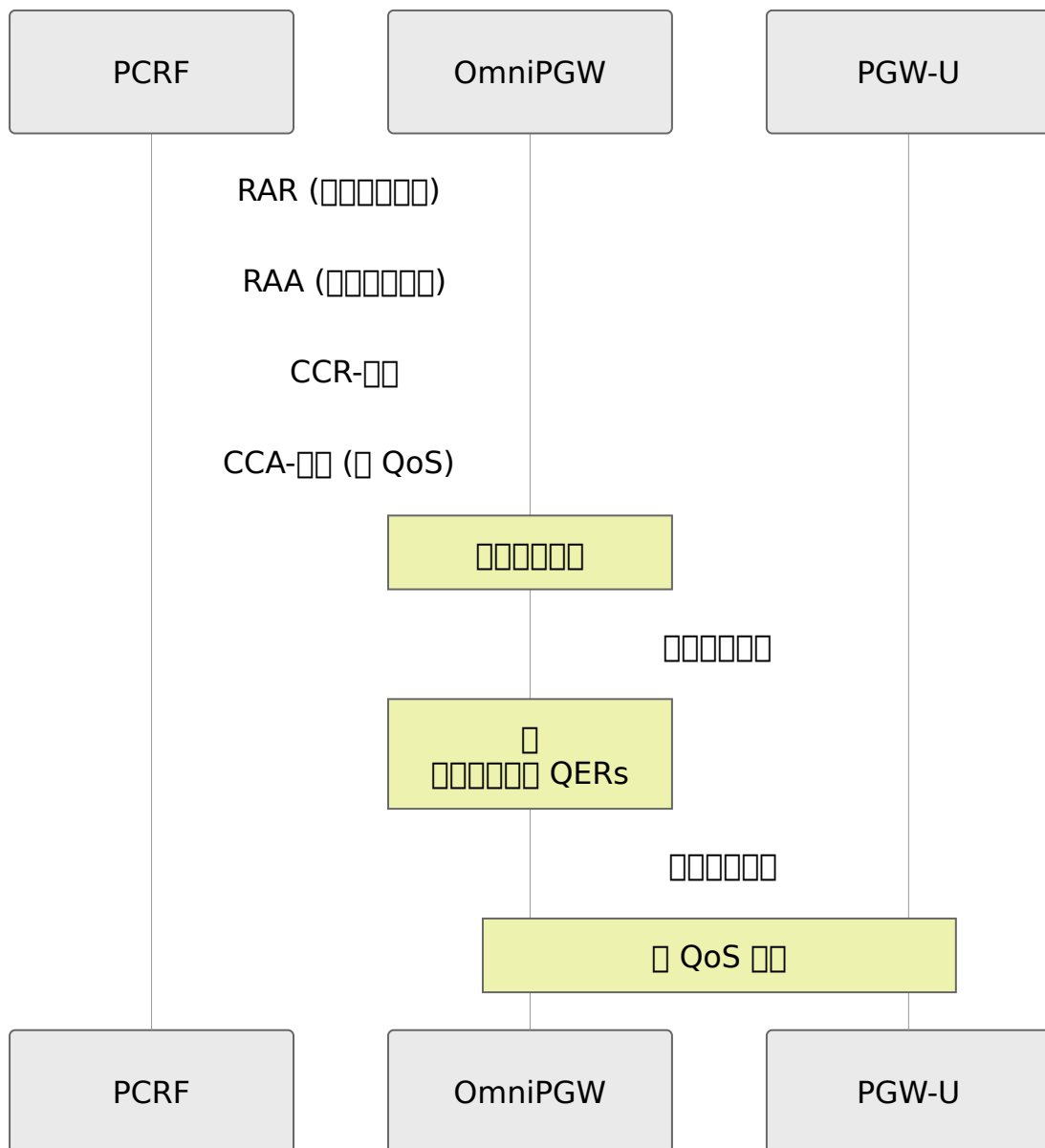
QoS Class

QoS

QoS Parameters

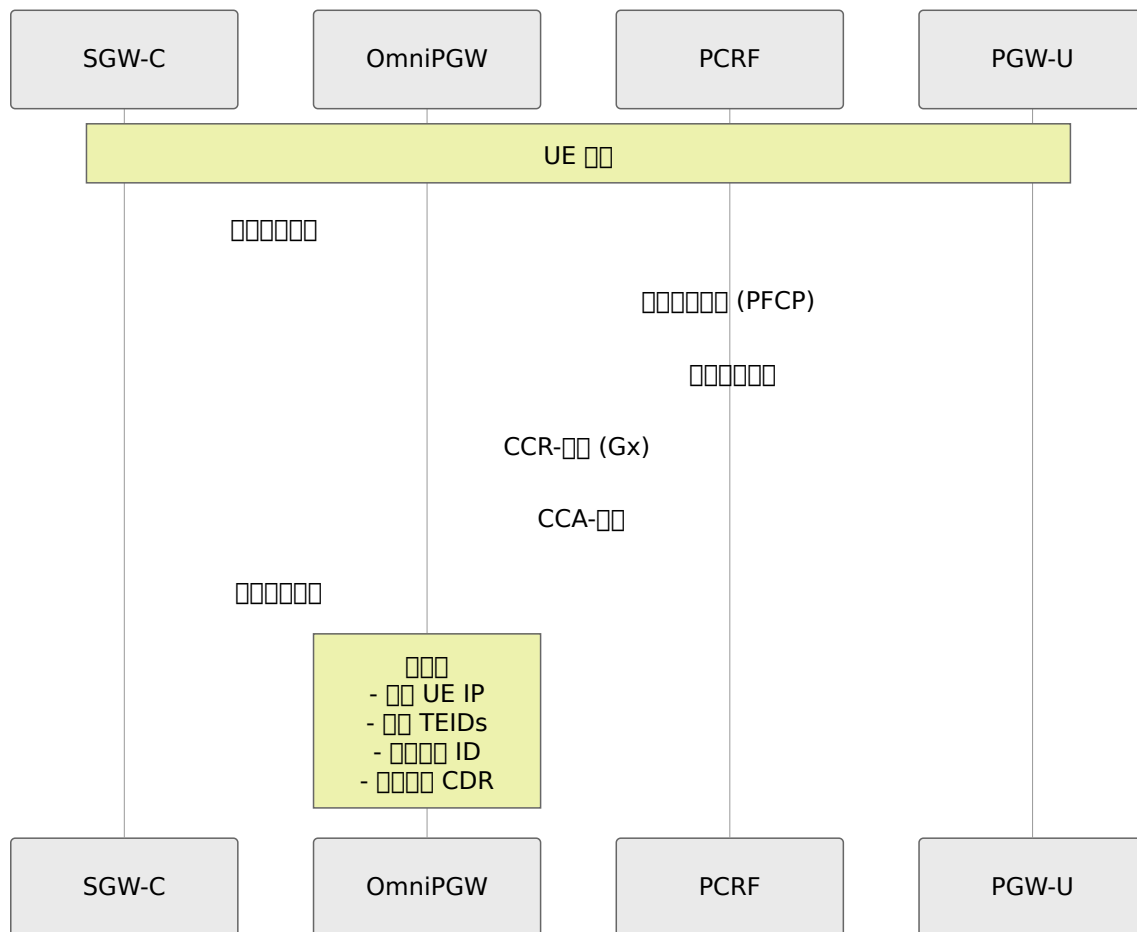
- **QoS** - PCRF QoS
- QoS - QoS/Service
- QoS - SGW
- QoS - PCRF QoS PCC

QoS □□□□



□□□□

□□□□



□□□□

□□□□□□

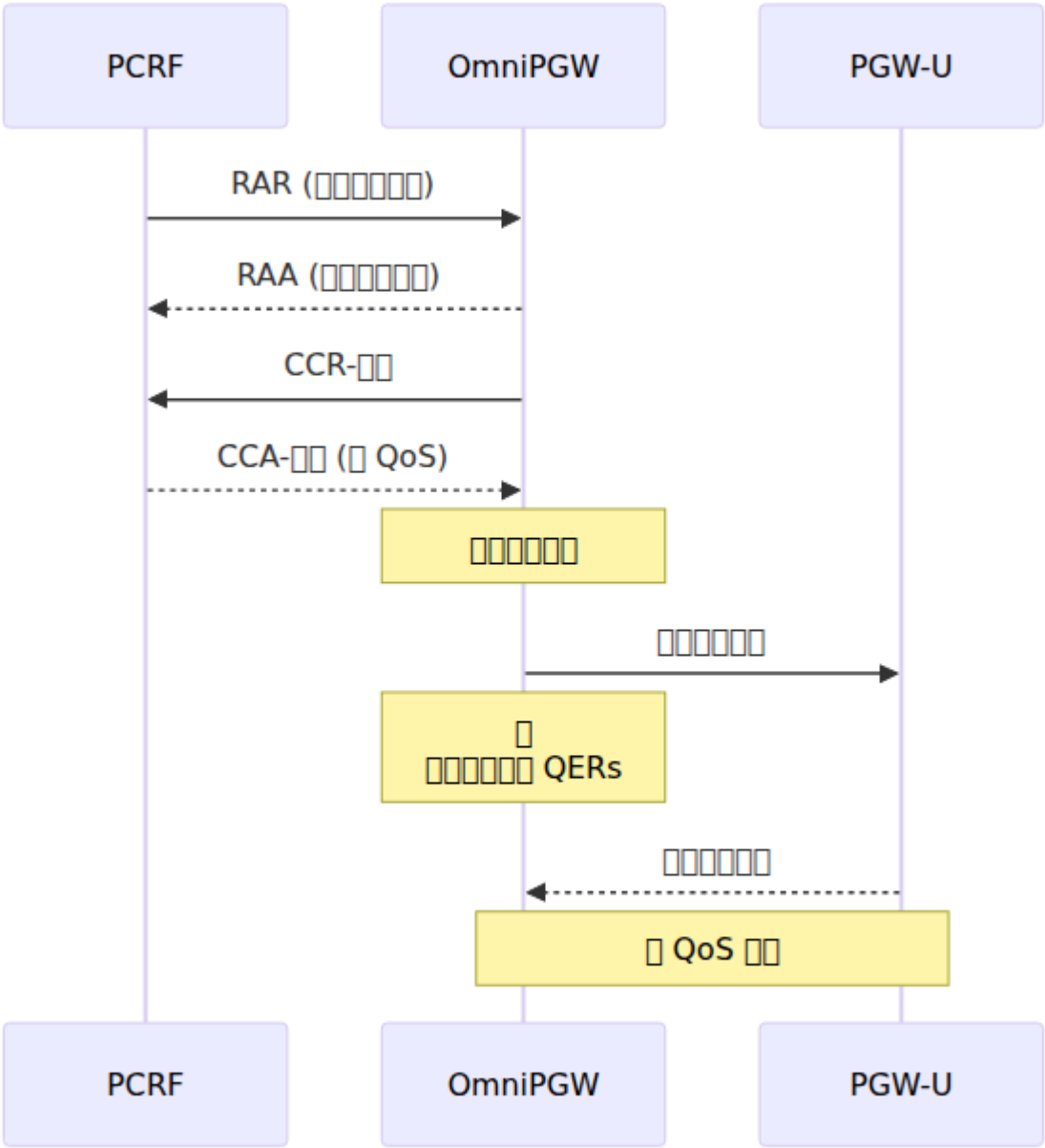
1. UE IP □□ → □□□□
2. TEID → □□□□□□□□
3. SEID → □□□□□□□□
4. □□-ID → □□□□□□□□
5. □□-ID → □□
6. □□□□□□

□□□□□□□□

- 3GPP CDR (3GPP) - 3GPP CDR

3GPP

3GPP



3GPP

3GPP

- TEID (S5/S8):
TEID 0x12345678 → □□ PID
- SEID (Sxb):
SEID 0xABCDEF → □□ PID
- □□-ID (Gx):
"pgw.example.com;123;456" → □□ PID
- UE IP:
100.64.1.42 → □□ PID
- IMSI + EBI:
"310260123456789" + EBI 5 → □□ PID

□□□□

□□□□□□

```
# □□□□□
teid_registry_count

# PFCP □□
seid_registry_count

# Gx □□
session_id_registry_count
```

Queries

```
# Session creation
rate(s5s8_inbound_messages_total{message_type="create_session_request"})

# Session deletion
rate(s5s8_inbound_messages_total{message_type="delete_session_request"})

# Session duration (p95)
histogram_quantile(0.95,

rate(s5s8_inbound_handling_duration_bucket{request_message_type="create_session_request",
[5m])
)
```

Tools

Network

Tools

1. **IP** - IP address
2. **PCRF** - PCRF (Policy and Charging Rules Function)
3. **PGW-U** - PGW-U (PDN Gateway - User Plane)
4. **PCRF** - PCRF (Policy and Charging Rules Function)

Tools

```
# IP
curl http://pgw:9090/metrics | grep address_registry_count

# PCRF
# PCRF Gx

# PGW-U
# PGW-U PFCP
```


設定/項目

項目

- 項目1
- 項目2
- 項目3

項目

- 項目1
- 項目2
- 項目3

項目

```
# 0mniPGW (項目)
```

```
# 項目
```

UE 項目

項目

- UE 項目
- 項目

項目

項目	項目	項目
項目	PCRF 項目 (IMSI 項目)	項目
項目	IP 項目	項目 IP 項目
項目	PCRF/PGW-U 項目	項目
項目	項目 APN	項目 APN 項目

□□□□

□□□□

□□□□□□□□

□□□□□□□10,000

□□□□□□□□~10KB RAM

□□□ RAM~100MB

Erlang VM □□□

- □□□□□262,144 (□□)
- □□□□□□□□□□□□

□□□□

□□□□□□□□

1. □□□□□□□□□□
2. □□□□□□□□□□□□□□
3. □□□□□□□□□□□□

□□□□

□□□□□

- □□□□□□□ (□□□□□□□□)
 - □□□□□□□□□□□□□□ (□□)
 - DNS/□□□□□□□□□□□□
-

□□□□□□

□□□□□□□□

□□□□ PDN □□□□□□□□

UE □□

- IMSI: "310260123456789" (□□□□)
- MSISDN: "14155551234" (□□□□)
- MEI/IMEI: □□□□

PDN □□□□

- APN: "internet" (□□□□)
- UE IP □□: 100.64.1.42 (□□□ IP)
- PDN □□: IPv4, IPv6 □ IPv4v6

□□□□□

- □□ ID: □□□□□□□□
- □□□□ EBI: EPS □□□□ (□□□ 5)

QoS □□

- APN-AMBR: □□□□□□□
 - □□□100 Mbps
 - □□□50 Mbps

□□□□□

- PDRs (□□□□□□□): □□□□□
- FARs (□□□□□□): □□/□□□□
- QERs (QoS □□□□): □□□□
- BAR (□□□□□□): □□□□

□□□□□□

- S5/S8 **接口**: **接口**/接口 TEIDs, SGW-C **接口**
- Sxb **接口**: **接口**/接口 SEIDs, PGW-U **接口**
- Gx **接口**: Diameter **接口**-ID, **接口**

Web UI - **接口**

OmniPGW **接口** **Web UI** **接口**

UE **接口**

接口: `http://<omnipgw-ip>:<web-port>/ue_search`

接口: **接口** UE **接口**

接口

1. **接口** **接口**

- **IMSI** (**接口**“310170123456789”)
- **MSISDN** (**接口**)

- **IP** 地址 (例如“100.64.1.42”)

2. 网络

- 网络地址
- 网络地址
- 网络地址

3. 网络地址 网络地址

a) 网络

- 网络地址
- IMSI, MSISDN, UE IP 地址
- APN, RAT 类型
- PGW TEID, SGW TEID

b) 网络地址 网络地址

- **TAC** (网络地址) - UE 网络地址
- **ID (ECI)** - E-UTRAN 网络地址
- **ECGI** - E-UTRAN 网络地址 (PLMN + ECI)
- **MCC/MNC** - 网络地址 / 网络地址

网络地址 网络地址 OpenCellID 网络地址

- 网络地址 (网络/网络)
- 网络 Google 网络地址
- UE 网络地址

网络地址 网络地址 网络地址

c) 网络 网络地址 QoS 网络

网络:

- EBI (EPS 网络地址)
- QCI (QoS 网络地址)
- 网络地址

- APN-AMBR (UL/DL)

QoS (QoS) 1

- EBI, QCI, 5QI
- MBR UL/DL (UL/DL)
- GBR UL/DL (UL/DL)

d) Gy (Gy 1)

- Gy 1 ID
- 1, 1
- 1

e) Gx (Gx 1)

- Gx 1 ID
- PCRF 1/1
- CC 1
- 1 (1 PCC 1)

f) 1

- 1
- 1/1/1

□□□

- □□□□□□□□
- □□□□□□
- □□□□□ IP □□
- □□□□□□

PGW □□□□

□□: `http://<omnipgw-ip>:<web-port>/pgw_sessions`

例: 無線インターネット PDN 例

例

1. 例

- 無線インターネット (2 例)
- 無線インターネット
- 無線 - 無線

2. 無線インターネット 例

- **IMSI** - 無線
- **UE IP** - 無線 IP 例
- **SGW TEID** - 無線 SGW 例 S5/S8 無線 ID
- **PGW TEID** - 無線 OmniPGW 例 S5/S8 無線 ID
- **APN** - 無線

3. 無線インターネット 例

- IMSI (無線“310260”)
- UE IP 例 (無線“100.64”)
- MSISDN / 無線

- APN □□

4. □□□□□□ □□□□□□□□□□□□□□□□

- IMSI, MSISDN, IMEI
- RAT (RAT, MCC/MNC)
- QoS (AMBR /)
- TEIDs
- ID
- ()


```
❏❏: http://<omnipgw-ip>:<web-port>/topology
```

00: 0000000000000000

000

1. 00000

- 0000000000
- 00 PGW-C (0000) 00
- 000 HSS (00000000) 000
- 000000000

2. 0000

- 0000 (+/-)
- 000000
- 00000000000
- 000000 (00 = 00, 00 = 00)

3. 0000

- 0000000000
- 0000
- 000000000

□□□

- □□□□□□□□□□
- □□□□□□
- □□□□□□
- □□□□□□□□

□□□□□□□□□□

□□: `http://<omnipgw-ip>:<web-port>/session_history`

□□: □□□□□□□□□□□□□□

□□□

1. □□□□

- □□□□□□□□ (□□□□, □□□□, □□□□, □)
- □□□□□□ (□□□□ / □□□□)
- □ IMSI, MSISDN, IP □□□□ TEID □□

2. □□□□

- 下載 CSV 檔案
- 註冊號碼
- 註冊號碼

3. 註冊號碼

- 註冊號碼
- 註冊號碼
- 註冊
- 註冊

註冊

- 註冊
- 註冊
- 註冊
- 註冊
- 註冊

註冊

註冊

1. 註冊
2. 註冊 IMSI 註冊 Web UI 註冊
3. 註冊 UE 註冊 IP 註冊
4. 註冊 QoS 註冊
5. 註冊

註冊

- 註冊
- 註冊
- 註冊 APN 註冊

註冊

- `ssh -p 22 root@<ip>`
- `ssh -p 22 root@<ip> SSH/IEx`
- `ssh -p 22 root@<ip> SGW PGW TEIDs`
- `ssh -p 22 root@<ip> PCRF AMBR`

Network Settings

- `ssh -p 22 root@<ip> (root@<ip>)`
- `ssh -p 22 root@<ip>`
- `ssh -p 22 root@<ip> (root@<ip> Mbps root@<ip> bps)`
- `ssh -p 22 root@<ip>`
- `ssh -p 22 root@<ip>`

OpenCellID

OmniPGW `ssh -p 22 root@<ip> OpenCellID` `ssh -p 22 root@<ip> UE` `ssh -p 22 root@<ip>`

UE

`ssh -p 22 root@<ip> UE` `ssh -p 22 root@<ip>`

- `ssh -p 22 root@<ip> (root@<ip>)`
- `ssh -p 22 root@<ip> Google`
- `ssh -p 22 root@<ip>`
- `ssh -p 22 root@<ip>`

UE

`ssh -p 22 root@<ip> http://<omnipgw-ip>:<web-port>/cell_towers` `ssh -p 22 root@<ip> "root@<ip>"`

UE

- `ssh -p 22 root@<ip> OpenCellID.org`

- 使用 SQLite
- 数据集 (约 10-15 万)
- 使用 Web 应用程序
- 使用 Python 进行数据分析和可视化

数据集 使用 Python 进行数据分析和可视化“数据集”数据集

数据集

数据集

- SQLite DB: `priv/cell_towers.db`
- CSV 文件 (约): `priv/data/cell_towers.csv.gz`
- 数据集 MCC, MNC, LAC, CellID 数据集

数据集

- 使用 OpenCellID.org 数据集 107 MB
- 数据集 10-15 万

数据集

- 数据集 (<1ms)
- 数据集
- 数据集 UE 数据集

数据集

数据集

UE 数据集

- 数据集
- 数据集 Google 数据集
- 数据集

使用 **Web UI**

- 電話番号 (市区, 市外, 携帯)
- 電話番号 - 電話番号 OpenCellID 番号
- 電話番号
- MCC, MNC, LAC, Cell ID 番号
- 電話番号
- 電話番号電話番号電話番号

電話番号

- 電話番号電話番号
- 電話番号
- 電話番号電話番号
- 電話番号電話番号

電話番号

OpenCellID 電話番号電話番号電話番号

電話番号

1. 番号 `http://<omnipgw-ip>:<web-port>/cell_towers`
2. 番号 “電話番号” 番号
3. 電話番号電話番号
4. 番号 10-15 電話番号電話番号/番号
5. 電話番号電話番号電話番号

電話番号 電話番号

番号 OpenCellID 電話番号電話番号電話番号電話番号電話番号電話番号

電話番号

電話番号

- 番号 OpenCellID.org 電話番号
- 電話番号 HTTPS 番号
- 電話番号 (番号 ~200 MB 電話番号)

- 00000000000000000000
- OpenCellID 00000000 - 0000000000
- 00 Web UI 0000000000000000

00000000

- 00 `priv/` 000000000000
- 0000000000 (~150 MB 000000)
- 000000000000 `priv/` 000/0000

000000

- 000000000000
- OpenCellID 0000000000000000
- 0000000000000000

000000

- 00000 JavaScript 00000000
- 00 Google 000000
- 0000000000



000000

- **PFCP** 00 - 00000000, PDRs, FARs, QERs, URRs
- **UE IP** 00 - IP 0000, APN 000
- **PCO** 00 - 000 UE 0 DNS, P-CSCF, MTU 00
- 0000 - UPF 00, 000000

000000

- **Diameter Gx** 00 - PCRF 0000, PCC 00, QoS 00
- **Diameter Gy** 00 - OCS 0000, 0000

- **CDR** -

- **S5/S8** - GTP-C , SGW-C
- **QoS** - QoS

- - , ,
- **P-CSCF** - IMS

OmniPGW - *Omnitouch*

OmniPGW 部署架构图

部署架构图

OmniPGW 部署架构图

部署架构图

1. 部署
2. 部署
3. 部署
4. PFCP / 部署
5. Diameter (Gx/Gy) 部署
6. IP 部署
7. 部署

部署架构图

部署架构图 OmniPGW 部署架构图

- 部署
- 部署
- 部署
- 部署
- 部署

部署架构图

- 部署 - Prometheus 部署
- 部署 - 部署

□□□□□□

Web UI

□□□□ `http://<omnipgw_ip>:4000`

□□□□

- **/pgw_sessions** - □□□□□□□□ IMSI□IP□MSISDN□APN □□□
- **/diameter** - Diameter □□□□Gx PCRF□Gy OCS□
- **/pfcpeers** - PFCP □□□□PGW-U □□□□
- **/logs** - □□□□□□□□□□

Prometheus □□

□□□□ `http://<omnipgw_ip>:9090/metrics`

□□□□

- `teid_registry_count` - □□□□
- `address_registry_count` - □□□ UE IP
- `sxb_inbound_errors_total` - PFCP □□
- `gx_inbound_errors_total` - Diameter Gx □□
- `gy_inbound_errors_total` - Diameter Gy □□

□□□ □□□□ □□□□□□□□□□

□□□□

Web UI□ □□ **/logs** □□□□□□□□□□

□□□□□□□□

- "create_session_request" - □□□□
- "Credit Control" - Gx/Gy □□
- "PFCP Session" - □□□□□□
- "error" □ "ERROR" - □□□□

- "timeout" - 0000

00000000

0000000000000000 "000000"

000

- SGW-C 000000000000 "000000" (73)
- 000000000000
- 0000000000
- 000 [PGW-C] 0000000000 - 000000

Wireshark 0000000000000000 "000000"

000000

- OmniPGW 0000000000
- 000000000000

000

1. 0000000000

license_status

- 0 0

2. 0

- "license" "License"
- "0" 0

3. 0

- config/runtime.exs :license_client URL
- https://localhost:10443/api

0

1. 0

```
curl -k https://<license_server_ip>:10443/api/status
```

2. 0 config/runtime.exs 0

```
config :license_client,  
  license_server_api_urls:  
  ["https://<license_server_ip>:10443/api"],  
  licensee: "0"
```

3. 0

- omnipgwc
- Omnitouch 0

4. 0 OmniPGW

0

- license_status 0
- 0

- 確認する

確認する

確認

- SGW-C 確認
- 確認 PDN 確認
- 確認 `s5s8_inbound_errors_total` 確認

確認

1. IP 確認
2. PCRF (Gx) 確認
3. PGW-U (PFCP) 確認
4. APN 確認

確認

1. 確認 **IP** 確認

`address_registry_count`

- 確認

2. 確認 **PCRF** 確認

- Web UI → **/diameter** 確認
- 確認 PCRF 確認 = "disconnected"
- 確認 "Credit Control Answer" 確認

3. 確認 **PFCP** 確認

- Web UI → **/pfcpeers** 確認
- 確認 "Association: DOWN"
- 確認 `pfcpeers_associated` = 0

4. 設定 **APN** 設定

- 設定 `config/runtime.exs` 設定 `ue.apn_map`
- 設定 APN 設定

設定

設定 **IP** 設定

1. 設定 Web UI → **/pgw_sessions** 設定
2. 設定 `config/runtime.exs` 設定 IP 設定

```
config :pgw_c,  
  ue: %{  
    subnet_map: %{  
      "internet" => "10.0.0.0/23" # 10.0.0.0 /24 10.0.0.0 /23  
    }  
  }  
}
```

3. 設定 OmniPGW

4. 設定 `curl http://<ip>:9090/metrics | grep address_registry_count`

設定 **PCRF** 設定

1. 設定 `ping <pcrf_ip>`
2. 設定 PCRF Diameter 設定 `telnet <pcrf_ip> 3868`
3. 設定 `config/runtime.exs` 設定 Diameter 設定
4. 設定 OmniPGW
5. 設定 Web UI → **/diameter** 設定 "connected"

設定 **PFCP** 設定

- 設定 **PFCP** / 設定 設定

設定

- 設定 IP 設定 80% 設定
- 設定 PCRF 設定 Diameter 設定
- 設定

실시간 모니터링

목적

- Web UI 모니터링
- 실시간 모니터링
- 실시간 모니터링

요구사항

1. PCFP 모니터링 S5/S8 모니터링
2. PCRF CCR-Initial
3. 실시간 모니터링
4. 실시간 모니터링

구현

1. **Web UI** 모니터링
 - **/pgw_sessions** → IMSI
 - **pcfcp_seid** PCFP
 - **gx_session_id** Gx
2. 실시간 **IMSI**
 - IMSI
 - "PCFP"
 - "Gx"
 - 실시간
3. 실시간

```
# TEID PCFP
teid_registry_count - seid_registry_count

# TEID Gx
teid_registry_count - session_id_registry_count
```


□□□□

1. □□ **PFCP** □□□□

- □□ PGW-U □□□□□□
- □□ PFCP □□□Web UI → **/pfcpeer**
- □ SGW-C □□□□□□□□□□

2. □□ **Gx** □□□□

- □□ PCRF □□□ `histogram_quantile(0.95, rate(gx_outbound_transaction_duration_bucket[5m]))`
- □□□□□ `config/runtime.exs` □□□ Gx □□
- □□□□□□□□□□□□

3. □□□□□□□□□□

- □□□□□ OmniPGW □□□□□□□
- □□□□□□ `teid_registry_count` □□□□□

□□□

- □□ PFCP □ Gx □□□□
- □□□□□□/□□□□□□□□□□□□
- □□□□□□□□□□□□

PFCP / □□□□□□

□□□PFCP □□□□

□□□

- Web UI → **/pfcpeer** □□ "Association: DOWN"
- □□□□□□□□□□□□
- □□□ `pfcpeer_associated` = 0
- □□□"PFCP □□□□" □ "□□□□□□"

□□□□❓❓❓

1. PGW-U □□□□□□□□
2. PGW-U □□□□
3. PFCP □□□□□IP□□□□
4. □□□□ UDP 8805

□□□

1. □□□□□□

```
ping <pgw_u_ip>
nc -u -v <pgw_u_ip> 8805
```

2. □□ **PFCP** □□□

- □□ `config/runtime.exs` □□ `upf.peer_list`
- □□ IP □□□□ ID □□□ PGW-U □□□□

3. □□ **PGW-U** □□□

- □□ PGW-U □□
- □□ PGW-U □□□□□□□ `systemctl status omnipgw_u` □□□□□□□

4. □□□□□

```
# □□□□
pfcpc_consecutive_heartbeat_failures

# PFCP □□□
rate(sxb_inbound_errors_total[5m])
```

□□□□□

1. □□□□□□

- □□□□□ `traceroute <pgw_u_ip>`
- □□□□□□□□□□□□ UDP 8805

- 配置参数

2. 配置 PGW-U 参数

- 配置 PGW-U 参数
- 配置 30 秒心跳间隔
- 配置 Web UI → **/pfcp_peers** 配置 "Association: UP"

3. 配置 PFCP

- 配置 `config/runtime.exs` 配置 PFCP 参数
- 配置 OmniPGW
- 配置 PFCP 参数

配置

- 配置 `pfcp_peer_associated` 配置参数
- 配置 `pfcp_consecutive_heartbeat_failures` 配置 > 2 配置参数
- 配置 PGW-U 参数
- 配置 PFCP 参数/配置参数

配置 PFCP 配置参数

配置

- 配置参数
- QoS 配置参数 PCRF RAR 配置
- 配置 "配置参数"
- 配置


```
sxb_inbound_errors_total{message_type="session_modification_response"} 配置
```

配置

1. PFCP 配置 PDR/FAR/QR 配置
2. PGW-U 配置
3. 配置 ID 配置

4. PGW-U 確認

確認

1. 確認

- 値 "SEID" の SEID
- 値 PFCP の PFCP 識別子
- 値 "ID" の ID 値 "ID"

2. 値 PGW-U 確認

- 値 PFCP 確認
- 値 CPU 確認

3. 値 Web UI 確認

- /pgw_sessions → の IMSI 確認
- 値 pdr_map, far_map, qer_map 確認
- 値 ID

確認

1. 確認

- 値 確認
- 値 UE 確認

2. 値 PGW-U 確認

- 値 PGW-U 確認 PDR 確認
- 値 PGW-U
- 値 PGW-U 確認

3. 確認

- 値 Web UI 確認
- 値 PFCP 確認
- 値 確認

확인

- PGW-U 확인
 - 확인
 - `sxb_inbound_errors_total` 확인
-

Diameter (Gx/Gy) 확인

확인 PCRF 확인 Gx

확인

- Web UI → **/diameter** 확인 PCRF 확인 "disconnected"
- 확인 QoS 확인 QCI=5
- 확인 "Diameter 확인" 확인 "CER/CEA 확인"

확인

1. PCRF 확인
2. PCRF 확인
3. Diameter 확인 Origin-Host Realm
4. 확인 TCP 3868

확인

1. 확인

```
ping <pcrf_ip>
telnet <pcrf_ip> 3868
```

2. 확인 **Diameter** 확인

- 확인 `config/runtime.exs` 확인 `diameter.peer_list`
- 확인 `host` `realm` `ip` 확인 PCRF 확인
- 확인 `origin_host` 확인 PCRF 확인

3. PCRF

- PGW-C CER
-

4.

```
# Diameter
diameter_peer_connected{peer="<pcrf_host>"}
```

1.

- PCRF
- TCP 3868
- `nc -v <pcrf_ip> 3868`

2. PCRF

- PCRF
- 30
- Web UI → **/diameter**

3.

- `config/runtime.exe` Diameter

```

config :pgw_c,
  diameter: %{
    host: "pgw-c.epc.mnc999.mcc999.3gppnetwork.org", # 3GPP
    PCRF 3GPP
    realm: "epc.mnc999.mcc999.3gppnetwork.org",
    peer_list: [
      %{
        host: "pcrf.epc.mnc999.mcc999.3gppnetwork.org",
        realm: "epc.mnc999.mcc999.3gppnetwork.org",
        ip: "192.168.1.100",
        initiate_connection: true
      }
    ]
  }
}

```

- 3GPP OmniPGW
- 3GPP 3GPP

3GPP

- 3GPP Diameter 3GPP 3GPP
- 3GPP PCRF 3GPP 3GPP
- 3GPP 3GPP Diameter 3GPP

3GPP CCR/CCA 3GPP Gx 3GPP

3GPP

- 3GPP 3GPP > 5 3GPP
- 3GPP "3GPP 3GPP"
- 3GPP `gx_outbound_transaction_duration` 3GPP > 5s
- 3GPP 3GPP QoS 3GPP 3GPP

3GPP

1. PCRF 3GPP
2. PCRF 3GPP 3GPP
3. 3GPP

4. PCRF 設定

設定

1. Gx 設定

```
# P95 設定
histogram_quantile(0.95,
rate(gx_outbound_transaction_duration_bucket[5m]))

# P99 設定
histogram_quantile(0.99,
rate(gx_outbound_transaction_duration_bucket[5m]))
```

2. PCRF 設定

- PCRF 設定
- CPU使用率設定
- PCRF 設定

3. 設定

```
ping -c 100 <pcrf_ip> | tail -1 # 設定
```

4. 設定

- CCR/CCA 設定 "設定"
- "CCR" と "CCA" 設定

設定

1. PCRF 設定

- PCRF設定
- CCR 設定
- PCRF 設定/設定

2. 設定

- 確認する
- 確認 PGW-C へ PCRF 接続する

3. 確認する

- 確認 config/runtime.exs

```
config :pgw_c,
  diameter: %{
    transaction_timeout_ms: 10000 # 確認 5000 秒
  }
```

- 確認 OmniPGW
- 確認 確認する

確認

- 確認 Gx 確認する > 1s 確認 > 5s
- 確認する PCRF 確認
- 確認 PCRF 確認

確認 OCS 確認 Gy

確認

- Web UI → /**diameter** 確認 OCS 確認 "disconnected"
- 確認
- 確認 "Gy 確認"

確認

確認 PCRF 確認 Gy 確認

確認

- 確認 TCP 3868 確認 Gx 確認
- 確認
- 確認 diameter.peer_list 確認 OCS 確認

□□□IP □□□

- `address_registry_count` を取得
- Web UI → **/pgw_sessions** を取得
- `"IP アドレス"` を取得

1. 00000000
2. 0000 IP00000000
3. 0000000000
4. IP 0000

1.

2. `config/runtime.exs` `ue.subnet_map`
 - `"10.0.0.0/24"` = 254 IP
3. **IP**

```
# 000000
teid_registry_count
address_registry_count
```

4. 00000000

- Web UI → **/pgw_sessions**
- 0000000000
- 0000000000000000

000000

0000000000

1. 00 config/runtime.exs

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      "internet" => "10.0.0.0/22" # 1022 0 IP0000 /24 = 254 0
    }
  }
  }
```

2. 00 OmniPGW

3. 000000000000

00000000

1. 0 Web UI 00000000
2. 0 SGW-C 000000000000
3. 0 PCRF/SGW 0000000000
4. 00 address_registry_count 000000000000

000

- 00 IP 000000000000
 - 000 > 70%
 - 000 > 85%

- 確認する
- 確認する
- 確認する

確認する IP 確認

確認

- UE 確認 IP 確認
- 確認"IP 確認" 確認
- Web UI 確認確認確認確認 IP 確認

確認

1. 確認確認確認
2. 確認確認確認確認
3. 確認確認

確認

1. 確認 Web UI 確認 IP
 - /pgw_sessions → 確認 IP 確認
 - 確認確認 IMSI 確認確認 IP
2. 確認
 - 確認 IP 確認
 - 確認 "IP 確認" 確認

確認

1. 確認確認確認
 - 確認確認 IP 確認 IMSI
2. 確認確認
 - 確認 SGW-C 確認確認確認確認確認 IMSI

- □□□□□□□□

3. UE □□□□□

- UE 00000000
- 00000000 IP

4. □□□□□□□□

- **OmniPGW** **IP**
-

111

- 
- 

□□□□

□□ Prometheus □□

```
# □□□□
teid_registry_count

# □□□□□□□□□□
rate(s5s8_inbound_messages_total{message_type="create_session_request

# IP □□□□□□□□ /24 □□□
(address_registry_count / 254) * 100

# P95 □□□□□□
histogram_quantile(0.95,
rate(s5s8_inbound_handling_duration_bucket{request_message_type="crea
[5m]))

# □□□
rate(s5s8_inbound_errors_total[5m])

# PCRF □□
histogram_quantile(0.95, rate(gx_outbound_transaction_duration_bucket

# PFCP □□□□
pfcpeer_associated
```

Web UI

項目	値
IMSI	15桁の数字
"create_session"	セッション作成
"delete_session"	セッション削除
"Credit Control"	Gx PCRF 信用制御
"PFCP Session"	PFCP セッション
"error"	エラー
"timeout"	タイムアウト
"Association"	PFCP アソシエーション

コマンド

```
# サービスの状態を確認
systemctl status omnipgw_c

# Web UI へのアクセス
curl http://<omnipgw_ip>:4000

# サービスのメトリクスを確認
curl http://<omnipgw_ip>:9090/metrics

# サービスのメトリクスを確認 (teid_registry_count を含む)
curl http://<omnipgw_ip>:9090/metrics | grep teid_registry_count

# PFCP サービスのメトリクスを確認
curl http://<omnipgw_ip>:9090/metrics | grep pfcpeer_associated

# IP アドレス登録のメトリクスを確認
curl http://<omnipgw_ip>:9090/metrics | grep address_registry_count
```

機能

- **監視** - Prometheus と Grafana を利用して監視
- **ログ** - ログ出力
- **メトリクス** - サービスのメトリクスを出力
- **PFCP サービス** - PFCP サービスを提供
- **Diameter Gx サービス** - Gx サービスを提供
- **Diameter Gy サービス** - Gy サービスを提供
- **QoS サービス** - QoS サービスを提供

インストール

OmniPGW のインストール - *OmniTouch* のインストール

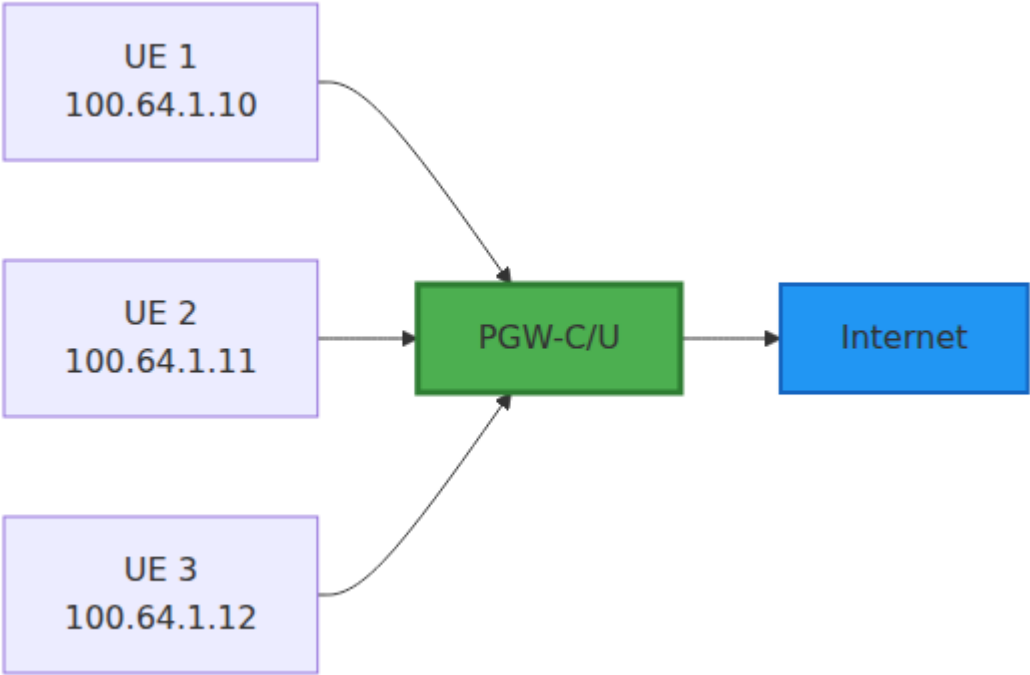
UE IP

IP

- 1.
2. IP
- 3.
- 4.
- 5.
- 6.
- 7.

PGW-C UE PDN IP

이제 IP 할당



이제 UE 가 PGW-C 할당 받은 IP 할당

- 할당된 IP
- 할당된 IP
- 할당된 IP
- 이 PDN 할당된 IP

이제 IP 할

IP 할	할	할
IPv4	할 할	할 IPv4 할
IPv6	할 할	IPv6 할 할
IPv4v6	할 할	할 IPv4 할 IPv6

IP 地址

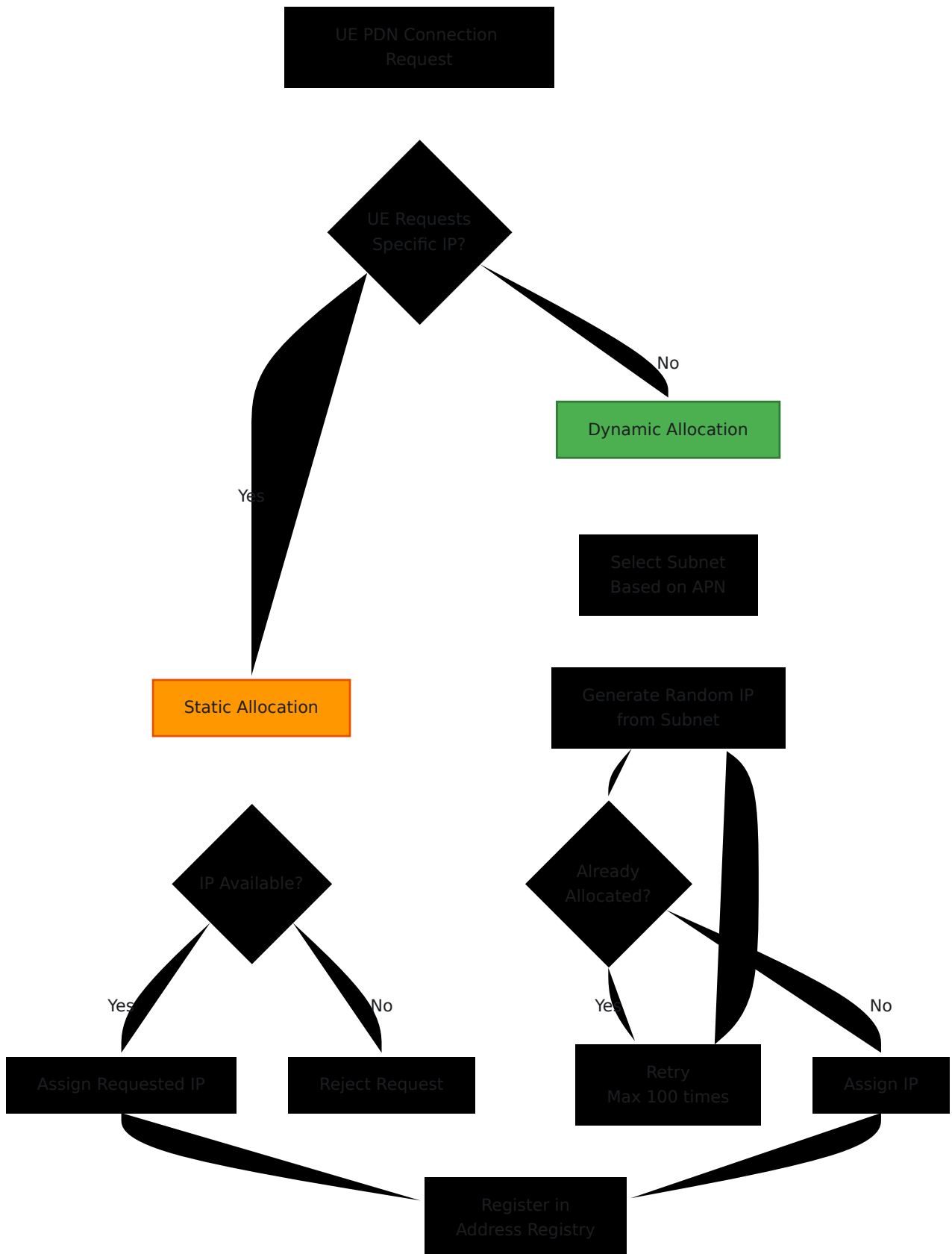
PDN 类型

UE 通过 PDN 类型选择 **PDN 类型**

PDN 类型	地址	地址范围
IPv4	IPv4 地址	IPv4 地址
IPv6	IPv6 地址	IPv6 地址/64
IPv4v6	地址	IPv4 地址 IPv6 地址

地址

PGW-C 地址 IP 地址



1. IP Allocation

- PGW-C allocates IP
- IP Allocation

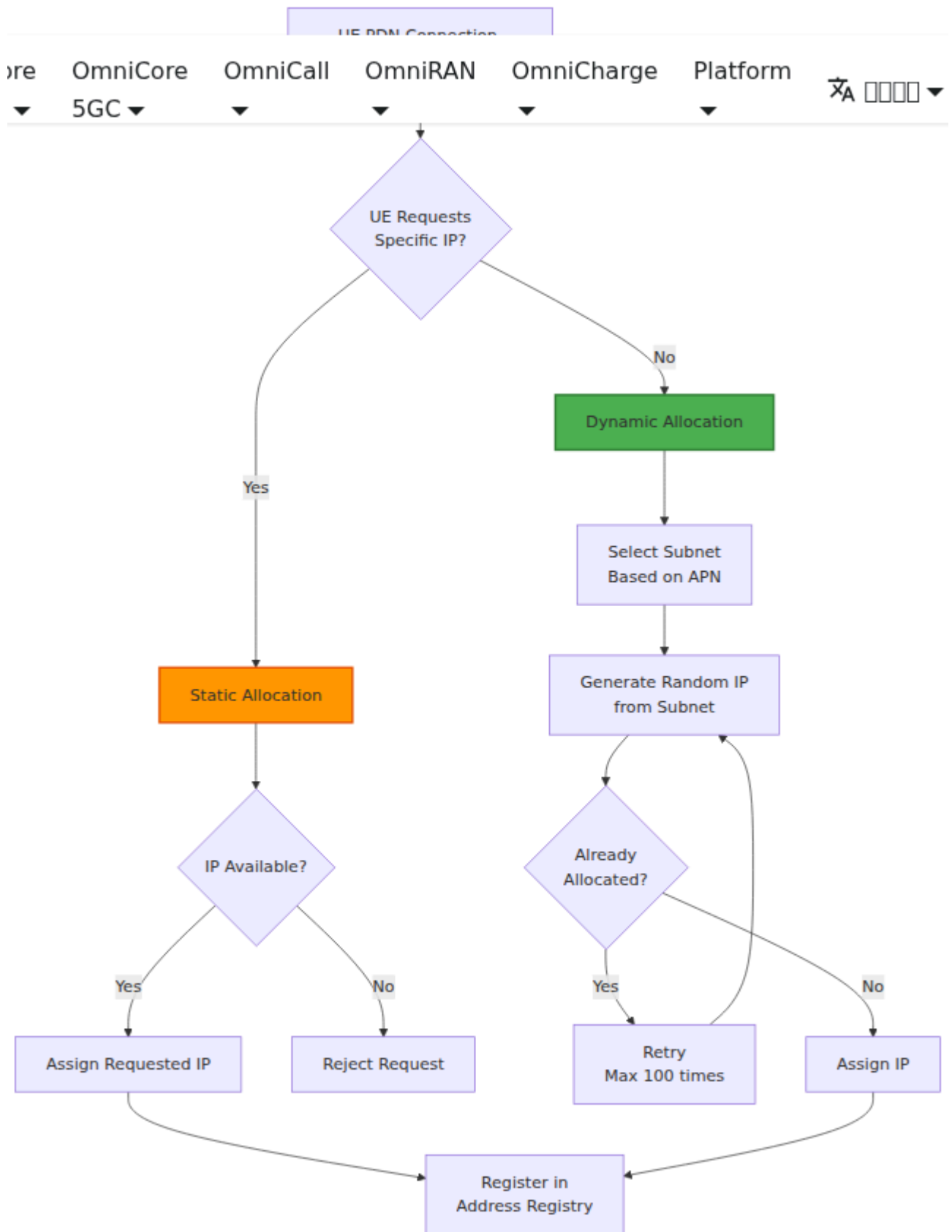
- 0000000000

2. 000000

- UE の GTP-C 00000000 IP
- PGW-C 000000
- 00000000 IP 000000

00 **APN** 000000

000 **APN**00000000 00000000 IP 00



□□□

- □□□□ - □□ APN □□□□□□□□
- □□❓❓□□ - □□□□ APN □□□□□□□□

- 0000 - 0000000000000000
- 00 - 000000000000

0000

0000 000000 IP

00	00
00	00 UE IP → 0000 PID
00	00 UE IP 0000
00	00000000 IP
0000	000000

00

0000

00 config/runtime.exs

```

config :pgw_c,
  ue: %{
    subnet_map: %{
      # APN "internet"
      "internet" => [
        "100.64.1.0/24",    # 254 IP
        "100.64.2.0/24"    # 254 IP
      ],

      # APN "ims"
      "ims" => [
        "100.64.10.0/24"
      ],

      # Default APN
      default: [
        "42.42.42.0/24"
      ]
    }
  }
}

```

APN

APN is a network identifier that identifies the network to which the device is connected. It is used to route traffic to the correct network.

Examples:

- `internet` - Default APN
- `ims` - IMS APN
- `default` - Default APN


```
config :pgw_c,
  ue: %{
    subnet_map: %{
      # [] [] "ims" [] APN[]
      [] "ims" [] "ims.apn" [] "ims.something.else" []
      "^ims" => [
        "100.64.10.0/24"
      ],

      # [] [] "m2m." [] APN[] [] "m2m.test" [] "m2m.prod" []
      "^m2m\." => [
        "100.64.20.0/24"
      ],

      # [] [] - "enterprise.corp" [] []
      "enterprise.corp" => [
        "10.100.0.0/16"
      ],

      # [] APN [] []
      default: [
        "42.42.42.0/24"
      ]
    }
  }
}
```

□□□□□

- 開発言語 Elixir/Erlang を採用
- Elixir を採用する理由 \ 何 \
- 何 ^ 開発言語 Elixir/Erlang を採用
- 開発言語 Elixir/Erlang を採用
- 開発言語 Elixir/Erlang を採用

Regex	Pattern	Matches
^ims	"^ims"	ims, ims.apn, ims.anything
^.*\.corp\$	"^.*\.corp\$"	foo.corp, bar.corp
test	"^.*test.*"	test, foo.test.bar, testing
internet.apn	"^internet\.apn\$"	internet.apn

Regex

Regex .corp APN ^.*\.suffix\$

```
subnet_map: %{\n  # .corp APN\n  "^.*\\.corp$" => ["10.100.0.0/16"],\n\n  # .iot APN\n  "^.*\\.iot$" => ["10.200.0.0/16"],\n\n  default: ["42.42.42.0/24"]\n}
```

Regex

APN 名称	正则表达式	IP 地址
ims	^ims	100.64.10.0/24
ims.apn	^ims	100.64.10.0/24
ims.something.else	^ims	100.64.10.0/24
m2m.test	^m2m\.	100.64.20.0/24
m2m	default	42.42.42.0/24
enterprise.corp	enterprise.corp	10.100.0.0/16
foo.corp	^.*\.corp\$	10.100.0.0/16
unknown.apn	default	42.42.42.0/24

IP 地址

CIDR 表示法 <network>/<prefix_length>

CIDR	可用 IP	IP 地址范围
/24	254	100.64.1.1 - 100.64.1.254
/23	510	100.64.0.1 - 100.64.1.254
/22	1022	100.64.0.1 - 100.64.3.254
/20	4094	100.64.0.1 - 100.64.15.254
/16	65534	100.64.0.1 - 100.64.255.254

子网

- 100.64.1.0/24
- 100.64.1.255/24
- PGW-C `<network> + 1` `<broadcast> - 1`

APN

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      "internet" => [
        "100.64.1.0/24",
        "100.64.2.0/24",
        "100.64.3.0/24",
        "100.64.4.0/24"
      ]
    }
  }
}
```

- PGW-C
-
-

-
-
-

配置示例

```
config :pgw_c,  
  ue: %{  
    subnet_map: %{  
      # 互联网  
      "internet" => [  
        "100.64.0.0/20"      # 4094 个 IP 地址  
      ],  
  
      # IMS  
      "ims" => [  
        "100.64.16.0/22"    # 1022 个 IP 地址  
      ],  
  
      # 企业 APN  
      "enterprise.corp" => [  
        "10.100.0.0/16"     # 65534 个 IP 地址  
      ],  
  
      # IoT  
      "iot.m2m" => [  
        "100.64.20.0/22"    # 1022 个 IP 地址  
      ],  
  
      # 默认  
      default: [  
        "42.42.42.0/24"     # 254 个 IP 地址  
      ]  
    }  
  }  
}
```

IPv6 配置

```
config :pgw_c,  
  ue: %{  
    subnet_map: %{  
      "internet" => [  
        # IPv4 网段  
        "100.64.1.0/24"  
      ],  
      "internet.ipv6" => [  
        # IPv6 网段  
        "2001:db8:1::/48"  
      ],  
      default: [  
        "42.42.42.0/24"  
      ]  
    }  
  }  
}
```

IPv6 地址规划

- UE 地址 /64 网段
- 每个 UE 地址 IP 地址
- 每个 UE 地址 `2001:db8:1:a::/64`

IPv4v6 配置

```
config :pgw_c,  
  ue: %{  
    subnet_map: %{  
      "internet" => [  
        "100.64.1.0/24",      # IPv4 网段  
        "2001:db8:1::/48"    # IPv6 网段  
      ],  
      default: []  
    }  
  }  
}
```

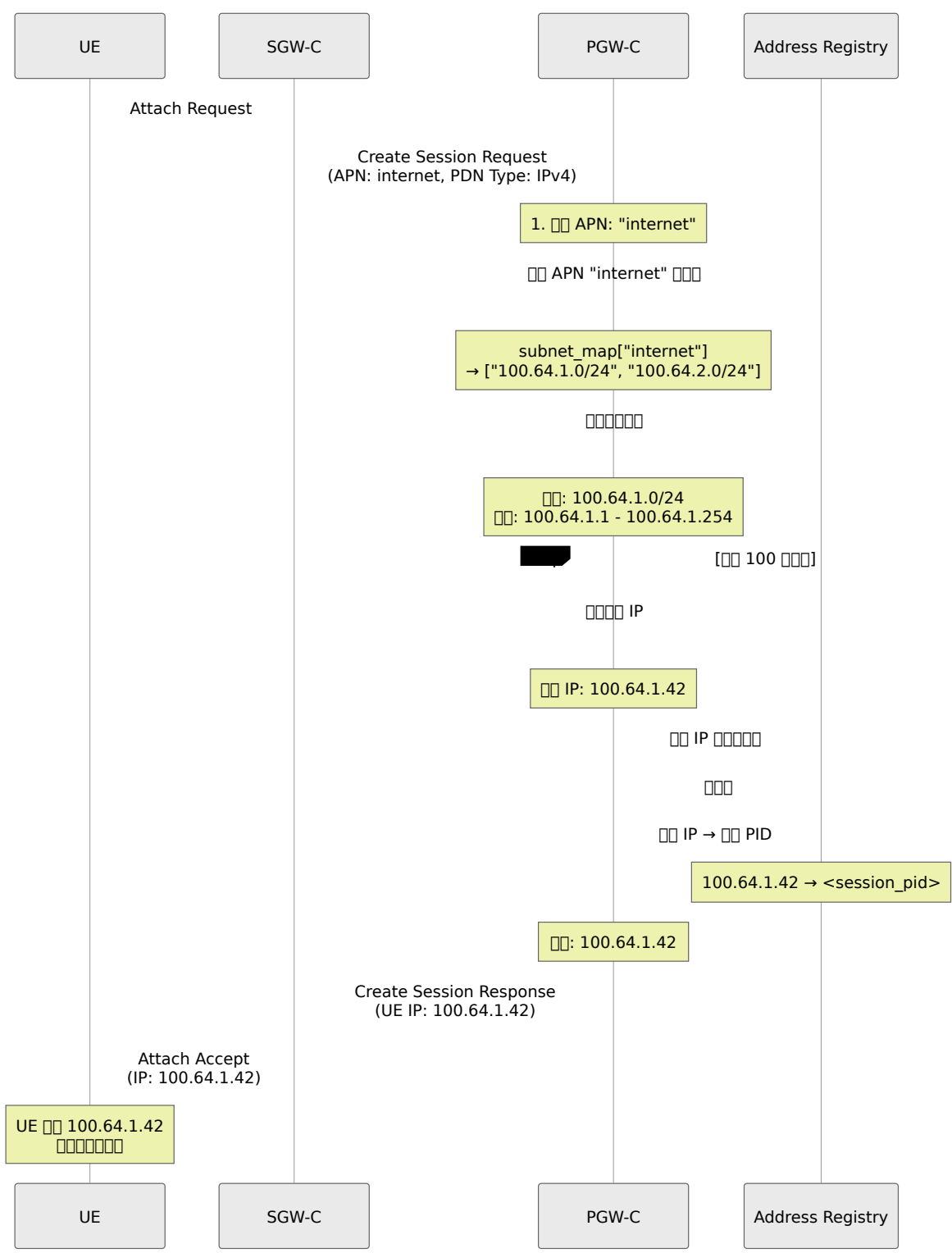
配置

- UE 通过 PDN 连接IPv4v6
 - PGW-C 支持 IPv4 和 IPv6 地址
 - 网络地址转换
-

网络

IP 地址由 PGW-C 通过 S5/S8 接口分配给 GTP-C 接口。S5/S8 接口由 PGW-C 通过 S5/S8 接口分配给 GTP-C 接口。

IPv4



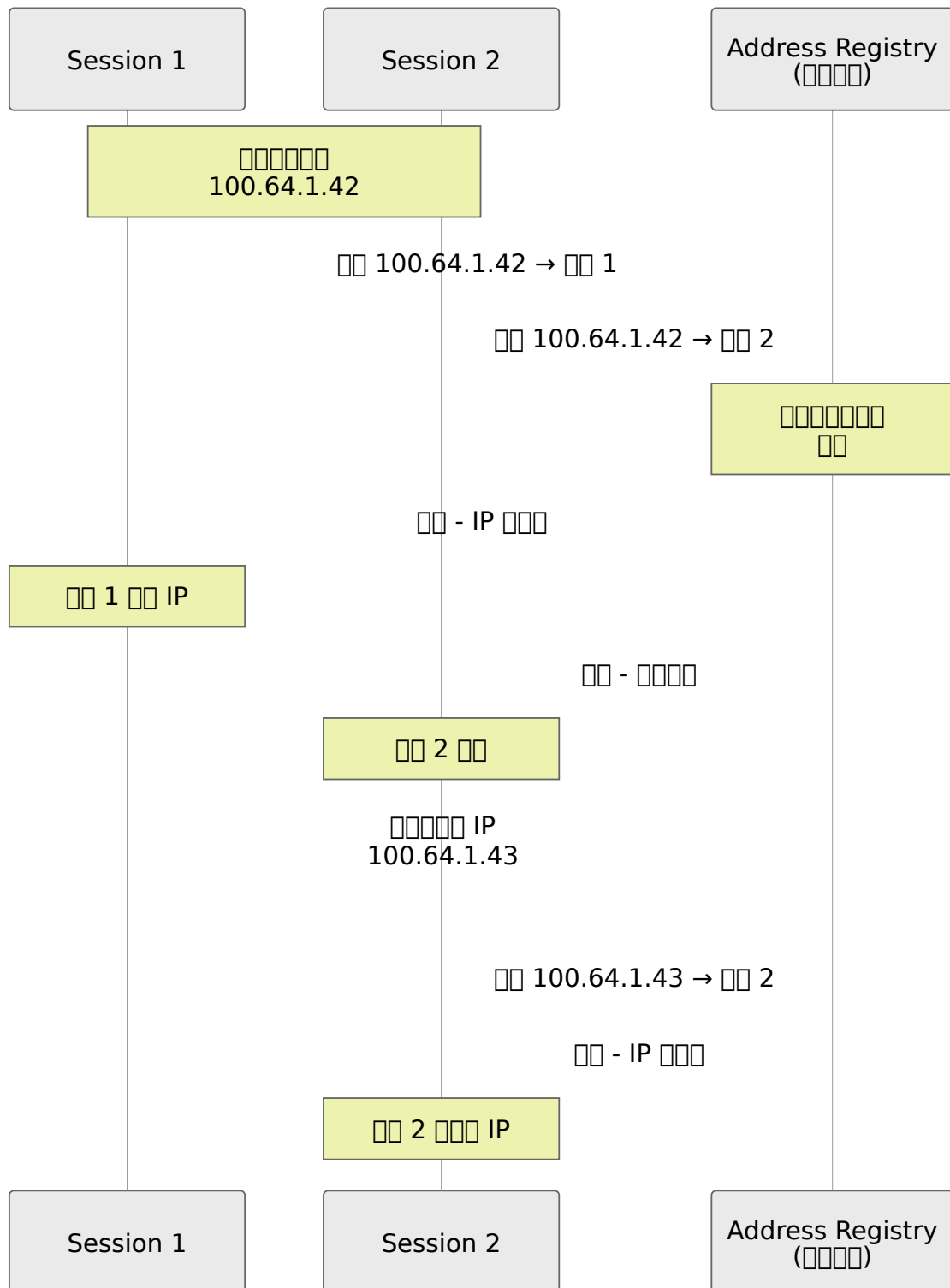
1. 查詢所有 網卡 APN 清單
2. 查詢所有 網卡 IP 清單
3. **IP** 查詢 網卡 IP 清單
4. 查詢所有 網卡 IP 清單
5. 查詢所有 網卡 IP 清單 100 個   網卡 IP
6. 查詢 網卡 IP 清單

查詢所有

- 查詢 **100** 個 網卡 IP 清單
- 查詢所有 網卡 IP 清單
- 查詢所有 網卡 IP 清單
- 查詢所有 網卡 IP 清單 APN 清單

查詢

查詢 網卡 IP 清單



IP address allocation

- IP address allocation
- IP address allocation
- IP address allocation IP 100
- IP address allocation IP 100
- IP address allocation IP 100

設定方法

UE の APN

設定

```
# 例
subnet_map: %{
  "internet" => ["100.64.1.0/24"],
  default: ["42.42.42.0/24"]
}
```

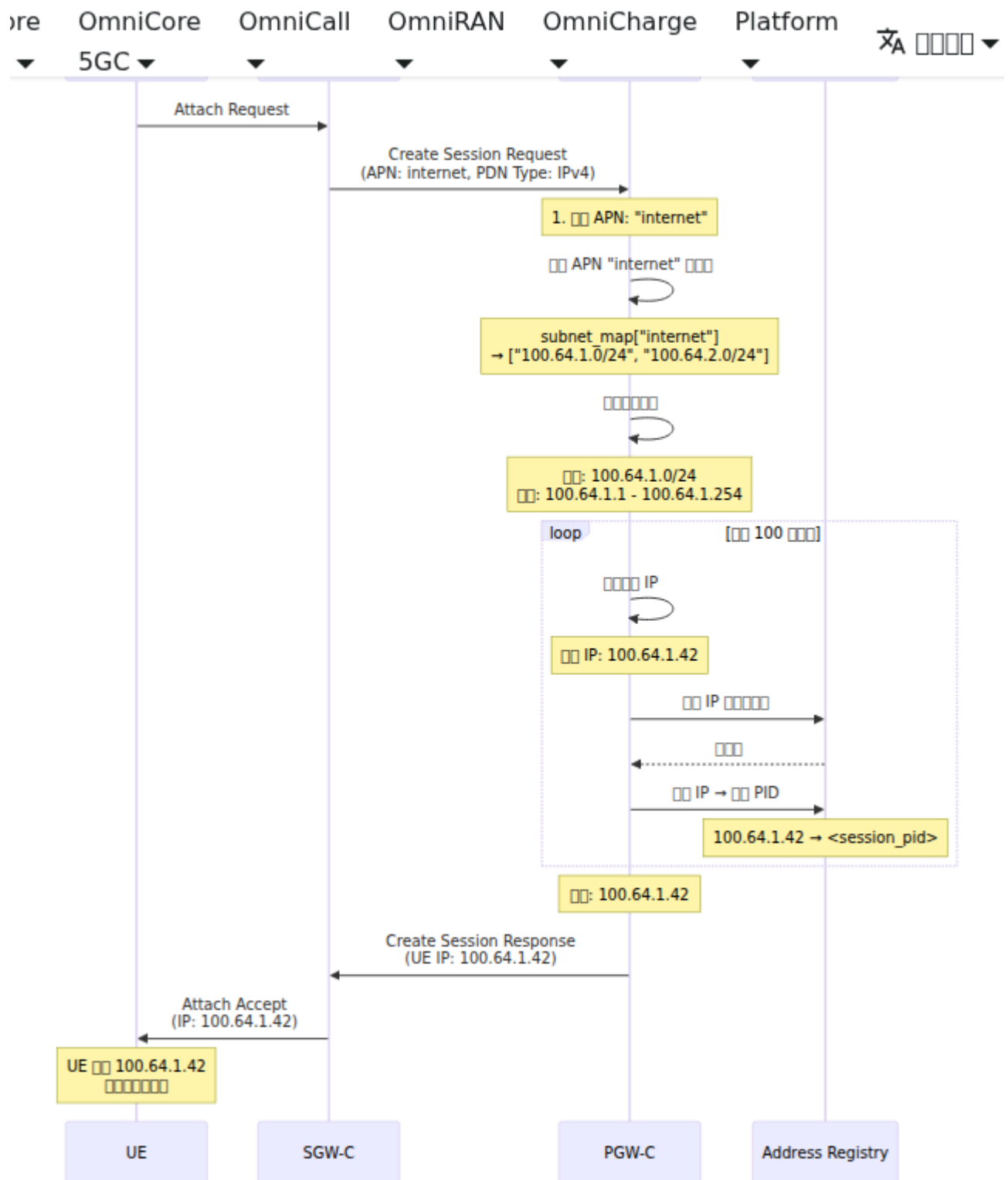
設定

- UE の APN: "unknown.apn"
- subnet_map の "unknown.apn"
- 設定
- 42.42.42.0/24 の IP

設定

1. 設定 APN
2. default
3. 設定

Sequence Diagram



Summary

- The diagram illustrates the process of IP address allocation for a UE.
- The IP address 100.64.1.42 is assigned to the UE.
- The IP address is mapped to a PID (100.64.1.42 → <session_pid>).

IP Pool

Overview

IP Pool Management IP Address

IP: 100.64.1.0/24 (254 IP)

Pool: 254 IP

Pool Size → 254

Configuration

1. PGW-C IP 100 IP Pool
2. IP Pool Management IP Address
3. Error: `{:error, :ue_ip_address_allocation_failed}`
4. IP Pool Management
5. SGW-C IP Pool Management

Configuration

```
# IP Pool Management
address_registry_count / total_pool_size > 0.8 # 80% IP Pool

# IP Pool Management
"internet" => [
  "100.64.1.0/24",
  "100.64.2.0/24", # IP Pool Management
  "100.64.3.0/24"
]
```

IP Pool

IP Pool Management IP Address

GTP-C IP Pool

Create Session Request

```
|— IMSI: 310260123456789
|— APN: enterprise.corp
|— PDN  (IE)
|   |— PDN : IPv4
|   |— IPv4 : 10.100.0.50 ← UE  IP
```

OmniPGW

1. PDN IP PDN IE
2. IP IP APN
3. IP
4.
 - IP
 -

- UE IP
- IP - IP
- IP - IP

IPv6

UE IPv6

Create Session Request

```
|— PDN : IPv6
```

PGW-C /64

Network: 2001:db8:1:a::/64

UE Network

- 2001:db8:1:a::1
- 2001:db8:1:a::2
- ... (18 quintillion)

Network

- UE Network IP Network
- SLAAC Network
- NAT

Network

UE IP IPv4v6

Create Session Request

PDN: IPv4v6

PGW-C Network

IPv4: 100.64.1.42

IPv6: 2001:db8:1:a::/64

Network

- IPv4 Network IPv4
- IPv6 Network IPv6
- Network
- GTP Network

Network IP

IP RFC 1918

```
# 配置子网
subnet_map: %{
  "internet" => [
    "10.0.0.0/8",
    "172.16.0.0/12",
    "192.168.0.0/16"
  ]
}
```

配置 **PGW-U** 的 **NAT** 规则

配置 **IP** 池

```
# 配置 IP 池
subnet_map: %{
  "internet" => [
    "203.0.113.0/24" # 公网 IP 池
  ]
}
```

配置 **NAT** - 规则

配置

- 配置 **IP** **RFC 6598** 的 **100.64.0.0/10** 公网 NAT
- 配置 **IP**

配置

Web UI - IP 配置

OmniPGW 配置 Web 接口 IP 地址

配置 `http://<omnipgw-ip>:<web-port>/ip_pools`

□□□

1. □□□

- □□□□□ IP □
- □□□□□□□□
- □□□□ IP
- □□□□□□□□

2. □□ **APN** □□□ □□□□□□□□

- □□□ - APN □□□□□□□□“default”□“ims.something.else”□“Internet”□
- **APN** □□ - □□□ APN □□□□
- **IP** □□ - □□□□□□□□ CIDR □□□
- □□□ - □□□□□□□□□□□□□□
- □□□□□
 - □□□□□□□ IP □□
 - □□□□□□□□□□ IP
 - □□□□□□□□□□ IP

3. □□□□

- 2 個
- 個
- 個

個

- 個
- 個
- 個
- 1 APN 個

個

個

```
# 個 IP
address_registry_count

# 個
address_registry_count / <total_pool_size> * 100
```

個

```
個: 100.64.1.0/24 (254 個 IP)
個: 150 個 IP
個: 150 / 254 = 59%
```

00

```
# 0000000000
- alert: UEIPPoolUtilizationHigh
  expr: address_registry_count > 200 # 00 /24 0
  for: 10m
  annotations:
    summary: "UE IP 000000 80%"
    description: "00: {{ $value }} / 254 0 IP 000"

# 00000000
- alert: UEIPPoolExhausted
  expr: address_registry_count >= 254 # 00 /24 0
  for: 1m
  annotations:
    summary: "UE IP 000 - 0000 IP"

# 0000000000
- alert: UEIPAllocationFailures
  expr: rate(ue_ip_allocation_failures_total[5m]) > 0
  for: 5m
  annotations:
    summary: "UE IP 000000"
```

Grafana 000

00 10IP 0000

```
# 0000000000
(address_registry_count / 254) * 100
```

00 200000000000 IP

```
# 0000
address_registry_count
```

00 30000

```
# 確認
rate(address_registry_count[5m])
```

約 4割増

```
# 確認
(254 - address_registry_count) / rate(address_registry_count[1h])
```

確認

約 1割増のIP

確認

- 確認 "確認"
- 確認 "UE IP 確認"

確認

1. 確認

```
# 確認
curl http://<pgw_c_ip>:42069/metrics | grep
address_registry_count
```

2. 確認

```
# 例
config :pgw_c,
  ue: %{
    subnet_map: %{
      "internet" => [
        "100.64.1.0/24" # 例 CIDR
      ]
    }
  }
}
```

3. APN 例

```
# 例 APN
# 例
subnet_map: %{
  default: ["42.42.42.0/24"]
}
```

例

- 例 例
- 例 PGW-C 例 IP
- 例 `runtime.exs` 例

2 IP 例

例

- 例 UE 例 IP
- 例

例

- 例

例

```
# 確認する IP
grep "already_registered" /var/log/pgw_c.log
```

確認

- 確認する IP
- 確認する IP

3. 確認する IP

確認

- UE 確認する IP
- APN "internet" の "ims" の IP

確認

- 確認する IP

確認

```
# 確認する APN 確認
subnet_map: %{
  "internet" => [...],      # 確認
  "Internet" => [...],     # 確認 APN!
}
```

確認

- 確認 APN 確認する IP
- 確認する IP

4. IPv6 確認

確認

- UE の IPv6 確認

□□□□□

1. □□□ IPv6 □

```
# □□ IPv6 □□
subnet_map: %{
  "internet" => [
    "100.64.1.0/24" # □ IPv4
  ]
}
```

2. □□□ IPv6 □□

```
# □□□□□□□ /48 □□□□
"internet" => [
  "2001:db8::/128" # □□ - □□□□□□
]
```

□□□□□

```
# □□ IPv6 □
subnet_map: %{
  "internet" => [
    "100.64.1.0/24",
    "2001:db8:1::/48" # IPv6 □
  ]
}
```

□□ 5□□□□□□□

□□□

- □□□□□
- `address_registry_count` □□□□□

□□□□□

1. □□□□□

```
"internet" => [
  "100.64.1.0/24",    # 1
  "100.64.2.0/24",    # 256 1 IP
  "100.64.3.0/24"     # 256 1 IP
]
```

2. 100.64.0.0/22

```
# 100.64.0.0/22
"internet" => [
  "100.64.0.0/22" # 1024 1 IP
]
```

3. 100.64.0.0/22

- 100.64.0.0/22
- 100.64.0.0/22

100.64.0.0/22

100.64.0.0/22

100.64.0.0/22

```
100.64.0.0/22 10,000
100.64.0.0/22 30% 3,000 100.64.0.0/22
100.64.0.0/22 50%
100.64.0.0/22 3,000 * 1.5 = 4,500 100.64.0.0/22
```

```
100.64.0.0/20 4,094 100.64.0.0/22 - 100.64.0.0/22
100.64.0.0/19 8,190 100.64.0.0/22 - 100.64.0.0/22
```

100.64.0.0/22

100.64.0.0/22

- 100.64.0.0/10 RFC 6598 - NAT
- 400 IP
- NAT

- IP
- VPN

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      # APN - 
      "internet" => [
        "100.64.0.0/18" # 16,382 IP
      ],

      # IMS - 
      "ims" => [
        "100.64.64.0/22" # 1,022 IP
      ],

      # - 
      "enterprise.corp" => [
        "100.64.68.0/22" # 1,022 IP
      ],

      # - 
      "iot.m2m" => [
        "100.64.72.0/20" # 4,094 IP
      ],

      # - 
      default: [
        "100.64.127.0/24" # 254 IP
      ]
    }
  }
}
```

□□□□

□□

- □□□□ - UE IP □□□□APN □□□□
- **PCO** □□ - □ IP □□□□□□□ DNS□P-CSCF□MTU
- □□□□ - □□□□□□□PDN □□□□□ IP □□
- **PFCP** □□ - □□ PFCP □ UPF □□ UE □□

□□□□

- **S5/S8** □□ - □□ GTP-C □□ IP □□
- **Diameter Gx** □□ - IP □□□□□□□

□□

- □□□□ - IP □□□□□□□□□□□□
- □□ **CDR** □□ - CDR □□ UE IP □□□□□□□□

□□□□□□