

無線通信記録ファイル (CDR) について

SGW-C について

OmniSGW と Omnitouch の関係

目次

1. 概要
 2. CDR の種類
 3. CDR の取得
 4. CDR の形式
 5. 取得方法
 6. 注意点
 7. CDR の利用
 8. 関連情報
 9. お問い合わせ
 10. 補足
-

概要

無線通信記録 (CDR) は、無線通信ネットワークにおいて、SGW-C (Serving Gateway - Control Plane) と CDR (Call Detail Record) の関係を示すファイル形式です。

無線通信記録 (PGW-C CDR) は、EPC (Evolved Packet Core) ネットワークにおいて、

取得方法

- 無線 CDR 取得 - 無線通信記録の取得方法
- 無線 CDR 取得 - 無線通信記録の取得方法

- **CDR** - **Call Detail Record**
- **CDR** - **Call Detail Record**
- **3GPP** **TS 32.251** **PS** **TS 32.298** **CDR**

CDR

CDR	CDR
CDR	CDR CDR
CDR	CDR CDR
CDR	CDR CDR
CDR	CDR CDR
CDR	CDR CDR
CDR	CDR CDR

CDR CDR

CDR CDR

<epoch_timestamp>

CDR

1726598022

CDR CDR Unix CDR CDR

□ □ □ □

5/5

- SGW-C: `/var/log/sgw_c/cdrs/`

CDR directory

111

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
```

□ □ □ □ □

- 00000000 - CDR 0000000000000000 Unix 0000
- 00000000 - 00000000000000000000 Unix 0000
- 000000 - SGW-C 00000000
- 00 - 000000 CSV 000000

CDR 表

表名

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

項目	項目名	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 項目

項目名

CDR 項目名一覧

項目名	型	型	説明
default_bearer_start	uint32_t	uint32_t	デフォルトベアラの開始時刻
default_bearer_update	uint32_t	uint32_t	デフォルトベアラの更新時刻
default_bearer_end	uint32_t	uint32_t	デフォルトベアラの終了時刻

項目名

- default - デフォルト PDN 項目名
- dedicated - デDICATED PDN 項目名

項目名

- default_bearer_start
- 開始時刻
- default_bearer_update
- 更新時刻
- default_bearer_end
- 終了時刻
- dedicated_bearer_start
- 開始時刻
- dedicated_bearer_update
- 更新時刻
- dedicated_bearer_end
- 終了時刻

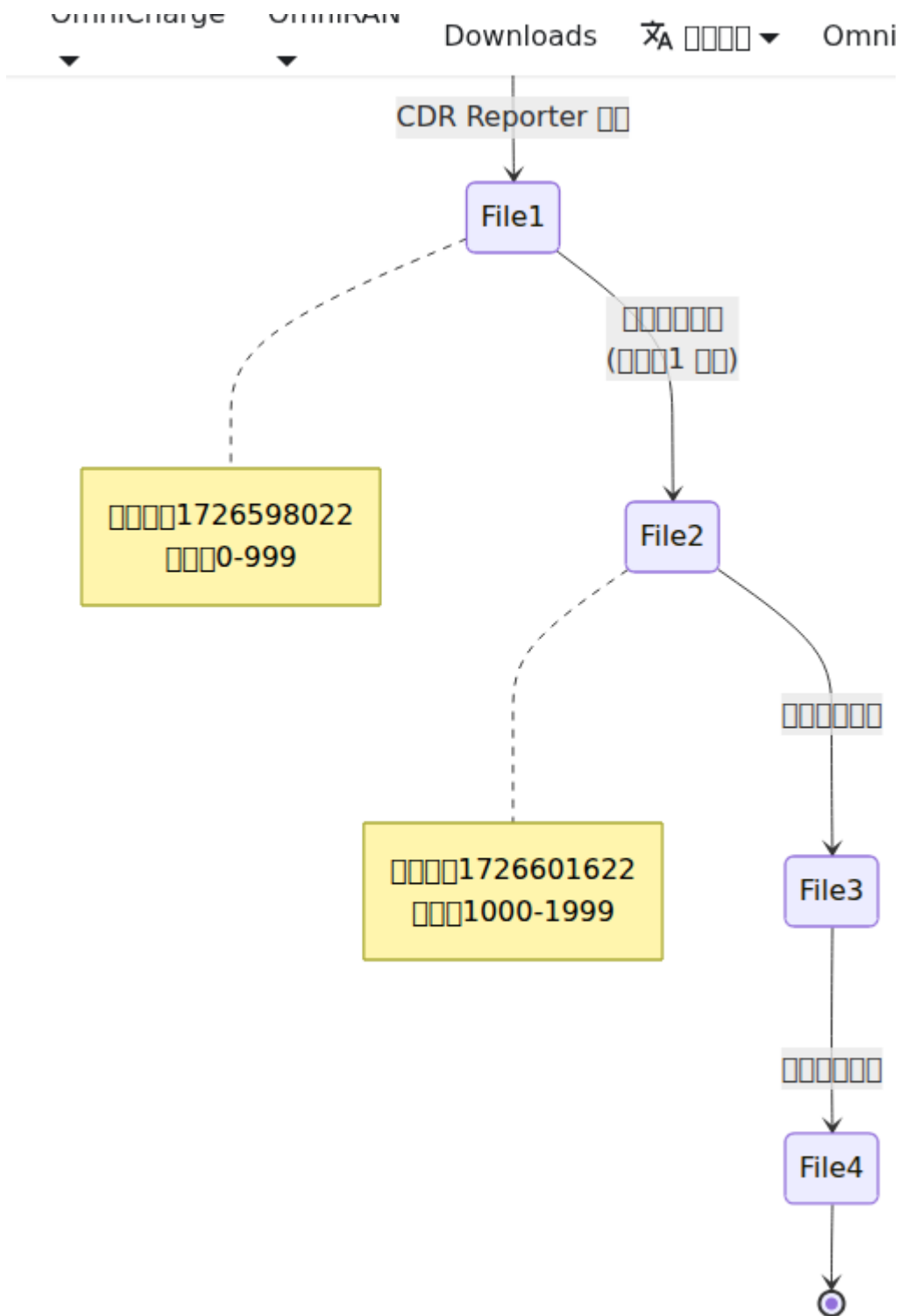
□□□□

□□ **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. □□□□ CDR □□
2. □□□□□□□□□□□□□□
3. □□□□□□□□
4. □□□ CDR □□□□□□□□

필수

선택

필드명	유형	설명	필수성	기본값
gateway_name	문자열	SGW-C 인스턴스명	-	인스턴스 ID
duration	정수	로그 수집 기간	-	3600000~1 일
directory	문자열	CDR 저장 경로	-	/var/log/sgw_c/cdrs

예시 1

설정

- **gateway_name:** "sgw-c-prod-01"
- **duration:** 3,600,000 ~1 일
- **directory:** "/var/log/sgw_c/cdrs"

예시 2

- **gateway_name:** "sgw-c-dev"
- **duration:** 300,000 ~5 분
- **directory:** "/tmp/sgw_c_cdrs"

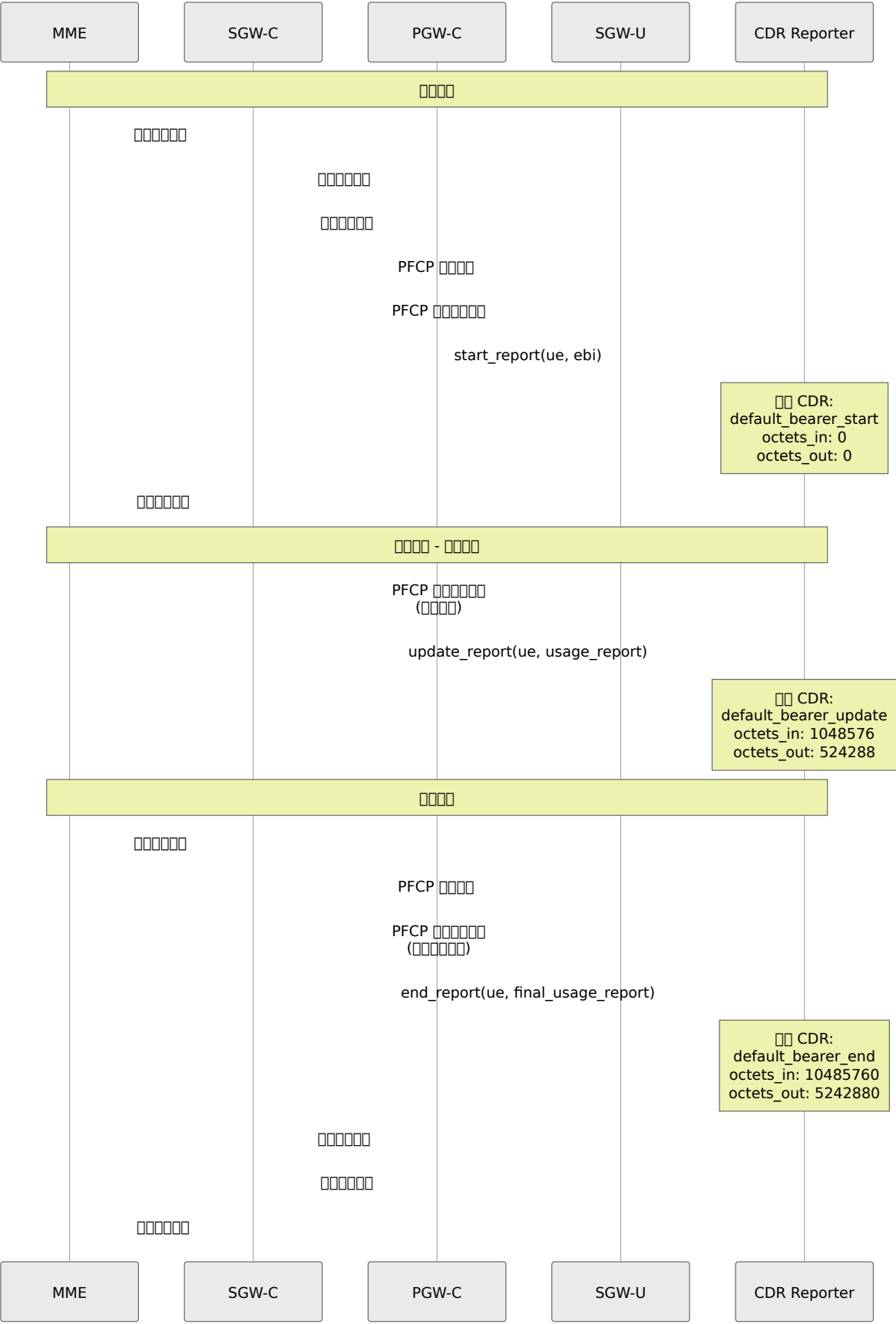
예시 3

- **gateway_name:** "sgw-c-prod-heavy"
 - **duration:** 1,800,000 ~30 분
 - **directory:** "/mnt/fast-storage/cdrs"
-

CDR □□□□

□□□□□□ **CDR** □□

SGW-C CDR □□□



CDR 項目

1. 項目

- 項目 項目項目項目
- 項目 項目項目項目項目
- **octets_in** 0
- **octets_out** 0

2. 項目

- 項目 項目項目項目 PFCP 項目項目
- 項目 項目項目項目
- **octets_in** 項目項目項目項目項目項目
- **octets_out** 項目項目項目項目項目項目

3. 項目

- 項目 項目 PFCP 項目項目項目項目項目項目
- 項目 項目項目項目項目項目項目
- **octets_in** 項目項目項目
- **octets_out** 項目項目項目

項目項目項目

1. epoch (項目)

項目 Unix 項目項目項目❓❓❓

項目 CDR 項目項目項目

項目

1726598022 → 2025-09-17 18:53:42 UTC

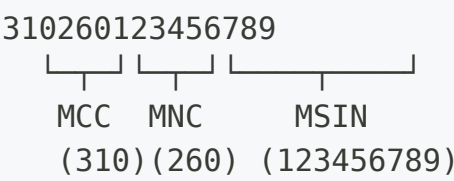
2. imsi (IMEI)

IMEI 15 자리

IMEI MCCMNC + MSIN

IMEI 0000000000000000

IMEI



IMEI UE 0000000000000000

3. event (CDR 이벤트)

IMEI 이벤트

IMEI <bearer_type>_bearer_<event>

IMEI

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

IMEI

- IMEI EBI EPS ID IMEI LBI ID default
- IMEI EBI ID LBI dedicated

4. charging_id (000000)

0000 0000 32 0000

0000 000000000000000000000000

0000

12345

0000 0 PGW-C 0000000000000000

0000

- 0 SGW 0 PGW 0000000000
- 00 Diameter Gy/Gz 0000
- 0000000000000000

5. msisdn (000000)

0000 00000 E.164 0000

0000 0000 ISDN 00000000000000

0000 00000 + 00000

0000

15551234567

└─┬────────┘

CC National

(1) (5551234567)

0000 UE 0000000000 MME 0 HSS 00

6. ue_imei (IMEI)

IMEI 15 数字

IMEI TAC 8 + SNR 6 + Spare 1

IMEI 15 数字

IMEI

123456789012345

12345678

901234

5

TAC SNR S

IMEI UE 15 数字 MME 15

7. timezone_raw (UE 时区)

时区 数字/数字

IMEI UE 15 数字

时区 数字 CSV 数字

时区 数字 数字

IMEI

, (数字)

8. plmn (PLMN)

PLMN 数字

PLMN 数字 数字

00000

MCC: 505, MNC: 57

↓

"50557"

↓

0000 "055570"

↓

000000000000x055570 = 349552

000

349552 → MCC: 505, MNC: 57

000 0 MME 000 UE 000000

000 00000000000000000000

9. tac (0000000)

000 000 16 000

000 00000000 UE 00000000

000 0 - 65535

000

1234

000 UE 00000000 MME 0000000000

000

- 00000000
- 00000000

- TAI

10. eci (E-UTRAN)

28

E-UTRAN UE

eNodeB ID20 + ID8

0 - 268,435,455

5678

MME UE

-
-
-

11. sgw_ip (SGW IP)

IPv4 IPv6

SGW-C S5/S8 IP F-TEID

IPv4IPv6

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

UE S5/S8

12. ue_ip (UE IP)

UE 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

- IPv4 IPv4
- IPv6 IPv6
- PDN

13. pgw_ip (PGW IP)

PGW IPv4 IPv6

PGW-C S5/S8 IP F-TEID

IPv4IPv6

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

14. apn ()

100

PDN

DNS

```
internet
ims
mms
enterprise.corporate
```

MME

-
-
- IP

15. qci (QoS)

8

QoS

1 - 9128-254

QCI

QCI	サービス	優先度	最大ビットレート	遅延許容度	サービス
1	GBR	2	100 Mbps	10^{-2}	VoLTE
2	GBR	4	150 Mbps	10^{-3}	VoLTE
3	GBR	3	50 Mbps	10^{-3}	VoLTE
4	GBR	5	300 Mbps	10^{-6}	VoLTE
5	Non-GBR	1	100 Mbps	10^{-6}	IMS
6	Non-GBR	6	300 Mbps	10^{-6}	VoLTE
7	Non-GBR	7	100 Mbps	10^{-3}	VoLTE
8	Non-GBR	8	300 Mbps	10^{-6}	VoLTE
9	Non-GBR	9	300 Mbps	10^{-6}	VoLTE

VoLTE

9 → VoLTE

VoLTEはPGW-CでQoS

16. octets_in (VoLTE)

VoLTEは64

VoLTE → UE

VoLTE

VoLTE

1048576 → 1 MB

SGW-U PCP

- update
- end
- start 0

17. octets_out ()

64

UE →

524288 → 512 KB

SGW-U PCP

- update
- end
- start 0



Step 1: Configure Network Settings

Configure

- Network
- 5 Gbps download 10 MB upload 5 MB upload
- Network

CDR Data

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

Step 2: Configure Network Settings

Configure

- Network IPv4 + IPv6
- Network
- Network

CDR Data

```
1726570800,310260222222222,default_bearer_start,10002,1555222222,222
1726571100,310260222222222,default_bearer_update,10002,1555222222,22
1726571400,310260222222222,default_bearer_update,10002,1555222222,22
1726571700,310260222222222,default_bearer_update,10002,1555222222,22
1726572000,310260222222222,default_bearer_end,10002,1555222222,2222
```

3GPP 5G NR

Overview

- 5G NR QCI 9
- 5G NR QCI 6
- 5G NR QCI 5
- 5G NR QCI 4
- 5G NR QCI 3

CDR Data

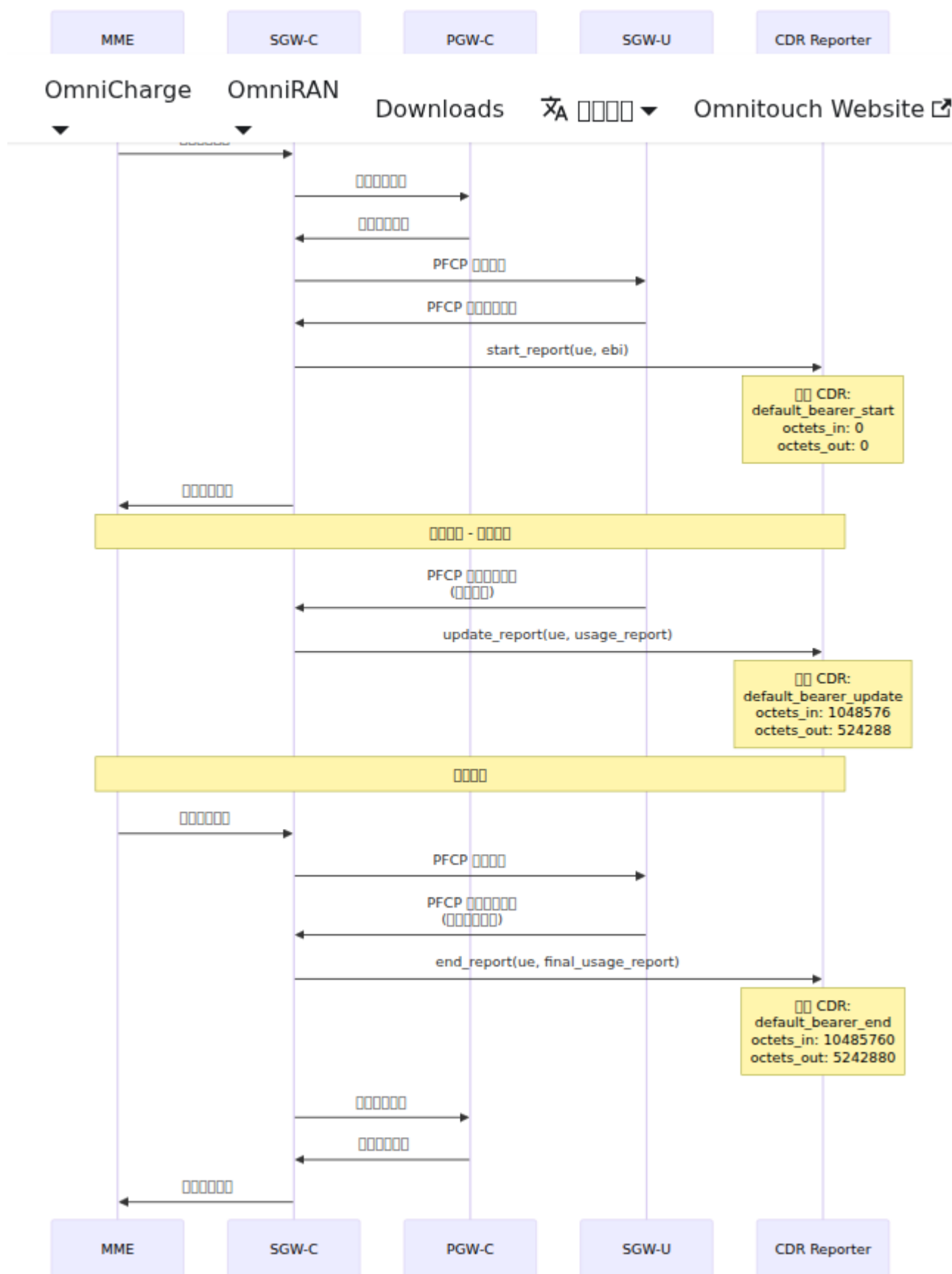
```
1726570800,3102603333333333,default_bearer_start,10003,15553333333,3333333333,1726571100,3102603333333333,dedicated_bearer_start,10004,15553333333,3333333333,1726571400,3102603333333333,default_bearer_update,10003,15553333333,3333333333,1726571400,3102603333333333,dedicated_bearer_update,10004,15553333333,3333333333,1726571700,3102603333333333,dedicated_bearer_end,10004,15553333333,3333333333,1726572000,3102603333333333,default_bearer_end,10003,15553333333,3333333333
```

Details

- 10003: 10 MB, 4 MB
 - 10004: 200 MB, 2 MB
 - QCI 9: 6 QoS
-

□□

CDR □□□□



CDR 配置

1. 配置守护进程

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

2. 配置

```
# 配置守护进程
tail -F /var/log/sgw_c/cdrs/* | process_cdr_stream
```

配置

- **配置** - 配置
- **Sxa** **配置** - 配置 SGW-U 配置
- **配置** - 配置

3GPP

- TS 32.251 - 配置PS配置
- TS 29.274 - 3GPP 配置EPS配置GTP-C 配置
- TS 29.244 - CP 配置 UP 配置PFCP
- TS 32.298 - CDR 配置

CDR 配置 - SGW-C 配置

配置 Omnitouch 配置

□□□□ 1.0 □□□□ 2025-12-10

SGW-C



runtime.exs

OmniSGW Omnitouch

- 1.
- 2.
- 3.
4. S11
5. S5/S8
6. Sxa
7. CDR
- 8.

OmniSGW config/runtime.exs

-
- MME PGW-C SGW-U
-
- CDR
-

- runtime.exs

Web UI

□□□□

□□□□

```
# config/runtime.exs
import Config

config :sgw_c,
  metrics: %{ ... },
  s11: %{ ... },
  s5s8: %{ ... },
  sxa: %{ ... },
  cdr: %{ ... }
```

□□□□

□□□□

```
config :sgw_c,  
  metrics: %{  
    # □□□□ HTTP □□  
    metrics_bind_address: "127.0.0.40",  
    metrics_port: 42068,  
  
    # □□□□□□□□□□  
    poll_interval_ms: 10000  
  }
```

□□□□

```
config :sgw_c,  
  metrics: %{  
    # □□□□□□□□□□□□  
    metrics_bind_address: System.get_env("MGT_IP") || "10.0.0.40",  
    metrics_port: 42068,  
  
    # □□□□□□□□□□□□  
    poll_interval_ms: 5000  
  }
```

□□□□

```
# □ Prometheus □□□□□□  
curl http://10.0.0.40:42068/metrics  
  
# □□□□□  
# - teid_registry_count: □□ S11/S5S8 TEID  
# - seid_registry_count: □□ PFCP □□  
# - s11_inbound_messages_total: S11 □□□□  
# - sxa_inbound_messages_total: Sxa □□□□
```

S11 📄📄📄

📄📄📄

```
config :sgw_c,  
  s11: %{  
    # S11 📄📄 IPv4 📄📄MME 📄📄  
    local_ipv4_address: "10.0.0.10",  
  
    # 📄📄📄 IPv6 📄📄📄📄  
    local_ipv6_address: nil,  
  
    # 📄📄📄📄  
    local_port: 2123,  
  
    # 📄📄📄📄  
    message_timeout_ms: 5000,  
  
    # 📄📄  
    max_retries: 3,  
    retry_backoff_ms: 1000  
  }
```

配置 S11

```
# 配置 S11
config :sgw_c,
  s11: %{
    local_ipv4_address: "10.0.0.10" # S11 地址
  }

# 配置 S11 和 Sxa
config :sgw_c,
  s11: %{
    local_ipv4_address: "10.0.0.10" # S11 地址
  },
  sxa: %{
    local_ip_address: "10.1.0.20" # Sxa 地址
  }
```

配置 S11 重试

```
config :sgw_c,
  s11: %{
    # S11 重试配置
    message_timeout_ms: 10000,
    max_retries: 5,
    retry_backoff_ms: 2000,

    # S11 重试配置
    message_timeout_ms: 3000,
    max_retries: 2,
    retry_backoff_ms: 500
  }
```

S5/S8 配置

配置

```
config :sgw_c,  
  s5s8: %{  
    # S5/S8 配置 IPv4 地址 PGW 配置  
    local_ipv4_address: "10.0.0.15",  
  
    # 配置 IPv6 地址  
    local_ipv6_address: nil,  
  
    # 配置端口  
    local_port: 2123,  
  
    # PGW-C 配置  
    pgw_peers: [  
      %{  
        ip_address: "10.0.0.20",  
        name: "pgw-c-primary"  
      },  
      %{  
        ip_address: "10.0.0.21",  
        name: "pgw-c-secondary"  
      }  
    ],  
  
    # 配置  
    message_timeout_ms: 5000,  
    max_retries: 3,  
    retry_backoff_ms: 1000  
  }
```

PGW 配置

PGW

```
config :sgw_c,  
  s5s8: %{  
    pgw_peers: [  
      %{  
        ip_address: "10.0.0.20",  
        name: "pgw-c-prod"  
      }  
    ]  
  }  
}
```

PGW

```
config :sgw_c,  
  s5s8: %{  
    pgw_peers: [  
      %{ip_address: "10.0.0.20", name: "pgw-c-1"},  
      %{ip_address: "10.0.0.21", name: "pgw-c-2"},  
      %{ip_address: "10.0.0.22", name: "pgw-c-3"}  
    ]  
  }  
}
```

PGW-

```
config :sgw_c,  
  s5s8: %{  
    pgw_peers: [  
      %{ip_address: "10.0.0.20", name: "pgw-c-primary"},  
      %{ip_address: "10.0.0.21", name: "pgw-c-backup"}  
    ]  
  }  
}
```

Sxa 配置

配置

```
config :sgw_c,  
  sxa: %{  
    # Sxa 配置 IP  
    local_ip_address: "10.0.0.20",  
  
    # 配置端口  
    local_port: 8805,  
  
    # SGW-U 配置  
    peers: [  
      %{  
        ip_address: "10.0.0.30",  
        node_id: "sgw-u-1.example.com"  
      }  
    ],  
  
    # 心跳间隔  
    heartbeat_interval_s: 20,  
  
    # 会话超时  
    session_timeout_ms: 5000,  
  
    # 重试次数  
    max_retries: 3  
  }
```

SGW-U

```
# SGW-U
config :sgw_c,
  sxa: %{
    peers: [
      %{
        ip_address: "10.0.0.30",
        node_id: "sgw-u-prod-01"
      }
    ]
  }
}
```

```
# SGW-U
config :sgw_c,
  sxa: %{
    peers: [
      %{
        ip_address: "10.0.0.30",
        node_id: "sgw-u-prod-01"
      },
      %{
        ip_address: "10.0.0.31",
        node_id: "sgw-u-prod-02"
      }
    ]
  }
}
```

□□□□

```
# □□□□□□□□
config :sgw_c,
  sxa: %{
    heartbeat_interval_s: 10,
    max_retries: 2
  }
```

```
# □□□□□□□□
config :sgw_c,
  sxa: %{
    heartbeat_interval_s: 20,
    max_retries: 3
  }
```

```
# □□□□□□□□
config :sgw_c,
  sxa: %{
    heartbeat_interval_s: 40,
    max_retries: 5
  }
```

CDR 設定

設定例

```
config :sgw_c,  
  cdr: %{  
    # CDR 設定  
    gateway_name: "sgw-c-prod-01",  
  
    # 回転間隔  
    rotation_interval_ms: 3600000, # 1 時間  
  
    # CDR 保存ディレクトリ  
    directory: "/var/log/sgw_c/cdrs"  
  }
```

設定例

```
config :sgw_c,  
  cdr: %{  
    # 設定 ID  
    gateway_name: System.get_env("HOSTNAME") || "sgw-c-prod-01",  
  
    # 回転間隔  
    rotation_interval_ms: 3600000,  
  
    # CDR 保存ディレクトリ  
    directory: System.get_env("CDR_DIR") || "/var/log/sgw_c/cdrs"  
  }
```

□□□□□

```
config :sgw_c,  
  cdr: %{  
    gateway_name: "sgw-c-prod-high-vol",  
  
    # □□□□□□□□□□□□□□  
    rotation_interval_ms: 1800000, # 30 □□  
  
    # □□□□□□□□  
    directory: "/mnt/fast-ssd/sgw_c/cdrs"  
  }
```

□□□□

□□□□□□□□

```
import Config

config :sgw_c,
  metrics: %{
    metrics_bind_address: "127.0.0.40",
    metrics_port: 42068,
    poll_interval_ms: 10000
  },
  s11: %{
    local_ipv4_address: "10.0.0.10",
    local_port: 2123,
    message_timeout_ms: 5000,
    max_retries: 3,
    retry_backoff_ms: 1000
  },
  s5s8: %{
    local_ipv4_address: "10.0.0.10",
    pgw_peers: [
      %{ip_address: "10.0.0.20", name: "pgw-c-prod"}
    ],
    message_timeout_ms: 5000,
    max_retries: 3,
    retry_backoff_ms: 1000
  },
  sxa: %{
    local_ip_address: "10.0.0.10",
    peers: [
      %{ip_address: "10.0.0.30", node_id: "sgw-u-prod-01"}
    ],
    heartbeat_interval_s: 20,
    session_timeout_ms: 5000,
    max_retries: 3
  },
  cdr: %{
    gateway_name: "sgw-c-prod-01",
    rotation_interval_ms: 3600000,
```

```
directory: "/var/log/sgw_c/cdrs"  
}
```

□□□□□□□□□□

```
import Config

sgw_s11_ip = System.get_env("SGW_S11_IP") || "10.0.0.10"
sgw_s5s8_ip = System.get_env("SGW_S5S8_IP") || "10.0.0.15"
sgw_sxa_ip = System.get_env("SGW_SXA_IP") || "10.0.0.20"
mgt_ip = System.get_env("MGT_IP") || "10.0.0.40"

config :sgw_c,
  metrics: %{
    metrics_bind_address: mgt_ip,
    metrics_port: 42068,
    poll_interval_ms: 5000
  },
  s11: %{
    local_ipv4_address: sgw_s11_ip,
    message_timeout_ms: 5000,
    max_retries: 3,
    retry_backoff_ms: 1000
  },
  s5s8: %{
    local_ipv4_address: sgw_s5s8_ip,
    pgw_peers: [
      %{ip_address: "10.0.0.20", name: "pgw-c-1"},
      %{ip_address: "10.0.0.21", name: "pgw-c-2"},
      %{ip_address: "10.0.0.22", name: "pgw-c-3"}
    ],
    message_timeout_ms: 5000,
    max_retries: 3,
    retry_backoff_ms: 1000
  },
  sxa: %{
    local_ip_address: sgw_sxa_ip,
    peers: [
      %{ip_address: "10.0.0.30", node_id: "sgw-u-1"},
      %{ip_address: "10.0.0.31", node_id: "sgw-u-2"},
      %{ip_address: "10.0.0.32", node_id: "sgw-u-3"}
    ],
    heartbeat_interval_s: 20,
    session_timeout_ms: 5000,
    max_retries: 3
  },
}
```

```
cdr: %{  
  gateway_name: System.get_env("HOSTNAME") || "sgw-c-prod-01",  
  rotation_interval_ms: 3600000,  
  directory: "/var/log/sgw_c/cdrs"  
}
```

□□□□□□

```
import Config

# □□□□□□□□□□□□□□□□
sgw_s11_ip = System.fetch_env!("SGW_S11_IP")
sgw_s5s8_ip = System.fetch_env!("SGW_S5S8_IP")
sgw_sxa_ip = System.fetch_env!("SGW_SXA_IP")
mgt_ip = System.fetch_env!("MGT_IP")
hostname = System.get_env("HOSTNAME")

# □□□□□ PGW □□□□JSON □□□
pgw_peers_env = System.get_env("PGW_PEERS", "[]")
{:ok, pgw_peers} = Jason.decode(pgw_peers_env)
pgw_peers = Enum.map(pgw_peers, &Map.to_atom/1)

# □□□□□ SGW-U □□□
sgwu_peers_env = System.get_env("SGWU_PEERS", "[]")
{:ok, sgwu_peers} = Jason.decode(sgwu_peers_env)
sgwu_peers = Enum.map(sgwu_peers, &Map.to_atom/1)

config :sgw_c,
  metrics: %{
    metrics_bind_address: mgt_ip,
    metrics_port: 42068,
    poll_interval_ms: 5000
  },
  s11: %{
    local_ipv4_address: sgw_s11_ip,
    message_timeout_ms: 5000,
    max_retries: 3,
    retry_backoff_ms: 1000
  },
  s5s8: %{
    local_ipv4_address: sgw_s5s8_ip,
    pgw_peers: pgw_peers,
    message_timeout_ms: 5000,
    max_retries: 3,
    retry_backoff_ms: 1000
  },
  sxa: %{
    local_ip_address: sgw_sxa_ip,
    peers: sgwu_peers,
```

```
heartbeat_interval_s: 20,
session_timeout_ms: 5000,
max_retries: 3
},
cdr: %{
gateway_name: hostname,
rotation_interval_ms: 1800000, # 30
directory: "/mnt/fast-ssd/sgw_c/cdrs"
}
```

SGW IP

IP

SGW IP	SGW IP	IP
SGW_S11_IP	S11 SGW IP	10.0.0.10
SGW_S5S8_IP	S5/S8 SGW IP	10.0.0.15
SGW_SXA_IP	Sxa SGW IP	10.0.0.20
MGT_IP	Management IP	10.0.0.40

環境変数

項目	説明	値
HOSTNAME	CDR 保存先	保存先
PGW_PEERS	PGW 接続先 JSON 形式	[]
SGWU_PEERS	SGW-U 接続先 JSON 形式	[]
CDR_DIR	CDR 保存先	/var/log/sgw_c/cdrs

起動

```
export SGW_S11_IP="10.0.0.10"
export SGW_S5S8_IP="10.0.0.15"
export SGW_SXA_IP="10.0.0.20"
export MGT_IP="10.0.0.40"
export HOSTNAME="sgw-c-prod-01"
export PGW_PEERS=' [{"ip_address": "10.0.0.20", "name": "pgw-c-1"} ] '
export SGWU_PEERS=' [{"ip_address": "10.0.0.30", "node_id": "sgw-u-1"} ] '

mix run --no-halt
```

確認

環境変数確認

起動確認

```
mix run --no-halt 2>&1 | grep -E "S11|S5/S8|Sxa|Metrics"
```

```
# 启动
# [info] Starting SGW-C...
# [info] Starting Metrics Exporter on 10.0.0.40:42068
# [info] Starting S11 Broker on 10.0.0.10
# [info] Starting S5/S8 Broker on 10.0.0.15
# [info] Starting Sxa Broker on 10.0.0.20
# [info] OmniSGW successfully started
```

验证

```
# 验证
curl http://10.0.0.40:42068/metrics | head -20

# 验证 S11 连接
netstat -an | grep 2123

# 验证 S11 日志
tail -f /var/log/sgw_c/sgw_c.log | grep "S11"
```

验证

"验证"

验证 验证

验证

```
# 確認する
lsof -i :2123

# 確認する
killall sgw_c
# 確認
config :sgw_c, s11: %{local_port: 2124}
```

"確認する" PGW

確認 S5/S8 確認 PGW-C

確認する

```
# 確認 PGW IP
ping 10.0.0.20

# 確認する
iptables -L | grep 2123

# 確認する
nc -u -v 10.0.0.20 2123
```

"確認する SGW-U"

確認 Sxa 確認

確認する

```
# 確認 SGW-U 確認
ping 10.0.0.30

# 確認 PFCP 確認
netstat -an | grep 8805

# 確認 PFCP 確認する
iptables -L | grep 8805
```

목차

Prometheus & Grafana 소개

OmniSGW & Omnitouch 소개

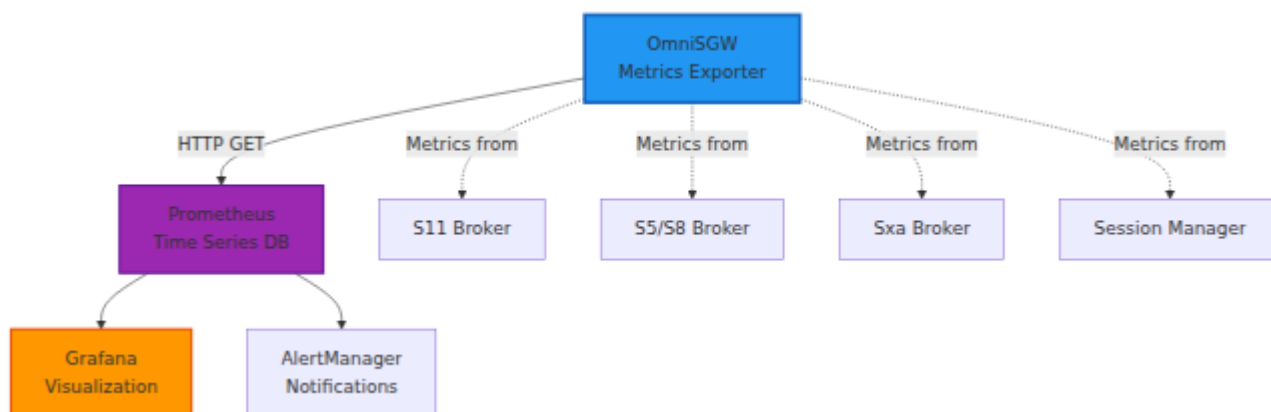
목차

1. 소개
 2. 설치
 3. 구성
 4. Prometheus 소개
 5. Grafana 소개
 6. 테스트
 7. 결론
-

목차

OmniSGW & Prometheus & Grafana & AlertManager & Session Manager & Sxa Broker & S5/S8 Broker & S11 Broker & Prometheus Time Series DB & Grafana Visualization & AlertManager Notifications

목차



□□□□

□□□□□□

□□□□

```
teid_registry_count
├─ □□□□ S11/S5S8 TEID □□
├─ □□□Gauge
├─ □□□0 □□□□□□
└─ □□□1234□1234 □□□□□□

seid_registry_count
├─ □□□□ PFCP □□□□ SGW-U □□□□
├─ □□□Gauge
├─ □□□peer_ip
└─ □□□seid_registry_count{peer_ip="10.0.0.30"} 1234

active_ue_sessions
├─ □□□□□□ UE □□
├─ □□□Gauge
└─ □□□5000

active_bearers
├─ □□□□□□□□□□ + □□□
├─ □□□Gauge
└─ □□□5500□1 □□ + □□□□ 0.1 □□□

charging_id_registry_count
├─ □□□□□□ ID
├─ □□□Gauge
└─ □□□5000
```

□□□□□

S11□**MME** □□□□

s11_inbound_messages_total

└─ Counter

└─ message_type

└─

│ └─ create_session_request

│ └─ delete_session_request

│ └─ modify_bearer_request

│ └─ create_bearer_request

│ └─ delete_bearer_request

│ └─ release_access_bearers_request

│ └─ downlink_data_notification

│ └─ echo_request

└─

s11_inbound_messages_total{message_type="create_session_request"}
5432

S5/S8 PGW

s5s8_inbound_messages_total

└─ Counter

└─ message_type

└─ S11

└─

s5s8_inbound_messages_total{message_type="create_session_request"}
4500

Sxa SGW-U

```

sxa_inbound_messages_total
├── Counter
├── message_type
├── 
│   ├── session_establishment_request
│   ├── session_modification_request
│   ├── session_deletion_request
│   ├── session_report_request
│   ├── association_setup_request
│   └── heartbeat_request
└── 
sxa_inbound_messages_total{message_type="session_report_request"}
67000

```

□□□□

□□□□□

```

s11_inbound_duration_seconds
├── Histogram
├── S11 
├── _count, _sum, _bucket
└── s11_inbound_duration_seconds_bucket{le="0.1"} 5000

s5s8_inbound_duration_seconds
├── Histogram
├── S5/S8 
sxa_inbound_duration_seconds
├── Histogram
├── Sxa 

```

PFCP □□□

```

pfcip_association_status
├── Gauge
├── 1000000 000000
├── peer_ip, node_id
└── pfcip_association_status{peer_ip="10.0.0.30"} 1

pfcip_heartbeat_latency_ms
├── Gauge
├── 000000000
├── peer_ip
└── pfcip_heartbeat_latency_ms{peer_ip="10.0.0.30"} 15

```

0000

00000

```

s11_inbound_errors_total
├── Counter
├── error_type
├── 
├── parse_error
├── validation_error
├── timeout
├── other
└── s11_inbound_errors_total{error_type="timeout"} 12

s5s8_inbound_errors_total
├── Counter
├── S5/S8 00

sxa_inbound_errors_total
├── Counter
├── Sxa 00

```

0000000

```

create_session_response_cause
├── Counter
├── cause_code
├── 3GPP
├──
├── cause_code="0":
├── cause_code="16":
├── cause_code="25":
└── cause_code="49":

```

Prometheus

```

# Prometheus
wget
https://github.com/prometheus/prometheus/releases/download/v2.45.0/pr
2.45.0.linux-amd64.tar.gz
tar xzf prometheus-2.45.0.linux-amd64.tar.gz
cd prometheus-2.45.0.linux-amd64

```

prometheus.yml:

```

global:
  scrape_interval: 15s
  evaluation_interval: 15s
  external_labels:
    monitor: 'sgw-c-prod'

scrape_configs:
  - job_name: 'sgw-c'
    static_configs:
      - targets: ['127.0.0.40:42068']
        labels:
          instance: 'sgw-c-prod-01'

  - job_name: 'sgw-c-backup'
    static_configs:
      - targets: ['127.0.0.41:42068']
        labels:
          instance: 'sgw-c-prod-02'

alerting:
  alertmanagers:
    - static_configs:
        - targets: ['127.0.0.50:9093']

```

❏ Prometheus

```

./prometheus --config.file=prometheus.yml \
  --storage.tsdb.path=/var/lib/prometheus \
  --web.console.libraries=consoles \
  --web.console.templates=console_templates

```

❏ Prometheus

http://localhost:9090

Grafana

```
# Docker
docker run -d \
  --name=grafana \
  -p 3000:3000 \
  -e GF_SECURITY_ADMIN_PASSWORD=admin \
  grafana/grafana
```

1. Grafana <http://localhost:3000>
2. →
3. → Prometheus
4. URL <http://prometheus:9090>

□ 1□

- └─ □□□□Gauge□
- └─ □□□□Gauge□
- └─ S11 □□/□□□□□
- └─ S5/S8 □□/□□□□□

□ 2□

- └─ Sxa □□/□□□□□
- └─ S11 □□ p95□□□□□
- └─ S5/S8 □□ p95□□□□□
- └─ Sxa □□ p95□□□□□

□ 3□

- └─ S11 □□/□□□□□□□
- └─ S5/S8 □□/□□□□□□□
- └─ Sxa □□/□□□□□□□
- └─ PFCP □□□□□□□

□□□□□□□□

□□□

□ 1□

- └─ S11 □□□□□□□□□
- └─ S5/S8 □□□□□□□□□
- └─ SGW-U □□□□□□□□□□□
- └─ □□□□□Gauge□

□ 2□

- └─ S11 □□□□□□□□□
- └─ S5/S8 □□□□□□□□□
- └─ Sxa □□□□□□□□□
- └─ □□□□□□□□□

□ 3□

- └─ □□□□□□□□□□□
- └─ □□□□□□□□□□□
- └─ □□□□□□□□□□□
- └─ □□□□□□□□□□□

仪表盘概览

仪表盘

```
1
├── 网络Gauge + 网络
├── 网络Gauge + 网络
├── PFCP 网络
└── APN 网络
```

```
2
├── 网络
├── 网络
├── 网络
└── 网络
```

网络概览

网络

```
teid_registry_count
```

网络

```
rate(s11_inbound_messages_total{message_type="create_session_request"
[5m])
```

S11 95th 网络

```
histogram_quantile(0.95,
rate(s11_inbound_duration_seconds_bucket[5m]))
```

网络

```
rate(s11_inbound_errors_total[5m]) +  
rate(s5s8_inbound_errors_total[5m]) +  
rate(sxa_inbound_errors_total[5m])
```

PFCP 配置

```
pfcip_association_status{peer_ip=~"10.0.0.3[0-2]"}
```

配置

配置

sgw-c-alerts.yml:

```

groups:
- name: sgw-c-alerts
  interval: 30s
  rules:
    # SGW Capacity High
    - alert: SGWCapacityHigh
      expr: (teid_registry_count / 100000) > 0.8
      for: 5m
      annotations:
        summary: "SGW 容量 80%"
        description: "teid_registry_count / 100000"

    # S11 Peer Down
    - alert: S11PeerDown
      expr: absent(s11_inbound_messages_total) > 0
      for: 2m
      annotations:
        summary: "S11 断连"

    - alert: PGWPeerDown
      expr: create_session_response_cause{cause_code="49"} > 100
      for: 2m
      annotations:
        summary: "PGW-C 断连"

    - alert: SGWUAssociationDown
      expr: pfcf_association_status == 0
      for: 1m
      annotations:
        summary: "SGW-U 断连"
        description: "peer_ip"

    # S11 Latency High
    - alert: S11LatencyHigh
      expr: histogram_quantile(0.95,
rate(s11_inbound_duration_seconds_bucket[5m])) > 1
      for: 5m
      annotations:
        summary: "S11 延迟 1s"
        description: "p95: {{ $value }}s"

    - alert: S5S8LatencyHigh
      expr: histogram_quantile(0.95,

```

```

rate(s5s8_inbound_duration_seconds_bucket[5m])) > 1
  for: 5m
  annotations:
    summary: "S5/S8 错误 1 次"

# 配置
- alert: S11ErrorRate
  expr: rate(s11_inbound_errors_total[5m]) > 10
  for: 3m
  annotations:
    summary: "S11 错误"
    description: "{{ $value }}" errors/sec

- alert: SessionEstablishmentFailure
  expr: rate(create_session_response_cause{cause_code!="0"}
[5m]) > 20
  for: 3m
  annotations:
    summary: "会话建立失败"
    description: "{{ $value }}" failures/sec

```

配置 AlertManager

alertmanager.yml:

```
global:
  resolve_timeout: 5m

route:
  receiver: 'sgw-alerts'
  group_by: ['alertname', 'instance']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 12h

receivers:
  - name: 'sgw-alerts'
    webhook_configs:
      - url: 'http://slack-webhook-url'
    email_configs:
      - to: 'noc@example.com'
        from: 'sgw-alerts@example.com'
        smarthost: 'smtp.example.com:587'
```

0000000

Slack 000

```
0 SGW 000
0000000
00000085,000 / 100,000 (85%)
0002025-12-10 15:30:00 UTC
0000000000
```

0000000

[[[[]]] S11 [[[[[[[[

SGW-C S11 [[[[2 [[[[[[[[[[[[[[
[[[[[[[[

- MME [[[[[[[[
- [[[[SGW-C
- S11 [[[[[[[[

[[[[[[[[[[S11 [[

[[[[[[

[[[[[[

[[[[[[[[[[[[404

[[[[

```
# [[[[[[[[[[[[[[[[[[
curl -v http://127.0.0.40:42068/metrics

# [[[[[[[[[[[[[[[[[[
tail -f /var/log/sgw_c/sgw_c.log | grep -i metric

# [[[[[[
cat config/runtime.exs | grep metrics
```

[[[[[[

1. [[SGW-C [[
2. [[[[IP/[[[[[[[[[[[[[[
3. [[[[[[[[[[
4. [[[[[[[[[[[[[[[[[[

環境構築

OS: S11 環境: S5/S8 Sxa 環境

環境

1. 環境構築
2. 環境構築
3. 環境構築

環境

- 環境構築
- 環境構築
- 環境構築

環境構築

環境 環境構築

環境

```
# 環境構築
ps aux | grep sgw_c | grep -v grep | awk '{print $6}'

# 環境構築
watch -n 5 'ps aux | grep sgw_c'
```

環境

1. 環境構築
 2. 環境構築
 3. 環境構築
 4. 環境構築
-

□□□□

□□□□

- □□□□ 15-30 □□□□□
- □□□ 15-30 □□□□□
- □□□ □□□□□□□
- □□□ □□□□□□□□□□□□□□□□

□□□□□

- □□□□ □□□□□□□□□□□□□
- □□□ □□ → □□ → □□
- □□□ □□□□□□□□□□
- □□□ □□□□□□

□□□□

- □□□ □□ → □□ → □□
- □□□ □□□□□□□□□□□□
- □□□□□ □□□□ +20% □□□□
- □□□□□ □□□□□□□□

❖❖□□□ - OmniSGW □□□□□□□

▮ Omnitouch □□□□□□

□□□□□ 1.0 □□□□□ 2025-12-10

S11 接口

MME 与 GTP-C 接口

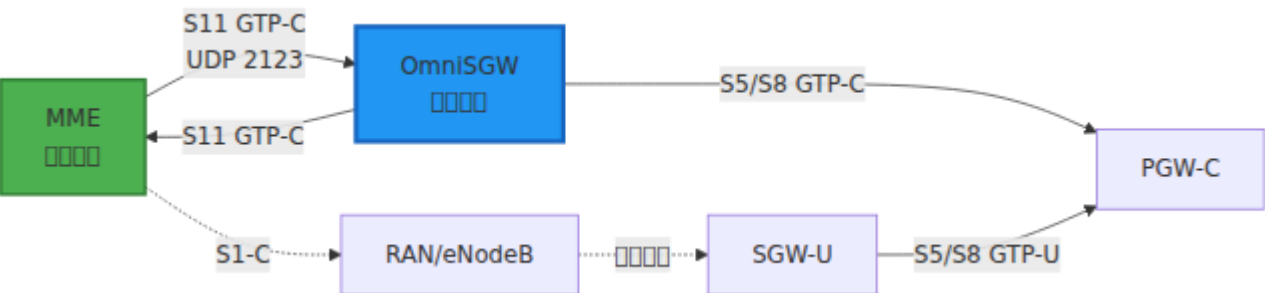
OmniSGW 与 Omnitouch 接口

接口

- 1. 接口
- 2. 接口
- 3. 接口
- 4. 接口
- 5. 接口
- 6. 接口
- 7. 接口
- 8. 接口
- 9. 接口

接口

S11 接口 与 GTP-C v2 接口 GPRS 接口 - 接口 OmniSGW 接口 MME 接口 UE 接口



概要

- **GTP-C v2** プロトコル - 3GPP TS 29.274
- **TEID** (Tunnel Endpoint Identifier) - トンネルの識別子
- ネットワーク - ユーザー UE 間の接続
- ネットワーク - ネットワーク MME (Mobility Management Entity) 間の接続
- ネットワーク - ネットワーク MME (Mobility Management Entity) 間の接続
- ネットワーク - ネットワーク MME (Mobility Management Entity) 間の接続

仕様

GTP-C プロトコル 2

- プロトコル GTP-C v2 (3GPP TS 29.274)
- プロトコル UDP
- ポート 2123
- ネットワーク MME (Mobility Management Entity)
- ネットワーク MME (Mobility Management Entity)

TEID (Tunnel Endpoint Identifier)

TEID (Tunnel Endpoint Identifier) は、トンネルの識別子です。

- **TEID** - ネットワーク MME (Mobility Management Entity) 間の接続
- **TEID** - ネットワーク MME (Mobility Management Entity) 間の接続

ネットワーク

MME → SGW: ネットワーク MME (Mobility Management Entity) 間の接続

SGW → MME: ネットワーク MME (Mobility Management Entity) 間の接続

結論

ネットワーク S11 ネットワーク GTP-C v2 プロトコル

GTP-C 消息 (12-16 字节)

- └─ 标志 (3 字节): 0x2 (GTP-C v2)
- └─ 序列号 (1 字节)
- └─ TEID 消息 (1 字节): 1 (TEID 消息)
- └─ 消息类型 (8 字节): 消息类型
- └─ 消息长度 (16 字节): 消息长度
- └─ TEID (32 字节): 消息长度
- └─ 消息 (24 字节): 消息/消息
- └─ 消息 (8 字节): 消息 0

消息 (消息)

- └─ 消息 (IE)
- | └─ IE 消息 (8 字节)
- | └─ 消息 (16 字节)
- | └─ 消息 (消息)
- └─ ... 消息 IE

□□

□□□□

```
# config/runtime.exs
config :sgw_c,
  s11: %{
    # S11 □□□□ IPv4 □□
    local_ipv4_address: "10.0.0.10",

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

    # □□□□□□□□
    local_port: 2123,

    # □□□□
    message_timeout_ms: 5000,

    # □□□□
    max_retries: 3,
    retry_backoff_ms: 1000
  }
```

□□□□

□□□□□□

```
# □□□□ MME □□□ GTP-C□□□□
iptables -A INPUT -p udp --dport 2123 -s <mme_network>/24 -j
ACCEPT

# □□□ MME □□□ GTP-C
iptables -A OUTPUT -p udp --dport 2123 -d <mme_network>/24 -j
ACCEPT
```

□□□

```
# MME
ip route add <mme_network>/24 via <gateway_ip> dev eth0
```

```
# MME GTP
# "S11 Broker connected"

# S11
curl http://127.0.0.40:42068/metrics | grep teid_registry_count
```

S11



S11

MME → OmniSGW

UE PDN

IE 名称	格式	说明
IMSI	IMSI	国际移动用户识别码
MSISDN	BCD	移动用户电话号码
MEI	IMSI	移动设备识别码
RAT 名称	名称	无线接入技术 EUTRAN
网络名称	名称	网络名称
UE 名称	名称	UE 名称
ULI	名称	用户本地标识符 TAI, ECGI
网络名称	PLMN	MCC/MNC
APN	名称	接入点名称

网络名称

IE 名称	格式	说明
名称	名称	网络名称/名称
网络名称	名称	网络名称 TEID
PDN 名称	名称	PGW 名称 IP 名称
APN 名称	名称	APN 名称

网络名称

网络名称 S11

UE MME → OmniSGW MME UE

UE MME MME

MME MME

IE	UE	UE
MEI	UE	UE
ULI	UE	UE
UE	UE	UE
TAI	TAI	UE
ECGI	ECGI	E-UTRAN UE

UE MME

IE	UE	UE
UE	UE	UE
UE	UE	UE

UE MME

UE MME

UE MME SGW UE

UE MME QoS MME

UE MME

- UE MME

- 
- 

000000/00

□□□ □□□ MME □ SGW□□□ PGW□□□

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □

00000000/00

□□□ MME → OmniSGW

[illegible]

111

- 1000000000000000000
- 100000000000000000
- 1 SGW-U 1000000000000000000
- UE 1000000000000000000

0000/00

```

graph LR
    OmniSGW -- "S-MB-MME" --> MME
    MME -- "MME-S-MB" --> OmniSGW

```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □

□□□□□□**S11**□

```
PGW-C → OmniSGW → MME
```

□□□ □□ MME □□ UE □□□□□□□□

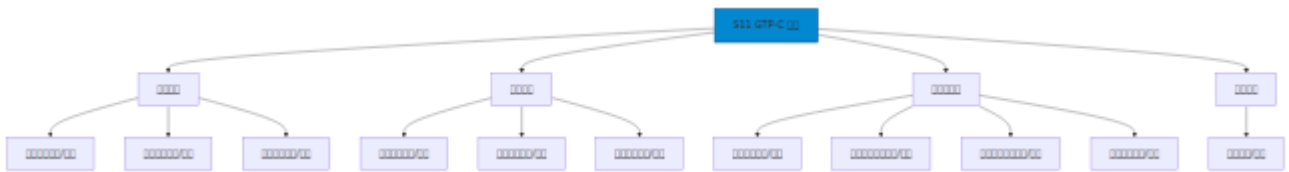
□□□□□□

IE 項目	項目	項目
EBI	項目	EPS 項目 ID
IMSI	項目	項目

項目 項目

項目

UE 項目

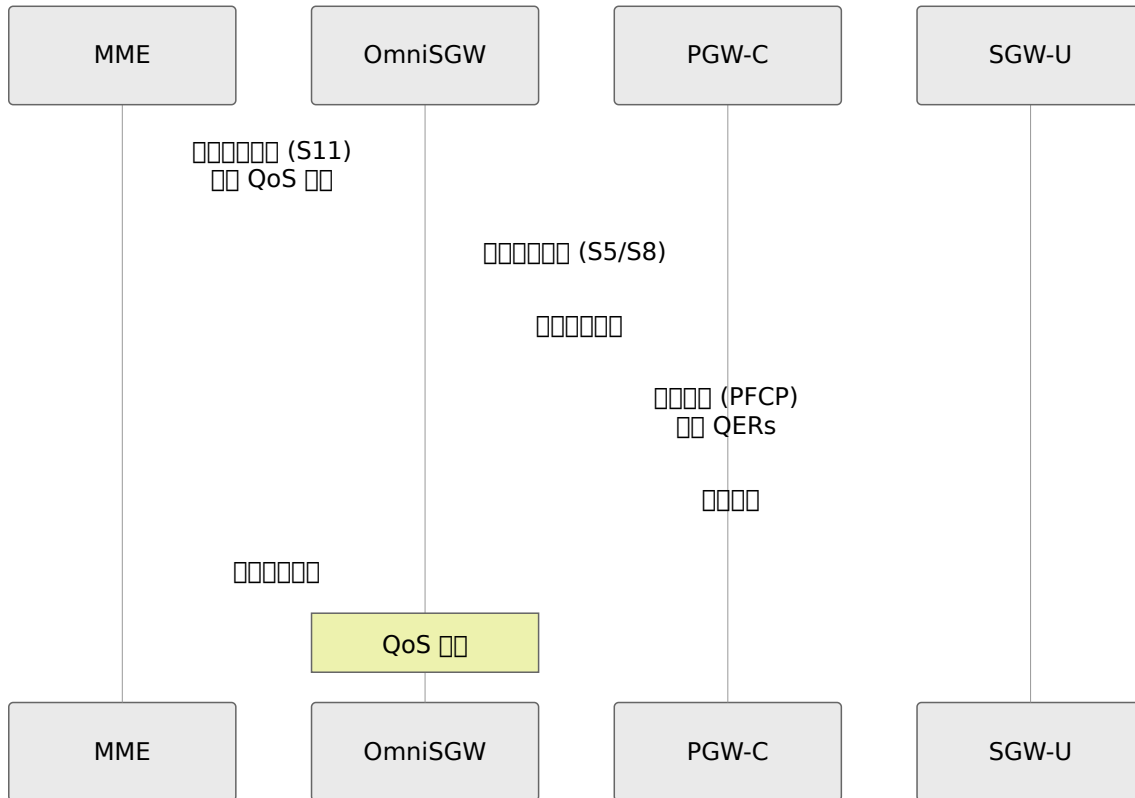


項目

[UE 項目]
↓ (項目)
[項目 PGW 項目]
↓ (PGW 項目)
[項目]
↓ (PCF 項目)
[項目]

QoS

QoS



TAU

SGW TAU

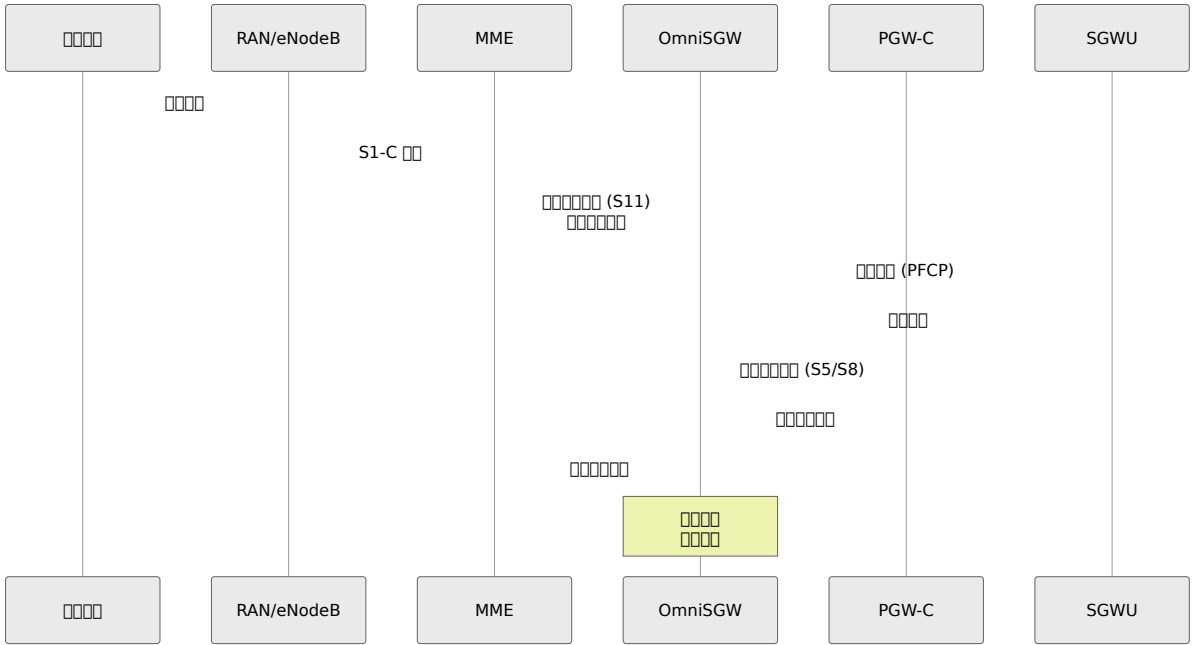
- MME UE
- ULI/TAI SGW
- SGW UE
- TAU

SGW TAU

- SGW MME
- SGW
- SGW SGW
- SGW

□□□□

□□□□□□



□□□□□

```
[□□□□]
  ↓ (□□□□□□)
[□□□□□□]
  ↓ (PFCP □□□□□)
[□□ PGW]
  ↓ (PGW □□□□□)
[□□□□]
```

□□□□

□□□□□

□□□□ S11 □□□□□

```
# S11
watch -n 1 'curl -s http://127.0.0.40:42068/metrics | grep
s11_inbound'

#
#
s11_inbound_messages_total{message_type="create_session_request"}
1245
#
s11_inbound_messages_total{message_type="delete_session_request"}
1200
# s11_inbound_messages_total{message_type="modify_bearer_request"}
3450
```

S11

Web UI → UE

- IMSI
- TAI
- TEID S11
- TEID MME
- QoS
- PGW-C

```
#
curl -s http://127.0.0.40:42068/metrics | grep
modify_bearer_request_total

#
# "TAU with SGW change"
```

테스트

테스트 방법

테스트 방법

테스트

- 1. Web UI → UE 테스트 방법
- 2. 테스트 `s11_inbound_errors_total`
- 3. 테스트 방법

테스트 방법

번호	이름	설명
16	테스트	테스트 SGW-U 테스트 PFCP 테스트
25	IE 테스트	테스트 방법
49	PGW 테스트	테스트 PGW-C 테스트 S5/S8 테스트
65	테스트 APN	테스트 APN 테스트

테스트 방법

테스트 "테스트 TEID"

테스트

```
# 테스트 TEID 테스트
curl -s http://127.0.0.40:42068/metrics | grep teid_registry_count

# 테스트 TEID 테스트
# Web UI → UE 테스트 → 테스트 IMSI 테스트
```

準備

- 仮想マシンを起動
- 仮想 TEID を取得
- 仮想 MME から仮想 TEID を取得

実行

仮想マシンを起動

仮想

1. 仮想マシンを起動/停止
2. 仮想マシン "TAU" を "TAU" に変更
3. 仮想マシン PFCP を実行

確認

- SGW-U から仮想マシンを起動
- 仮想マシンを起動
- 仮想マシンを起動

結果

仮想マシン S11 を実行

仮想マシン

```
# 仮想マシンを起動
curl -s http://127.0.0.40:42068/metrics | grep
s11_inbound_duration_seconds

# 仮想マシンを起動
curl -s http://127.0.0.40:42068/metrics | grep active_ue_sessions

# 仮想マシンを起動
curl -s http://127.0.0.40:42068/metrics | grep active_bearers
```

目標

1. 環境構築
2. 3GPP PFCP 実装
3. SGW-C 実装
4. CPU 負荷削減

監視 Prometheus 実装

概要

環境

- S11 実装
- RTT 実装
- 負荷削減

実装

- 環境構築
- MME 実装
- S11 実装
- 負荷削減

検証

- S11 実装
 - S11 MME IP
 - 負荷削減
-

□□□□□□

□□	□□	□□	□□□
□□□□□□/□□	MME → SGW	□□□□	□
□□□□□□/□□	MME → SGW	□□□□	□
□□□□□□/□□	MME ↔ SGW	QoS □□□TAU	□
□□□□□□/□□	MME ↔ SGW	□□□□	□
□□□□□□/□□	MME ↔ SGW	□□□□	□
□□□□□□□□/□□	MME → SGW	□□□□	□
□□□□□□□□/□□	MME ↔ SGW	□□□□	□
□□□□□□/□□	SGW → MME	□□□□	□
□□□□/□□	MME ↔ SGW	□□□□	□

S11 □□ - MME □ SGW-C □□□□□□

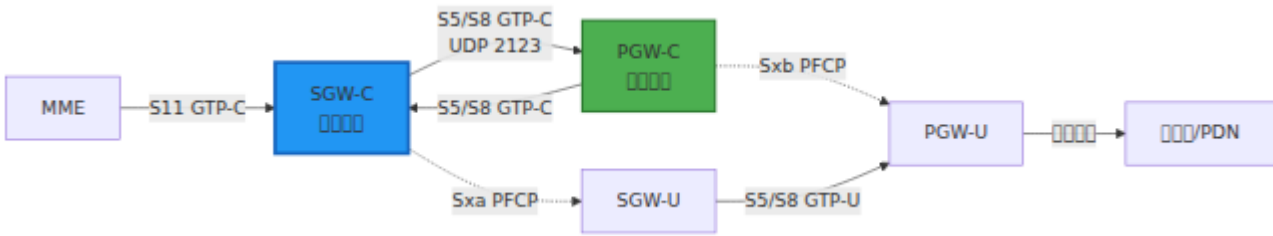
S5/S8 配置

PGW-C GTP-C

OmniSGW Omnitouch

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.

S5/S8 GTP-C v2GPRS - OmniSGW PGW-C PDN



TEID

- **GTP-C v2** TEID - TEID
- **TEID** TEID - TEID
- **PDN** TEID - TEID/TEID/TEID PDN TEID
- TEID - TEID
- **ID** TEID - TEID
- **IP** TEID - PGW TEID UE IP

TEID

GTP-C TEID 2

- TEID GTP-C v2 (3GPP TS 29.274)
- TEID UDP
- TEID 2123TEID
- TEID TEID
- TEID TEID/TEID

TEIDTEIDTEIDTEIDTEID

TEID PDN TEIDTEIDTEIDTEIDTEID TEID

- **SGW TEID** - SGW-C TEID PGW TEID S5/S8 TEID
- **PGW TEID** - PGW-C TEID SGW TEID S5/S8 TEID

TEID

SGW-C → PGW-CTEIDTEID PGW-C TEID

PGW-C → SGW-CTEIDTEID SGW-C TEID

TEID ID

TEID ID TEIDTEIDTEIDTEIDTEID

- PGW-C
 - SGW-C CDR
 - SGW PGW CDR
 - 32 PDN
-

```
# config/runtime.exs
config :sgw_c,
  s5s8: %{
    # S5/S8 IPv4
    local_ipv4_address: "10.0.0.15",

    # IPv6
    local_ipv6_address: nil,

    #
    local_port: 2123,

    # PGW-C
    pgw_peers: [
      %{
        ip_address: "10.0.0.20",
        name: "pgw-c-primary"
      },
      %{
        ip_address: "10.0.0.21",
        name: "pgw-c-secondary"
      }
    ],

    #
    message_timeout_ms: 5000,
    max_retries: 3,
    retry_backoff_ms: 1000
  }
```

□□□□

□□□□□□

```
# □□□□ PGW-C □□□ GTP-C
iptables -A INPUT -p udp --dport 2123 -s <pgw_network>/24 -j
ACCEPT

# □□□ PGW-C □□□ GTP-C
iptables -A OUTPUT -p udp --dport 2123 -d <pgw_network>/24 -j
ACCEPT
```

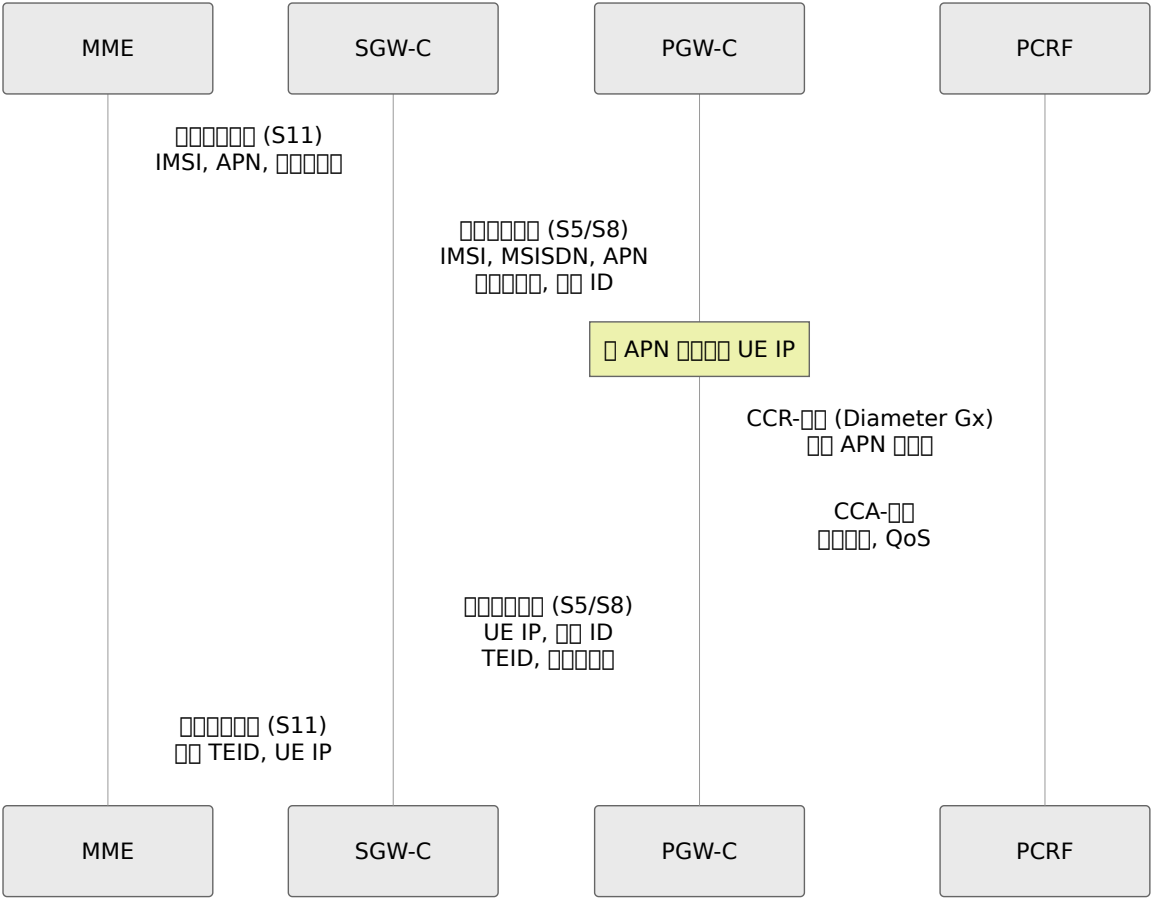
□□□

```
# □□□ PGW-C □□□□□□
ip route add <pgw_network>/24 via <gateway_ip> dev eth0
```

□□□□□

□□ **PDN** □□□□

□ MME □□ S11 □□ PDN □□□□SGW-C □□ S5/S8 □□□ PGW-C□



SGW-C → PGW-C

IE 項目	項目	項目
IMSI	MME	国際移動番号
MSISDN	MME	電話番号
MEI	MME	端末識別子
QoS 値	MME	QoS 値 QCI, ARP
APN	MME	インターネットプロキシアドレス, ims, mms
PLMN	MME	PLMN 識別子 MCC/MNC
RAT 種別	MME	無線アクセス技術 EUTRAN
ULI	MME	ユーザローカル識別子 TAI, ECGI
IP ID	SGW	SGW 識別子

PGW-C → SGW-C

IE 名称	名称	内容
网络	PGW	网络/网络
网络地址	PGW	网络地址 TEID
PDN 名称	PGW	网络 UE IP 地址
APN 名称	PGW	网络 APN 名称
网络 ID	PGW	PGW 网络 ID
TEID	PGW	网络 S5/S8 网络

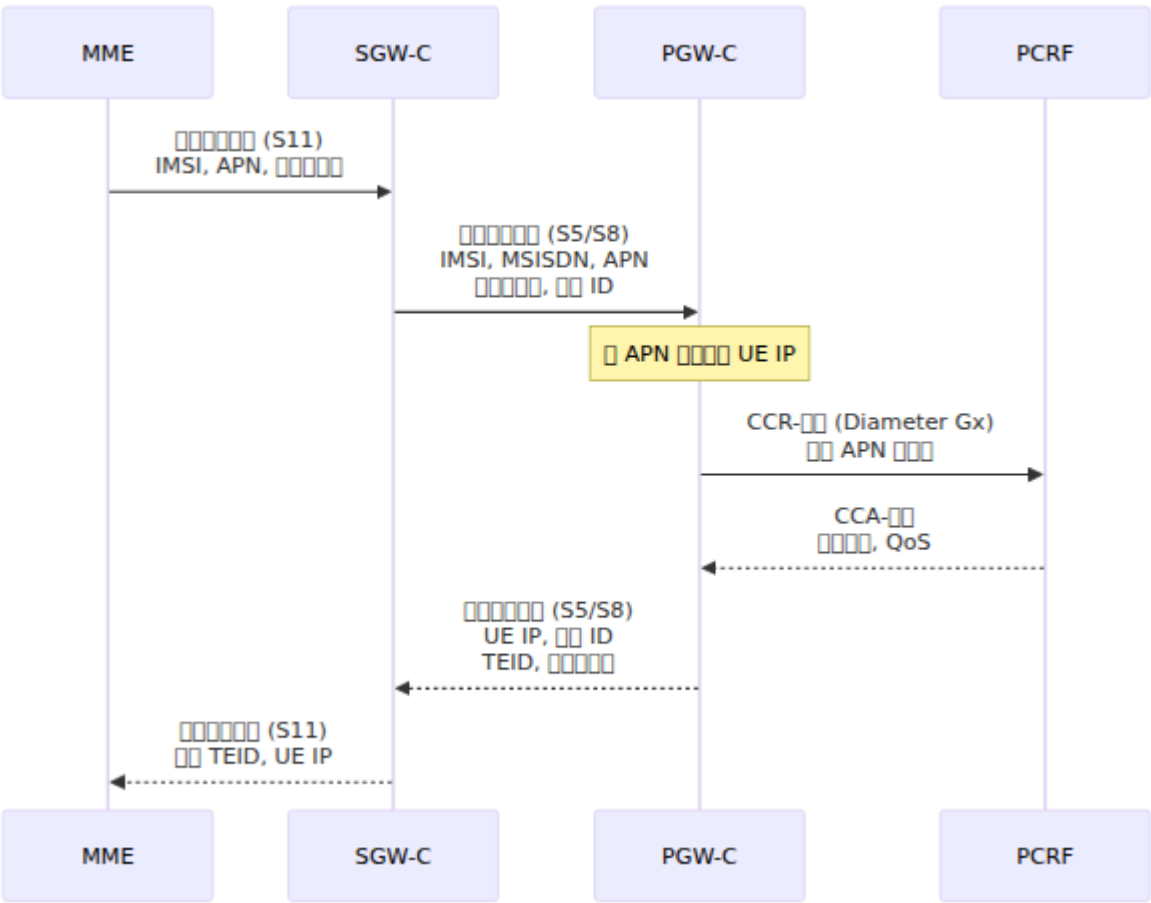
网络

网络	名称	内容
0	网络	网络
16	网络	网络 MME 网络
25	IE 网络	网络
49	网络	PGW-C 网络
64	网络	网络
65	网络	PGW 网络
72	网络/网络 IE	网络

□□□□

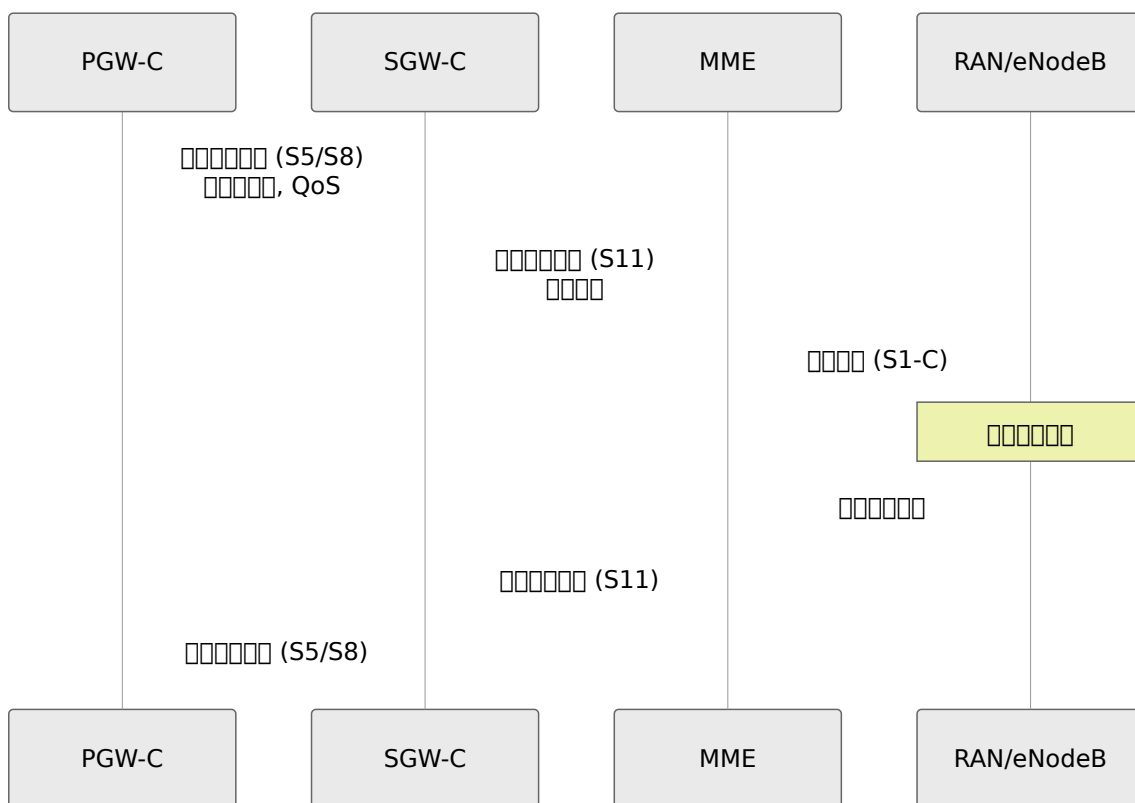
□□ QoS □□

□ MME □□ S11 □□ QoS □□□□SGW-C □□ S5/S8 □□□ PGW-C□



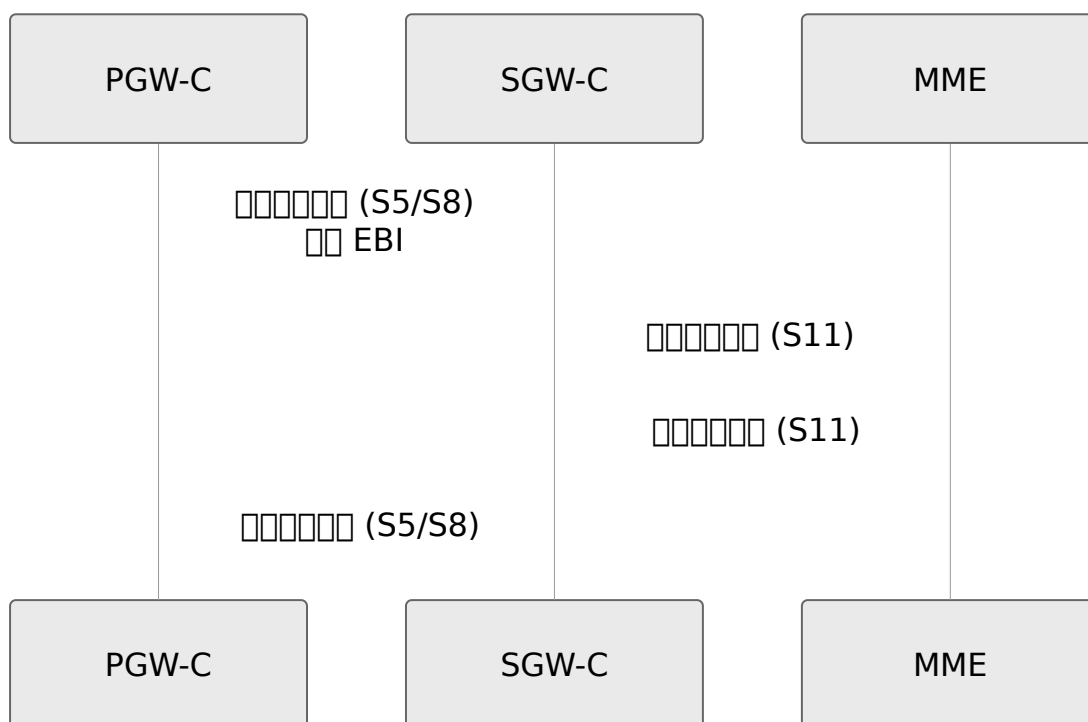
□□□□□□□□□□

PGW-C □□□□ S5/S8 □□□□□□□□□□



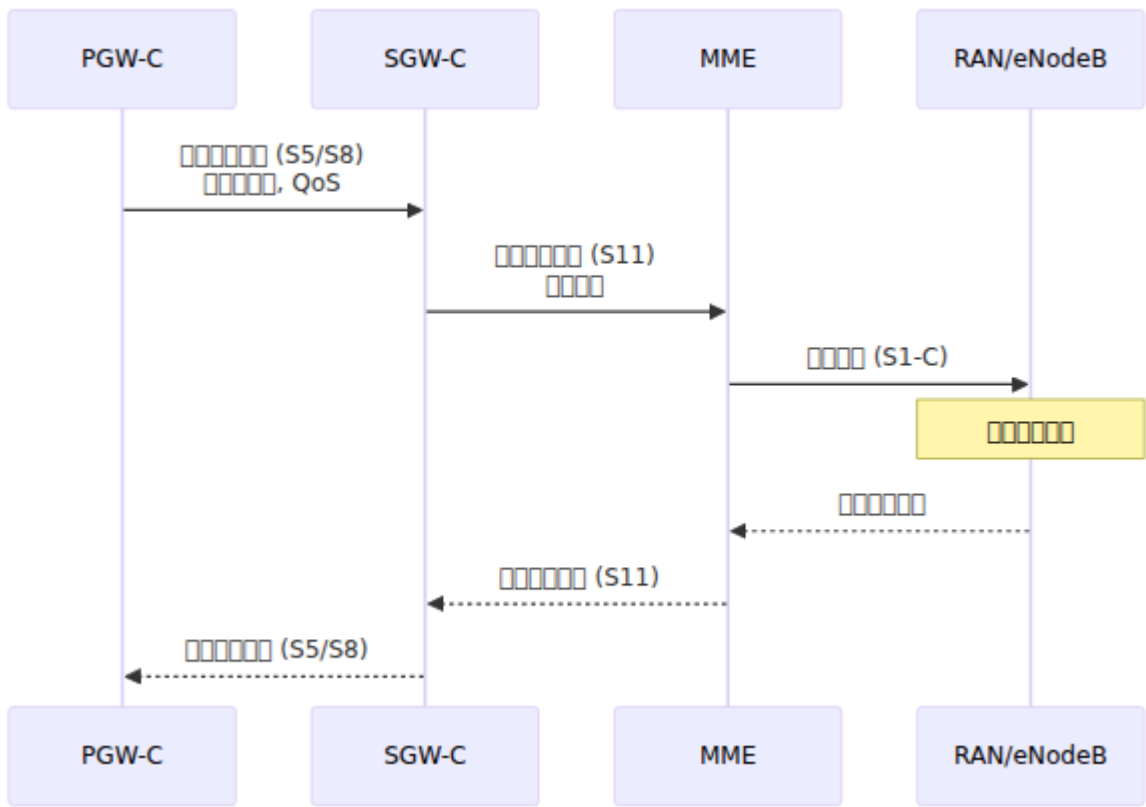
Network Architecture

Network Architecture



□□□□

□□ **PDN** □□□□

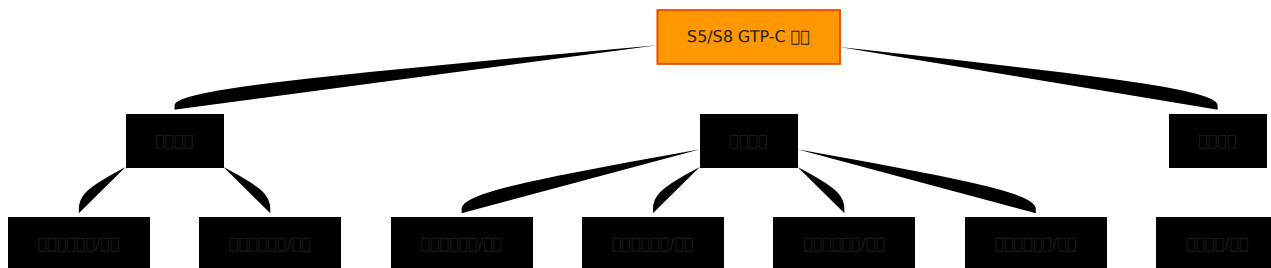


□□□□□

[PDN □□]
↓ (□□ MME □□□□□□□)
[□□ PGW □□]
↓ (□□ PGW □□□□)
[□□ SGW □□]
↓ (TEID □□□□CDR □□)
[PDN □□]

□□□□

S5/S8 □□□□



□□□□

□□□□□□/□□

- □□□□ □□□□□□PDN □□□□
- □□□ □□ UE □□ PDN □□□□ 1 □
- □□□□ □□

□□□□□□/□□

- □□□□ □□□□□□PDN □□□□
- □□□ □□ PDN □□□□□□ 1 □
- □□□□ □□

□□□□□□/□□

- □□□□ QoS □□□□□□□□
- □□□ □□□□□□□□□□ 0 □□□□
- □□□□ □□

□□/□□□□□□□□/□□

- □□□□ □□□□□□□□□□/□□
- □□□ □□□□□□□□□□ 0 □□□□
- □□□□ □□

□□□□□□/□□

- 5G/4G/LTE
- 5G/LTE 1 5G
- 5G 5G

5G

5G

5G PGW-C 5G

```
# 5G S5/S8 TEID
curl -s http://127.0.0.40:42068/metrics | grep s5s8_teid

# 5G S5/S8 5G
curl -s http://127.0.0.40:42068/metrics | grep
s5s8_inbound_messages_total

# 5G/4G/LTE/5G
```

PDN 5G

5G PDN 5G

```
Web UI → UE 5G
├── 5G UE 5G
│   ├── 5G PGW-C 5G
│   ├── 5G ID 5G PGW
│   ├── UE IP 5G PGW
│   ├── 5G QoS 5G
│   └── S5/S8 TEID 5G
```

5G

5G S5/S8 5G

```
# 创建会话
curl -s http://127.0.0.40:42068/metrics | grep
create_session_request_total

# 修改 Bearer
curl -s http://127.0.0.40:42068/metrics | grep modify_bearer

# 5s8 入站错误
curl -s http://127.0.0.40:42068/metrics | grep
s5s8_inbound_errors_total
```

PGW 配置

配置 PGW-C 配置

```
配置
├── 配置 PGW-C 配置
├── 配置 APN 配置 PGW
├── 配置-配置 PGW-C 配置
└── 配置 PGW-C 配置
```

配置

```
# PGW 配置
curl -s http://127.0.0.40:42068/metrics | grep session_by_pgw_peer
```

配置

配置

配置 "配置 PGW 配置"

配置

```
# 创建会话
curl -s http://127.0.0.40:42068/metrics | grep
create_session_response_cause

# PGW 状态
curl -s http://127.0.0.40:42068/metrics | grep s5s8_peer_status
```

配置项说明

配置项	说明	配置项
16	创建会话	PGW IP 地址
25	创建会话	PGW 名称
49	创建会话	PGW APN 名称
72	PGW IE	MME 名称

配置项说明

配置项 "创建会话"

配置项

- 创建会话
- PGW QoS 配置
- PGW 名称

配置项

- PGW QoS 配置
- PGW 名称 QoS
- PGW 名称 PCRF/名称

🔍 🔍 🔍 🔍 🔍

🔍 "S5/S8 🔍 🔍"

🔍

```
# 🔍  
curl -s http://127.0.0.40:42068/metrics | grep  
s5s8_inbound_duration_seconds  
  
# 🔍  
curl -s http://127.0.0.40:42068/metrics | grep s5s8_timeout_total
```

🔍 🔍 🔍 🔍

- 🔍 RTT 🔍 🔍 message_timeout_ms
- 🔍 🔍 🔍
- 🔍 PGW CPU/🔍 🔍 🔍
- 🔍 🔍 🔍 🔍

🔍 **ID** 🔍 🔍

🔍 "CDR 🔍 🔍 ID 🔍"

🔍

- 🔍 PGW 🔍 🔍 🔍 🔍 🔍 ID
- 🔍 CDR 🔍 🔍 🔍 🔍 🔍 ID
- 🔍 SGW 🔍 PGW CDR

🔍 🔍 🔍 🔍

- 🔍 PGW 🔍 🔍 🔍 🔍 🔍 ID
- 🔍 CDR 🔍 🔍 🔍 🔍 🔍 🔍 🔍 ID

🔍 🔍 🔍 🔍 🔍 Prometheus 🔍 🔍 🔍 🔍 🔍 🔍 🔍 🔍

□□□□

□□

- **PGW** □□□ □□□□ PGW-C □□□□□□□□□□
- □□□□□ □□□□□□□□□□
- □□□ □□ WAN RTT □□□□□□□□5-10 □□
- □□□ 2-3 □□□□□□□□□□

□□

- □□□□□□ □□□□□□□□□□
- **APN** □□□ □□ SGW APN □□□□ PGW APN □□
- □□□□□ □□□□ S5/S8 □□□□□□□□
- □□□□□ □□ PGW □□ IP □□□□□□

□□□□

- □□□□□ □□□□□□□□ PGW □□□□□□
 - □□□□□ □□□□□ + □□□□□□□□
 - **QoS** □□□ □□ PGW □□□□ QoS □□
 - □□□ □□□□□□□□□□□□ ID
-

□□□□□□□□

S11 ↔ S5/S8 □□

S11 □□□□□□ MME□
↓
SGW-C □□□□
↓
S5/S8 □□□□□□ PGW-C□
↓
□□□□
↓
S11 □□□□□□ MME□

S5/S8 ↔ Sxa □□

S5/S8 □□□□□□□□ PGW□
↓
□□□□/QoS □□
↓
Sxa □□□□□□□□ SGW-U□
↓
□□□□□□□□□□
↓
□□ S11 □□□□□□



UE □□□□□□□□

OmniSGW □ *Omnitouch* □□□□□□



1. □□
 2. □□□□□□
 3. □□□□
 4. □□□□
 5. □□□□□□
 6. □□□□
 7. □□□□
 8. □□□□
 9. □□□□
-

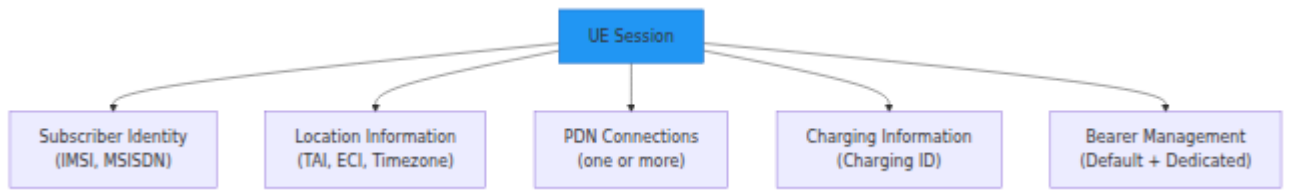


UE □□□□□□□□□□□□□□□□ SGW-C □□□□□□□□□□□□□□□□

- **MME** - □□□□□□□□ S11□
- **PGW-C** - □□□□□□□□□□ S5/S8□
- **SGW-U** - □□□□□□□□ Sxa□

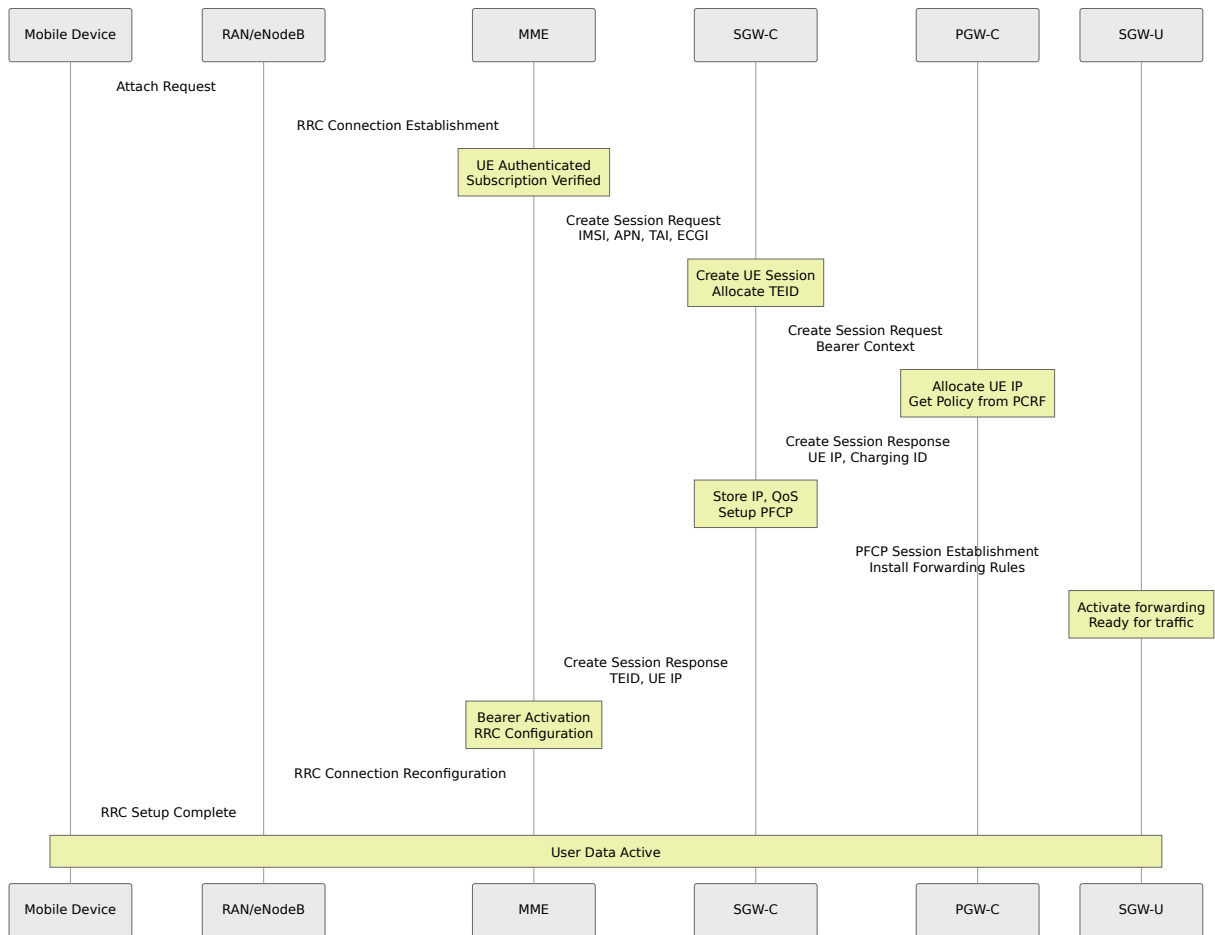
□□□□□□□□□□ IMSI□□□□□□□□□□□□□□□□ PDN □□□

□□□□

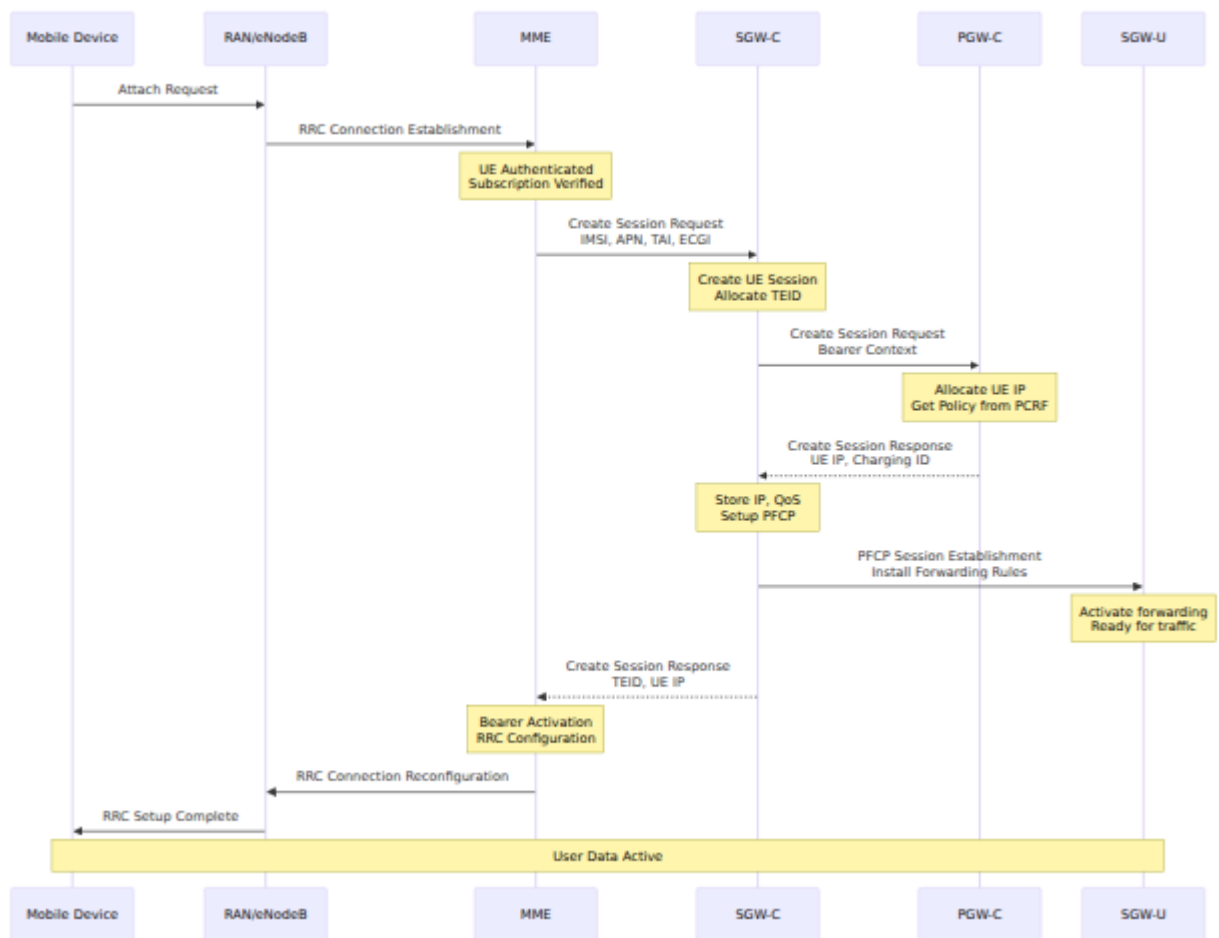


□□□□□□

□□□□□UE□□□



UE



□□□□

UE □□□□□

```
[No Session]
  ↓ (Create Session Request from MME)
[Creating Session - PGW]
  ↓ (Create Session Response from PGW)
[Creating Session - User Plane]
  ↓ (PFCP Session Establishment Response)
[Session Active]
  ↓ (Modify Bearer Request or bearer changes)
[Session Modifying]
  ↓ (Modification Complete)
[Session Active]
  ↓ (Delete Session Request or network error)
[Session Terminating]
  ↓ (All responses received, CDR logged)
[Session Terminated]
```

Session State

Session State:

- └─ IMSI: Mobile subscriber identity
- └─ GUTI: Temporary ID from MME
- └─ Location:
 - └─ TAI: Current tracking area
 - └─ ECI: Current cell
 - └─ Timezone: UE timezone
- └─ PDN Connections: Array of PDN connection contexts
 - └─ APN: Access Point Name
 - └─ TEID (S11): To MME
 - └─ TEID (S5/S8): To PGW-C
 - └─ Charging ID: From PGW-C
 - └─ UE IP: From PGW-C
 - └─ PGW-C Address: S5/S8 peer
 - └─ Bearers: Default + Dedicated
 - └─ EBI: Bearer ID
 - └─ QCI: QoS class
 - └─ ARP: Priority
 - └─ GBR: Guaranteed rate
 - └─ MBR: Maximum rate
- └─ Charging: Charging ID, event log

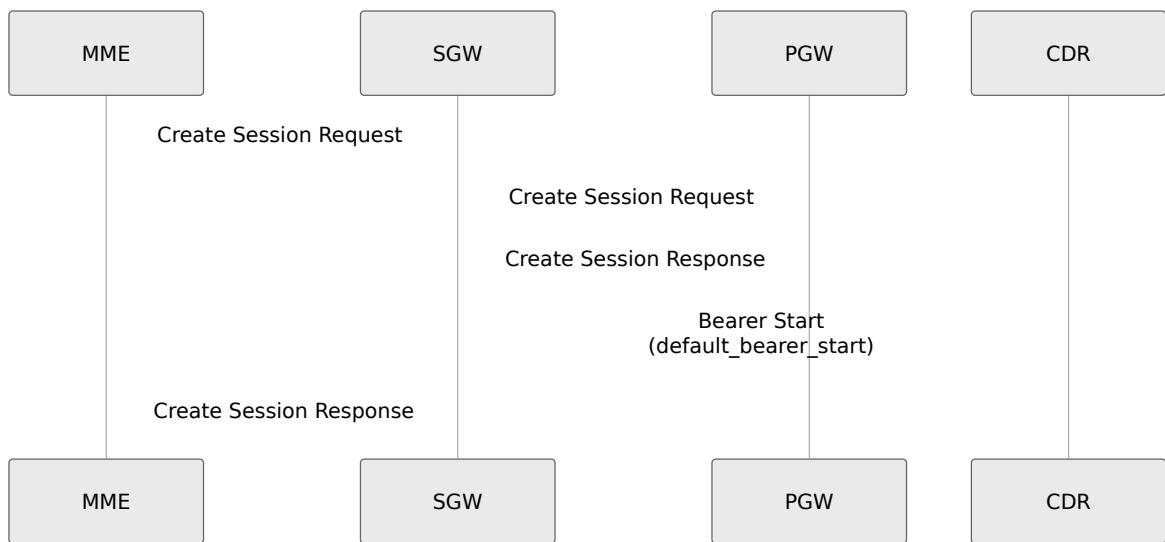
PDN Context

PDN Context

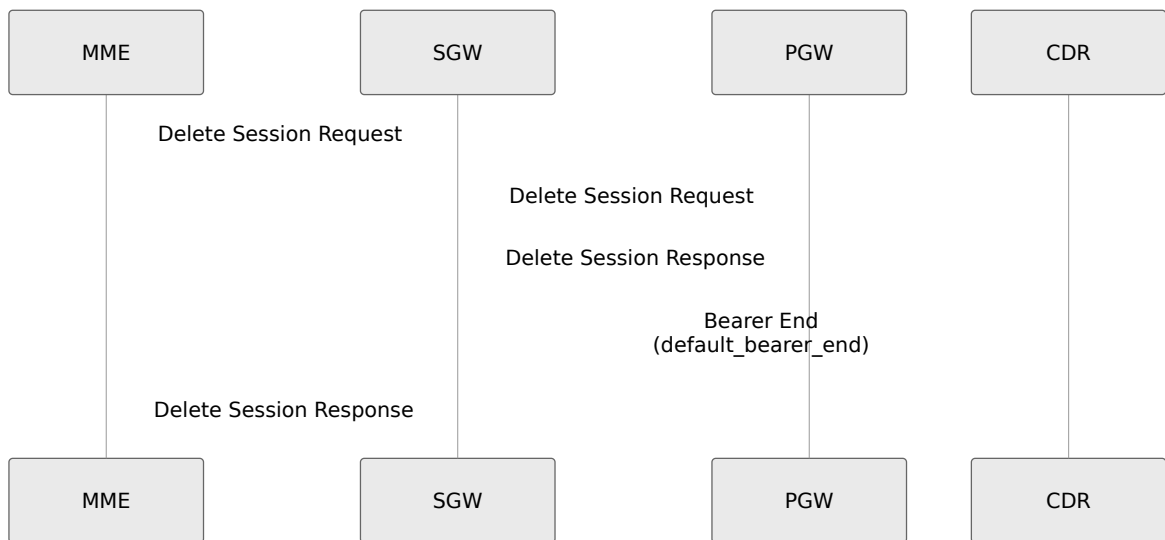
PDN Context PDN Context

- **QoS:** QCI 9 (Default)
- **APN:** PDN APN
- **IP:** IP address
- **TEID:** PDN TEID

PDN Context:



□□□□□□:



□□□□

□□□□□□□□□□□□□□ QoS□

- □□: □□□□□□□□□□□□
- **QoS**: QCI 1-8□□□□□□□□
- □□□□: □□□□ PDN □□
- □□: □□ PDN □□□□□□□□□□

□□□□□□:

Application Trigger

↓

PGW-C Policy Decision (via PCRF)

↓

Create Bearer Request (S5/S8)

↓

SGW forwards to MME (S11)

↓

MME activates bearer on RAN

↓

Create Bearer Response back through SGW to PGW

□□□□□□:

Network or Application Decision

↓

Delete Bearer Request (S5/S8)

↓

SGW forwards to MME (S11)

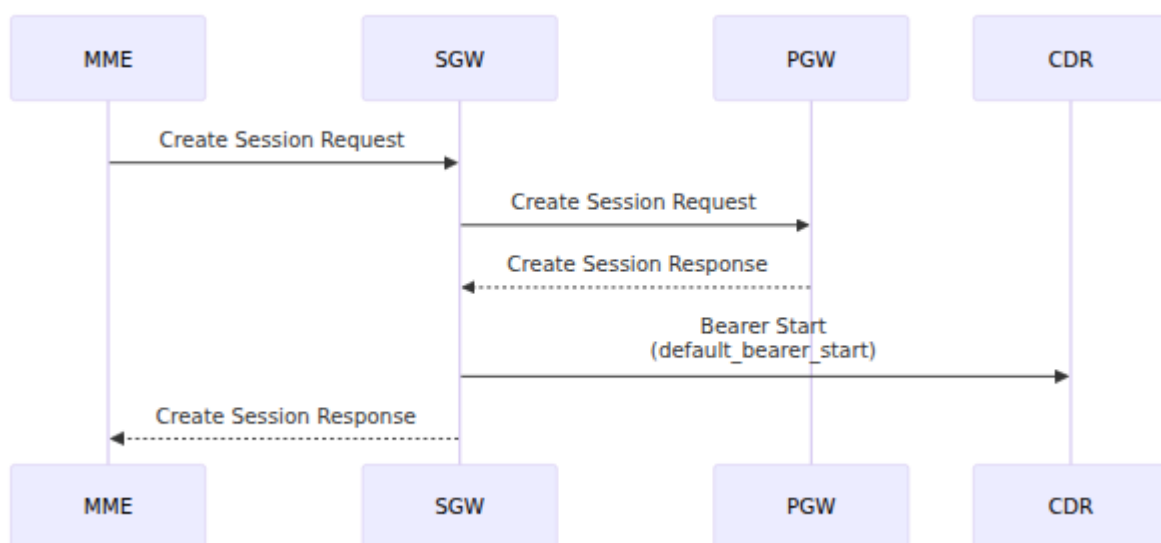
↓

MME deactivates bearer on RAN

↓

Delete Bearer Response back through SGW to PGW

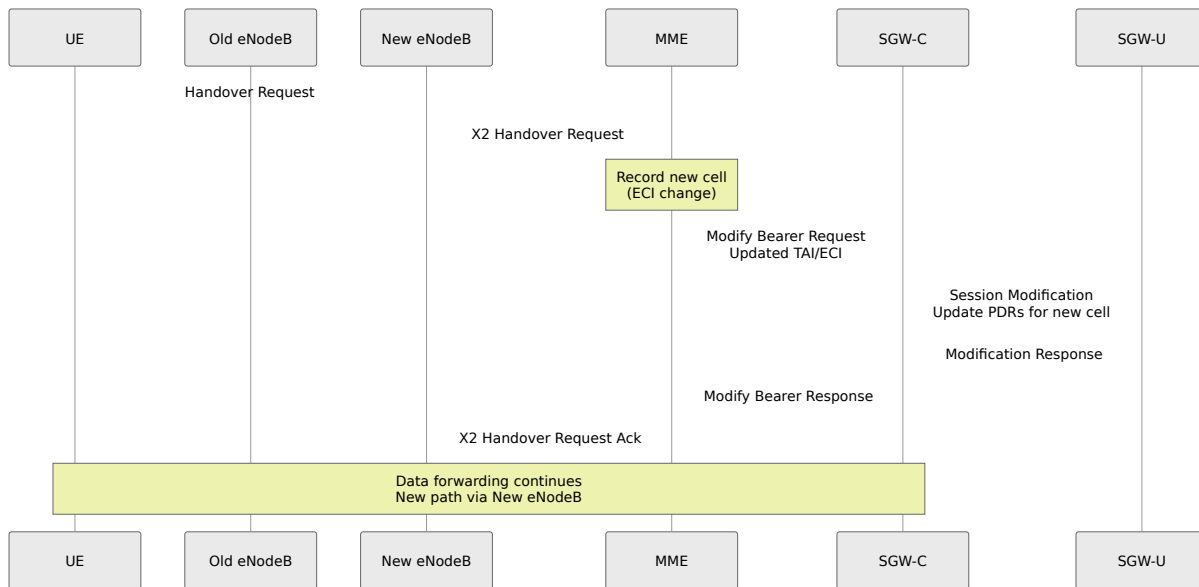
□□ **QoS** □□



Handover

MME and SGW

UE MME

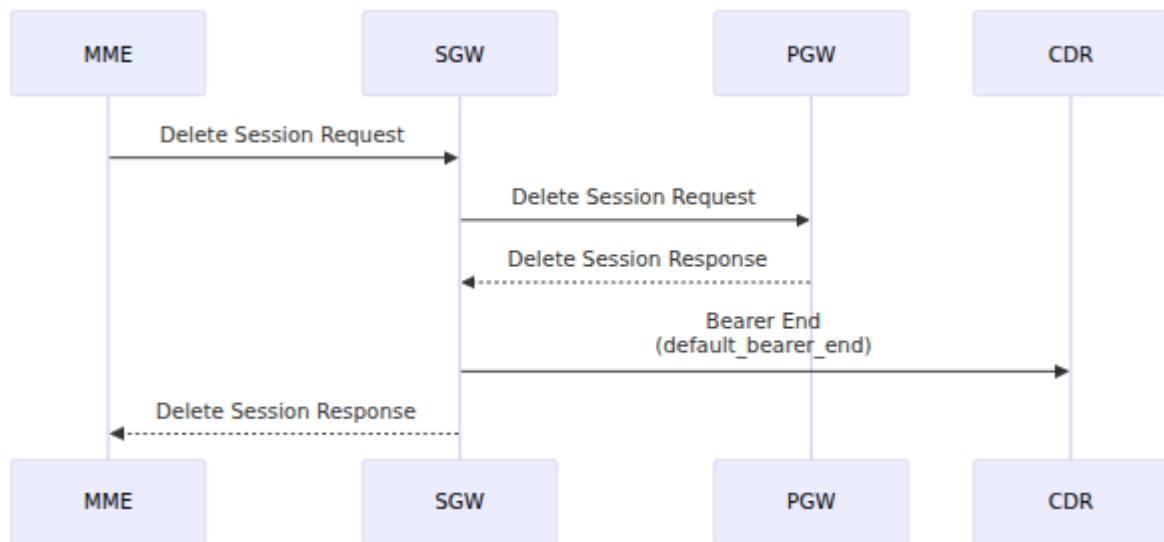


Handover

- Handover
- TEID
- Handover
- CDR Charging ID

MME and SGW

UE MME SGW



CDR

- CDR “”
- Charging ID
-
- SGW-U

TAU

SGW TAU:

```

UE
↓
MME TAU
↓
MME SGW
↓
SGW TAI ECI
↓

```

SGW TAU:

- MME
- SGW
- CDR SGW SGW

□□□□

□□□□

□□□□□□□□

1. □ **SGW-U** □□ - □□□□□□
2. **PDR** □□ - □□□□□□□□
3. □□□□ - □□□□□□□□□□□□
4. □□□□ - S11/S5/S8 □□□□

□□□□□□

□□□□□□□□

- □ **SGW-U** □□□□ - □□□□□□□□
- □ **SGW-U** □□□□ - □□□□
- **GTP** □□ - □□□□□□□□□□□□
- □□□□□□ - □□□□

□□□□

□□□□□□□□

1. □□□□ - □□□□□□□□
 2. □□□□ - □□□□□□□□
 3. □□□□□□ - □□□□□□□□
 4. □□□□ - □□□□□□ TEID
-

□□□□

□□□□

□□ Web UI □□□□□□

1. □□ http://<sgw-ip>:<port>/ue_sessions
2. □□□□□□ UE □□
3. □ IMSI□GUTI □□□□□□
4. □□□□□□□□□□
 - □□ (TAI, ECI)
 - □□□□□ QoS
 - PGW-C □□
 - TEID □□□
 - Charging ID

UE □□□□□□□□□□□□□□□□

□□□□□□□□□□ TEID□□□□□□□□ PDN □□□□□□□□

📌 📌 📌 Web UI 📌📌📌📌

📌📌📌

📌📌📌📌

```
# 📌📌📌📌  
curl -s http://10.0.0.40:42068/metrics | grep active_ue_sessions  
  
# 📌📌📌📌  
curl -s http://10.0.0.40:42068/metrics | grep active_bearers  
  
# 📌 APN 📌  
curl -s http://10.0.0.40:42068/metrics | grep sessions_by_apn  
  
# 📌📌📌📌  
curl -s http://10.0.0.40:42068/metrics | grep  
s11_inbound_messages_total
```

📌📌📌📌Prometheus 📌📌📌📌📌📌📌📌📌📌📌📌📌

📌📌📌📌📌

📌📌📌📌📌📌

1. 📌 **API** 📌📌 📌📌📌📌
2. 📌📌📌 📌📌📌📌

3. 確認する
4. 確認 **CDR** 確認する

確認

確認

```
# 確認
curl -s http://10.0.0.40:42068/metrics | \
  grep -E "active_ue_sessions|active_bearers" | \
  awk '{print $NF}'

# 確認 80% 確認
# 確認
```

確認

確認

確認 確認

確認

1. 確認
2. 確認 S11 確認
3. 確認 PGW 確認
4. 確認 Charging ID 確認

確認

項目	説明
PGW アドレス	ネットワーク S5/S8 アドレス
ネットワーク IP	ネットワーク PGW IP アドレス
APN アドレス	ネットワーク PGW の APN
ネットワーク SGW-U	ネットワーク SGW-U アドレス
ネットワーク	ネットワーク PGW アドレス

ネットワーク

ネットワーク アドレス

ネットワーク

- ネットワーク
- ネットワーク SGW-U アドレス
- ネットワーク PGW アドレス
- ネットワーク

ネットワーク

項目	説明
SGW-U アドレス	ネットワーク SGW-U アドレス
ネットワーク	ネットワーク
PGW アドレス	ネットワーク PGW
ネットワーク	ネットワーク RTT

1. 1. 1. 1.

2. 2. 2. 2.

3. 3. 3. 3.

1. 1. 1. 1.
2. 2. 2. 2.
3. 3. 3. 3.
4. 4. 4. 4.

5. 5. 5. 5.

1. 1. 1. 1.	2. 2. 2. 2.
3. 3. 3. 3.	4. 4. 4. 4.
5. 5. 5. 5.	6. 6. 6. 6.
7. 7. 7. 7.	8. 8. 8. 8.
9. 9. 9. 9.	10. 10. 10. 10.
11. 11. 11. 11.	12. 12. 12. 12.

13. 13. 13. 13.

14. 14. 14. 14.

15. 15. 15. 15.

```
# 確認
curl -s http://10.0.0.40:42068/metrics | \
  grep "inbound_duration_seconds"

# 確認
curl -s http://10.0.0.40:42068/metrics | \
  grep queue_depth

# 確認
top -n1 | head -1
```

確認

1. 確認 RTT 確認
2. 確認 SGW-C 確認
3. 確認
4. 確認

CDR 確認

確認 CDR 確認

確認

1. 確認 CDR 確認
2. 確認
3. 確認
4. 確認

確認

```
# 查看 CDR 文件
tail -f /var/log/sgw_c/cdrs/*

# 查看目录
ls -la /var/log/sgw_c/cdrs/

# 设置权限
chmod 755 /var/log/sgw_c/cdrs/
```

查看 CDR 文件 查看 CDR 文件

查看

查看

- 查看 查看
- 查看 70-80% 查看
- 查看 查看
- 查看 查看

查看

- 查看 查看
- 查看 查看
- 查看 SGW-U 查看
- 查看 查看

查看

- 查看 **Charging ID** 查看 PGW 查看
- **CDR** 查看 查看 SGW 查看 PGW 查看 CDR
- 查看 查看 CDR 查看
- 查看 查看 CDR

Sxa



PFCP  **SGW-U**  

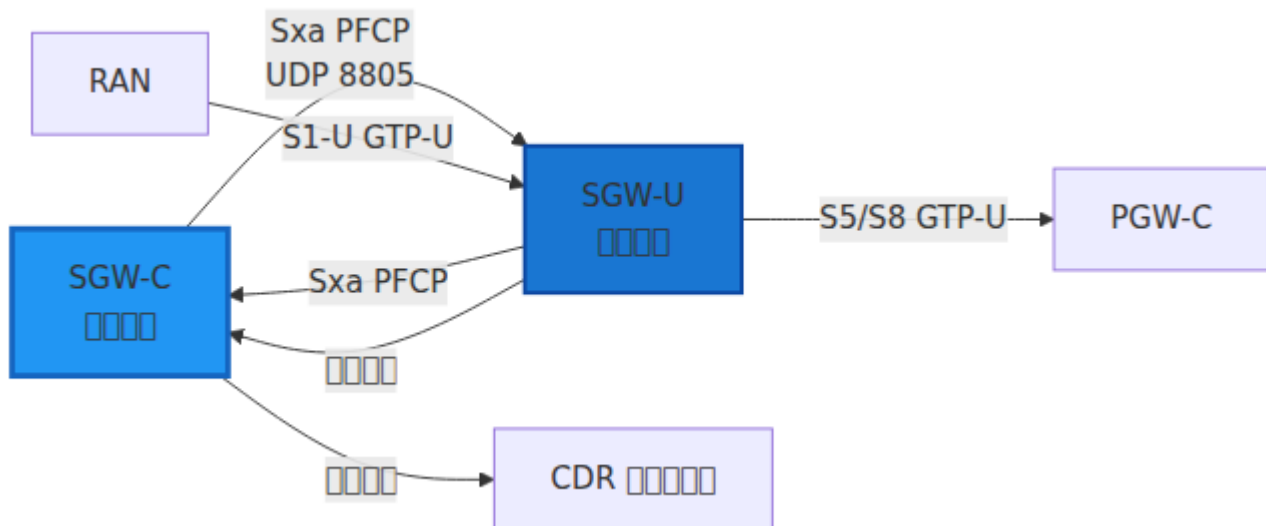
OmniSGW  *Omnitouch*    



1.  
 2.    
 3.  
 4. **PFCP**  
 5.    
 6. **PFCP**  
 7.    
 8.    
 9.    
-



Sxa   **PFCP**                                                



SGW-U

- **PFCP v1.0** - 3GPP TS 29.244
- **SEID** - 32-bit identifier
- **QoS** - 3GPP TS 23.287
- **CP** - Control Plane
- **UP** - User Plane

SGW-C

PFCP v1.0

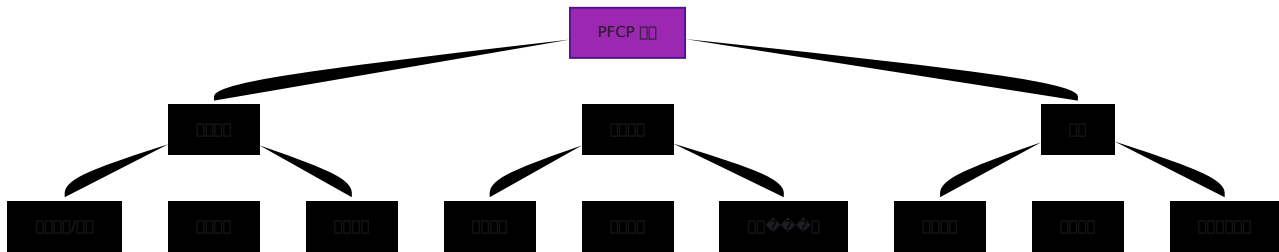
- PFCP v1.0 (3GPP TS 29.244)
- UDP
- 8805
- CP
- UP

SEID

SEID

- **CP SEID** - SGW-C SGW-U
- **UP SEID** - SGW-U SGW-C

SGW-C → SGW-U: SGW-U UP SEID
SGW-U → SGW-C: SGW-C CP SEID



□□

□□□□

```
# config/runtime.exs
config :sgw_c,
  sxa: %{
    # Sxa □□□□□ IP □□
    local_ip_address: "10.0.0.20",

    # □□□□□□□□□□
    local_port: 8805,

    # □□□□ SGW-U □□□
    peers: [
      %{
        ip_address: "10.0.0.30",
        node_id: "sgw-u-1.example.com"
      },
      %{
        ip_address: "10.0.0.31",
        node_id: "sgw-u-2.example.com"
      }
    ],

    # □□□□□□□□□□
    heartbeat_interval_s: 20,

    # □□□□□□□□□□
    session_timeout_ms: 5000,

    # □□□□□□□□□□□□
    max_retries: 3
  }
```

□□□□

□□□□□□

```
# SGW-U PFCP
iptables -A INPUT -p udp --dport 8805 -s <sgwu_network>/24 -j
ACCEPT

# SGW-U PFCP
iptables -A OUTPUT -p udp --dport 8805 -d <sgwu_network>/24 -j
ACCEPT
```

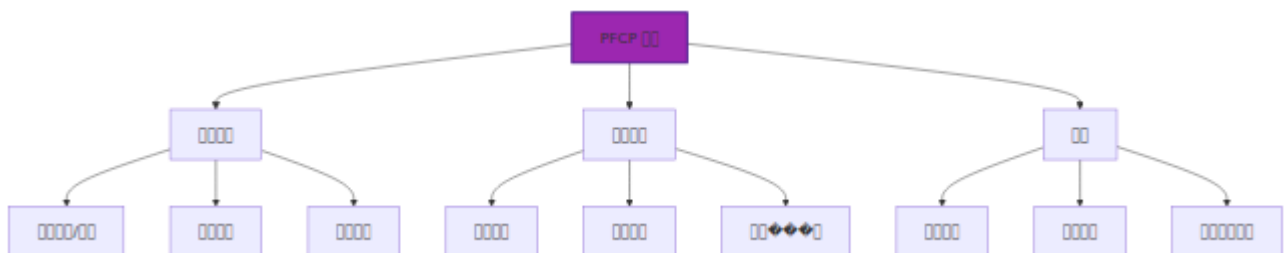
```
# SGW-U
ip route add <sgwu_network>/24 via <gateway_ip> dev eth0
```

```
# PFCP
# " "

# PFCP
curl http://127.0.0.40:42068/metrics | grep seid_registry_count
```

PFCP

SGW-C SGW-U PFCP



□□□□□

```
[□□□]
  ↓ (□□□□)
[□□□]
  ↓ (□□□□ OK)
[□□□]
  ↓ (□□□□)
[□□□□]
  ↓ (□□□□)
[□□□□□]
  ↓ (□□ OK □□□)
[□□□□□□□]
```

□□□□

□□ PFCP □□□□□□□□

1. □□□□□

- □□□□□□
- □□□□□□□□□□
- □□□□□□□

2. □□□□□

- □□□□□□□□□□□□□□□□
- □□ SGW-U □□□□□□□□
- □□□□□□□□□□□□□□□□

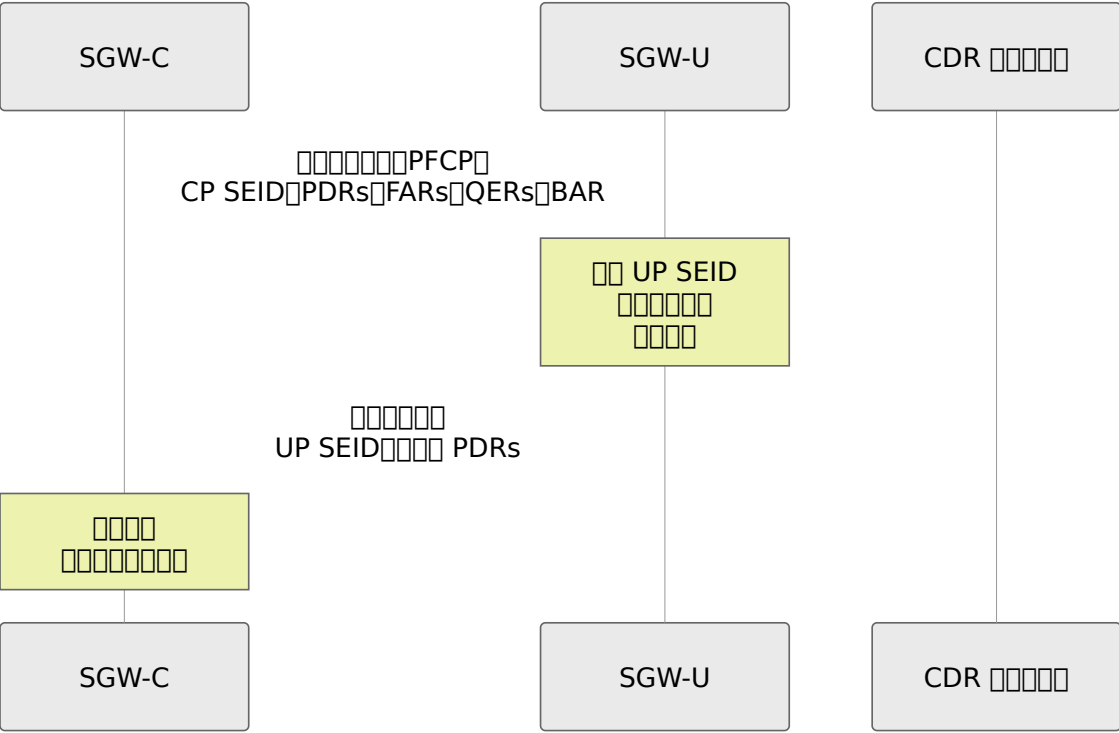
3. □□□□□

- □□□□□□□ SGW-U □□□□□□□□□□
 - PDR □□□□□□□□□□□□□□
 - □□□□□□□□□□□□□□□□
-

4.1.1

4.1.1.1

4.1.1.1.1 MME 4.1.1.1.1.1 S11 4.1.1.1.1.1.1



4.1.1.1.1.1

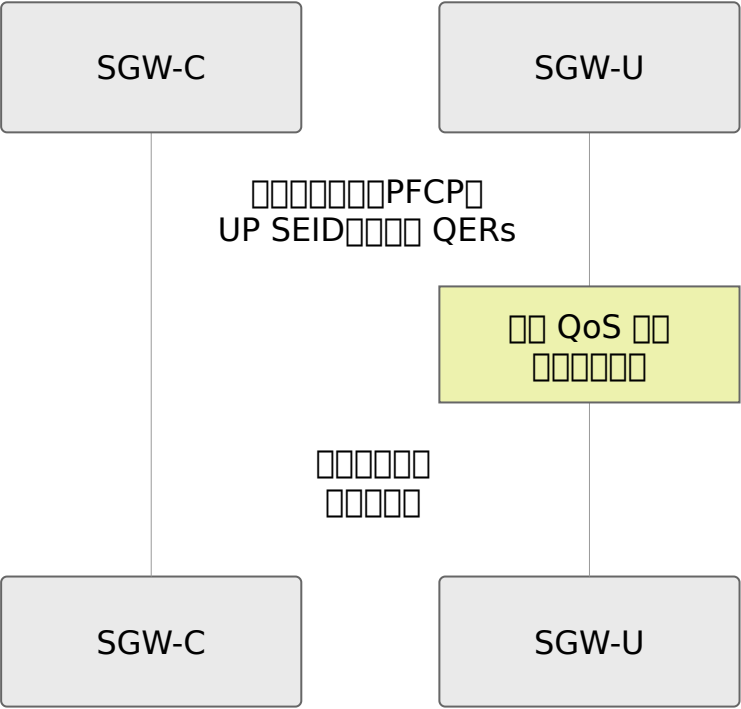
CP SEID	SGW-C CP SEID
PDRs	PDRs
FARs	FARs
QERs	QoS
BAR	BAR
UP PDR	UP PDR

□□□□

[□□□]
↓ (□□□□)
[□□□]
↓ (□□□□)
[□□□□]

□□□□

□□□ □□ MME □□□□□□□□□□ QoS □□□□□□

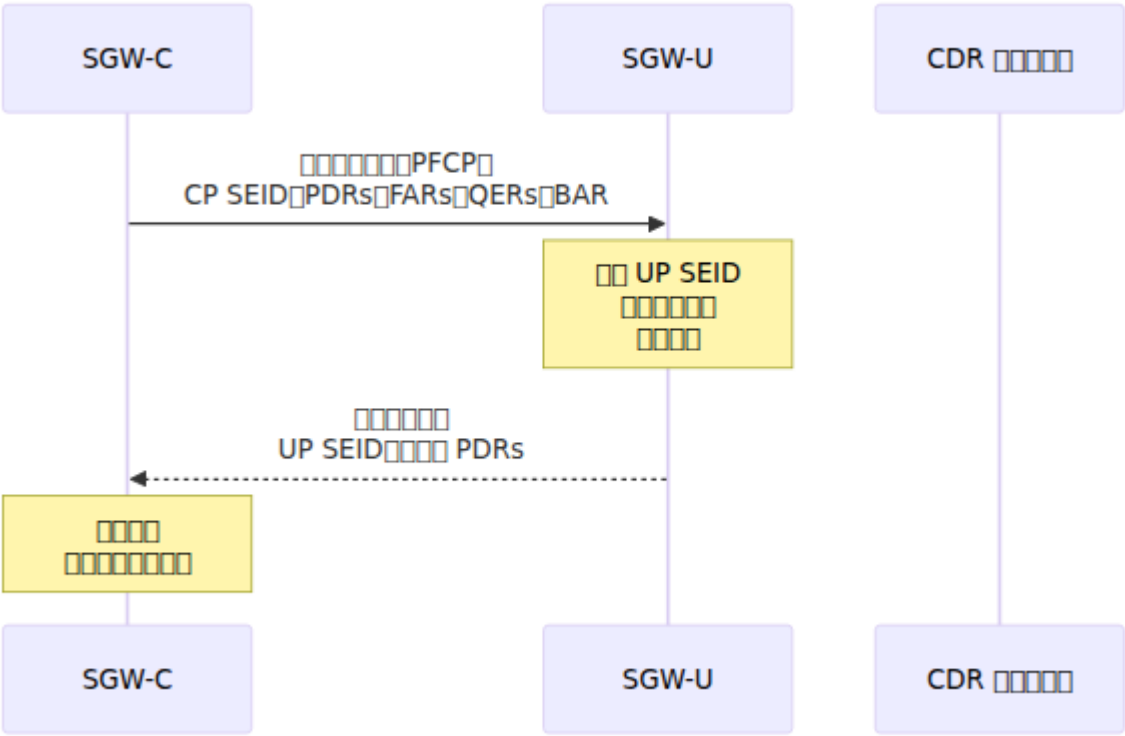


□□□□

項目	項目名	項目
QoS 項目	QERs	項目/項目
項目	PDRs/FARs	eNodeB 項目SGW-U 項目
項目	項目 PDR/FAR/QER	項目
項目	項目 PDR/FAR/QER	項目

項目

項目 項目 MME 項目



項目項目

[項目項目]
↓ (項目項目)
[項目項目]
↓ (項目項目)
[項目項目]

PFCP

PDR

??

		S1-U S5/S8
IP	UE IP	10.45.0.50
IP	IP	8.8.8.8
	IP	TCP6UDP17
		1024-65535
		80HTTP443HTTPS
TEID	GTP-U	

PDR

```
PDR
├─ PDR ID
├─
├─
├─
│   ├─
│   └─ APN
│       └─ UE IP / IP
├─ FAR ID
├─ QER ID QoS
└─
```

UE - E-UTRAN

- UE IP address
- PDN type PGW-U type
- QoS parameters

UE - E-UTRAN

- UE capabilities
- UE location
- QoS parameters GBR

FAR parameters

UE capabilities

UE

UE	UE	UE
UE	UE capabilities	UE
UE	UE location	UE/UE
UE	UE	UE capabilities
UE	UE capabilities	UE

UE

- **GTP-U** - GTP-U S1-U/S5/S8
- UE - UE capabilities
- **IPv4** - IPv4 address
- **IPv6** - IPv6 address

UE - UE

PDR 000000 = 000UE IP = 10.45.0.50

FAR 000

- 00 = 0
- 00000 = 00000000
- 0000 = 00000

QERQoS 00000

00000000000000

QoS 000

00	00	00
QCI	00	QoS 0000001-90
MBR00000000	000	00000000
GBR00000000	000	0000000
ARP	00	00000000001-150

QoS 000QCI00

QCI	サービス	制限
1	GBR	MBR: 64 kbps
2	GBR	MBR: 256 kbps
3	GBR	MBR: 50 kbps
4	GBR	GBR: 128 kbps MBR: 256 kbps
5	IMS	GBR: 100 kbps MBR: 256 kbps
6		MBR: 10 Mbps
7	GBR	GBR: 64 kbps MBR: 384 kbps
8		MBR: 5 Mbps
9		MBR: 3 Mbps

- QCI 9

QCI: 9
 MBR: 100 Mbps
 GBR: GBR
 ARP: 15

- QCI 1

QCI: 1
 MBR: 128 kbps +
 GBR: 64 kbps
 ARP: 1

BAR

1.

- UE eNodeB
-
-

2.

- UE
-
- UE

3. SGW

- SGW MME
- SGW SGW
-

BAR

BAR

└─
└─
└─
└─ CP

□□□□

□□□□□□

SGW-U □ SGW-C □□□□□□□□□□□□



□□□□□□□□

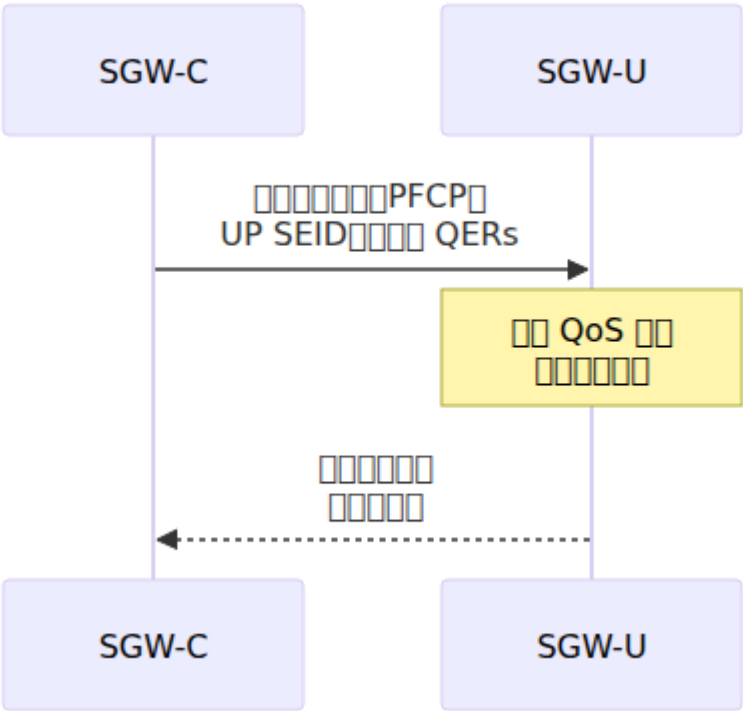
□□□□□□□□□□□□

□□□	□□
□□□□	□ N □□□□□
□□□□	□□ N □□□
□□□□□□	□□ N □□
□□□□	□□□□□□□
□□	□□□□□□□
□□□□	□□□□□□□□

□□□□□□

□□□□□□
├─ □□□□□□□□□□□□□□
├─ UR-SEQN□□□□□□□□□□
├─ □□□□□□□□□□
├─ ┌─ EBI□□□□□□□□
│ └─ □□□□□
│ ├── UL□□□□□□
│ ├── DL□□□□□□
│ ├── □□□□□□□□
│ └─ □□□□□□□□□□
├─ □□□□□□□□□□□□□□
├─ □□□□/□□□□□□□□□□□□□□
└─ □□ UR□□□□□□□□□□

CDR □□□□



실습

PFCP 실습

실습 PFCP 실습

```
# 실행
curl -s http://127.0.0.40:42068/metrics | grep pfcf_association

# 결과
# pfcf_association_status{peer_ip="10.0.0.30"} 1 (실행)
# pfcf_association_status{peer_ip="10.0.0.31"} 1 (실행)

# Web UI → SGW-U 실행
# 실행결과 "실행" 실행결과
```

실습

실습 PFCP 실습

```
# 실행
curl -s http://127.0.0.40:42068/metrics | grep seid_registry_count

# 실행 SGW-U 실행
curl -s http://127.0.0.40:42068/metrics | grep seid_by_peer

# 실행결과/실행
curl -s http://127.0.0.40:42068/metrics | grep usage_octets_rate
```

실습

실습 PFCP 실습

```
# PFCP
watch -n 1 'curl -s http://127.0.0.40:42068/metrics | grep sxa_inbound'

#
#
sxa_inbound_messages_total{message_type="session_establishment_response"} 5432
#
sxa_inbound_messages_total{message_type="session_modification_response"} 12100
# sxa_inbound_messages_total{message_type="session_report_request"} 6
```

SGW-U

SGW-U

```
#
curl -s http://127.0.0.40:42068/metrics | grep
sxa_session_establishment

# PDR
curl -s http://127.0.0.40:42068/metrics | grep pdr_installation

#
curl -s http://127.0.0.40:42068/metrics | grep sxa_timeout_total
```

SGW-U

SGW-U

SGW-U

SGW-U

1. ping <sgwu_ip>
2. netstat -an | grep 8805

3. 確認する

確認する

項目	項目	確認事項
確認事項	確認事項	確認 SGW-U 確認
確認事項	確認事項	確認事項
SGW-U 確認	確認	確認 SGW-U 確認
確認 ID 確認	確認事項	確認???

確認事項

確認 "確認事項"

確認

```
# 確認
curl -s http://127.0.0.40:42068/metrics | grep seid_registry_count

# 確認事項
tail -f /var/log/sgw_c/sgw_c.log | grep "確認"
```

確認

項目	パラメータ	説明
SGW-U 項目	"XXXX"	項目 SGW-U 項目
項目 PDR	"XXXX IE"	項目項目
SEID 項目	"SEID 項目"	項目項目
項目	"XXXXXXXX"	XXXXXXXX SGW-U

XXXXXXXX

項目 "XXXXXXXX"

項目

```
# XXXXXXXX
curl -s http://127.0.0.40:42068/metrics | grep
session_report_request_total

# 項目 CDR 項目
tail -f /var/log/sgw_c/cdrs/<timestamp>
```

項目項目

- 項目 SGW-U 項目項目
- XXXXXXXXXXXX
- 項目 CDR 項目項目
- 項目 SGW-U 項目項目

項目項目

項目 PFCP 項目項目

項目項目

```
# 確認コマンド
curl -s http://127.0.0.40:42068/metrics | grep
sxa_inbound_duration_seconds

# 確認コマンド
curl -s http://127.0.0.40:42068/metrics | grep seid_by_peer

# 確認コマンド
curl -s http://127.0.0.40:42068/metrics | grep pfcp_queue_depth
```

確認

1. 確認 SGW-U 確認コマンド
2. 確認コマンド
3. 確認コマンド
4. 確認 SGW-C 確認コマンド

確認コマンド確認コマンド確認コマンド 確認コマンド

確認

確認

- 確認コマンド 20-30 確認コマンド
- 確認コマンド RTT 確認 5-10
- 確認コマンド 2-3 確認コマンド
- 確認コマンド 確認 SGW-U 確認コマンド

確認

- 確認コマンド 確認 SGW-U 確認コマンド
- 確認コマンド 確認コマンド
- 確認コマンド 確認コマンド
- 確認 確認コマンド

□□□□

- □□□□ □□ PFCP □□□□□□□□
 - □□□ □ S11 □□□ PFCP □□□□□□
 - □□□□□ □□□□□□□□
 - □□□□□□□ □□ SGW-U □□□□□□
-

OmniSGW 白皮书

OmniSGW - 白皮书 (SGW)

Omnitouch 白皮书

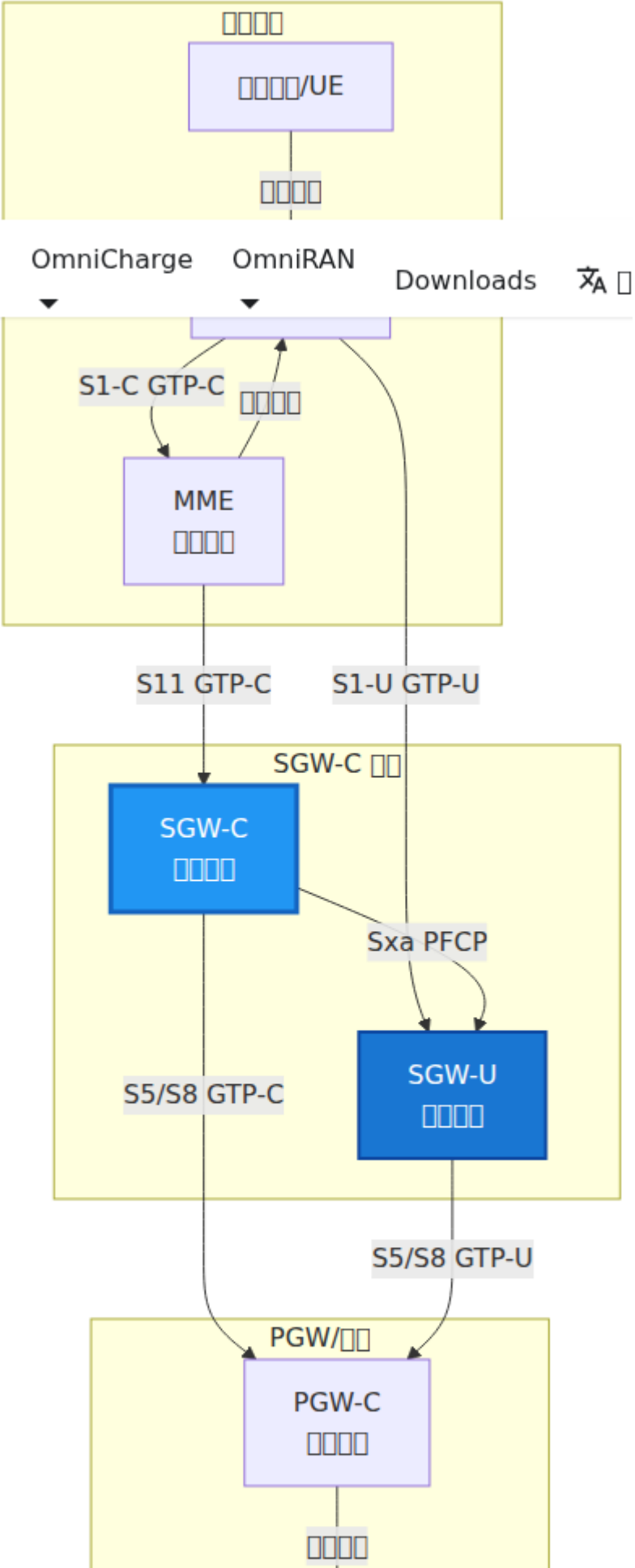
目录

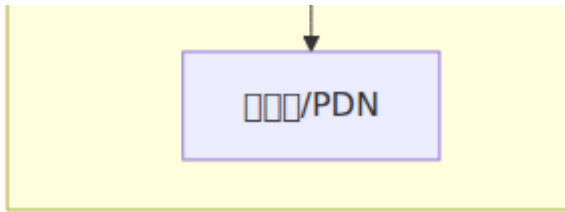
1. 简介
2. 背景
3. 目标
4. 架构
5. 功能
6. 部署
7. Web UI - 管理界面
8. 性能
9. 安全
10. 兼容性
11. 测试
12. 结论

简介

OmniSGW 是 3GPP LTE 网络 (EPC) 中的核心网元 (SGW) 的下一代 Omnitouch 版本。它旨在为 UE 提供无缝连接和最佳性能。

- 支持 - 提供 UE (用户) 的连接
- 支持 - 通过 eNodeB 进行数据传输
- 支持 - 提供 QoS 服务质量
- 支持 - 提供网络切片
- 支持 - 提供 SGW-U (用户面) 功能



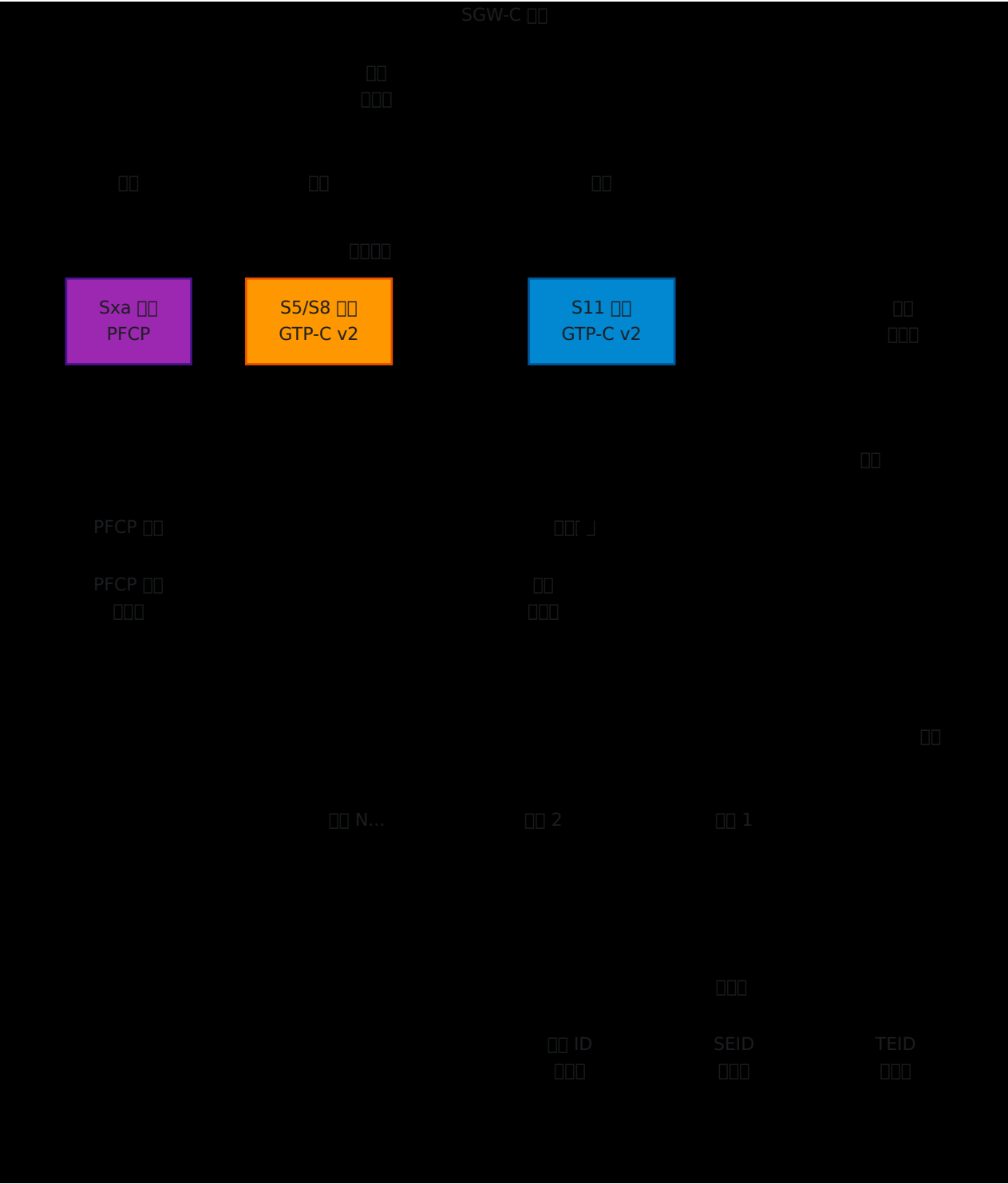


SGW-C

- MME S11 (GTP-C)
 - PGW-C S5/S8 (GTP-C) PDN
 -
 - SGW-U Sxa (PFCP)
 - eNodeB UE
 -
 -
-

□□

□□□□



概要

SGW-C には Elixir/OTP による 2 つのコンポーネントがあります

- コア - コアコンポーネント
- フロントエンド - フロントエンド (S11, S5/S8, Sxa)
- ユーティリティ - ユーティリティ UE として GenServer
- プロトコル - プロトコル (TEID, SEID, ID など)
- **PFCP** プロトコル - プロトコル SGW-U として PFCP

コアコンポーネントは、コアコンポーネントとして実行されます

フロントエンド Web UI は、フロントエンドとして実行されます

アーキテクチャ

SGW-C は 3GPP 仕様

S11 プロトコル (GTP-C v2)

コア MME と SGW-C の間には

コア プロトコル UDP を GTP-C として 2

ポート

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

000 000 S11 0000

Sxa 00 (PFCP)

000 SGW-C 0 SGW-U 0000000000

000 00 UDP 0 PFCP (0000000000)

000000

- 000000/00
- 000000/00
- 000000/00
- 000000/00
- 000000/00
- 0000/00

000 000 PFCP/Sxa 0000

S5/S8 00 (GTP-C v2)

000 SGW-C 0 PGW-C 000000000000 PDN 00

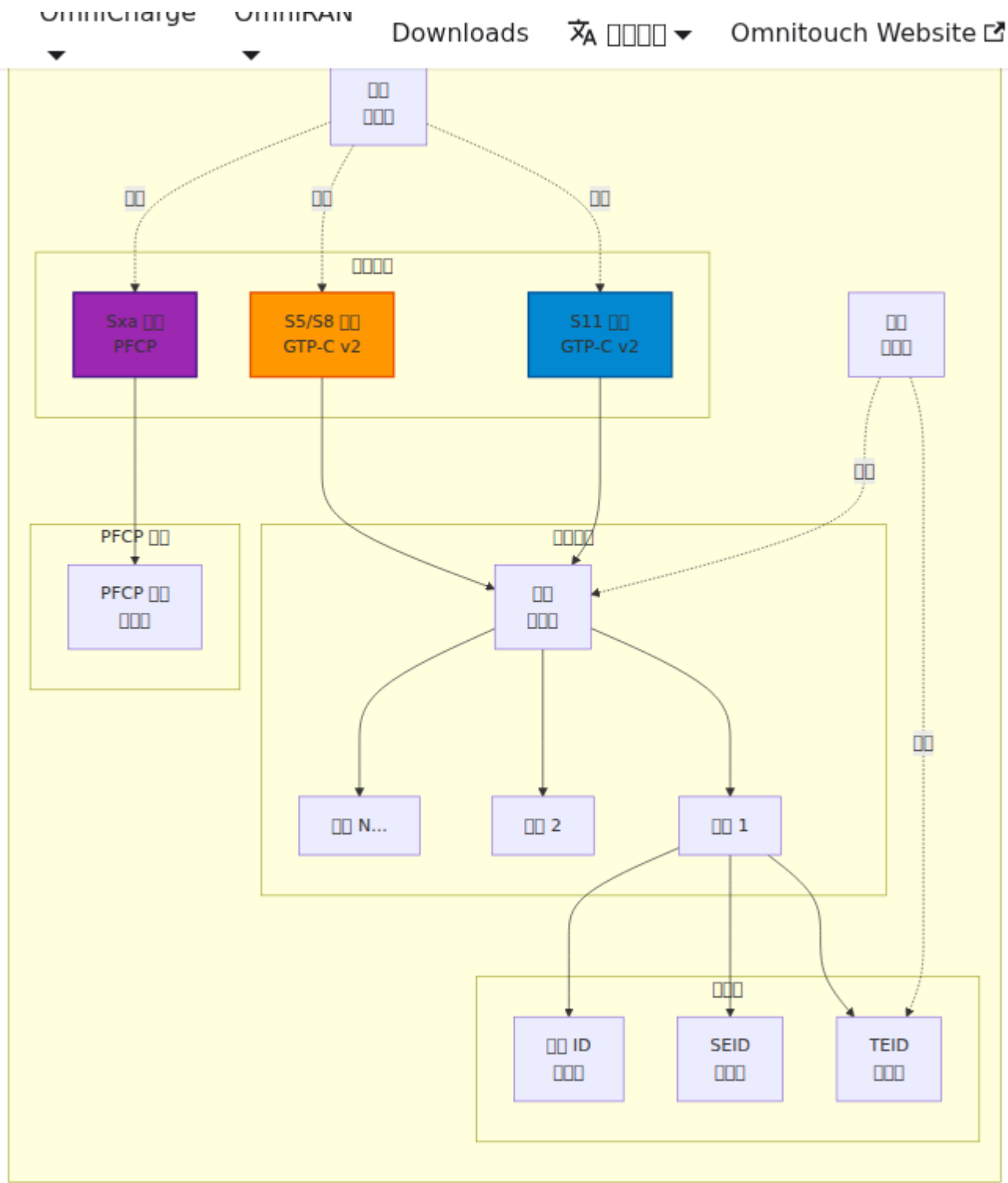
000 00 UDP 0 GTP-C 00 2

000000

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

- 000000/00
- 000000/00

000 000 S5/S8 0000



UE ID

UE ID

UE ID (UE ID) - UE ID

- **IMSI** (International Mobile Subscriber Identity) - IMSI
- **GUTI** (Globally Unique Temporary Identity) - MME UE ID
- **MSISDN** - MSISDN
- **TAI** (Tracking Area Identity) - TAI
- **TEID** - S11 & S5/S8 TEID
- **QoS** - QoS

PDN ID

PDN (PDN) - UE ID PGW-C

- **APN** (Access Point Name) - APN
- **ID** - SGW & PGW ID
- **TEID** (TEID ID) - S5/S8 TEID
- **SEID** (SEID ID) - Sxa SEID
- **QoS** - PDN QoS
- **QoS** - QoS QoS

QoS

QoS (QoS) - QoS

- **QoS** - PDN QoS
- **QoS** - QoS (QoS)
- **EBI** (EPS ID) - EBI
- **QoS** - QCI (QoS) ARP (MBR, GBR)

PFCP

SGW-C と SGW-U の関係

- **PDR** (Policy Data Rule) - トラフィック (IP/ポート)
- **FAR** (Forwarding Action Rule) - 転送ルール
- **QER** (QoS Enforcement Rule) - QoS 設定
- **BAR** (Bearer Allocation Rule) - 転送リソースの割り当て

SGW-C から SGW-U へ Sxa メッセージ

SGW-C

SGW-C から UE と eNodeB へ

- **MME** へ - MME へ通知 (IP SGW)
- **MME** へ - MMEs から SGW へ
- **UE** - UE へ通知
- **UE** - UE へ通知

UE

SGW-U

- Elixir ~1.16
- Erlang/OTP 26+
- MME と SGW-U と PGW-C の関係
- LTE EPC

SGW-C

SGW-U

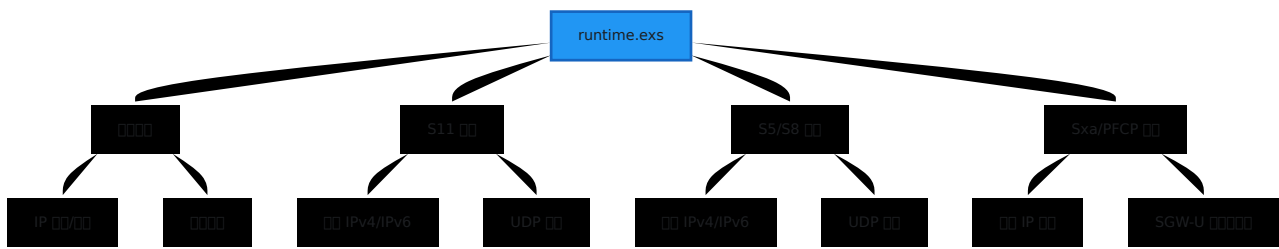
```
[info] [OmniSGW...]
[info] [127.0.0.40:42068]
[info] [127.0.0.10] [S11]
[info] [127.0.0.15] [S5/S8]
[info] [127.0.0.20] [Sxa]
[info] [PCF]
[info] OmniSGW
```

Access `http://127.0.0.40:42068/metrics` (Access)

Access

Access `config/runtime.exs` Access

Access



Access

metrics	Prometheus	
s11	GTP-C MME	S11
s5s8	GTP-C PGW-C	S5/S8
sxa	PCF SGW-U	Sxa

Access

Web UI -

OmniSGW  Web UI

Web UI

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

	URL		
UE	/ue_sessions	UE	2
PFCP	/pfcp_sessions	SGW-U PFCP	2
SGW-U	/sgwu_status	PFCP	2
	/logs		

- ()
- OmniSGW
- (/)

- IMSI/GUTI
-

- 网络拥塞避免
- 网络拥塞避免 QoS 策略
- 网络拥塞避免

网络拥塞避免 (拥塞避免)

- 网络拥塞避免
- 网络拥塞避免 NOC/网络拥塞避免
- 网络拥塞避免 IP 地址

网络拥塞避免

网络拥塞避免

1. 网络拥塞避免
2. 网络拥塞避免 UE 地址
3. 网络拥塞避免 IMSI 地址
4. 网络拥塞避免 QoS 策略
 - 网络拥塞避免
 - 网络拥塞避免 QoS
 - 网络拥塞避免
 - 网络拥塞避免 PGW-C 地址
5. 网络拥塞避免 → 网络拥塞避免

网络拥塞避免

1. 网络拥塞避免 SGW-U 地址 → 网络拥塞避免 SGW-U 地址“网络拥塞避免”
2. 网络拥塞避免 UE 地址 → 网络拥塞避免
3. 网络拥塞避免 APN 地址

网络拥塞避免

- 网络拥塞避免 UE 地址
- 网络拥塞避免/网络拥塞避免
- 网络拥塞避免
- 网络拥塞避免

Web UI

Web UI

-
-
-
-
-

Prometheus

-
-
-
-
-

- Web UI Prometheus

Web UIOmniSGW Prometheus

- - `teid_registry_count` - S11/S5S8 TEID
 - `seid_registry_count` - PFCP
 - `charging_id_registry_count` - ID
 - `active_ue_sessions` - UE
 - `active_bearers` -
-

- `s11_inbound_messages_total` - S11 接收 GTP-C 消息
- `s5s8_inbound_messages_total` - S5/S8 接收 GTP-C 消息
- `sxa_inbound_messages_total` - 接收 PFCP 消息
- 接收其他消息

- 错误

- `s11_inbound_errors_total` - S11 错误
- `s5s8_inbound_errors_total` - S5/S8 错误
- `sxa_inbound_errors_total` - Sxa 错误

配置

配置 HTTP 监听地址

```
curl http://127.0.0.40:42068/metrics
```

配置 Prometheus 监控

部署

部署 OmniSGW 到 Kubernetes 集群

0000

OmniSGW 00

└─ OPERATIONS.md (000)

└─ docs/

└─ 00000

└─ configuration.md

000 runtime.exs 00

└─ 0000

└─ sxa-interface.md

Sxa/PFCP (SGW-U 00)

└─ s11-interface.md

S11 (MME 00)

└─ s5s8-interface.md

S5/S8 (PGW-C 00)

└─ 00

└─ session-management.md

UE 000000

000└─ bearer-management.md

0000

└─ cdr-format.md

000000

└─ monitoring.md

Prometheus 00000

0000000000

0 00

00	00	00
OPERATIONS.md	0000000 (000)	00000000

⚙ 00

00	00
configuration.md	000 runtime.exs 0000

0 0000

파일명	설명
sxa-interface.md	PFCP/Sxa 인터페이스 SGW-U
s11-interface.md	GTP-C S11 인터페이스 MME
s5s8-interface.md	GTP-C S5/S8 인터페이스 PGW-C

기타 문서

파일명	설명
session-management.md	UE 세션 관리
bearer-management.md	Bearer 관리
cdr-format.md	CDR 형식
monitoring.md	Prometheus 및 Grafana 모니터링

참고

주요 문서

1. [OPERATIONS.md](#) - 개요 (🔍🔍🔍)
2. [configuration.md](#) - 구성
3. [monitoring.md](#) - 모니터링
4. [session-management.md](#) - 세션 관리

주요 인터페이스

1. [OPERATIONS.md](#) - 개요 (주요)
2. [sxa-interface.md](#) - PFCP/Sxa
3. [s11-interface.md](#) - GTP-C S11
4. [s5s8-interface.md](#) - PDN S5/S8
5. [session-management.md](#) - 세션 관리

6. [bearer-management.md](#) - 3GPP

3GPP

- 1. [configuration.md](#) - 3GPP
- 2. [monitoring.md](#) - 3GPP

3GPP

3GPP

TS	TS
TS 29.274	GTP-C v2 (S11 & S5/S8 TS)
TS 29.244	PCF (Sxa TS)
TS 32.251	TS
TS 32.298	CDR TS
TS 23.401	EPC TS