

OmniSGW 白皮书

OmniSGW - 概述 (SGW)

OmniSGW 概述

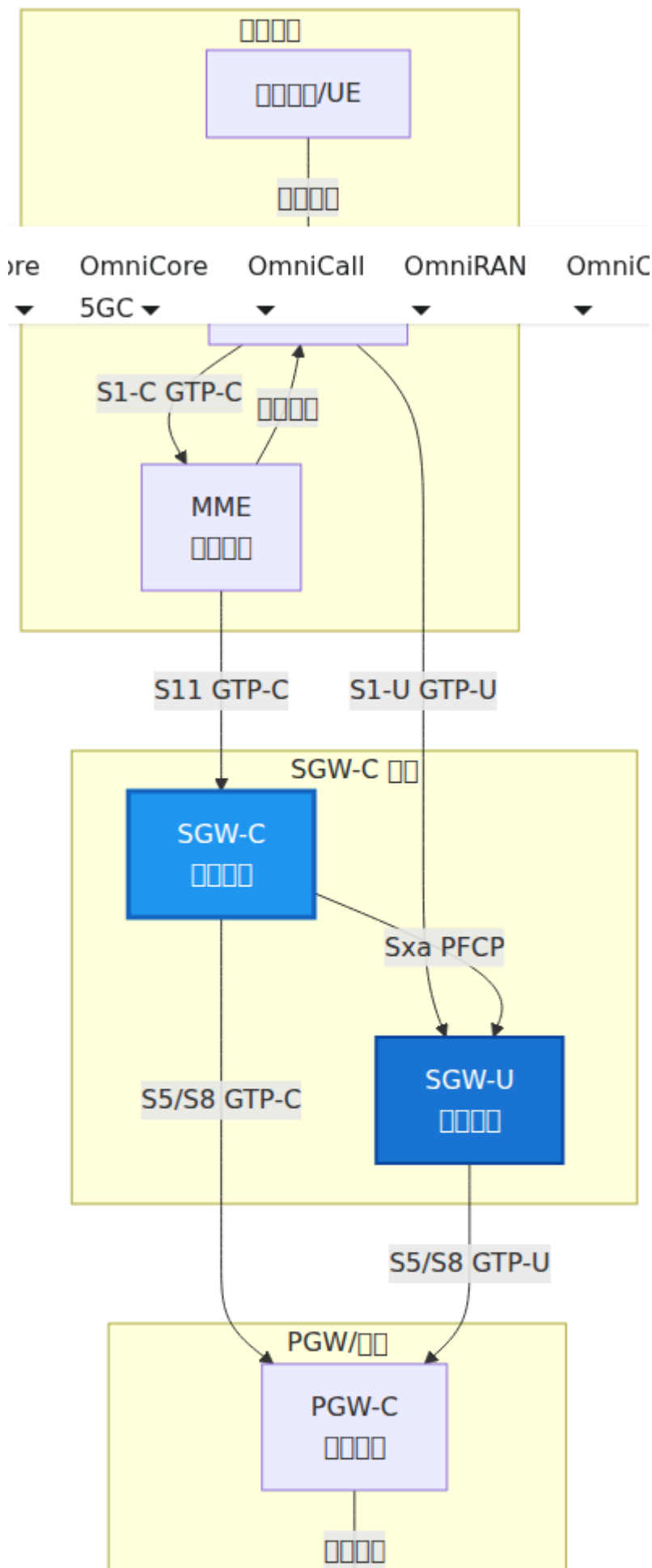
目录

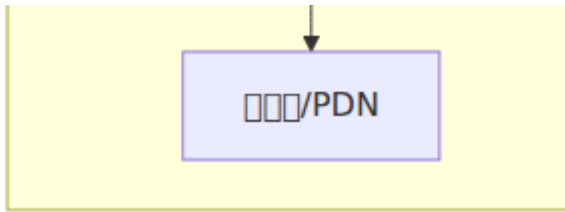
1. 简介
2. 背景
3. 架构
4. 功能
5. 部署
6. 性能
7. Web UI - 管理界面
8. 安全
9. 兼容性
10. 附录

简介

OmniSGW 是 Omnitouch 网络架构中的核心组件 (SGW)，负责管理 3GPP LTE 网络中的用户设备 (UE) 和提供网络服务。

- 负责管理 UE (用户设备) 的连接
- 负责管理 eNodeB 的连接
- 负责管理 QoS (服务质量) 策略
- 负责管理网络资源
- 负责管理 SGW-U (用户面) 的连接



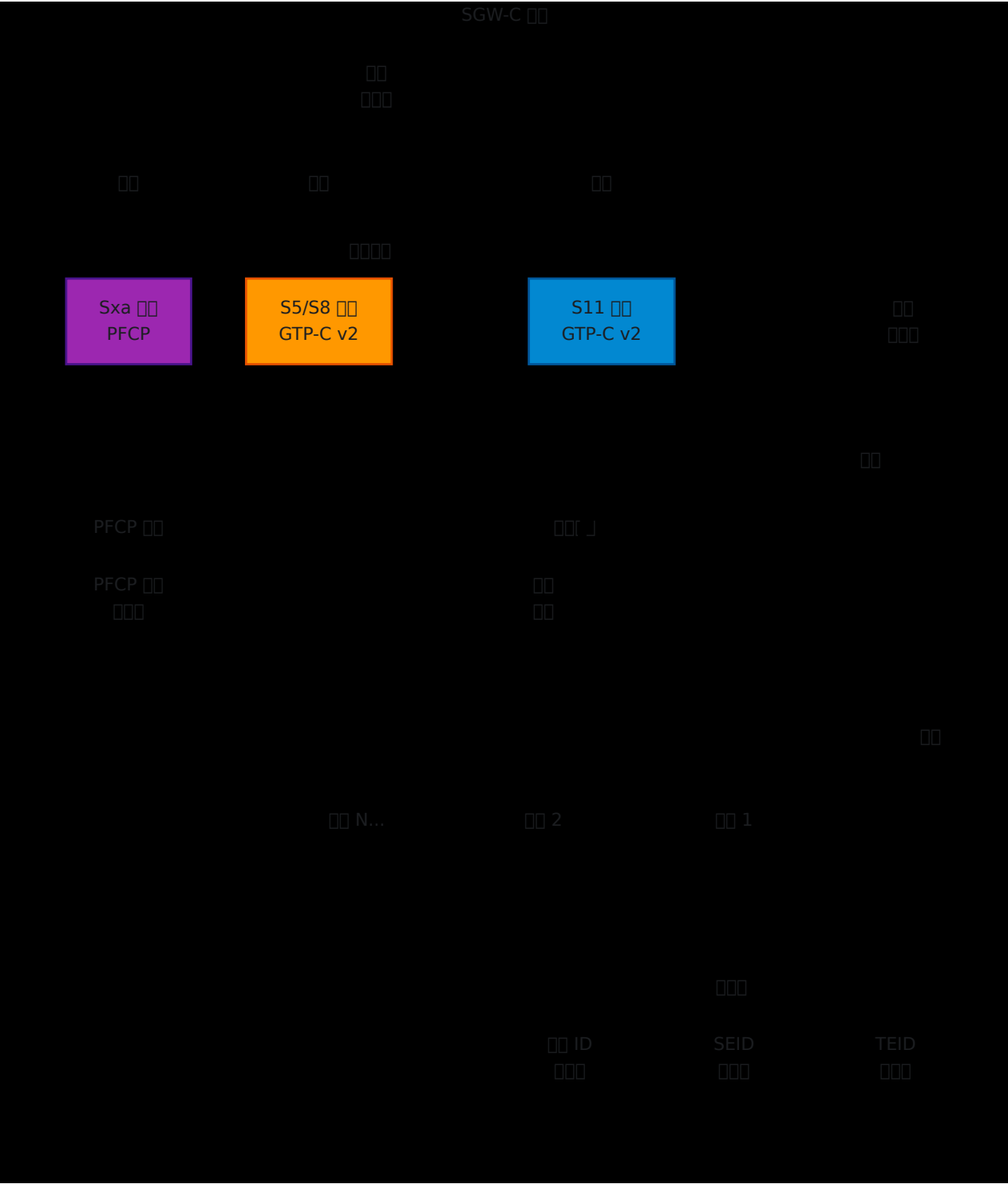


SGW-C

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

□□

□□□□



Overview

SGW-C uses Elixir/OTP for building and deployment

- **Architecture** - Scalable and modular
- **Interfaces** - 3GPP (S11, S5/S8, Sxa)
- **Integration** - Integrates UE through GenServer
- **Security** - Implements (TEID, SEID, ID, etc.)
- **PFCP** - Supports SGW-U through PFCP

Supports various network configurations and protocols

Includes Web UI for configuration and monitoring

Architecture

SGW-C implements 3GPP standards

S11 Interface (GTP-C v2)

MME connects to SGW-C for S11 interface

Supports UDP for GTP-C over port 2152

Configuration

- 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

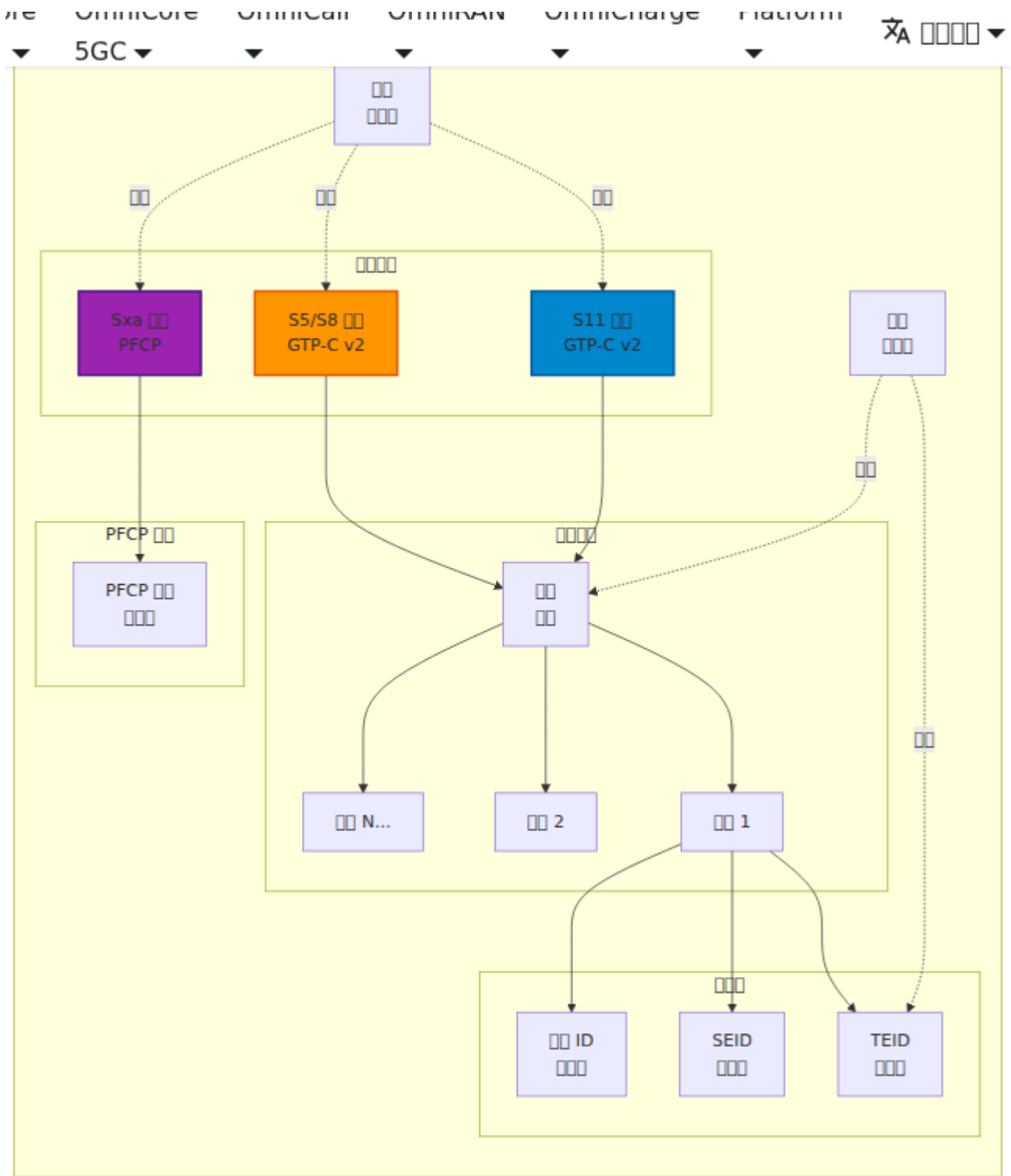
000 00 UDP 0 GTP-C 00 2

000000

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

- □□□□□□/□□
- □□□□□□/□□

□□□ □□□ S5/S8 □□□□



UE ID

UE ID

UE ID는 UE를 식별하는 데 사용되는 값입니다.

- **IMSI** (International Mobile Subscriber Identity) - IMSI는 UE의 고유한 식별자입니다.
- **GUTI** (Globally Unique Temporary Identity) - GUTI는 MME가 UE에 할당하는 임시 식별자입니다.
- **MSISDN** - MSISDN은 UE의 전화번호입니다.
- **TAI** (Tracking Area Identity) - TAI는 UE가 현재 위치한 추적 영역을 식별합니다.
- **TEID** - S11 및 S5/S8 인터페이스에서 사용되는 식별자입니다.
- **IP Address** - UE의 IP 주소입니다.

PDN ID

PDN ID는 UE가 연결되는 PDN을 식별하는 데 사용됩니다.

- **APN** (Access Point Name) - APN은 UE가 연결되는 PDN을 식별합니다.
- **ID** - ID는 SGW 및 PGW가 UE를 식별하는 데 사용됩니다.
- **TEID** (Temporary Endpoint Identifier) - S5/S8 인터페이스에서 사용되는 식별자입니다.
- **SEID** (Service Identifier) - Sxa 인터페이스에서 사용되는 식별자입니다.
- **QoS** - UE의 QoS (Quality of Service)는 PDN에 따라 달라집니다.

QoS

QoS는 서비스의 품질을 보장하는 데 사용됩니다.

- **QoS** - QoS는 PDN에 따라 달라집니다.
- **QoS** - QoS는 서비스 (예: 음성, 데이터)에 따라 달라집니다.
- **EBI** (EPS Bearer ID) - EBI는 EPS Bearer를 식별합니다.
- **QoS** - QoS는 QCI (QoS Class Identifier) 및 ARP (Allocation Retention Priority)에 따라 달라집니다 (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-U から SGW-C へ Sxa メッセージ

SGW-C から SGW-U へ

SGW-C から UE へ eNodeB へ

- **MME** へ - MME へ MME 情報 (IP SGW 情報)
- **MME** へ - MMEs へ MME 情報 SGW 情報
- **QoS** - QoS 情報
- **UE** - UE 情報

SGW

SGW

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

SGW

SGW

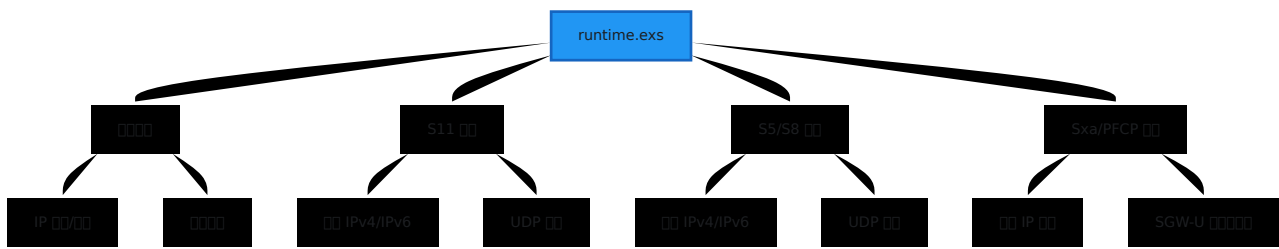
```
[info] 00 OmniSGW...
[info] 127.0.0.40:42068 0000000000
[info] 127.0.0.10 00 S11 00
[info] 127.0.0.15 00 S5/S8 00
[info] 127.0.0.20 00 Sxa 00
[info] 00 PFCP 000000
[info] OmniSGW 0000
```

00 `http://127.0.0.40:42068/metrics` 0000 (000000)0

00

0000000000 `config/runtime.exs` 00000000000000000000

000000



00000000

00	00	00
metrics	Prometheus 000000	00000
s11	GTP-C 0000 MME	S11 00
s5s8	GTP-C 0000 PGW-C	S5/S8 00
sxa	PFCP 0000 SGW-U	Sxa 00

000000000000 0000000000

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
-

- 網路服務品質管理
- 網路服務品質 QoS 管理
- 網路服務品質

網路服務 (QoS) 管理

- 網路服務品質
- 網路 NOC/網路服務
- 網路服務 IP 網路服務

網路服務

網路服務

1. 網路服務
2. 網路 UE 網路
3. 網路 IMSI 網路
4. 網路服務品質管理
 - 網路
 - 網路服務 QoS
 - 網路服務
 - 網路 PGW-C 網路
5. 網路服務 → 網路服務

網路服務

1. 網路 SGW-U 網路 → 網路 SGW-U 網路 "網路"
2. 網路 UE 網路 → 網路服務
3. 網路 APN 網路

網路服務

- 網路 UE 網路
- 網路/網路服務
- 網路服務
- 網路服務

Web UI

Web UI

-
-
-
-
-

Prometheus

-
-
-
-
-

- Web UI Prometheus

Web UI OmniSGW 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 `metrics` 监听地址

部署

部署 OmniSGW 到 Kubernetes 集群

📁📁📁📁

OmniSGW 📁
├── OPERATIONS.md (📄📄)
└── docs/
 ├── 📄📄📄
 │ ├── configuration.md
 │ └── 📄📄📄 📄📄
 ├── 📄📄📄 📄📄
 │ ├── sxa-interface.md
 │ ├── s11-interface.md
 │ └── s5s8-interface.md
 └── 📄
 ├── session-management.md
 ├── bearer-management.md
 ├── cdr-format.md
 └── monitoring.md

📄📄 runtime.exs 📄

Sxa/PFCP (SGW-U 📄)
S11 (MME 📄)
S5/S8 (PGW-C 📄)

UE 📄📄📄
📄📄
📄📄📄
Prometheus 📄📄📄

📄📄📄📄📄📄

📄 📄

📄	📄	📄
OPERATIONS.md	📄📄📄📄 (📄📄)	📄📄📄📄📄

⚙️ 📄

📄	📄
configuration.md	📄📄 runtime.exs 📄📄

📄 📄📄📄

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
session-management.md	세션 관리
cdr-format.md	CDR 형식
monitoring.md	Prometheus 및 Grafana 모니터링

참고

주요 문서

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

주요 문서

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

6. session-management.md - 会话管理

目录结构

- 1. configuration.md - 配置
- 2. monitoring.md - 监控

3GPP

3GPP 规范

规范	描述
TS 29.274	GTP-C v2 (S11 与 S5/S8 接口)
TS 29.244	PCF (Sxa 接口)
TS 32.251	计费数据记录
TS 32.298	CDR 格式
TS 23.401	EPC 架构

SGW-C CDR (CDR) 概要

SGW-C 概要

OmniSGW と Omnitouch の関係

目次

1. 概要
 2. CDR の種類
 3. CDR の取得
 4. CDR の利用
 5. 注意事項
 6. 参考
 7. CDR の取得方法
 8. CDR の利用方法
 9. 参考
 10. 参考
-

概要

SGW-C CDR は、SGW-C において発生する CDR の一覧です。SGW-C CDR は、SGW-C において発生する CDR の一覧です。

SGW-C CDR は、PGW-C CDR と EPC CDR の一覧です。

取得方法

- CSV 形式 - 取得方法
- API 形式 - 取得方法
- 参考 - 取得方法

- 0000 - 0000000000000000
- 00 **3GPP** 00 - 00 3GPP TS 32.251 PS 00000 TS 32.298 CDR 000

0000

0000	00
0000	00000000 CDR
00	00000000
0000	0000000000
0000	0000000000
0000	0000000000

CDR 0000

0000000

<epoch_timestamp>

000

1726598022

0000000000 Unix 00000000000000

0000

00000

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

CDR directory

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
 - SGW-C
 - 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 優先度

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

項目	フィールド名	型	説明
1	<type>_bearer_start	uint32_t	開始時刻
2	<type>_bearer_update	uint32_t	更新時刻
3	<type>_bearer_end	uint32_t	終了時刻/秒

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
- default_bearer_update - デフォルト
- default_bearer_end - デフォルト
- dedicated_bearer_start - デフォルト
- dedicated_bearer_update - デフォルト
- dedicated_bearer_end - デフォルト

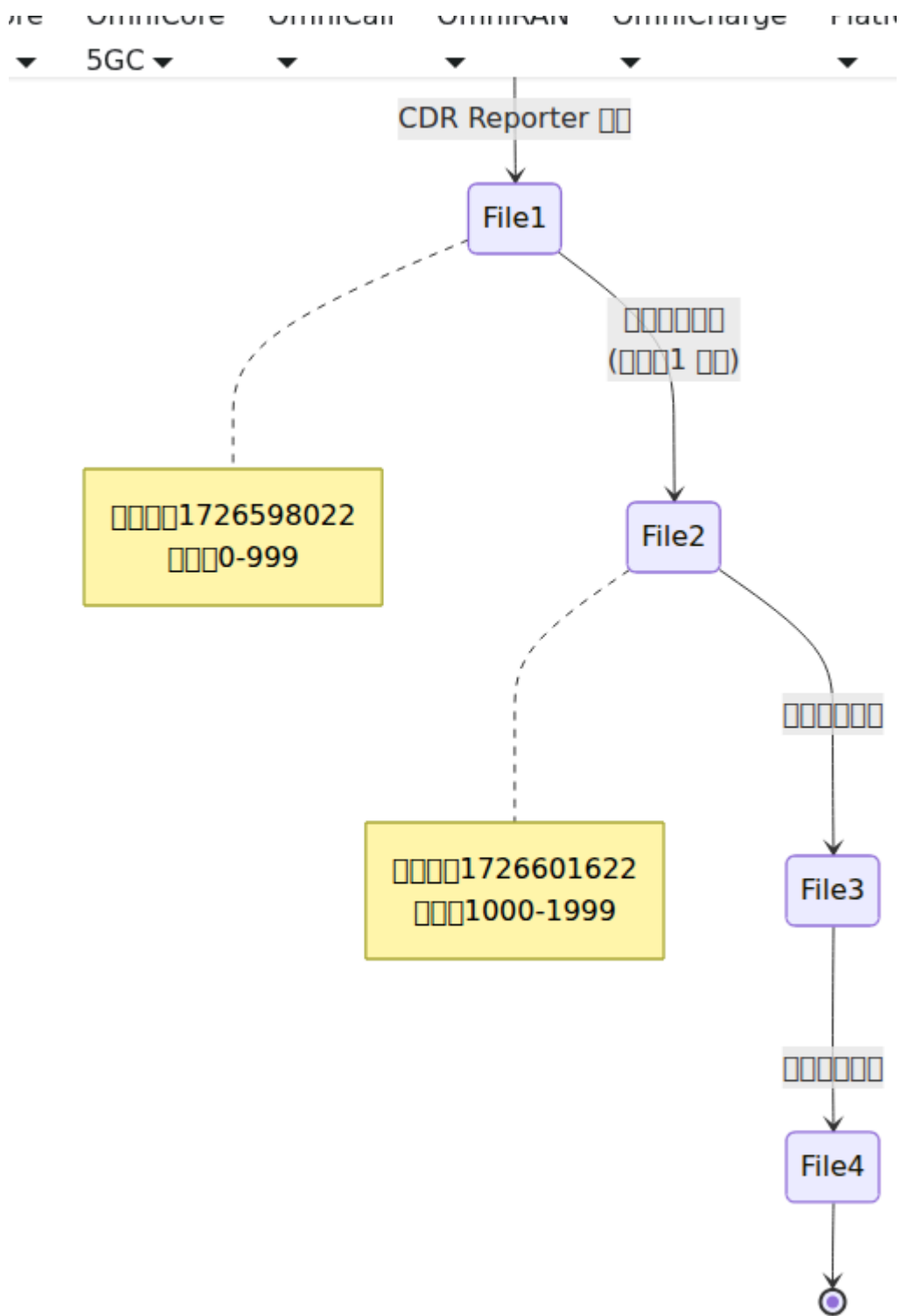
□□□□

□□ **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 □□□□□□□□□□□□□□□□



000000

1. 0000 CDR 00
2. 0000000000000000
3. 0000000000
4. 000 CDR 00000000

필수

선택

필드명	유형	설명	필수성	기본값
gateway_name	문자열	SGW-C 인스턴스명	-	인스턴스 ID
duration	정수	로그 수집 기간 (초)	-	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 초~1 일
- **directory:** "/var/log/sgw_c/cdrs"

예시 2

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

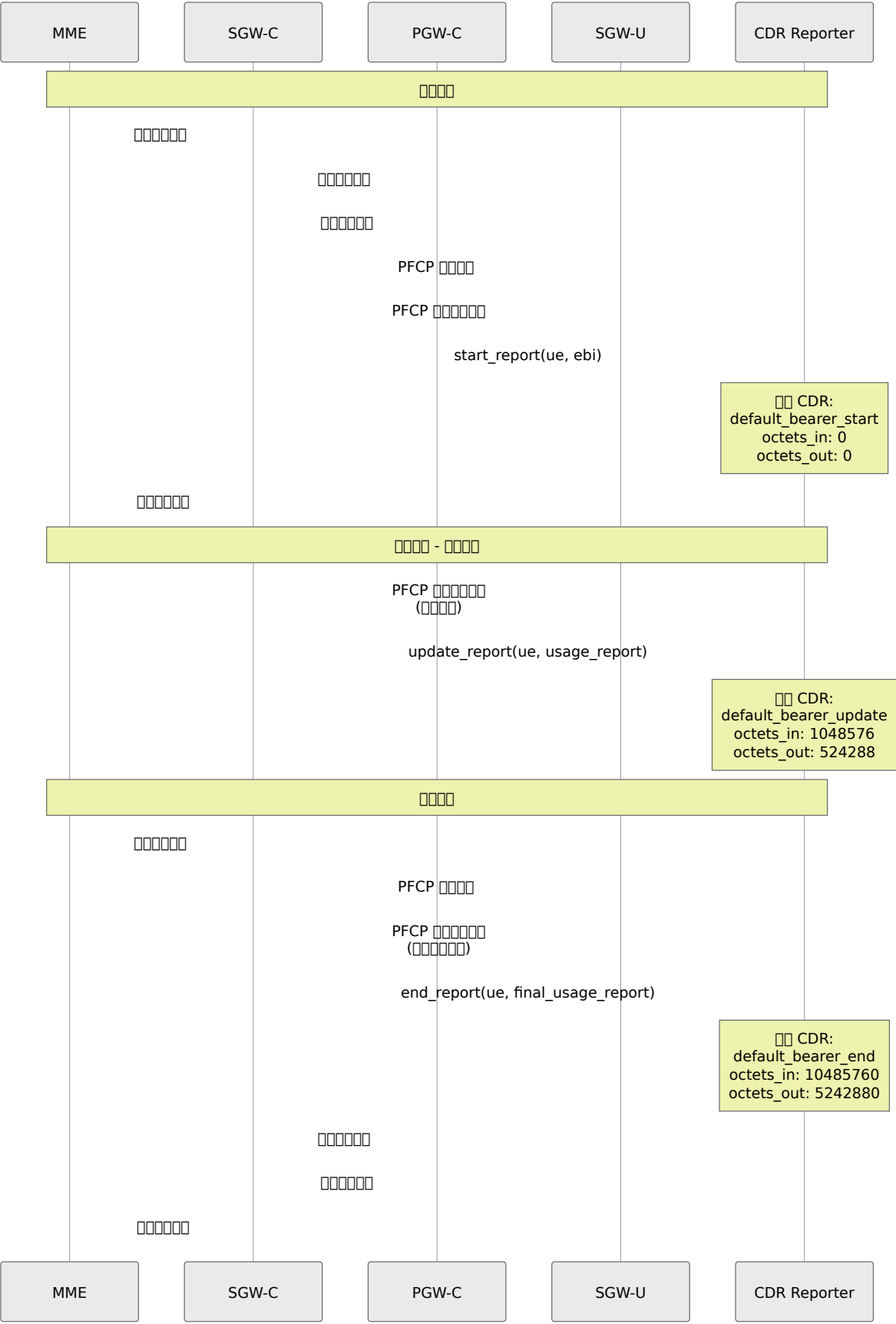
예시 3

- **gateway_name:** "sgw-c-prod-heavy"
 - **duration:** 1,800,000 초~30 일
 - **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
- octets_out** 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
- octets_in** 項目項目
- octets_out** 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 (15桁)

15桁の数字

MCCMNC + MSIN

3桁のMCC、2桁のMNC、10桁のMSIN

例

310260123456789

MCCMNCMSIN

(310)(260)(123456789)

UEのIMSI

3. event (CDR 1桁)

1桁の数字

<bearer_type>_bearer_<event>

例

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

例

- EBI EPS ID LBI ID default
- EBI LBI dedicated

□□□ □□□ 32 □□□

[illegible]

□ □ □

12345

PGW-C ██████████

111

- SGW ↔ PGW 信令面
- 信令面 Diameter Gy/Gz 信令面
- 数据面 数据面

□□□ □□□□E.164 □□□

□□□ □□□ ISDN □□□□□□□□□□

$$\square\square\square\square\square + \square\square\square\square\square$$

111

15551234567

CC National

(1) (5551234567)

```

UE
MME
HSS

```

6. ue_imei (IMEI)

IMEI 15 digits

IMEI TAC8+ SNR6+ Spare1

IMEI is a unique identifier for the device

IMEI

123456789012345

1234567890

12345

TAC SNR S

IMEI UE identifier MME

7. timezone_raw (UE)

IMEI is a unique identifier for the device

IMEI UE identifier

IMEI is a unique identifier for the device

IMEI is a unique identifier for the device

IMEI

, (IMEI)

8. plmn (PLMN)

IMEI is a unique identifier for the device

IMEI is a unique identifier for the device

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 0000000

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 (kbps)	遅延時間 (ms)	サービスタイプ
1	GBR	2	100	10^{-2}	リアルタイム
2	GBR	4	150	10^{-3}	リアルタイム
3	GBR	3	50	10^{-3}	リアルタイム
4	GBR	5	300	10^{-6}	リアルタイム
5	Non-GBR	1	100	10^{-6}	IMS
6	Non-GBR	6	300	10^{-6}	リアルタイム
7	Non-GBR	7	100	10^{-3}	リアルタイム
8	Non-GBR	8	300	10^{-6}	リアルタイム
9	Non-GBR	9	300	10^{-6}	リアルタイム

サービス

9 → リアルタイム

サービス PGW-C サービス QoS

16. octets_in (サービス)

サービス 64 サービス

サービス サービス → UEサービス

サービス

サービス

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

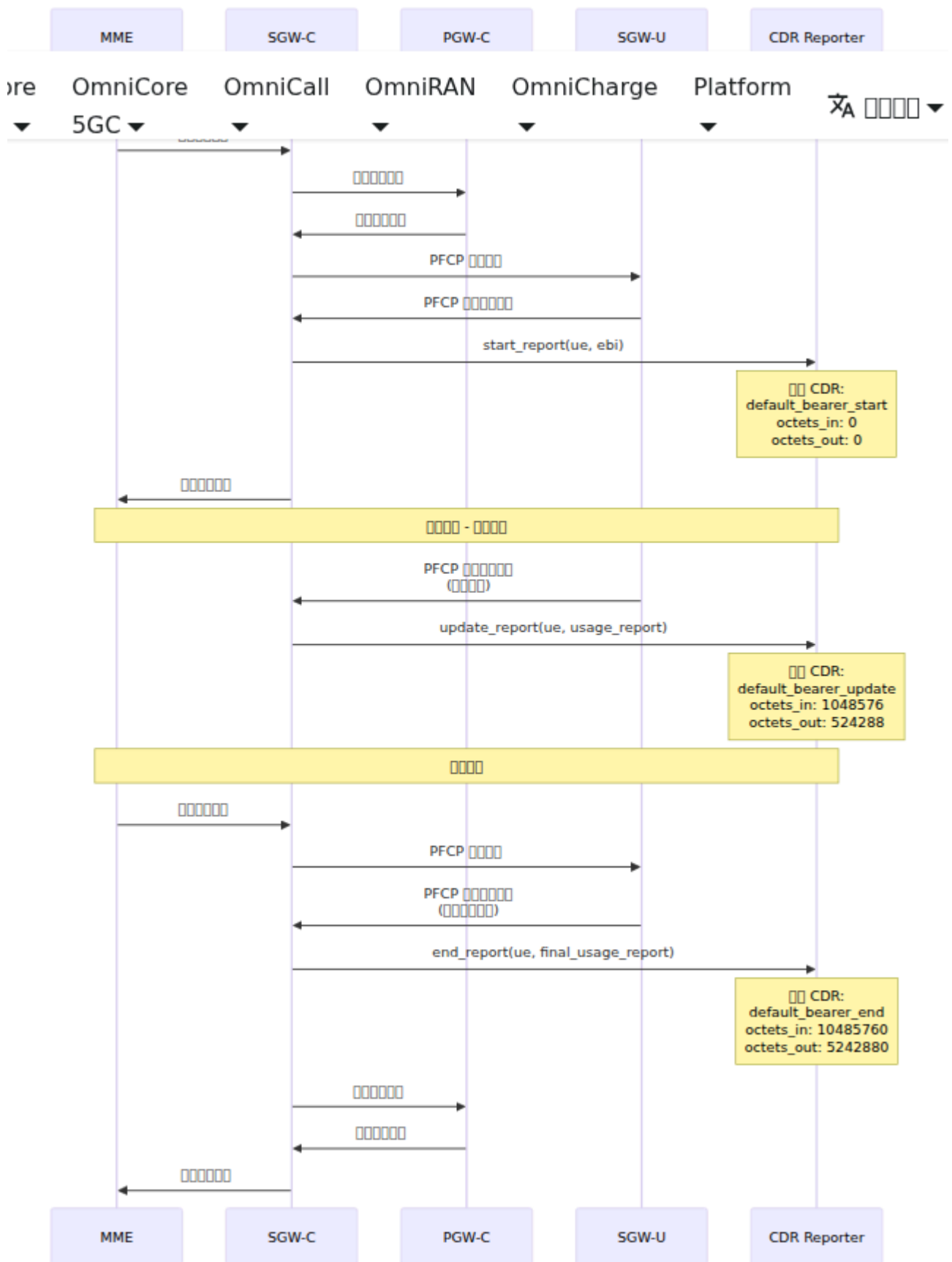
```
1726570800,3102603333333333,default_bearer_start,10003,15553333333,3333333333,1726571100,3102603333333333,dedicated_bearer_start,10004,15553333333,3333333333,1726571400,3102603333333333,default_bearer_update,10003,15553333333,3333333333,1726571400,3102603333333333,dedicated_bearer_update,10004,15553333333,3333333333,1726571700,3102603333333333,dedicated_bear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

Summary

- 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

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

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

配置 S11

```
config :sgw_c,
  s11: %{
    # S11 配置 > 100ms RTT
    message_timeout_ms: 10000,
    max_retries: 5,
    retry_backoff_ms: 2000,

    # S11 配置 < 50ms RTT
    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 地址  
    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"  
      }  
    ],  
  
    # 配置消息超时、重试和退避  
    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-PROD
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 分钟
directory: "/mnt/fast-ssd/sgw_c/cdrs"
}
```

配置项

配置表

名称	描述	值
SGW_S11_IP	S11 接口 IP	10.0.0.10
SGW_S5S8_IP	S5/S8 接口 IP	10.0.0.15
SGW_SXA_IP	Sxa 接口 IP	10.0.0.20
MGT_IP	管理接口 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

実行コマンド

```
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

環境変数の確認

実行結果の確認

```
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" PGW

確認 S5/S8 確認 PGW-C

確認

```
# 確認 PGW IP
ping 10.0.0.20

# 確認
iptables -L | grep 2123

# 確認
nc -u -v 10.0.0.20 2123
```

"確認" SGW-U

確認 Sxa 確認

確認

```
# 確認 SGW-U 確認
ping 10.0.0.30

# 確認 PFCP 確認
netstat -an | grep 8805

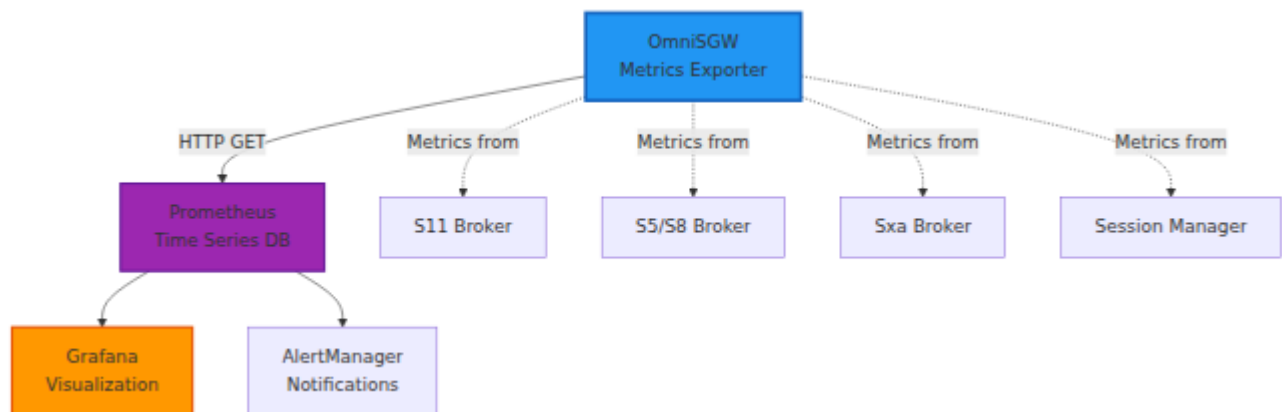
# 確認 PFCP 確認
iptables -L | grep 8805
```



11

- 11

--	--	--	--



□□□□

□□□□□□

□□□□

```
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

```

00000000

```

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

```
pfcf_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:
    # 容量告警
    - alert: SGWCapacityHigh
      expr: (teid_registry_count / 100000) > 0.8
      for: 5m
      annotations:
        summary: "SGW 容量告警 80%"
        description: "容量告警 {{ $value }} / 100000"

    # S11接口告警
    - alert: S11PeerDown
      expr: absent(s11_inbound_messages_total) > 0
      for: 2m
      annotations:
        summary: "S11 接口告警"

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

    - alert: SGWUAssociationDown
      expr: pfcf_association_status == 0
      for: 1m
      annotations:
        summary: "SGW-U 接口告警"
        description: "接口告警 {{ $labels.peer_ip }}"

    # S11接口延迟告警
    - alert: S11LatencyHigh
      expr: histogram_quantile(0.95,
rate(s11_inbound_duration_seconds_bucket[5m])) > 1
      for: 5m
      annotations:
        summary: "S11 接口延迟 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 に Sxa 対応

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

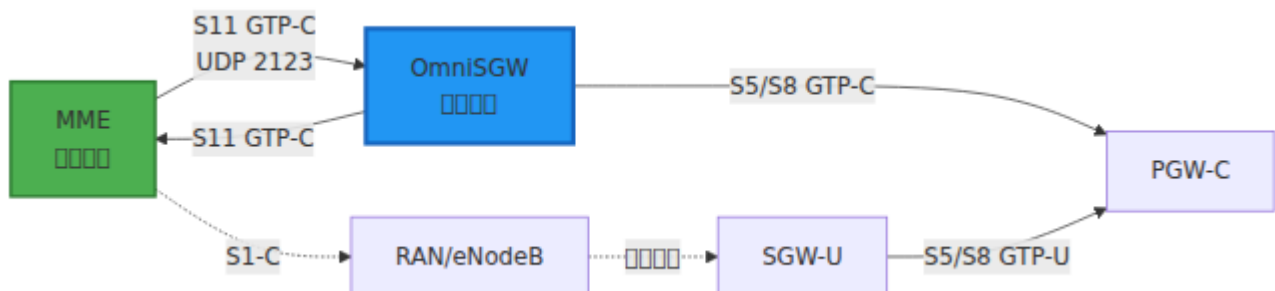
OmniSGW 与 Omnitouch 接口

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
接口 UE 接口



TEID

- **GTP-C v2** 프로토콜 - IPsec 터널링
- 프로토콜 **TEID** 프로토콜 - 프로토콜 프로토콜 프로토콜
- 프로토콜 프로토콜 - 프로토콜 프로토콜 UE 프로토콜
- 프로토콜 - 프로토콜 MME 프로토콜 MME 프로토콜
- 프로토콜 - 프로토콜 프로토콜 프로토콜
- 프로토콜 프로토콜 - 프로토콜 프로토콜

TEID

GTP-C 프로토콜 2

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

TEID 프로토콜 프로토콜

프로토콜 프로토콜 TEID 프로토콜

- 프로토콜 **TEID** - 프로토콜 OmniSGW 프로토콜 MME 프로토콜
- 프로토콜 **TEID** - 프로토콜 MME 프로토콜 MME 프로토콜

프로토콜

MME → SGW: 프로토콜 OmniSGW 프로토콜 TEID

SGW → MME: 프로토콜 MME 프로토콜 TEID

TEID

프로토콜 S11 프로토콜 GTP-C v2 프로토콜

GTP-C 消息 (12-16 字节)

- ├─ 消息头 (3 字节): 0x2 (GTP-C v2)
- ├─ 消息类型 (1 字节)
- ├─ TEID 消息头 (1 字节): 1 (TEID 消息)
- ├─ 消息体 (8 字节): 消息体
- ├─ 消息体 (16 字节): 消息体
- ├─ TEID (32 字节): 消息体
- ├─ 消息体 (24 字节): 消息体/消息体
- └─ 消息体 (8 字节): 消息体 0

消息体 (消息)

- ├─ 消息体 (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	SGW
MEI	UE	UE
ULI	UE	UE
UE	UE	UE
TAI	TAI	UE
ECGI	ECGI	E-UTRAN

UE MME

IE	UE	SGW
UE	UE	UE
UE	UE	UE

UE MME

UE MME

UE MME SGW

UE MME QoS

UE MME

- UE MME

- 3GPP
- 3GPP

3GPP/3GPP

3GPP MME 3GPP SGW 3GPP PGW

3GPP 3GPP

3GPP

3GPP/3GPP

3GPP MME → OmniSGW

3GPP 3GPP

3GPP

- 3GPP
- 3GPP
- 3GPP-SGW-U 3GPP
- UE 3GPP

3GPP/3GPP

3GPP OmniSGW → MME 3GPP MME → OmniSGW

3GPP 3GPP

3GPP

3GPP **S11**

3GPP PGW-C → OmniSGW → MME

3GPP MME 3GPP UE 3GPP

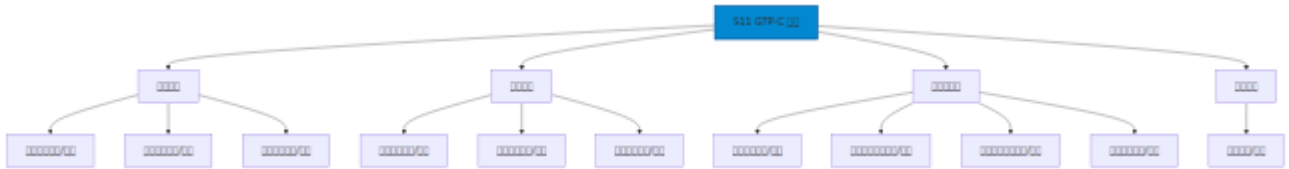
3GPP

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

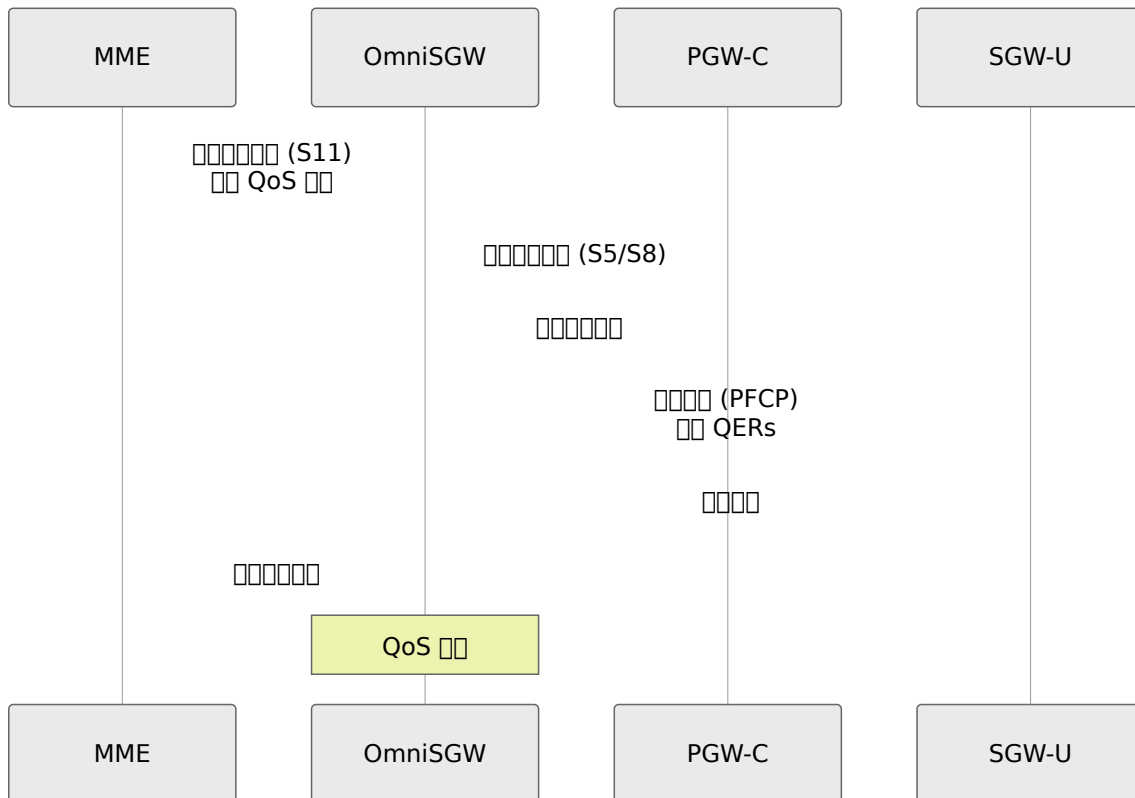


項目

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

QoS

QoS



TAU

SGW TAU

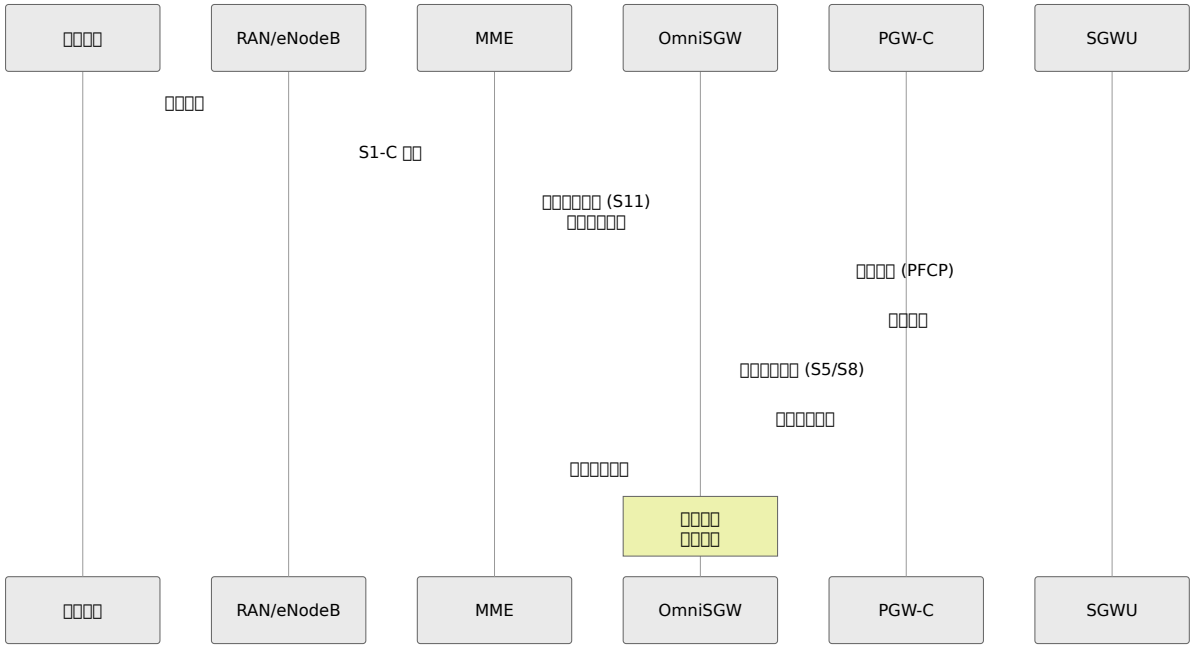
- MME UE
- ULI/TAI SGW
- SGW UE
- TAU

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
```

UE

UE S11

Web UI → UE

- UE
- IMSI & GUTI
 - TAI
 - TEID S11
 - TEID MME
 - QoS
 - PGW-C

UE

UE

```
#
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

- 環境構築が完了していること
- 端末 TEID が一意に決まっていること
- 端末 MME が一意に決まっていること

確認

端末 環境構築状況

確認

1. 環境構築が完了していること
2. 端末が "TAU" の状態にあること
3. PFCP が正常に動作していること

確認

- SGW-U が正常に動作していること
- 環境構築が完了していること
- 環境構築が完了していること

確認

端末 S11 環境構築

確認

```
# 環境構築状況
curl -s http://127.0.0.40:42068/metrics | grep
s11_inbound_duration_seconds

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

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

概要

1. 概要
2. 3GPP R16 PFCP 仕様
3. 3GPP SGW-C 実装
4. 3GPP CPU 実装

3GPP R16 PFCP 仕様 Prometheus 監視

3GPP R16

概要

- 3GPP R16 S11 仕様
- 3GPP R16 RTT 仕様
- 3GPP R16 仕様

3GPP

- 3GPP R16 仕様
- 3GPP R16 MME 仕様
- 3GPP R16 S11 仕様
- 3GPP R16 仕様

3GPP

- 3GPP R16 S11 仕様
 - 3GPP R16 S11 仕様 MME IP
 - 3GPP R16 仕様
-

□□□□□□

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

S11 □□ - MME □ SGW-C □□□□□□

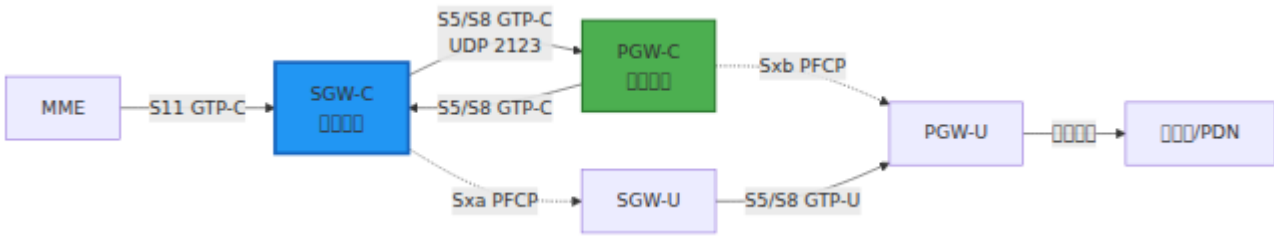
S5/S8 配置

PGW-C GTP-C

OmniSGW Omnitouch

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

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



IP

- **GTP-C v2** - 3GPP TS 29.274
- **TEID** - Tunnel Endpoint Identifier
- **PDN** - PDN Type / PDN Address
- **IP** - IP Address
- **ID** - IP Address
- **IP** - PGW IP Address UE IP

IP

GTP-C 2

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

TEID

PDN TEID

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

TEID

SGW-C → PGW-C PGW-C TEID
PGW-C → SGW-C SGW-C TEID

ID

ID

- 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
```

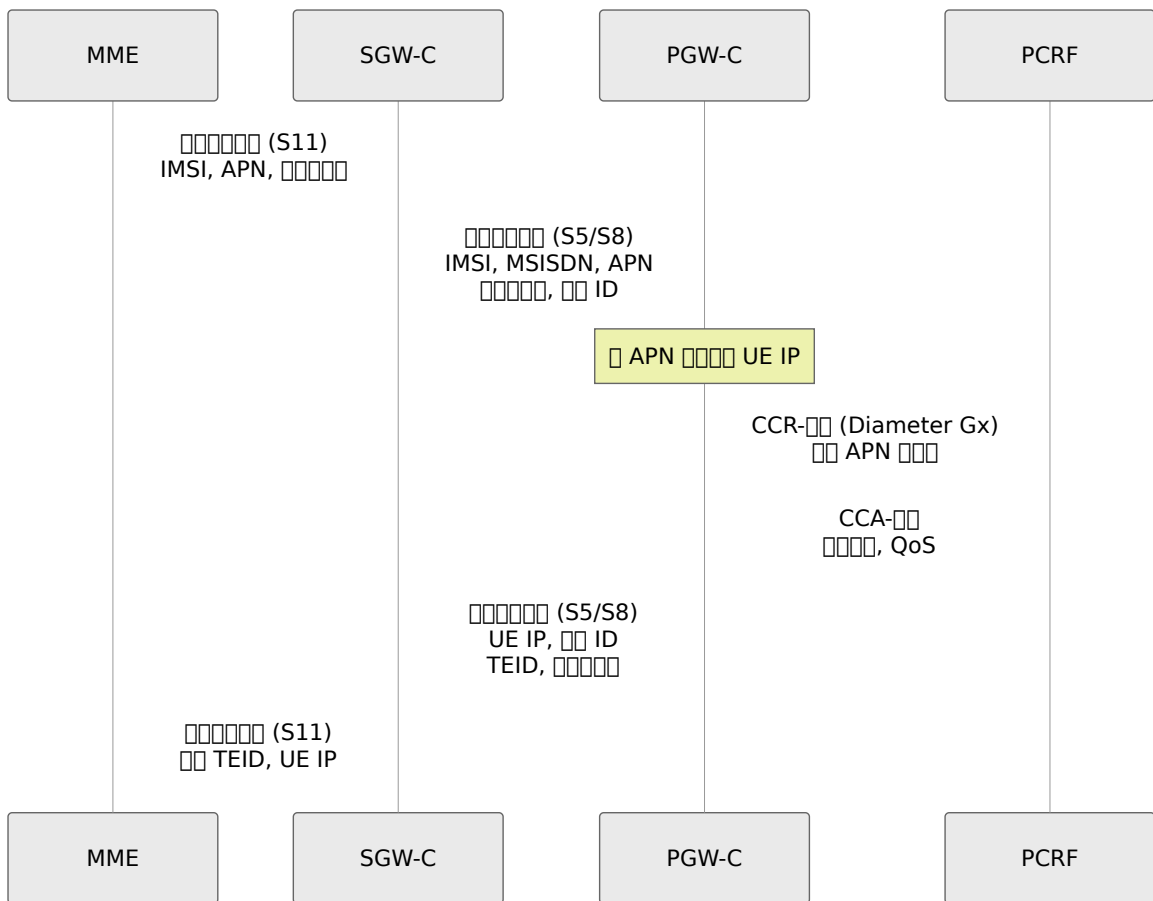
□□□

```
# □□□ 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 属性	MME	QoS 属性 QCI, ARP
APN	MME	アクセスポイント名, 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 IP アドレス

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

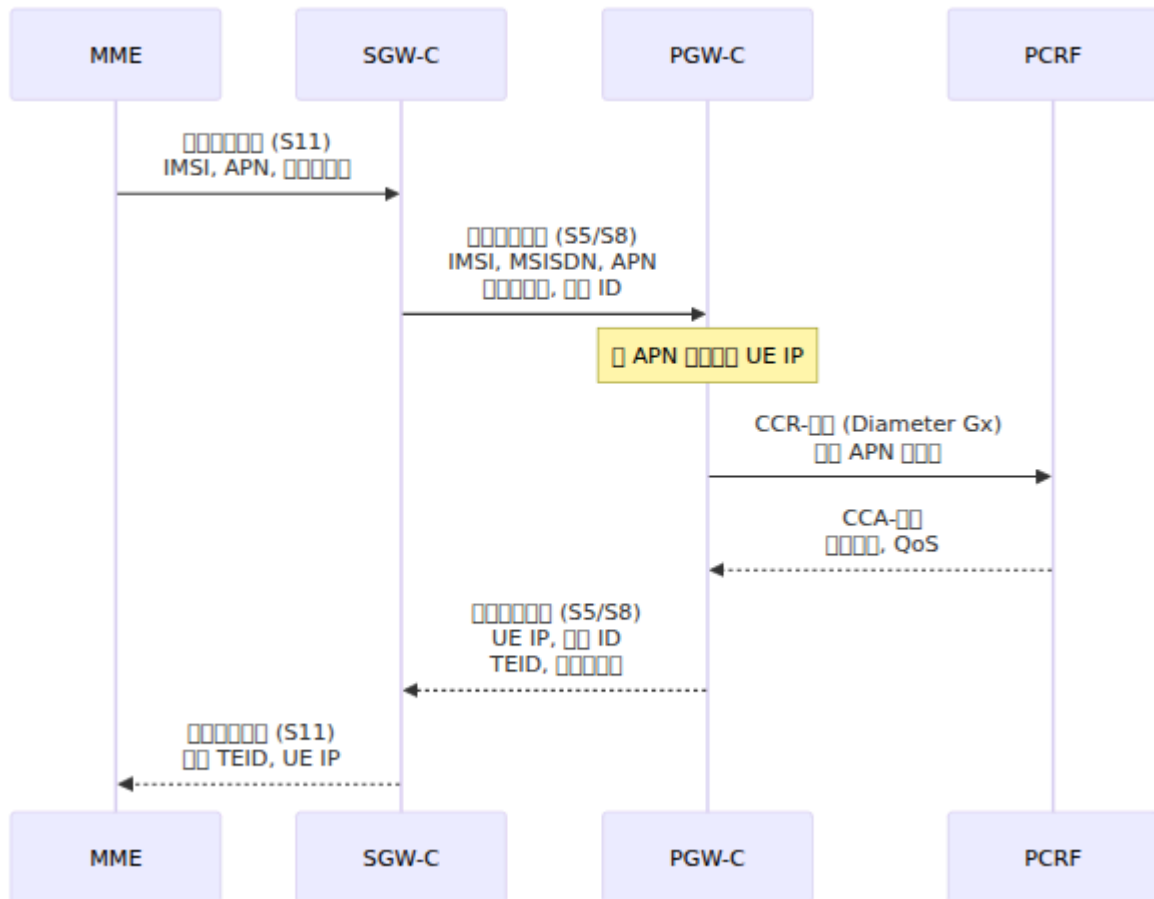
网络

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

□□□□

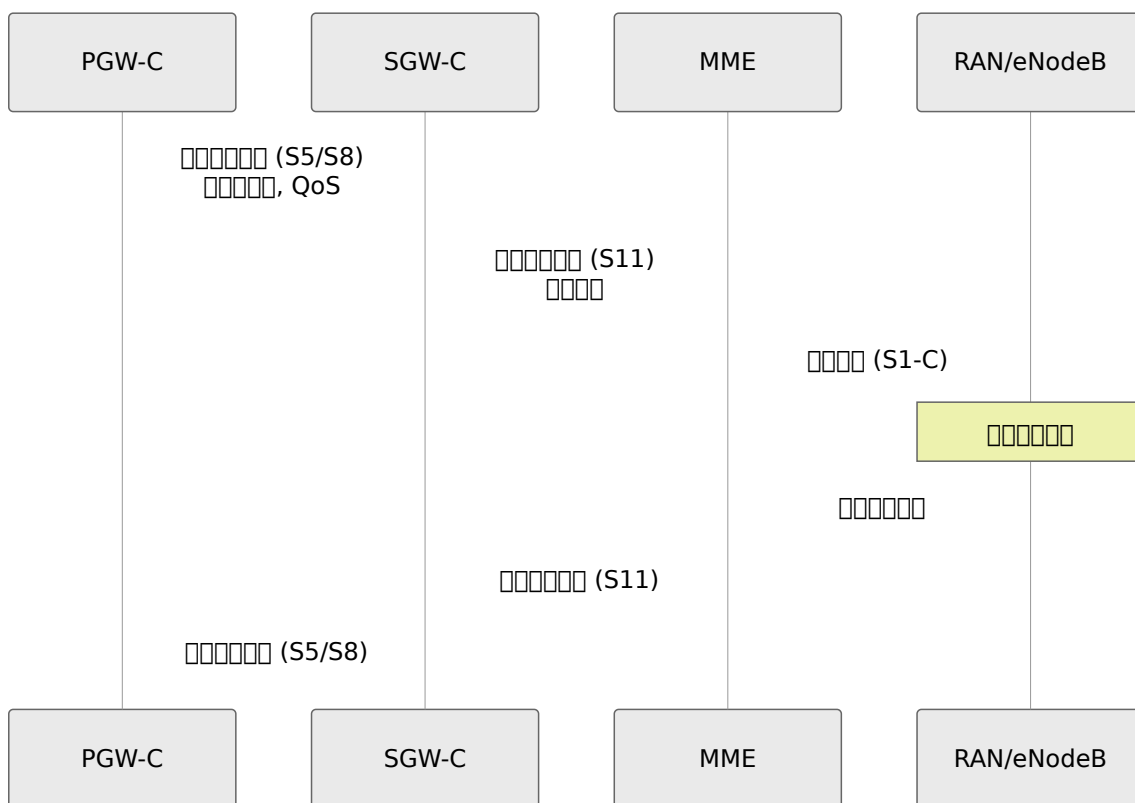
□□ QoS □□

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



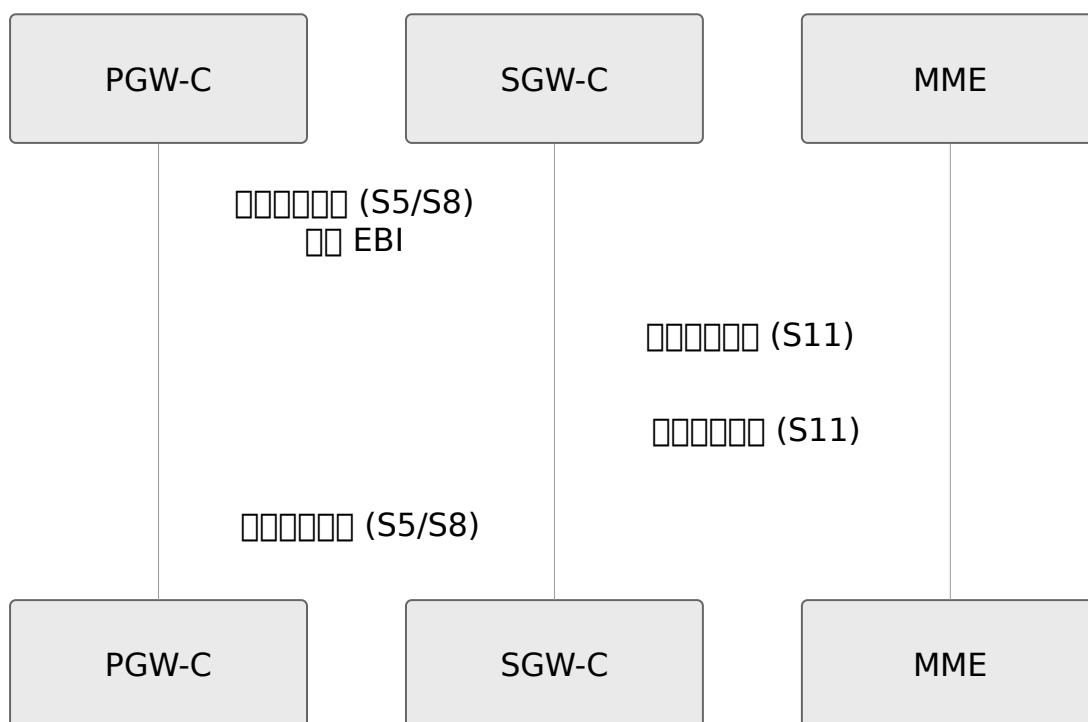
□□□□□□□□□□

PGW-C □□□□ S5/S8 □□□□□□□□□□



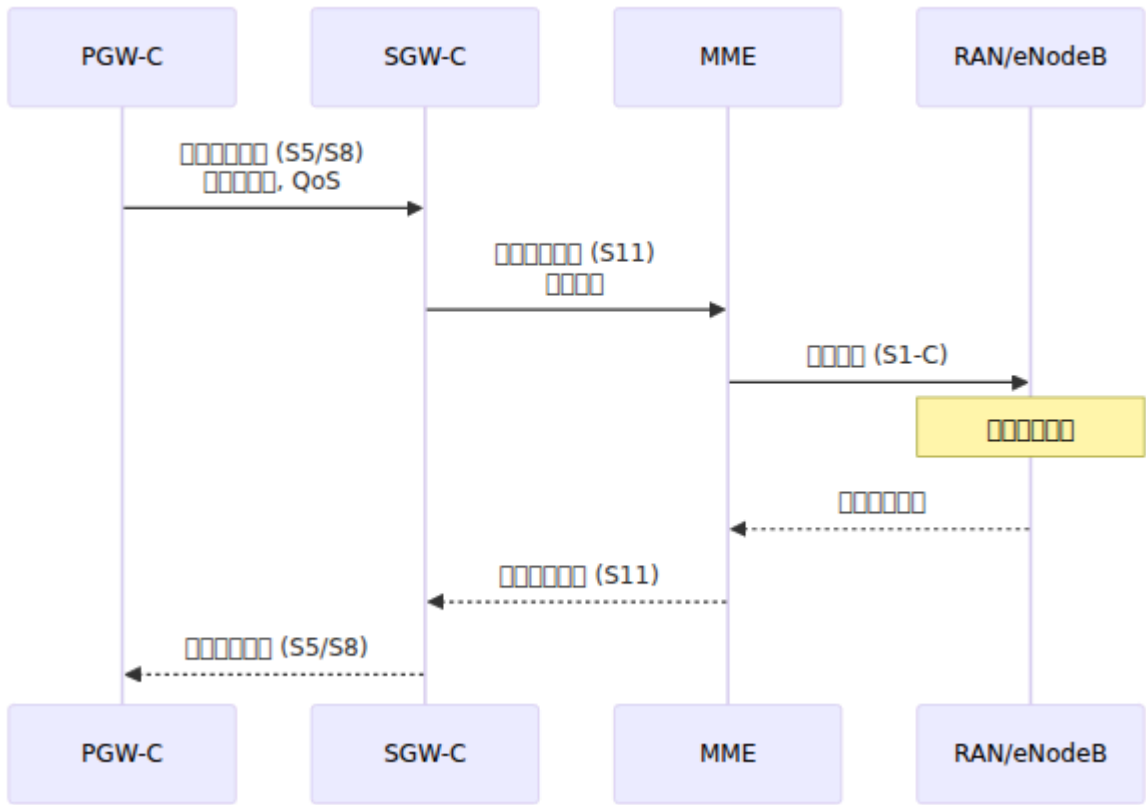
Network components and connections

Network components and connections



□□□□

□□ **PDN** □□□□

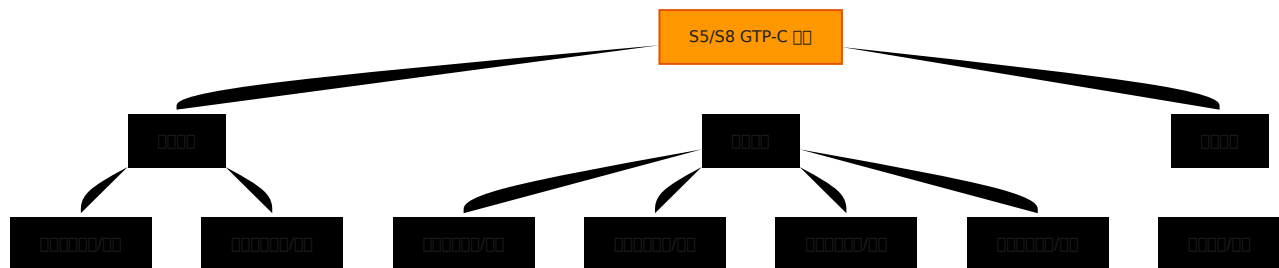


□□□□□

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

□□□□

S5/S8 □□□□



□□□□

□□□□□□/□□

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

□□□□□□/□□

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

□□□□□□/□□

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

□□/□□□□□□□□/□□

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

□□□□□□/□□

- S5/S8 TEID
- S5/S8 TEID 1
- S5/S8 TEID

PGW-C

PGW-C

PGW-C

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

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

# S5/S8
```

PDN

PDN

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

PGW-C

S5/S8

```
# 確認セッション
curl -s http://127.0.0.40:42068/metrics | grep
create_session_request_total

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

# 確認
curl -s http://127.0.0.40:42068/metrics | grep
s5s8_inbound_errors_total
```

PGW 確認

確認 PGW-C 確認

```
確認
├─ 確認
├─ 確認 APN 確認 PGW
├─ 確認-確認
└─ 確認
```

確認

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

確認

確認

確認 "確認 PGW 確認"

確認

```
# 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
```

创建会话

序号	指标	说明
16	创建会话	PGW 创建 IP 地址
25	创建会话	PGW 创建
49	创建会话	PGW 的 APN 名称
72	PGW IE	MME 创建会话

创建会话

创建 "创建会话"

创建

1. 创建会话
2. PGW QoS 创建
3. PGW 创建

创建

- PGW QoS 创建
- PGW 创建 QoS
- PGW 创建 PCRF/创建

🔍 🔍 🔍 🔍 🔍

🔍 "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.

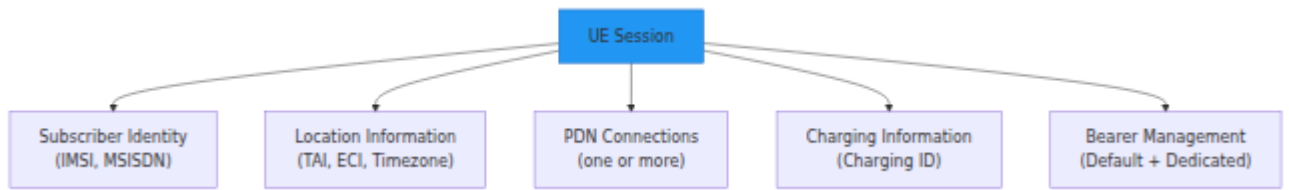


UE □□□□□□□□□□□□□□□□ SGW-C □□□□□□□□□□□□□□□□

- **MME** - 管理平面 S11
- **PGW-C** - 控制平面 S5/S8
- **SGW-U** - 用户平面 Sxa

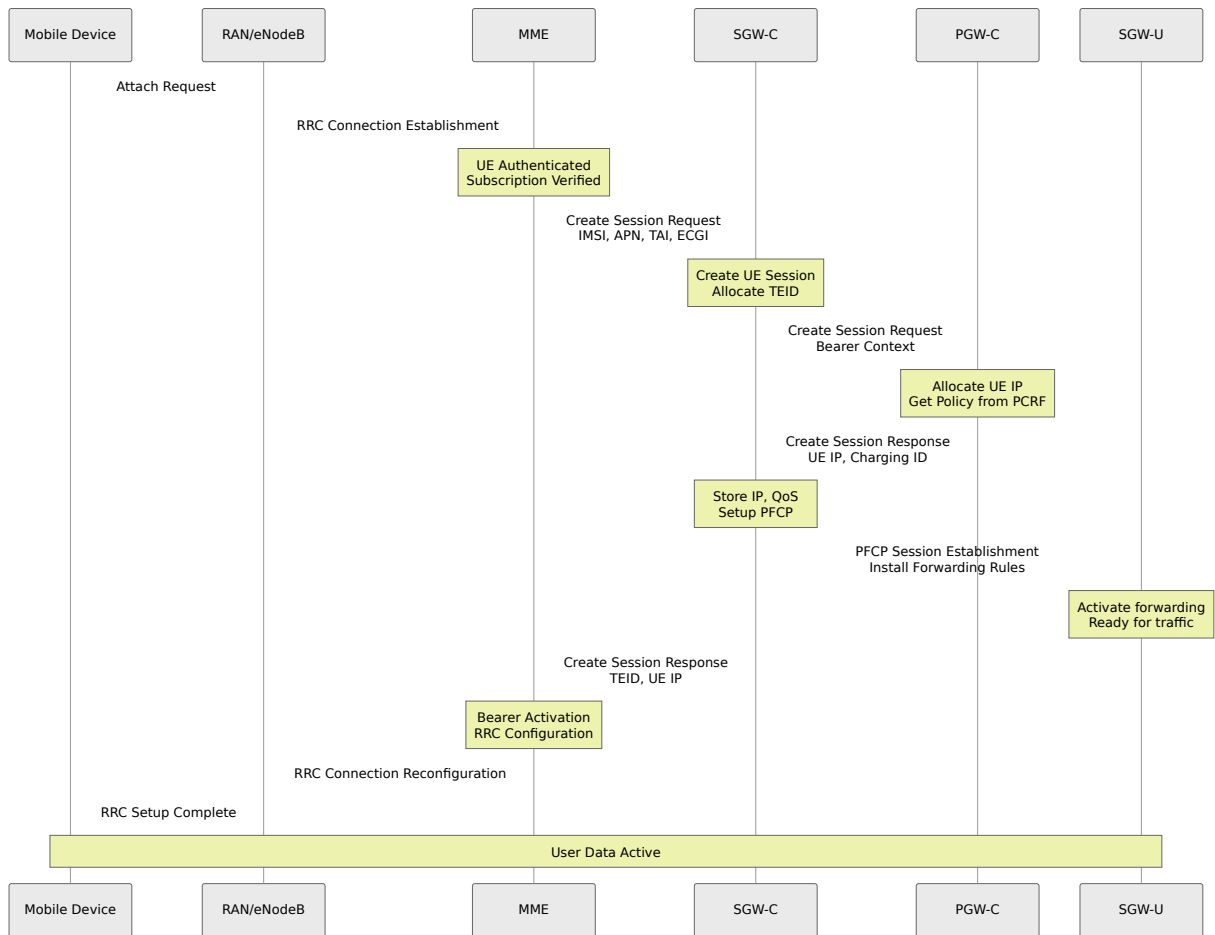
□□□□□□□□ IMSI □□□□□□□□□□□□□□□□ PDN □□□

□□□□

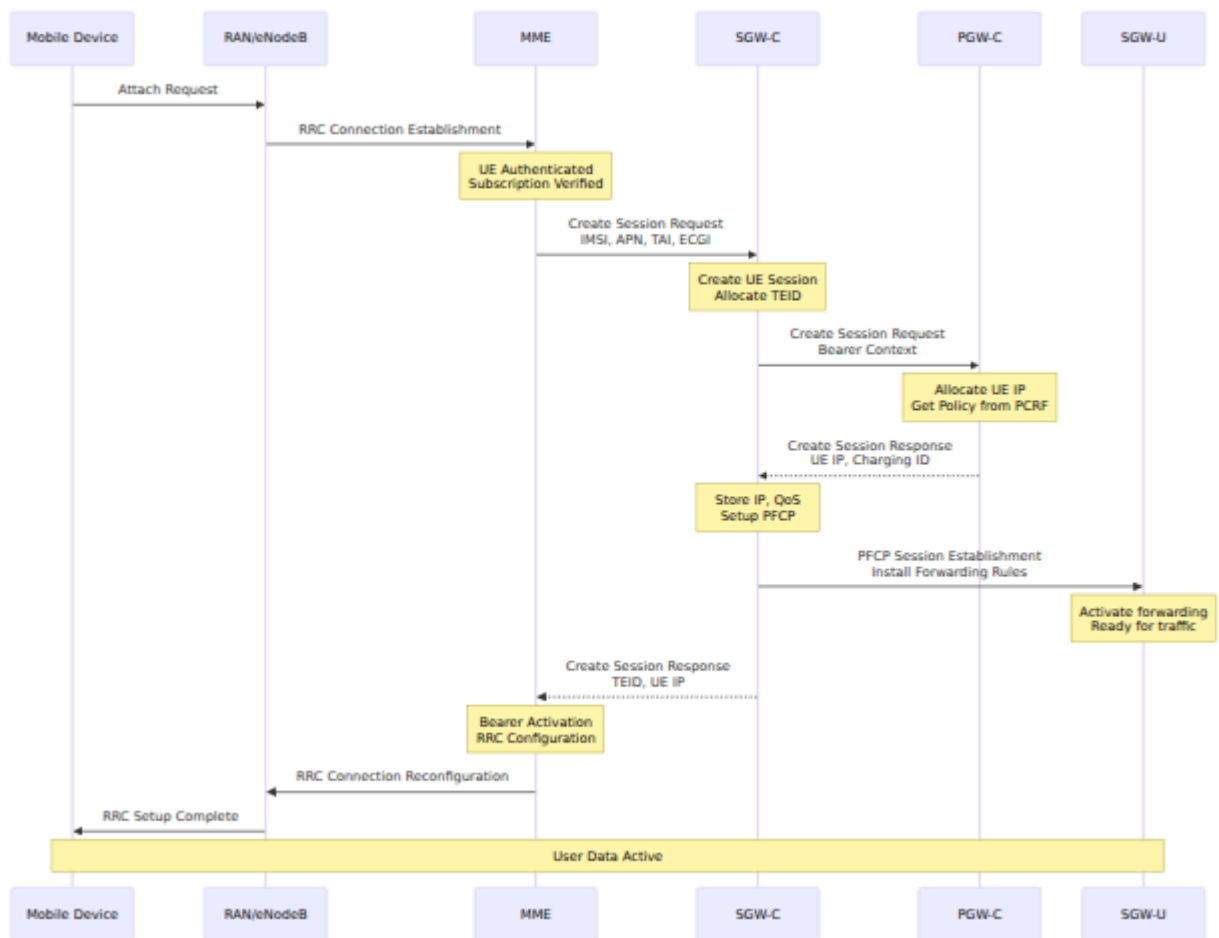


□□□□□□

□□□□□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

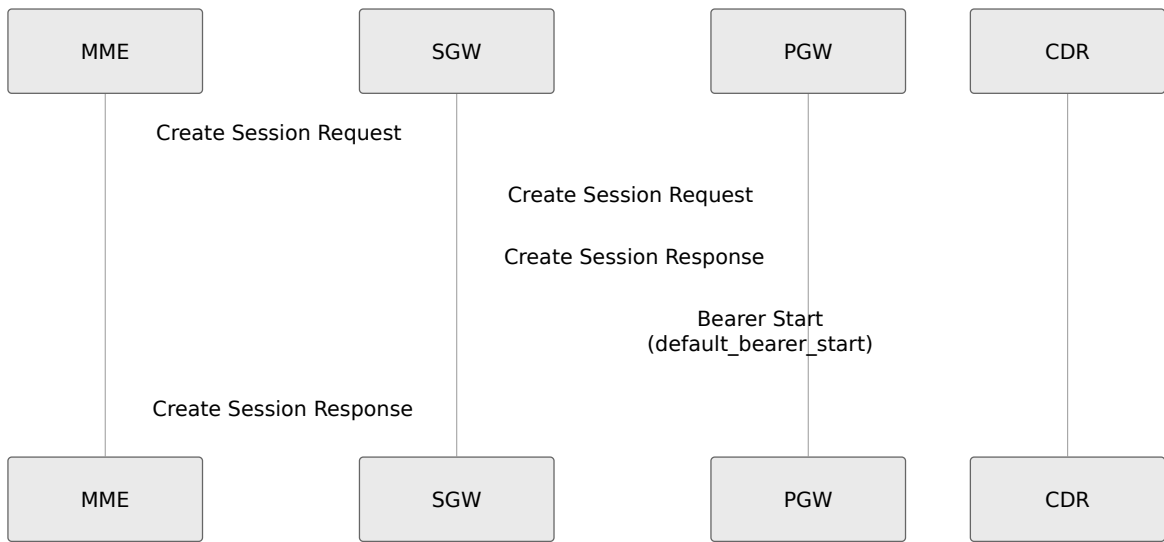
PDN Context

PDN Context

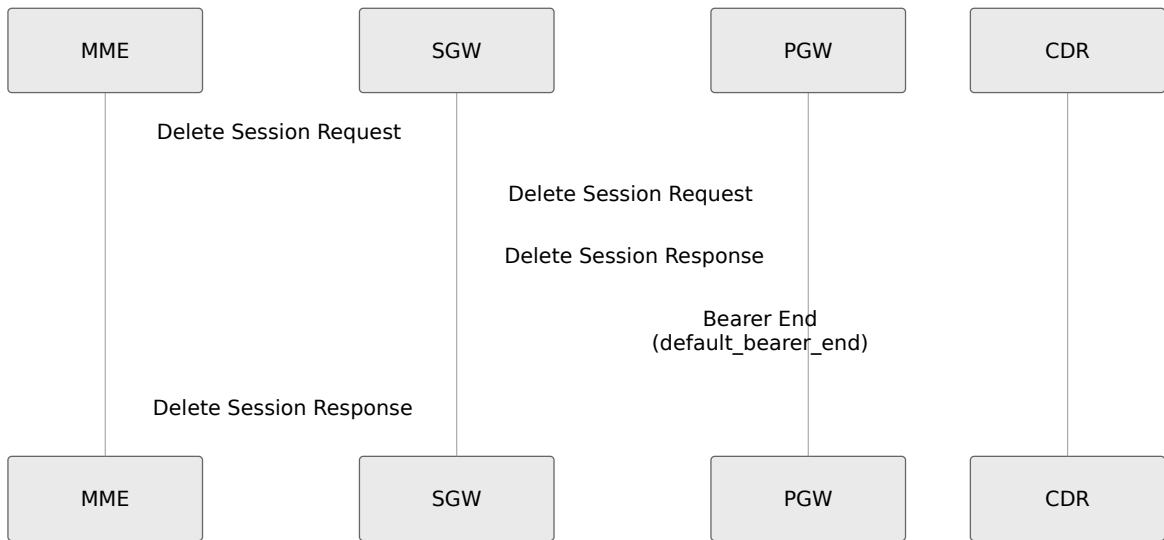
PDN Context PDN Context

- **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)

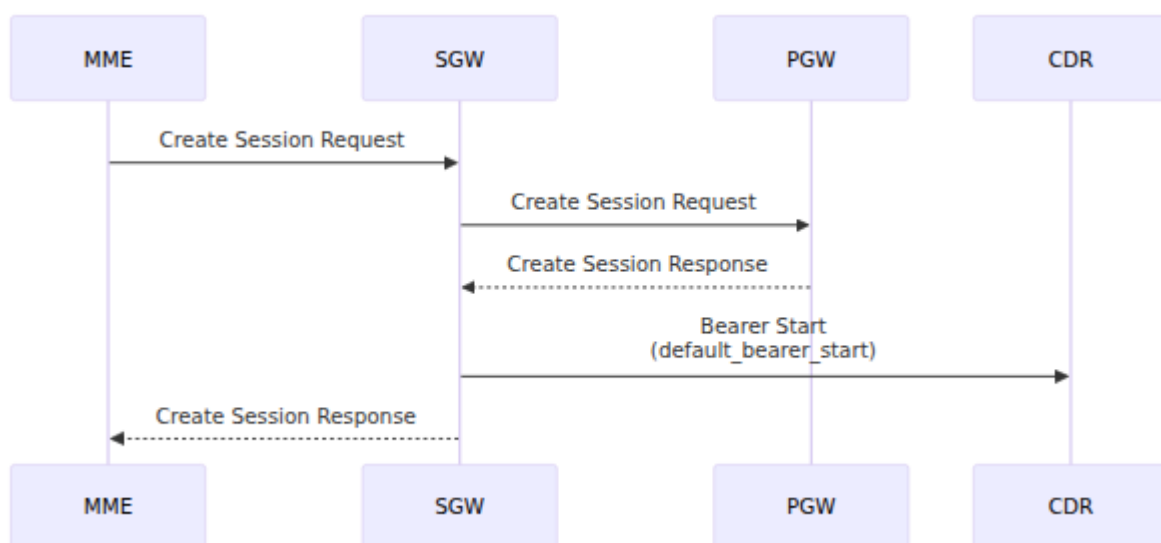
↓

MME deactivates bearer on RAN

↓

Delete Bearer Response back through SGW to PGW

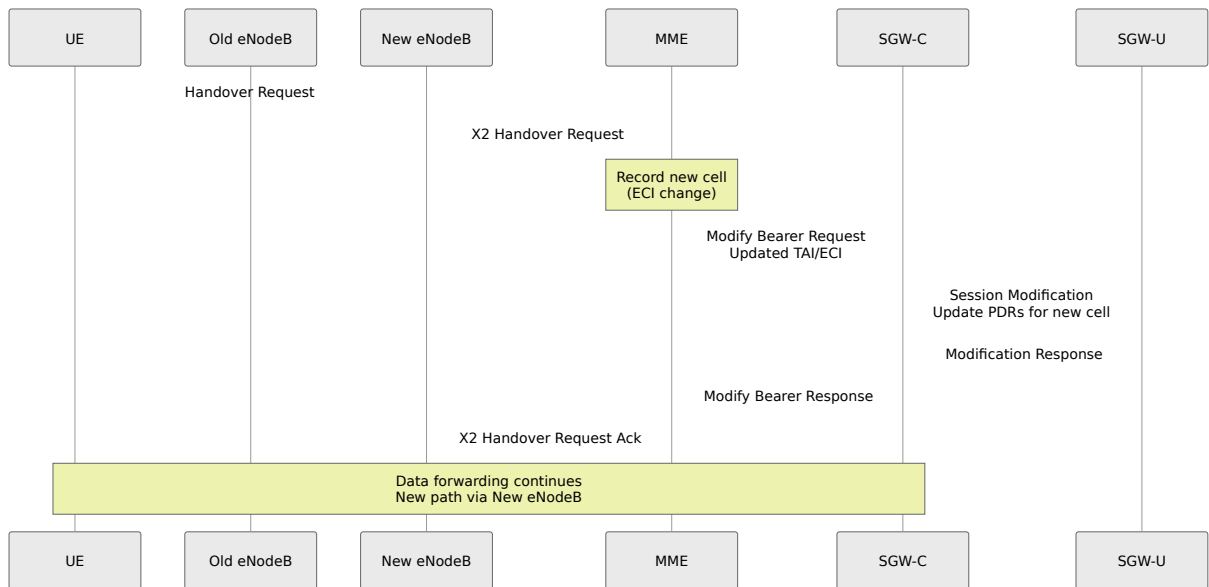
□□ **QoS** □□



Handover

Handover from MME to SGW

Handover from UE to MME

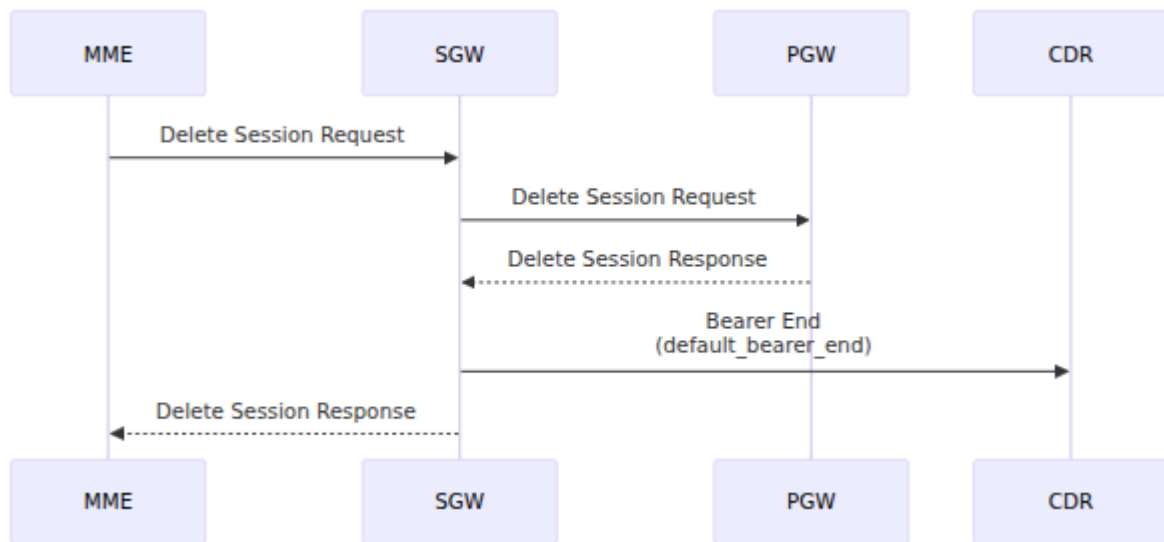


Handover from MME to SGW

- Handover from MME to SGW
- TEID Handover
- Handover from MME to SGW
- CDR Handover Charging ID

Handover from MME to SGW

Handover from UE to MME



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 確認
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.

2.

3.

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

4.

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

5.

6.

7.

```
# 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**  

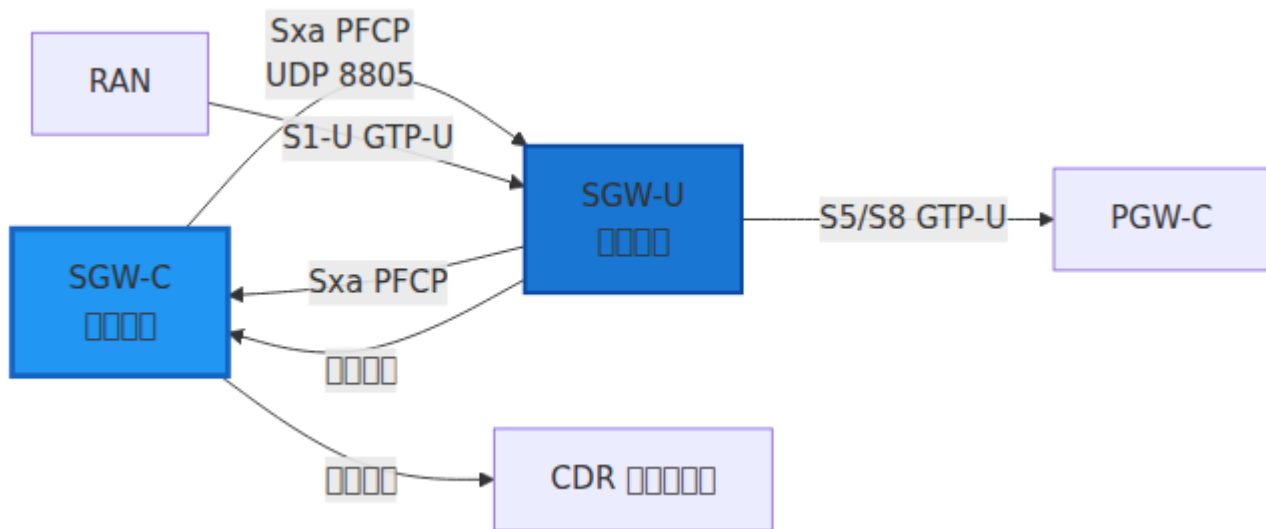
OmniSGW  *Omnitouch*    



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



Sxa   **PFCP**                                                



SGW-C

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

SGW-U

PFCP v1.0

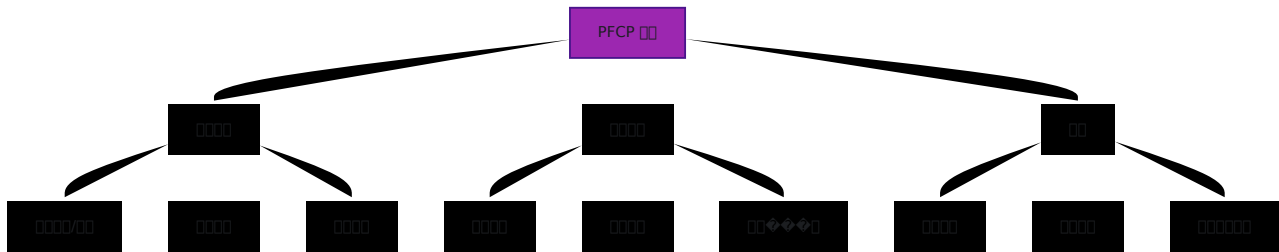
- PFCP v1.0 (3GPP TS 29.244)
- UDP
- 8805 port
- CP & UP

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
```

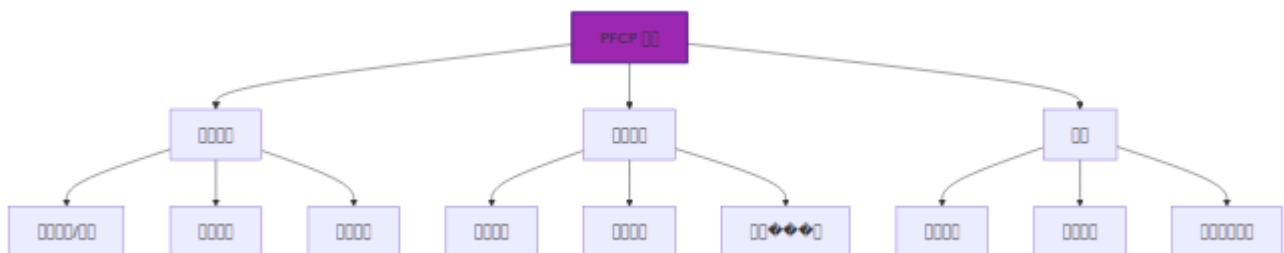
```
# SGW-U
ip route add <sgwu_network>/24 via <gateway_ip> dev eth0
```

```
# 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. □□□□□

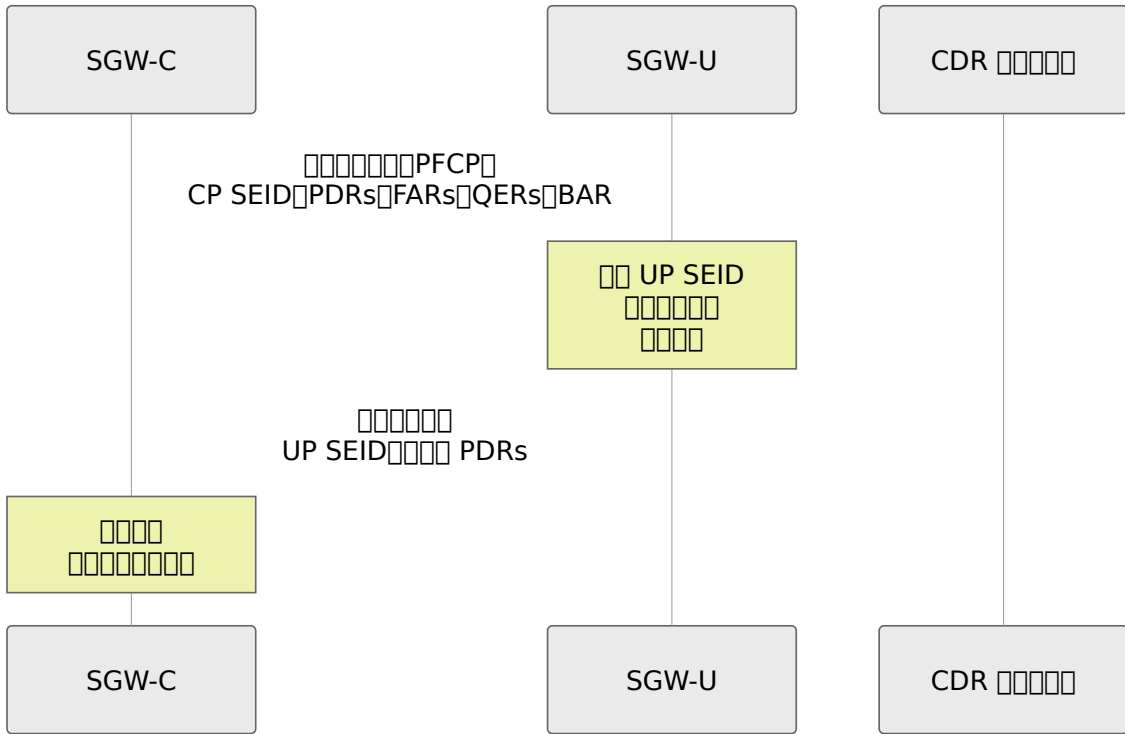
- □□□□□□□□□□□□□□□□
- □□ SGW-U □□□□□□□□
- □□□□□□□□□□□□□□□□

3. □□□□□

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

--	--	--	--

000 00 MME 0000000000 S11 0000



□□□□□□□□

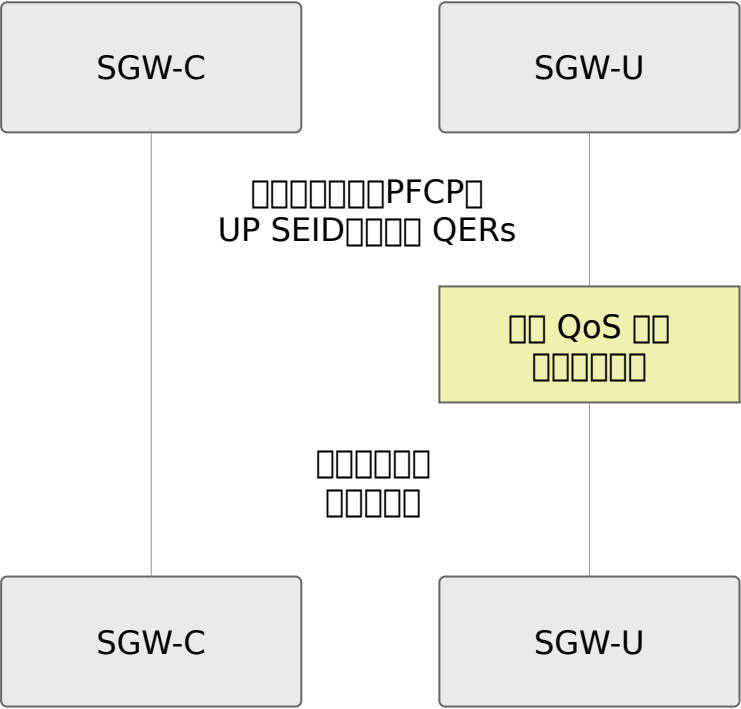
項目	内容
CP SEID	SGW-C 識別番号
PDRs	パケット定義規則
FARs	フィルタルール
QERs	QoS 設定
BAR	帯域幅保証
他の PDR	その他のパケット定義規則

□□□□

[□□□]
↓ (□□□□)
[□□□]
↓ (□□□□)
[□□□□]

□□□□

□□□ □□ MME □□□□□□□□□□ QoS □□□□□□

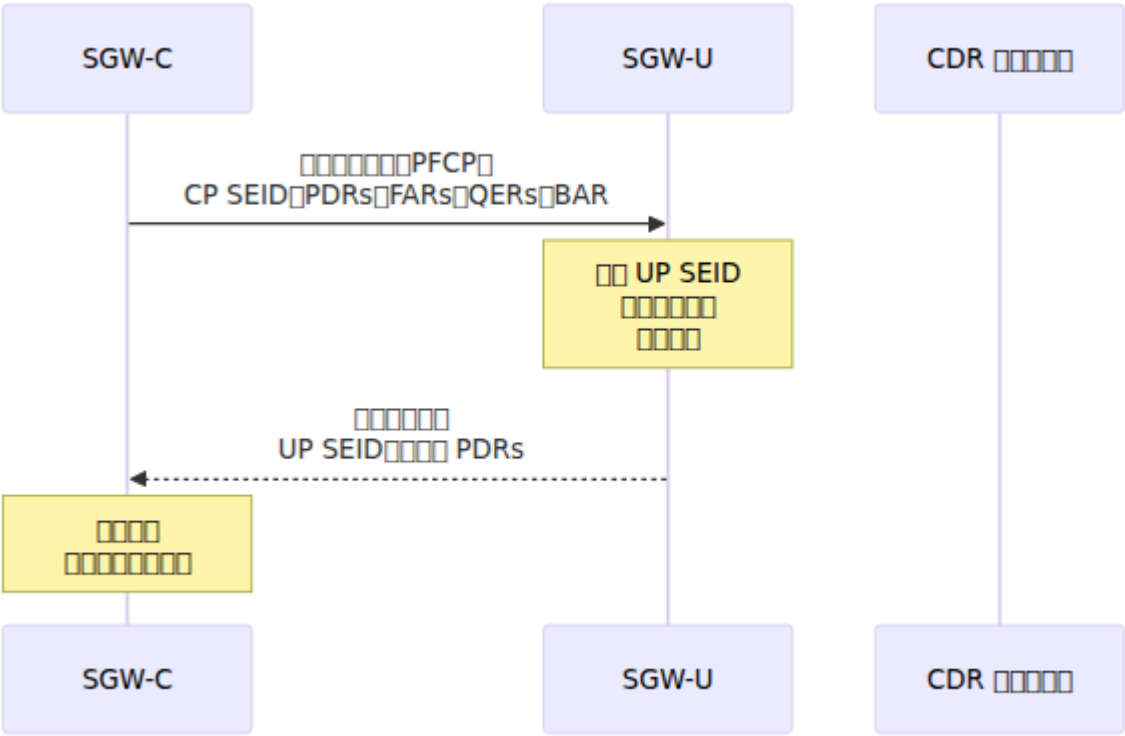


□□□□

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

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)
[項目項目]
↓ (項目項目)
[項目項目]

PFCP ☐ ☐

PDR

□ □ □ □ □ □ □ □ □ □ □ □ ◆ ◆

□ □ □ □ □ □ □ □

項目	項目	項目
IP アドレス	IP アドレス	IP アドレス S1-U から S5/S8 へ
IP アドレス	UE IP アドレス	10.45.0.50
IP アドレス	IP アドレス	8.8.8.8
ポート番号	IP アドレス	TCP 6 UDP 17
ポート番号	ポート番号	1024-65535
ポート番号	ポート番号	80 HTTP 443 HTTPS
TEID	GTP-U テンプレート	テンプレート

PDR ☐☐☐

```

PDR 000
├── PDR ID 0000000000
├── 0000000000000000
├── 00000000
├──   ├── 000
├──   ├── 000000APN 0
├──   └── UE IP 00 / 00 IP
├── FAR ID 000000000000
├── QER ID 000000 QoS 000
└── 00000000

```

UE - EPC

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

UE - EPC

- UE IP address
- PDN type
- QoS parameters GBR

FAR

UE IP address

UE IP address

UE	UE	UE
UE	UE IP address	UE IP address
UE	UE IP address	UE IP address
UE	UE IP address	UE IP address
UE	UE IP address	UE IP address

UE IP address

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

UE - UE IP address

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	性能
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 □□□□□□□□□□□□



□□□□□□□□

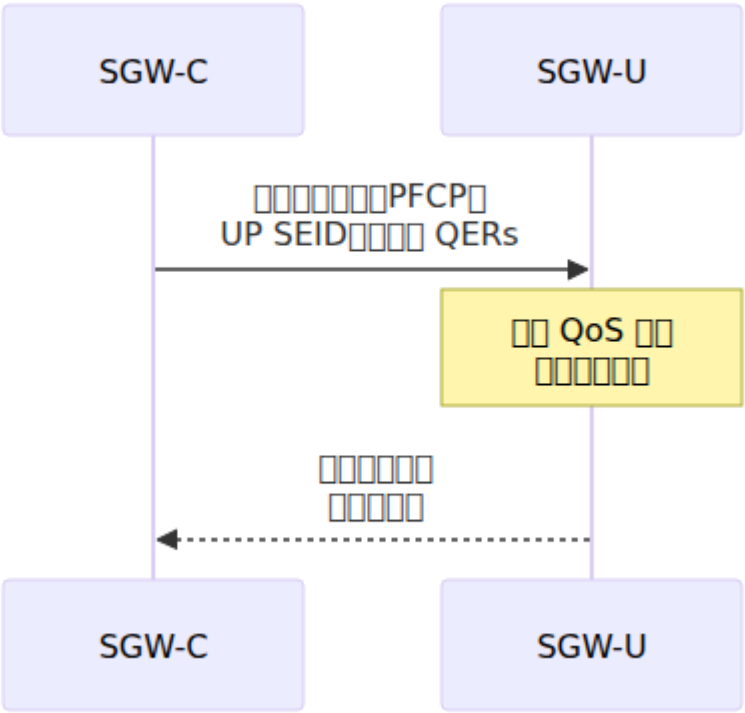
□□□□□□□□□□□□

□□□	□□
□□□□	□ 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	項目名	説明
SGW-U 項目	"項目名"	項目 SGW-U 項目
項目 PDR	"項目名 IE"	項目項目
SEID 項目	"SEID 項目"	項目項目
項目	"項目項目"	項目項目項目 SGW-U

項目項目項目

項目 "項目項目"

項目

```
# 項目項目  
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
- 項目項目項目項目
- 項目 CDR 項目項目
- 項目 SGW-U 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 □□□□□□
-