

OmniSGW 関数

OmniSGW - 機能 (SGW)

OmniSGW 機能

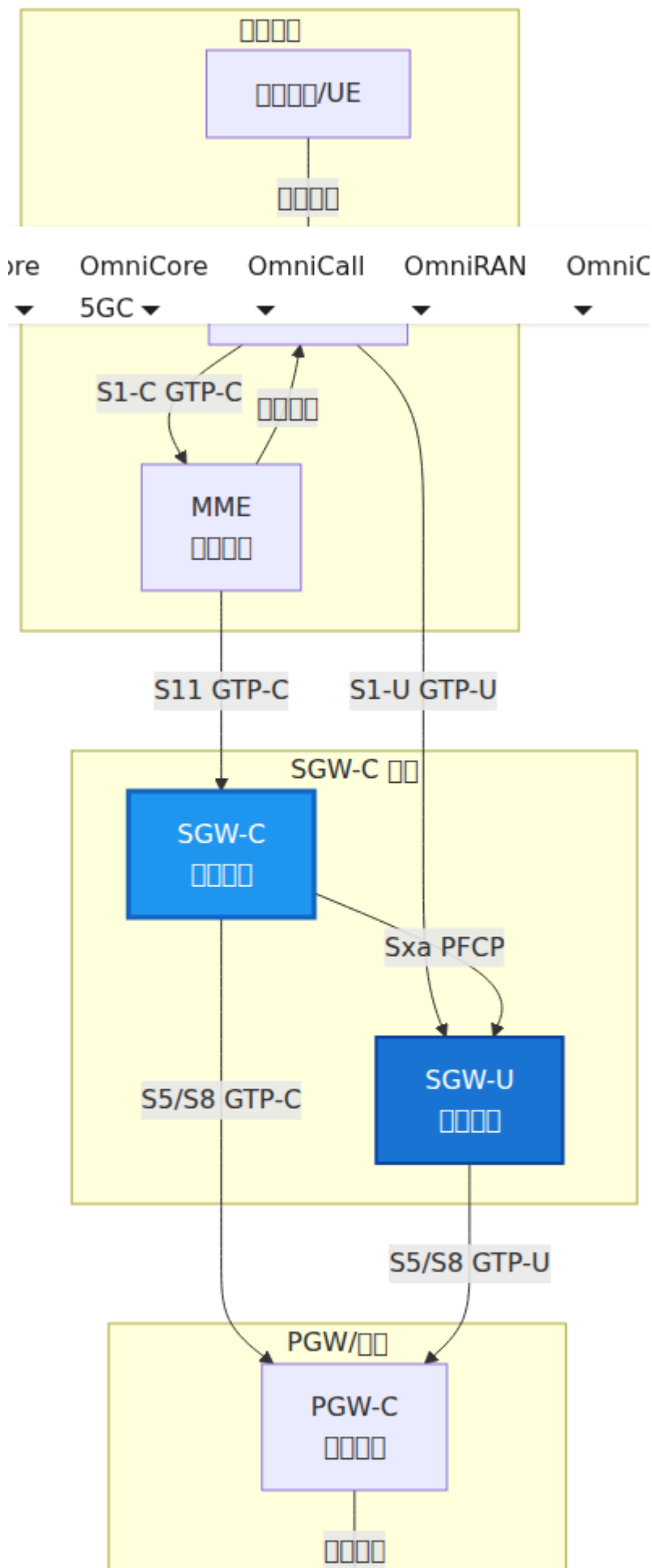
機能

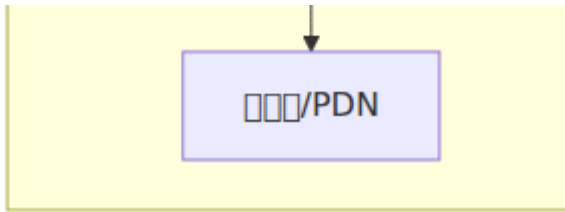
1. 機能
2. 機能
3. 機能
4. 機能
5. 機能
6. 機能
7. Web UI - 機能
8. 機能
9. 機能
10. 機能

機能

OmniSGW は OmniSGW 機能 (SGW) 3GPP LTE ネットワーク (EPC) ネットワーク UE ネットワーク

- 機能 - ネットワーク UE (機能) ネットワーク
- 機能 - ネットワーク eNodeB ネットワーク
- 機能 - ネットワーク QoS ネットワーク
- 機能 - ネットワーク
- 機能 - ネットワーク SGW-U (機能) ネットワーク



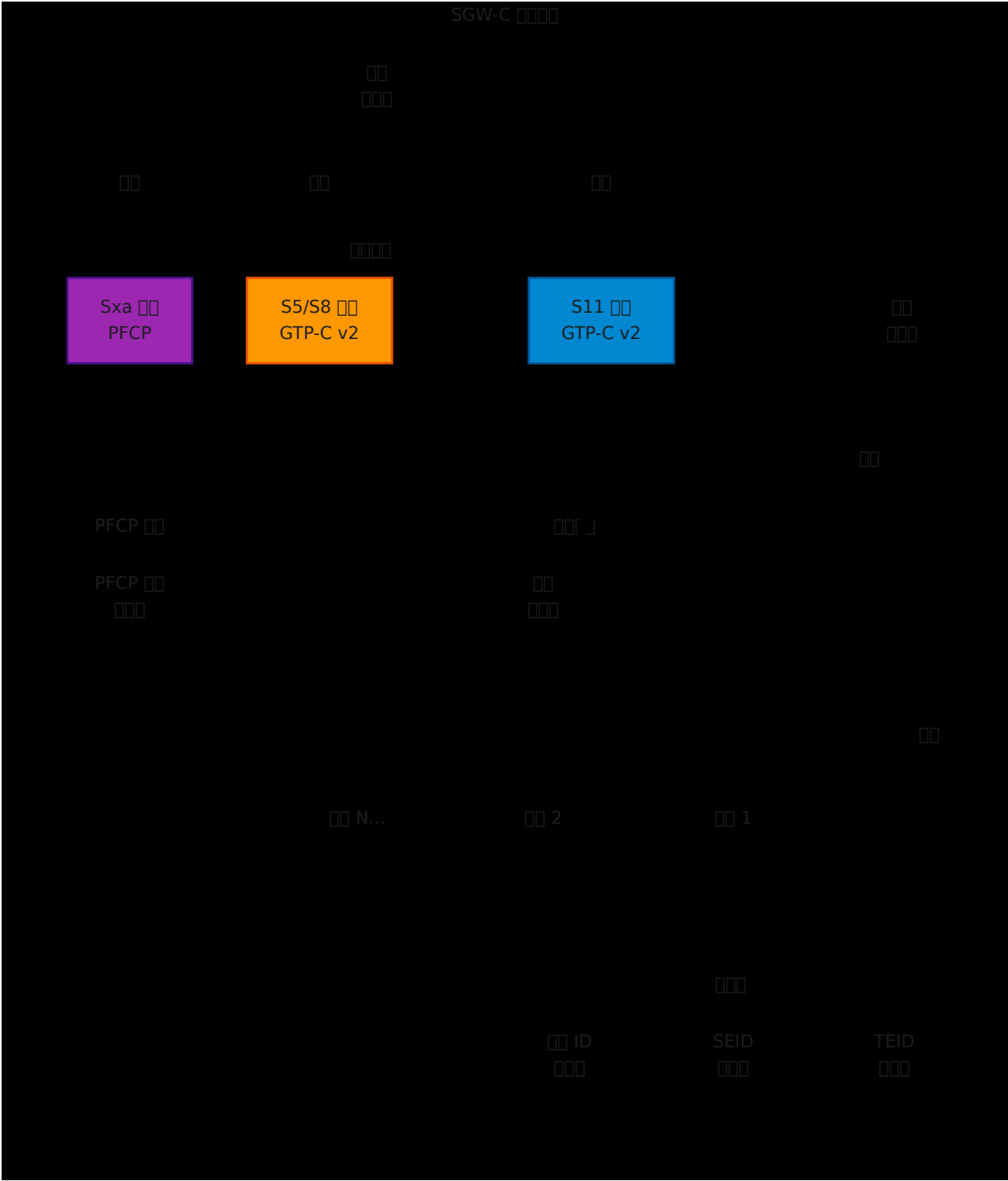


SGW-C

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

□□

□□□□



Overview

SGW-C uses Elixir/OTP for process management

- Processes - lightweight
- Interfaces - S11, S5/S8, Sxa
- UE - GenServer
- TEID/SEID ID
- **PFCP** - SGW-U PFCP

Web UI

Architecture

SGW-C 3GPP

S11 (GTP-C v2)

MME SGW-C

UDP GTP-C 2

- 3GPP TS 23.060
- 3GPP TS 23.061
- 3GPP TS 23.062
- 3GPP TS 23.063
- 3GPP TS 23.064
- 3GPP TS 23.065

3GPP TS S11 3GPP

Sxa 3GPP (PFCP)

3GPP SGW-C 3GPP SGW-U 3GPP

3GPP TS UDP 3GPP PFCP (3GPP)

3GPP

- 3GPP TS 23.060
- 3GPP TS 23.061
- 3GPP TS 23.062
- 3GPP TS 23.063
- 3GPP TS 23.064
- 3GPP TS 23.065
- 3GPP TS 23.066

3GPP TS PFCP/Sxa 3GPP

S5/S8 3GPP (GTP-C v2)

3GPP SGW-C 3GPP PGW-C 3GPP PDN 3GPP

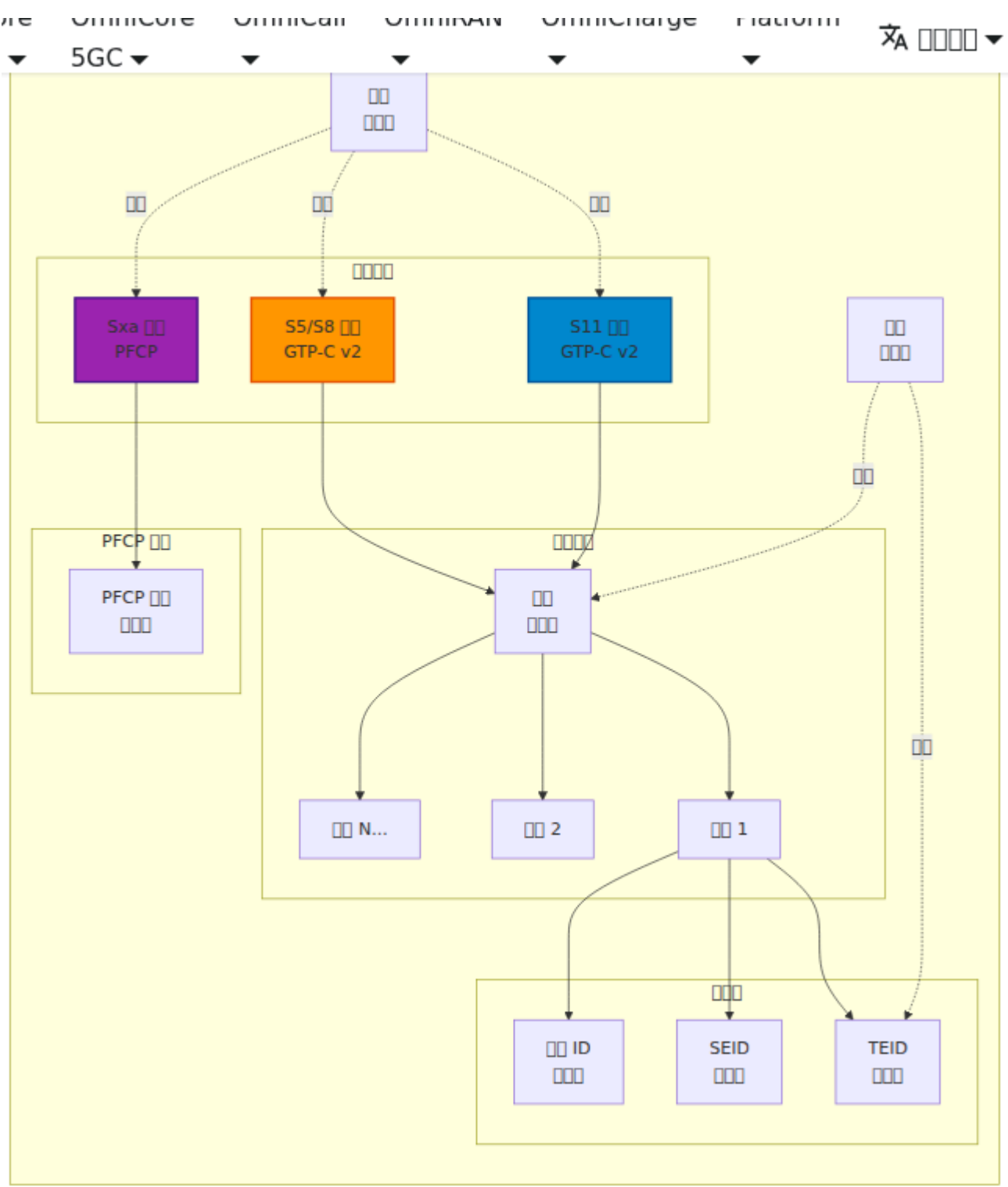
3GPP TS UDP 3GPP GTP-C 3GPP 2

3GPP

- 3GPP TS 23.060
- 3GPP TS 23.061
- 3GPP TS 23.062

- 网络切片/网络
- 网络切片/网络

网络切片 S5/S8 网络切片



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 ID

- **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 ID

QoS ID - QoS ID

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

PFCP

SGW-C と SGW-U の間で行われる

- **PDR** (Policy Data Rule) - ポリシー (何/何)
- **FAR** (Forwarding Action Rule) - 転送ルール
- **QER** (QoS Enforcement Rule) - QoS 強制ルール
- **BAR** (Bearer Authorization Rule) - 転送承認ルール

転送承認ルール **Sxa** を含む

転送承認

SGW-C と UE と eNodeB の間で行われる

- **MME** と - MME と MME (と SGW と)
- **MME** と - MMEs と MMEs SGW と
- MME - MMEs と MMEs
- MMEs - MMEs と UE と

転送

転送

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

転送

転送

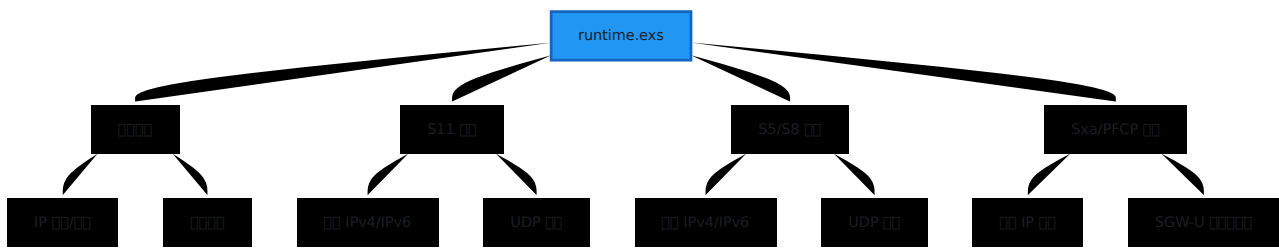
```
[info] [OmniSGW...]
[info] [127.0.0.40:42068]
[info] [127.0.0.10] [S11]
[info] [127.0.0.15] [S5/S8]
[info] [127.0.0.20] [Sxa]
[info] [PCF]
[info] OmniSGW
```

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
-

- 網路服務品質管理
- 網路服務品質 QoS 管理
- 網路服務品質

網路服務品質 (QoS) 管理

- 網路服務品質
- 網路 NOC/網路服務品質
- 網路服務 IP 網路服務品質

網路服務品質

網路服務品質

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

網路服務品質

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

網路服務品質

- 網路 UE 網路服務品質
- 網路/網路服務品質
- 網路服務品質
- 網路服務品質

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 抓取地址 `http://127.0.0.40:42068/metrics`

部署

部署 OmniSGW 到 Kubernetes 集群

0000

OmniSGW 00

└─ OPERATIONS.md (000)

└─ docs/

└─ 00000

└─ configuration.md

000 runtime.exs 00

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

└─ 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
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 - 3GPP

3GPP

- 1. configuration.md - 3GPP
- 2. monitoring.md - 3GPP

3GPP

3GPP

TS	Interface
TS 29.274	GTP-C v2 (S11 & S5/S8)
TS 29.244	PFPCP (Sxa)
TS 32.251	3GPP
TS 32.298	CDR
TS 23.401	EPC

SGW-C CDR (CDR) 說明

SGW-C 說明

OmniSGW 及 Omnitouch 說明

目錄

1. 說明
2. CDR 說明
3. CDR 說明
4. CDR 說明
5. 說明
6. 說明
7. CDR 說明
8. 說明
9. 說明
10. 說明

目錄

SGW-C (SGW-C) 說明 (SGW-C) 說明 CDR 說明

PGW-C CDR 說明 EPC 說明

說明

- CSV 說明 - 說明
- 說明 - 說明
- 說明 - 說明

- 0000 - 0000000000000000
- 00 **3GPP** 00 - 00 3GPP TS 32.251 (PS 000) 0 TS 32.298 (CDR 00)

00

00	00
0000	00000000 CDR
00	00000000
0000	000000000000
0000	0000000000
0000	0000000000

CDR 0000

00000000

<epoch_timestamp>

00:

1726598022

0000000000 Unix 0000000000000000

0000

0000:

- 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 項目

項目名

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

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

CDR 項目

項目名

CDR 項目名一覧

項目名	フィールド名	型	説明
default_bearer_start	<type>_bearer_start	uint32_t	デフォルトベアラ開始時刻
default_bearer_update	<type>_bearer_update	uint32_t	デフォルトベアラ更新時刻
default_bearer_end	<type>_bearer_end	uint32_t	デフォルトベアラ終了時刻

項目名:

- default - デフォルト PDN 項目名
- dedicated - デディケート 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

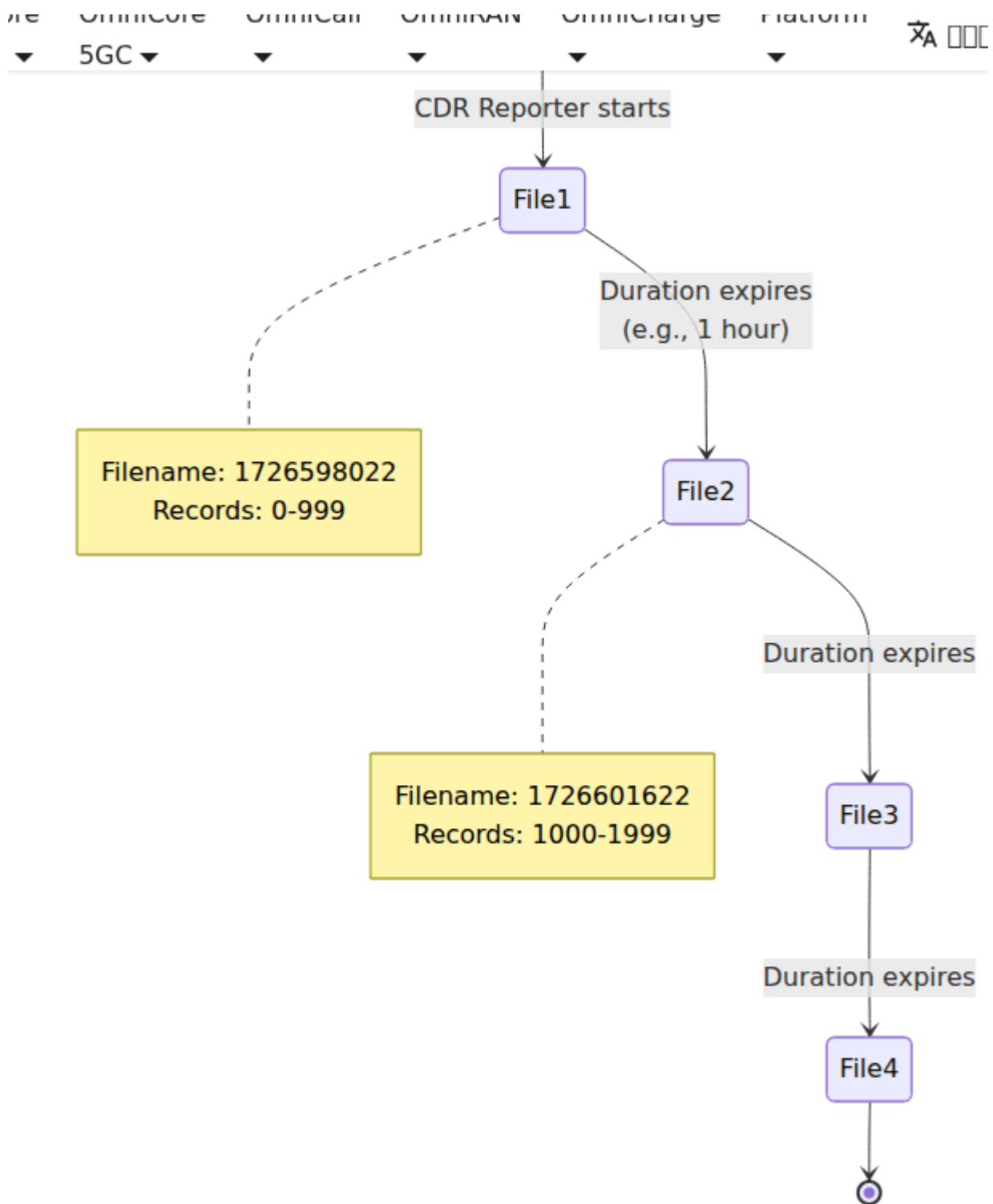
□□□□

□□ **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

필드	유형	설명	기본값	참고
gateway_name	문자열	SGW-C 인스턴스명	-	필수적으로 ID
duration	정수	로그 수집 기간 (초)	-	3600000 (1 시간)
directory	문자열	CDR 저장 경로	-	/var/log/sgw_c/cdrs

예시

예시 1:

- **gateway_name:** "sgw-c-prod-01"
- **duration:** 3,600,000 초 (1 시간)
- **directory:** "/var/log/sgw_c/cdrs"

예시 2:

- **gateway_name:** "sgw-c-dev"
- **duration:** 300,000 초 (5 시간)
- **directory:** "/tmp/sgw_c_cdrs"

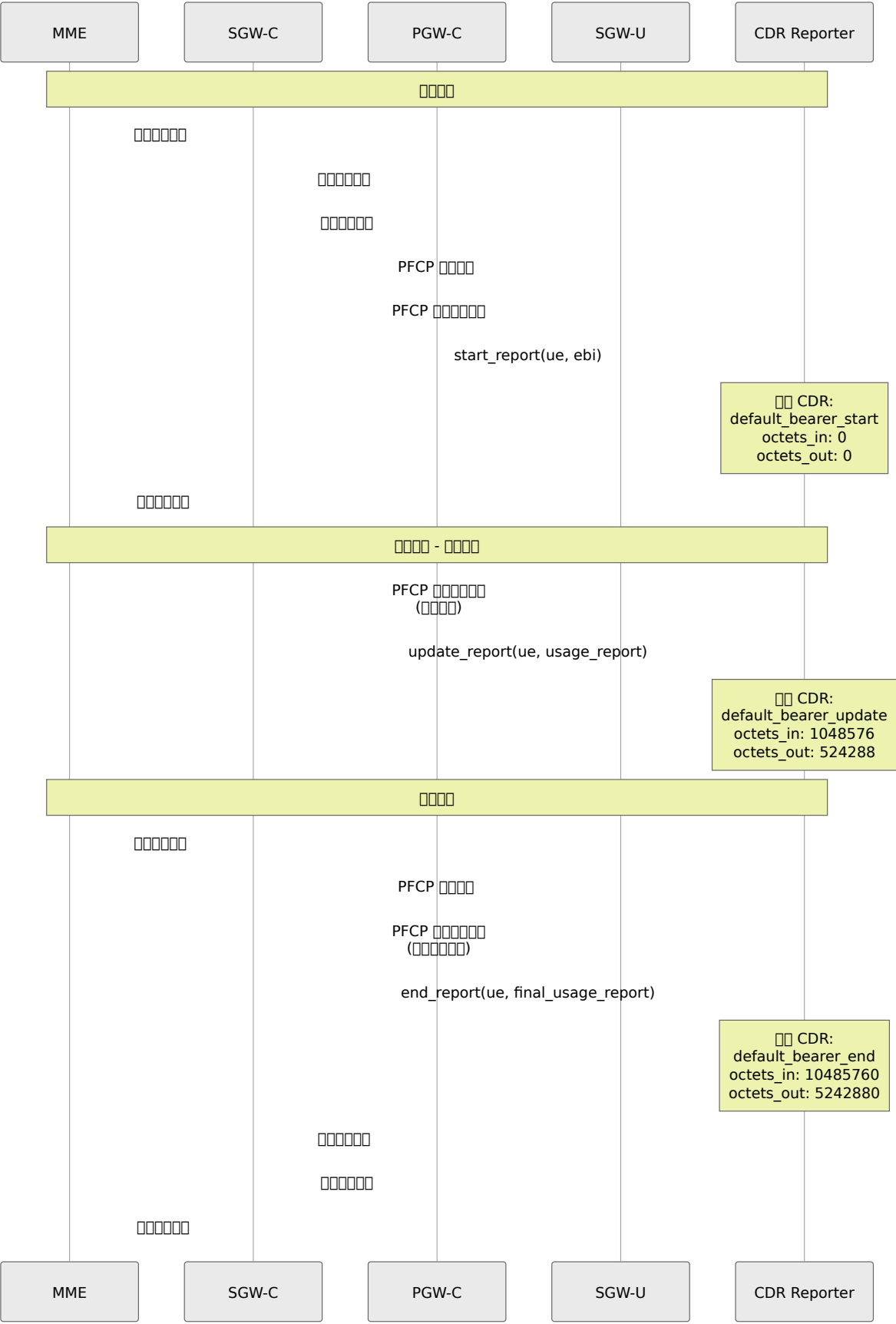
예시 3:

- **gateway_name:** "sgw-c-prod-heavy"
 - **duration:** 1,800,000 초 (30 시간)
 - **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 digits

IMEI: MCCMNC + MSIN

IMEI: 0000000000000000

IMEI:

```
310260123456789
  |  |  |  |  |  |  |  |  |
  MCC MNC  MSIN
  (310)(260) (123456789)
```

IMEI: UE 0000000000000000

3. event (CDR event)

event: event

event: <bearer_type>_bearer_<event>

event:

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

event:

- event EBI (EPS event ID) event LBI (event ID): default
- event EBI event LBI: dedicated

00: 000000EBI 0 LBI 000

4. charging_id (000000)

00: 000 32 000

00: 00000000000000000000

00:

12345

00: 0 PGW-C 00000000000000

00:

- 00 SGW 0 PGW 00000000
 - 00 Diameter Gy/Gz 00000000
 - 000000
-

5. msisdn (00000)

00: 0000E.164 000

00: 000 ISDN 000000000000

00: 0000 + 0000

00:

15551234567

└─┬────────┘

CC National

(1) (5551234567)

00: UE 00000000 MME 0 HSS 00

6. ue_imei (IMEI)

IMEI: 15 digits

IMEI: TAC (8) + SNR (6) + Spare (1)

IMEI: 000000000000000

IMEI:

123456789012345

12345678

901234

5

TAC SNR S

IMEI: UE IMEI MME ID

7. timezone_raw (UE TZ)

IMEI: 00000000/000

IMEI: 00000000 UE IMEI

IMEI: 0000CSV 000000

IMEI: 000000000000000

IMEI:

, (IMEI)

8. plmn (PLMN)

IMEI: 00000000

IMEI: 00000000000000000000

例:

MCC: 505, MNC: 57
↓
"50557"
↓
例: "055570"
↓
例: 0x055570 = 349552

例:

349552 → MCC: 505, MNC: 57

例: MME UE 例

例: 例

9. tac (例)

例: 例 16 例

例: 例 UE 例

例: 0 - 65535

例:

1234

例: UE 例 MME 例

例:

- 例
- 例

- TAI (TAI) 0000

10. eci (E-UTRAN 000000)

eci: 000 28 000

eci: E-UTRAN 0000000000 UE 00000000

eci: eNodeB ID (20 0) + 00 ID (8 0)

eci: 0 - 268,435,455

eci:

5678

eci: 0 MME 000 UE 0000

eci:

- 000000000000
- 0000000000
- 00000000

11. sgw_ip (SGW 0000 IP)

sgw_ip: 0000IPv4 0 IPv6 000

sgw_ip: SGW-C 0 S5/S8 0000 IP 00 (F-TEID)

sgw_ip: 000000 (IPv4) 0 00-000000 (IPv6)

sgw_ip:

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

例: 5S/S8 例

12. ue_ip (UE IP 例)

例: IPv4|IPv6 例

例: UE 例 PDN 例 IP 例

例: <ipv4>|<ipv6>

例:

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

例: PGW-C 例 PDN 例 (PAA)

例:

- 例 IPv4: 例 IPv4 例
- 例 IPv6: 例 IPv6 例
- 例: 例 PDN 例

13. pgw_ip (PGW 例 IP)

例: IPv4 例 IPv6 例

例: PGW-C 例 S5/S8 例 IP 例 (例 F-TEID)

例: (IPv4) 例 -例 (IPv6)

例:

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

例: PGW-C 設定

14. apn (アクセスポイント名)

例: デフォルト 100 設定

例: デフォルト (PDN) 設定

例: DNS 設定

例:

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

例: MME 設定

例:

- デフォルト
 - デフォルト
 - IP 設定
-

15. qci (QoS 設定)

例: 8 設定

例: QoS 設定

例: 1 - 9 (優先度), 128-254 (ビットレート)

例: QCI 設定

QCI	Service Type	Priority	Packet Delay Budget (ms)	Packet Error Rate	Allocation and Retention Priority
1	GBR	2	100 ms	10^{-2}	High
2	GBR	4	150 ms	10^{-3}	High
3	GBR	3	50 ms	10^{-3}	High
4	GBR	5	300 ms	10^{-6}	Medium
5	Non-GBR	1	100 ms	10^{-6}	IMS Voice
6	Non-GBR	6	300 ms	10^{-6}	Video
7	Non-GBR	7	100 ms	10^{-3}	Video
8	Non-GBR	8	300 ms	10^{-6}	Video
9	Non-GBR	9	300 ms	10^{-6}	Default

QoS:

9 → Default

QoS: PGW-C Default QoS

16. octets_in (Usage)

QoS: Default 64 Kbps

QoS: Default → UE Default

QoS: Default

QoS:

1048576 → 1 MB

SGW-U PFCP

:

- update
- end
- start 0

17. octets_out ()

: 64

: UE →

:

:

524288 → 512 KB

SGW-U PFCP

:

- update
- end
- start 0

00

00 1: 0000000000

000:

- 00000
- 5 000: 0000010 MB 0005 MB 000
- 0000

CDR 00:

```
# 00 CDR 00:
# 000000: 10:00:00 (1726570800)
# 000000: 11:00:00 (1726574400)
# 0000: sgw-c-01
# epoch,imsi,event,charging_id,msisdn,ue_imei,timezone_raw,plmn,tac,e
1726570800,310260111111111,default_bearer_start,10001,1555111111,111
1726571100,310260111111111,default_bearer_update,10001,1555111111,11
1726571400,310260111111111,default_bearer_end,10001,1555111111,11111
```

00 2: 0000000000

000:

- 00000000IPv4 + IPv6
- 000000
- 0000

CDR 00:

```
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,10002,1555222222,22222
```

3:

□□□■

1. □□□□□□□□□□QCI 9□
2. □□□□□□□□□□□□QCI 6□
3. □□□□□□□□□□
4. □□□□□□□□
5. □□□□□□□□

CDR :

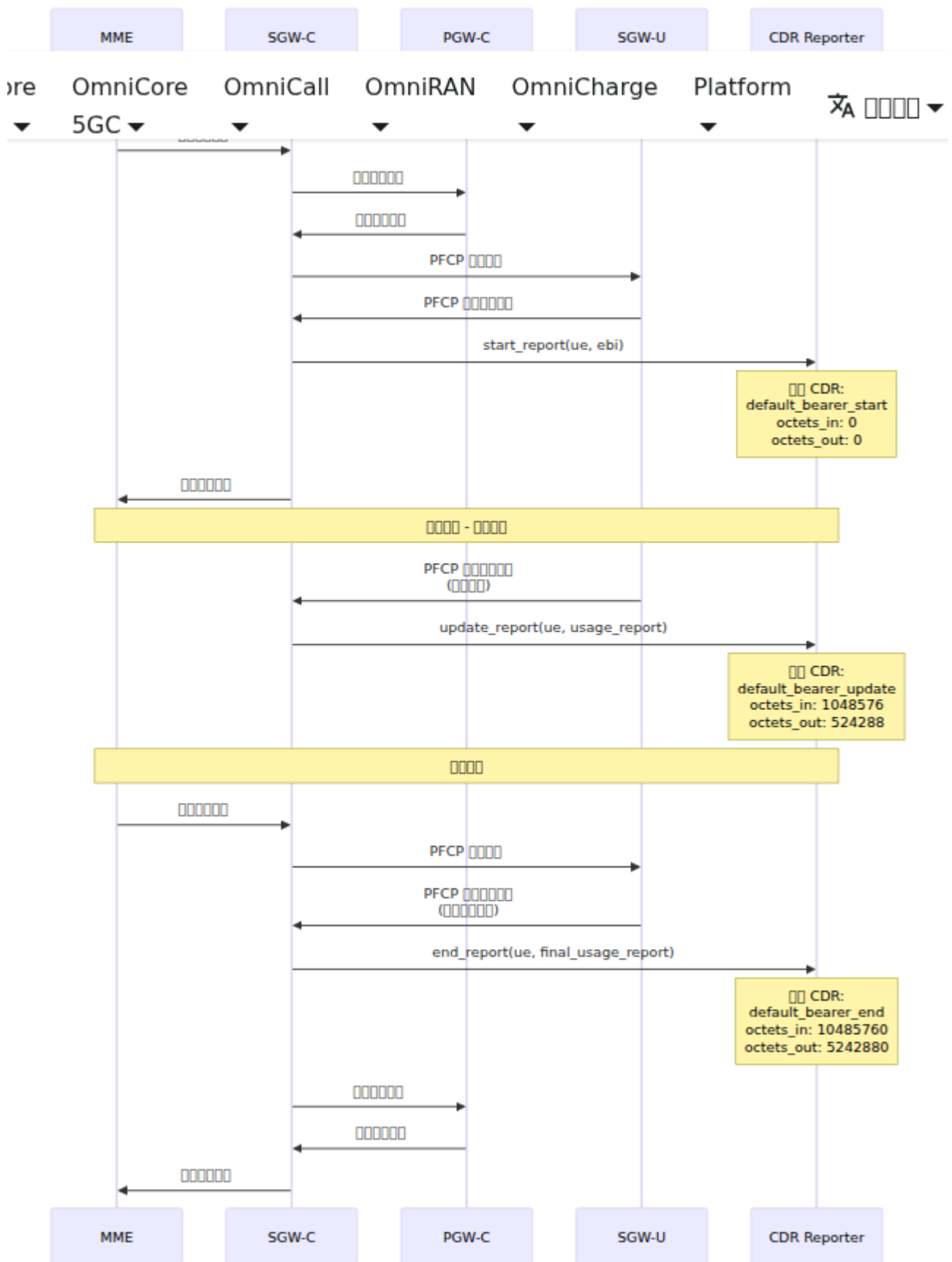
```
1726570800,3102603333333333,default_bearer_start,10003,15553333333,333
1726571100,3102603333333333,dedicated_bearer_start,10004,15553333333,3
1726571400,3102603333333333,default_bearer_update,10003,15553333333,33
1726571400,3102603333333333,dedicated_bearer_update,10004,15553333333,
1726571700,3102603333333333,dedicated_bearer_end,10004,15553333333,333
1726572000,3102603333333333,default_bearer_end,10003,15553333333,33333
```

11

- 10003 (10003) 10 MB 4 MB
- 10004 (10004) 200 MB 2 MB
- QCI (9 6) QoS

□□

CDR □□□□



CDR 配置

1. 配置脚本:

```
# CDR 脚本 (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:

```
# 配置脚本
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 配置

- TS 32.251 - 配置 (PS) 配置
- TS 29.274 - 3GPP 配置 (EPS); 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

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 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-

```
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  
    max_retries: 3  
  }
```

SGW-U 配置

```
# 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, keys: :atoms)

# □□□□□ SGW-U □□□
sgwu_peers_env = System.get_env("SGWU_PEERS", "[]")
{:ok, sgwu_peers} = Jason.decode(sgwu_peers_env, keys: :atoms)

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"
}
```

Network

IPs

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

Config

Key	Value	Default
HOSTNAME	CDR	
PGW_PEERS	PGW JSON	[]
SGWU_PEERS	SGW-U JSON	[]
CDR_DIR	CDR	/var/log/sgw_c/cdrs

□□□□

```
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
```

□□

□□□□□□□□

□□□□□□□□

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

# □□□□□
# [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
```

確認

```
# 確認メトリクス
curl http://10.0.0.40:42068/metrics | head -20

# S11 ネットワーク
netstat -an | grep 2123

# S11 ログを確認
tail -f /var/log/sgw_c/sgw_c.log | grep "S11"
```

確認

"確認"

確認

確認

```
# 確認
lsof -i :2123

# S11 サービスを再起動
killall sgw_c
# 
config :sgw_c, s11: %{local_port: 2124}
```

"確認" **PGW**

確認 S5/S8 サービス PGW-C

確認

```
# [] PGW IP
ping 10.0.0.20

# []
iptables -L | grep 2123

# []
nc -u -v 10.0.0.20 2123
```

"[] **SGW-U**"

[] Sxa []

[]

```
# [] SGW-U []
ping 10.0.0.30

# [] PFCP []
netstat -an | grep 8805

# [] PFCP []
iptables -L | grep 8805
```


OmniSGW

Prometheus Grafana

OmniSGW & Omnitouch

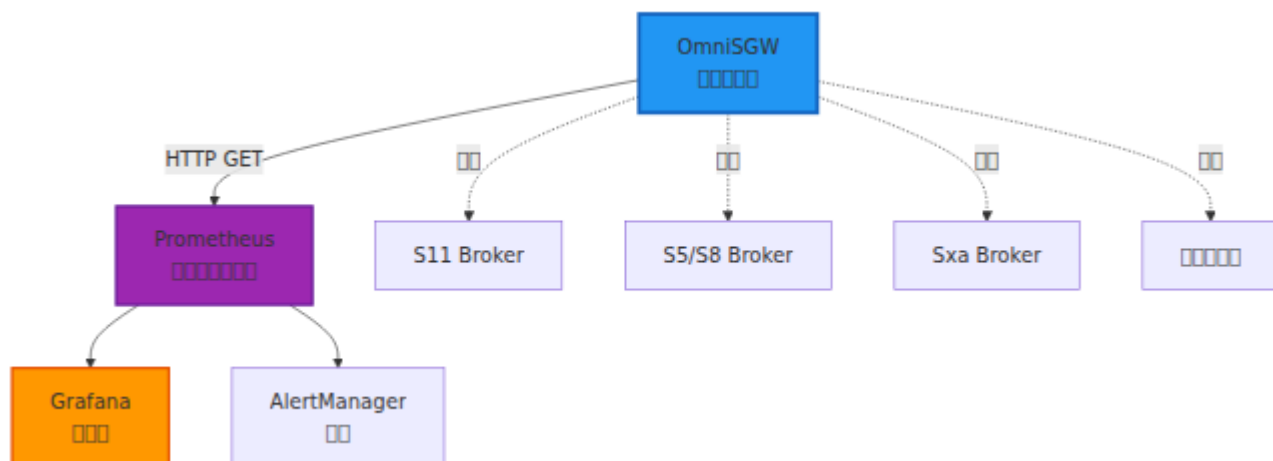
1. 1

- 1.
- 2.
- 3.
4. Prometheus
5. Grafana
- 6.
- 7.

2. 2

OmniSGW Prometheus

3. 3



部署

安装

安装 HTTP 代理

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

# 安装
curl http://127.0.0.40:42068/metrics > metrics.txt

# 安装
watch -n 5 'curl -s http://127.0.0.40:42068/metrics | head -30'
```

安装 Prometheus 代理

配置

配置 Prometheus 代理

```
# HELP teid_registry_count TEID 数量
# TYPE teid_registry_count gauge
teid_registry_count 1234

# HELP s11_inbound_messages_total S11 消息总数
# TYPE s11_inbound_messages_total counter
s11_inbound_messages_total{message_type="create_session_request"}
5432
s11_inbound_messages_total{message_type="delete_session_request"}
5100
s11_inbound_messages_total{message_type="modify_bearer_request"}
12000
```

□□□□

□□□□□□

□□□□

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

└─ 00: Counter

└─ 00: message_type

└─ 0:

| └─ session_establishment_request

| └─ session_modification_request

| └─ session_deletion_request

| └─ session_report_request

| └─ association_setup_request

| └─ heartbeat_request

└─ 00:

sxa_inbound_messages_total{message_type="session_report_request"}
67000

00000

000000

s11_inbound_duration_seconds

└─ 00: Histogram (00)

└─ 00: S11 0000000

└─ 000: _count, _sum, _bucket

└─ 00: s11_inbound_duration_seconds_bucket{le="0.1"} 5000

s5s8_inbound_duration_seconds

└─ 00: Histogram

└─ 00: S5/S8 0000000

sxa_inbound_duration_seconds

└─ 00: Histogram

└─ 00: Sxa 0000000

PFCP 000

```

pfcg_association_status
├── type: Gauge
├── unit: 1 (seconds) 0 (seconds)
├── labels: peer_ip, node_id
└── value: pfcg_association_status{peer_ip="10.0.0.30"} 1

pfcg_heartbeat_latency_ms
├── type: Gauge
├── unit: milliseconds
├── labels: peer_ip
└── value: pfcg_heartbeat_latency_ms{peer_ip="10.0.0.30"} 15

```

0000

00000

```

s11_inbound_errors_total
├── type: Counter
├── labels: error_type
├── values:
│   ├── parse_error
│   ├── validation_error
│   ├── timeout
│   └── other
└── value: s11_inbound_errors_total{error_type="timeout"} 12

s5s8_inbound_errors_total
├── type: Counter
├── labels: S5/S8 type

```

```

sxa_inbound_errors_total
├── type: Counter
├── labels: Sxa type

```

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 (□□)
- 00└─ S5/S8 □□ p95 (□□)
- └─ Sxa □□ p95 (□□)

□ 3:

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

□□□□□□□□

□□□

□ 1:

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

□ 2:

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

□ 3:

- └─ □□□□□□□ (□□)
- └─ □□□□□□ (□□)
- └─ □□□□□□ (□□)
- └─ □□□□□□ (□□)

📊📊📊📊📊

📊

📊 1:

- └─ 📊📊📊 (Gauge + 📊)
- └─ 📊📊📊 (Gauge + 📊)
- └─ PFCP 📊📊 (📊)
- └─ 📊 APN 📊📊 (📊)

📊 2:

- └─ 📊📊📊📊 (📊)
- └─ 📊📊📊📊 (📊)
- └─ 📊📊📊📊 (📊)
- └─ 📊📊📊📊📊 (📊)

📊📊📊📊📊

📊📊📊

teid_registry_count

📊📊📊📊

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

S11 📊📊**95th** 📊📊📊

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

📊📊📊

```
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:
    # 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: {{ $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: {{ $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
000: 00
0000: 85,000 / 100,000 (85%)
00: 2025-12-10 15:30:00 UTC
00: 0000000
```

00000000

00: [00] S11 000000

SGW-C S11 000 2 0000000000
000000

- MME 000000
- 0000 SGW-C
- S11 00000000

0000000000 S11 00

0000

000000

00: 00000000 404

00:

```
# 000000000000
curl -v http://127.0.0.40:42068/metrics

# 0000000000000000
tail -f /var/log/sgw_c/sgw_c.log | grep -i metric

# 0000
cat config/runtime.exs | grep metrics
```

0000:

1. 00 SGW-C 00
2. 0000 IP/000000000000
3. 0000000000
4. 0000000000000000

環境構築

OS: S11 (Ubuntu S5/S8) | Sxa (x86_64)

環境:

- Python 3.8.10
- MySQL 8.0.28
- Redis 6.0.16

インストール:

- Python 3.8.10
- MySQL 8.0.28
- Redis 6.0.16

確認

OS: Ubuntu S5/S8

環境:

```
# Python 3.8.10
ps aux | grep sgw_c | grep -v grep | awk '{print $6}'

# MySQL 8.0.28
watch -n 5 'ps aux | grep sgw_c'
```

確認:

- Python 3.8.10
 - MySQL 8.0.28
 - Redis 6.0.16
 - 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

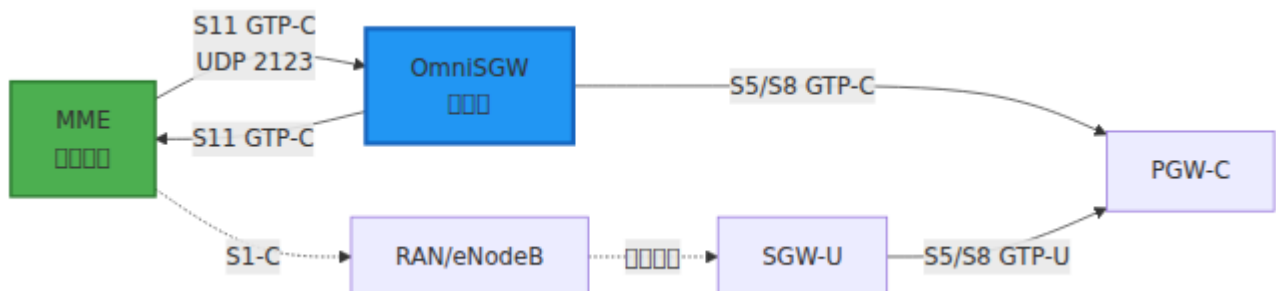
OmniSGW 与 Omnitouch 接口

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 - 接口 OmniSGW 接口 MME 接口
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
- 送信元 - 送信先 UE 識別
- ネットワーク - MME (Mobility Management Entity) 識別
- 送信元 - 送信先 MME 識別
- ネットワーク - MME 識別
- ネットワーク - MME 識別

仕様

GTP-C プロトコル 2

- プロトコル GTP-C v2 (3GPP TS 29.274)
- 送信元 UDP
- 送信元 2123ポート
- 送信元 IP
- 送信元ポート

TEID (Tunnel Endpoint Identifier)

送信元 TEID (Tunnel Endpoint Identifier)

- **TEID** - 送信元 OmniSGW (Serving Gateway) MME 識別
- **TEID** - 送信元 MME 識別

送信元

MME → SGW: 送信元 OmniSGW MME TEID

SGW → MME: 送信元 MME MME TEID

結論

送信元 S11 プロトコル GTP-C v2 プロトコル

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
- 000└─ 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	无线电接入技术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

```

[UE] [UE] MME [SGW] [PGW]

```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □

□□□□□□□□/□□

□□□ MME → OmniSGW

[illegible]

111

- 10000000000000000
- 1000000000000000
- 1 SGW-U 1000000000000000
- UE 1000000000000000

□□□□□□□□/□□

OmniSGW → MME → OmniSGW

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □

□□□□□□ (S11)

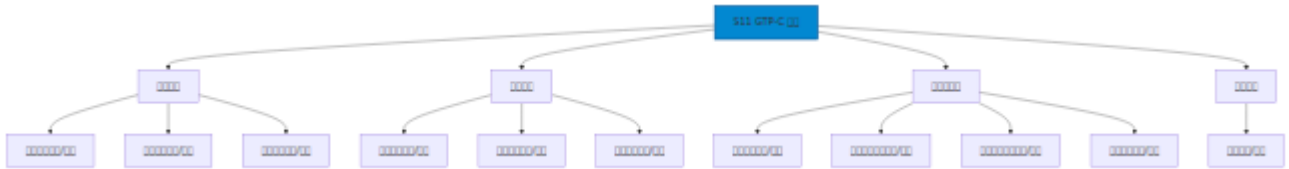
PGW-C → OmniSGW → MME

□□□ □□ MME □□□□□□□□□□□□□□□□ UE

□□□□□□□

IE		
EBI		EPS ID
IMSI		

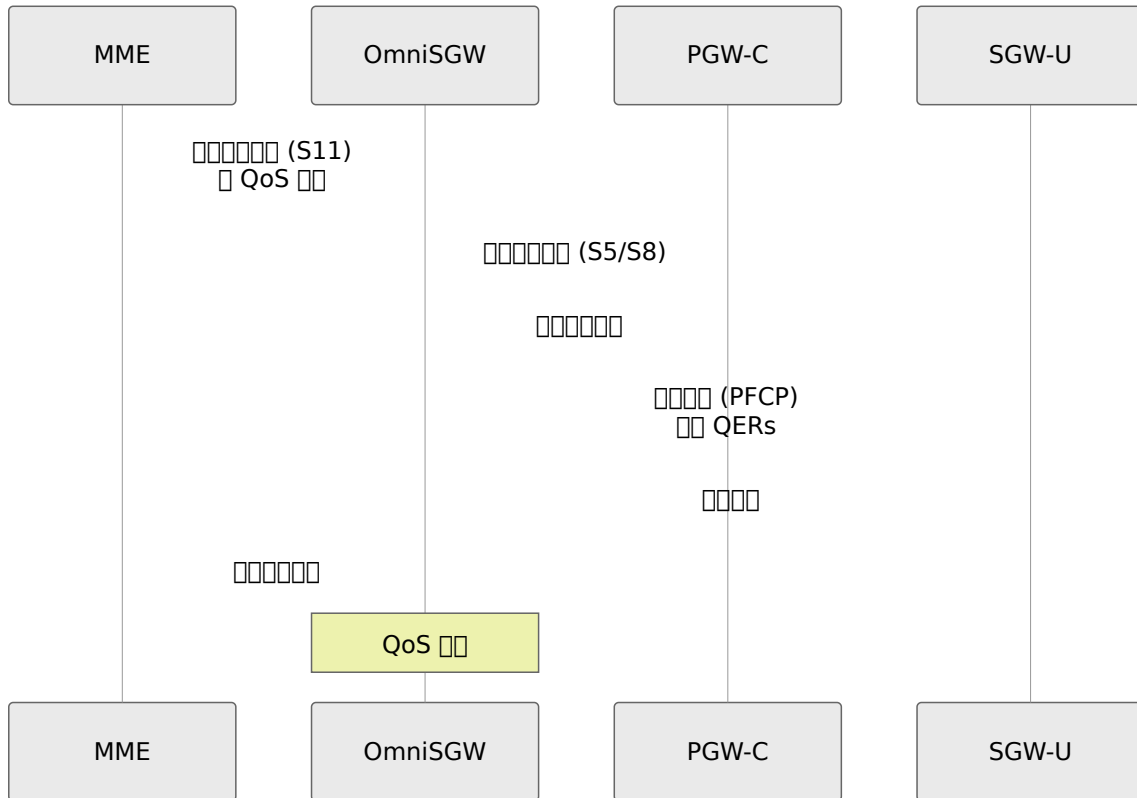
UE



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

QoS

QoS



TAU

SGW TAU

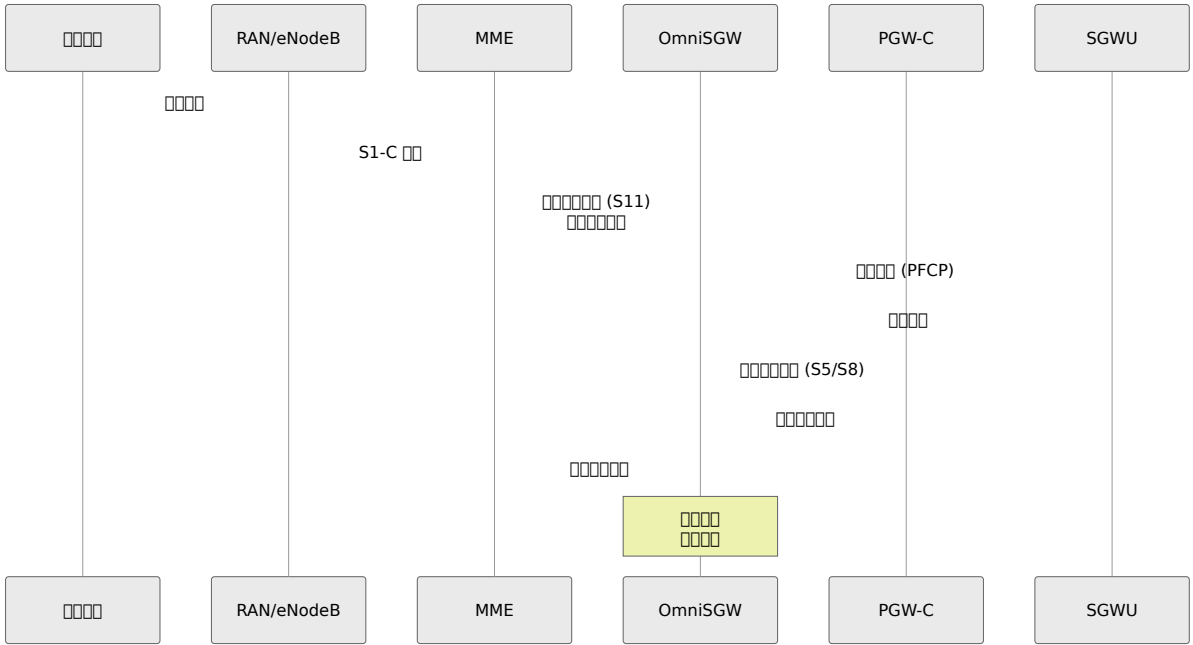
- MME UE
- ULI/TAI SGW
- SGW UE
- TAU

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"
```

1. 確認

2. 確認

確認 確認

確認

- 1. Web UI → UE 確認
- 2. `s11_inbound_errors_total`
- 3. 確認

確認

確認	確認	確認
16	確認	確認 SGW-U 確認PCFP 確認
25	IE 確認	確認
49	PGW 確認	確認 S5/S8 確認 PGW-C 確認
65	確認 APN	確認 APN 確認

3. 確認

確認 "確認 TEID"

確認

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

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

準備

- 環境構築が完了していること
- 仮想 TEID 番号を準備
- 仮想 MME 番号と仮想 TEID

確認

仮想環境を確認

確認

1. 仮想環境の起動/停止
2. 仮想環境 "TAU" の確認
3. 仮想環境 PFCEP の確認

確認

- SGW-U 仮想環境の確認
- 仮想環境の確認
- 仮想環境の確認

確認

仮想 S11 確認

確認

```
# 仮想環境の確認
curl -s http://127.0.0.40:42068/metrics | grep
s11_inbound_duration_seconds

# 仮想環境
curl -s http://127.0.0.40:42068/metrics | grep active_ue_sessions

# 仮想環境
curl -s http://127.0.0.40:42068/metrics | grep active_bearers
```


目標

1. 環境構築
2. 5G Core ネットワーク構築
3. 5G Core ネットワークの動作確認
4. 5G Core ネットワークの最適化

5G Core ネットワークの構築と動作確認に Prometheus を活用する

前提

環境

- 5G Core ネットワークの構築
- 5G Core ネットワークの動作確認
- 5G Core ネットワークの最適化

手順

- 5G Core ネットワークの構築
- 5G Core ネットワークの動作確認
- 5G Core ネットワークの最適化
- 5G Core ネットワークの動作確認

結果

- 5G Core ネットワークの構築
 - 5G Core ネットワークの動作確認
 - 5G Core ネットワークの最適化
-

□□□□□□

□□	□□	□□	□□□
□□□□□□/□□	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 网络

PGW-C 与 GTP-C 网络

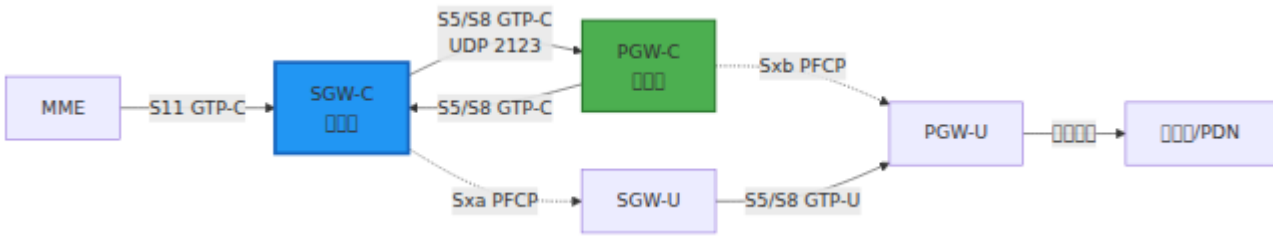
OmniSGW 与 Omnitouch 网络

网络

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

网络

S5/S8 网络 与 GTP-C v2 网络 - 网络 OmniSGW 网络 PGW-C 网络 PDN 网络



□□□□

- **GTP-C v2** 消息 - 消息格式
- 消息 **TEID** 消息格式 - 消息格式
- **PDN** 消息格式 - 消息格式 PDN 消息
- 消息格式 - 消息格式
- 消息 **ID** 消息 - 消息格式
- **IP** 消息格式 - 消息 PGW 消息 UE 消息 IP

GTP-C 2

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

TEID

PDN TEID

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

□□□□

SGW-C → PGW-C□□□□□□ PGW-C □ TEID

PGW-C → SGW-C□□□□□□ SGW-C □ TEID

□□ ID

ID 0000000000000000

- 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
```

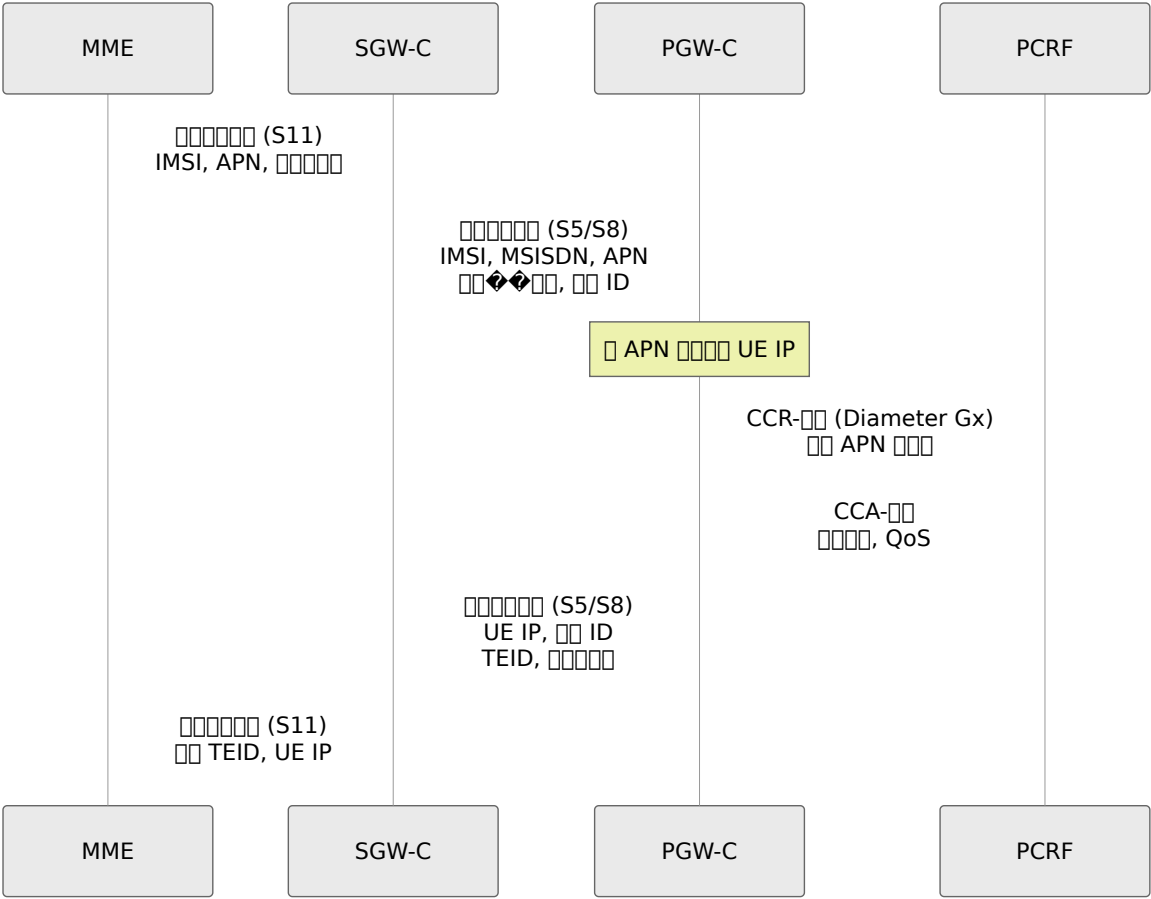
□□□

```
# □□□ 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	送信元	内容
IMSI	MME	国際移動番号
MSISDN	MME	電話番号
MEI	MME	機器識別番号
QoS 値	MME	QoS 値 (QCI, ARP)
APN	MME	アクセスポイント名 (internet, ims, mms)
PLMN 値	MME	PLMN 値 (MCC/MNC)
RAT 値	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

IP アドレス (PGW-C → SGW-C)

IP アドレス

IE 項目	項目	項目
項目	PGW	項目/項目
項目	PGW	項目項目 TEID
PDN 項目	PGW	項目 UE IP 項目
APN 項目	PGW	項目 APN 項目
項目 ID	PGW	PGW 項目項目 ID
TEID	PGW	項目 S5/S8 項目

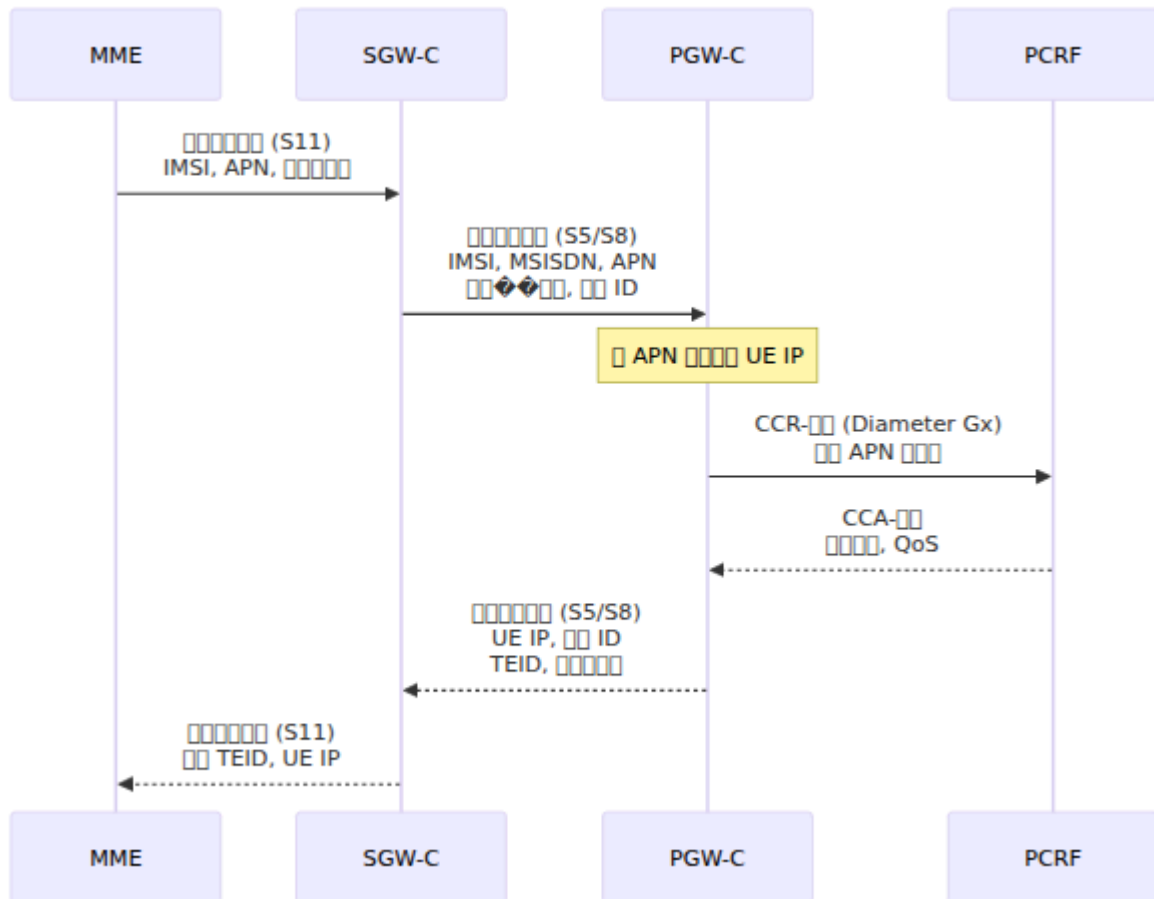
項目項目

項目項目	項目項目項目	項目
0	項目項目	項目項目
16	項目項目	項目 MME項目項目
25	IE 項目項目	項目項目
49	項目項目	PGW-C 項目項目
64	項目項目	項目項目
65	項目項目項目	PGW 項目項目
72	項目/項目項目 IE	項目項目

□□□□

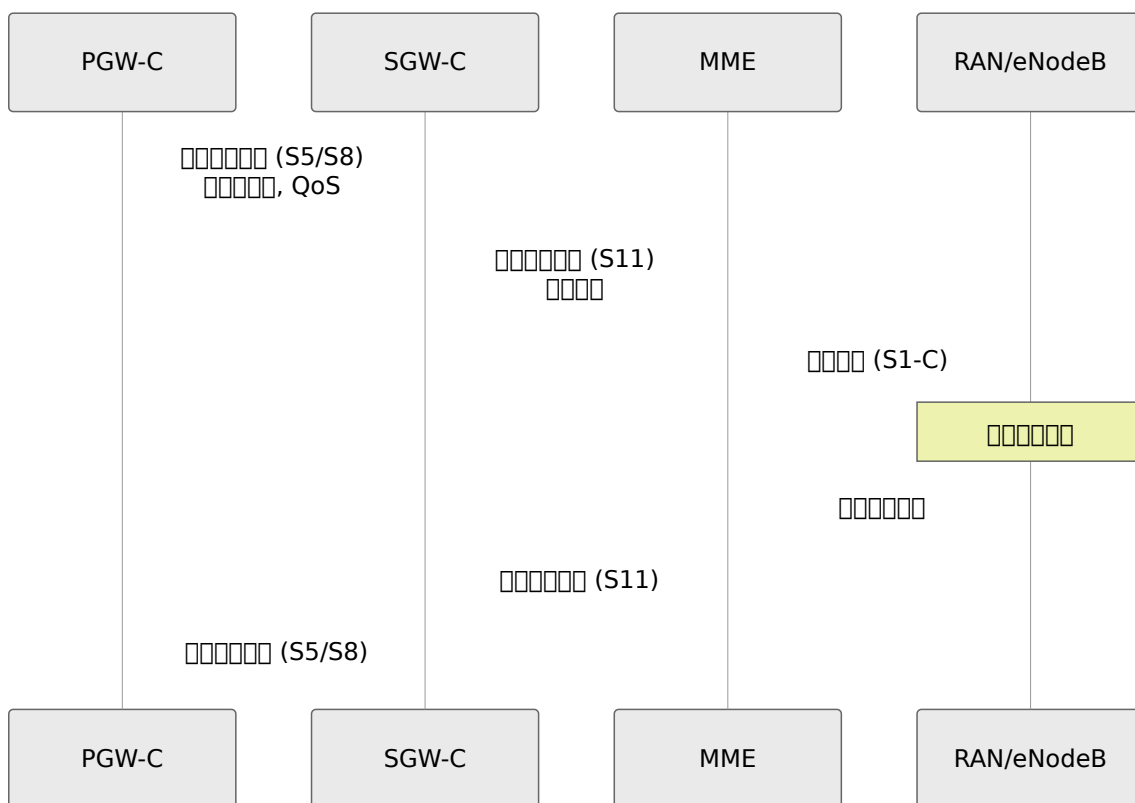
□□ QoS □□

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



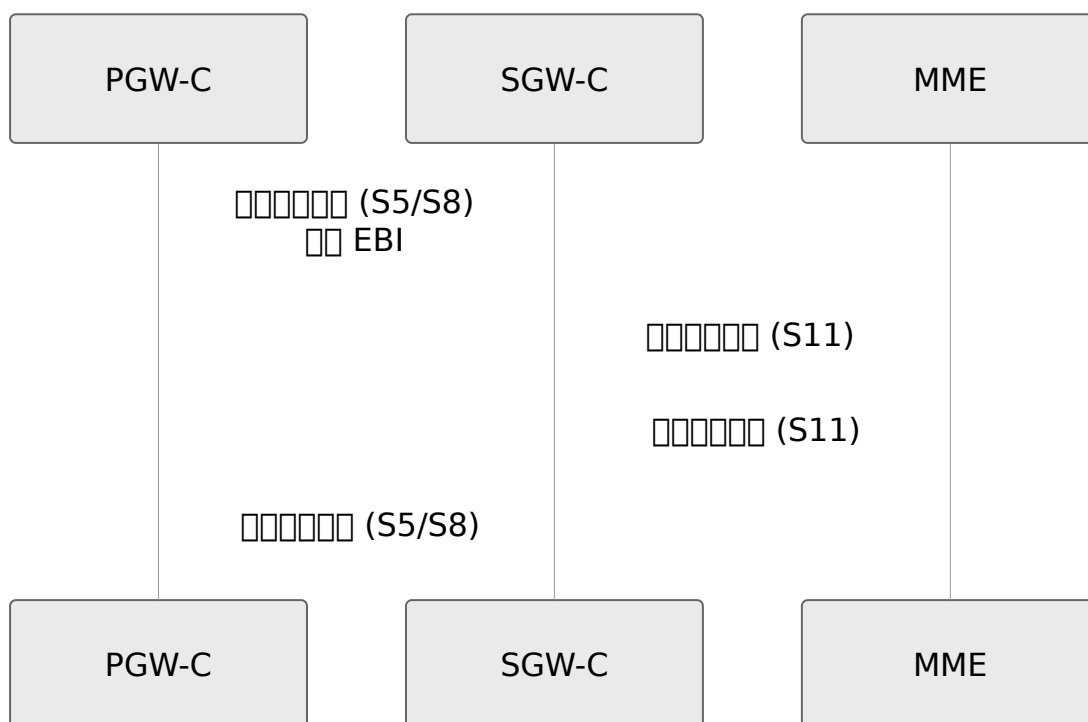
□□□□□□□□□□

PGW-C □□□□ S5/S8 □□□□□□□□□□



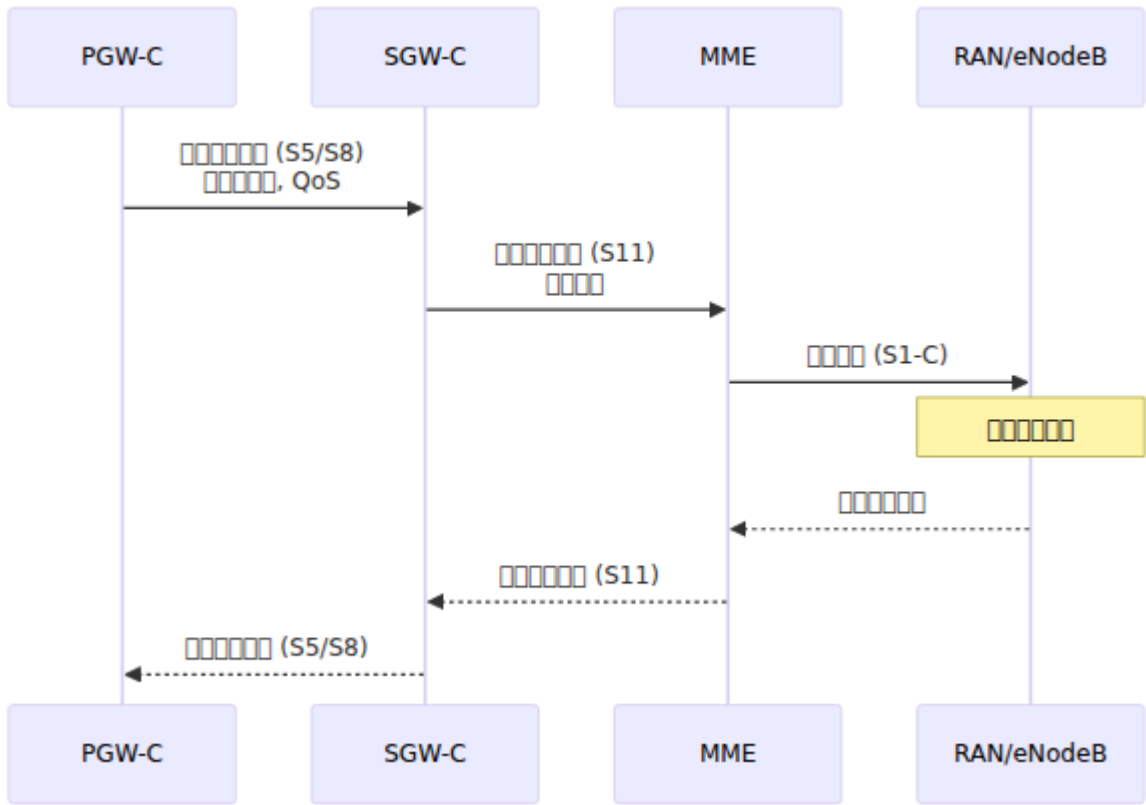
Network components and interfaces

Network components and interfaces



□□□□

□□ **PDN** □□□□

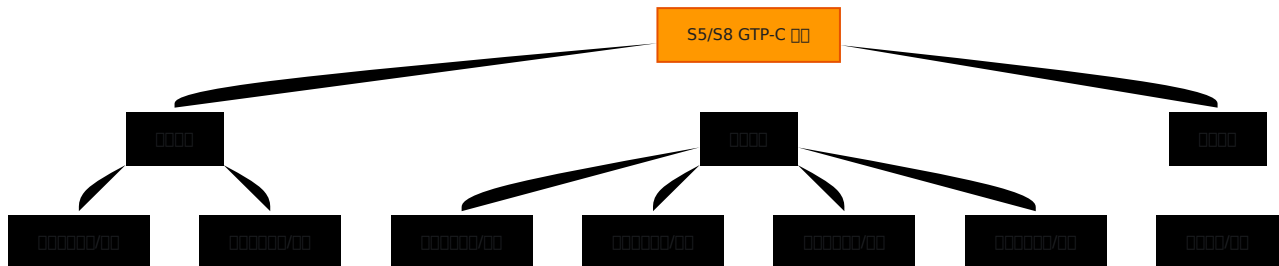


□□□□□

[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

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

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

# 5s8 inbound errors 总数
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
└─ 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 "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
```

创建会话

序号	指标	说明
16	创建	PGW 创建 IP 地址
25	失败	PGW 失败
49	失败原因	PGW 失败原因 APN 失败
72	失败原因 IE	MME 失败原因 IE

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. □□□□
-

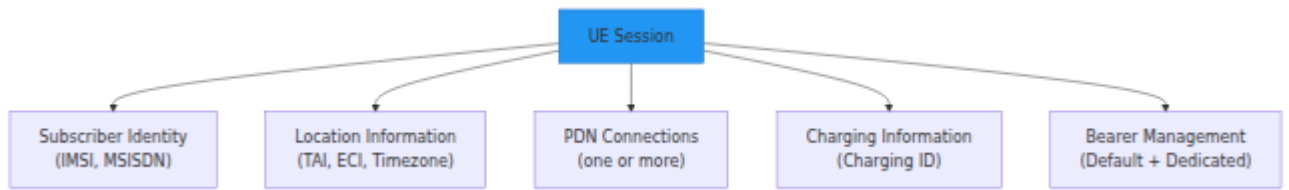


UE □□□□□□□□□□□□□□□□ SGW-C □□□□□□□□□□□□□□□□

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

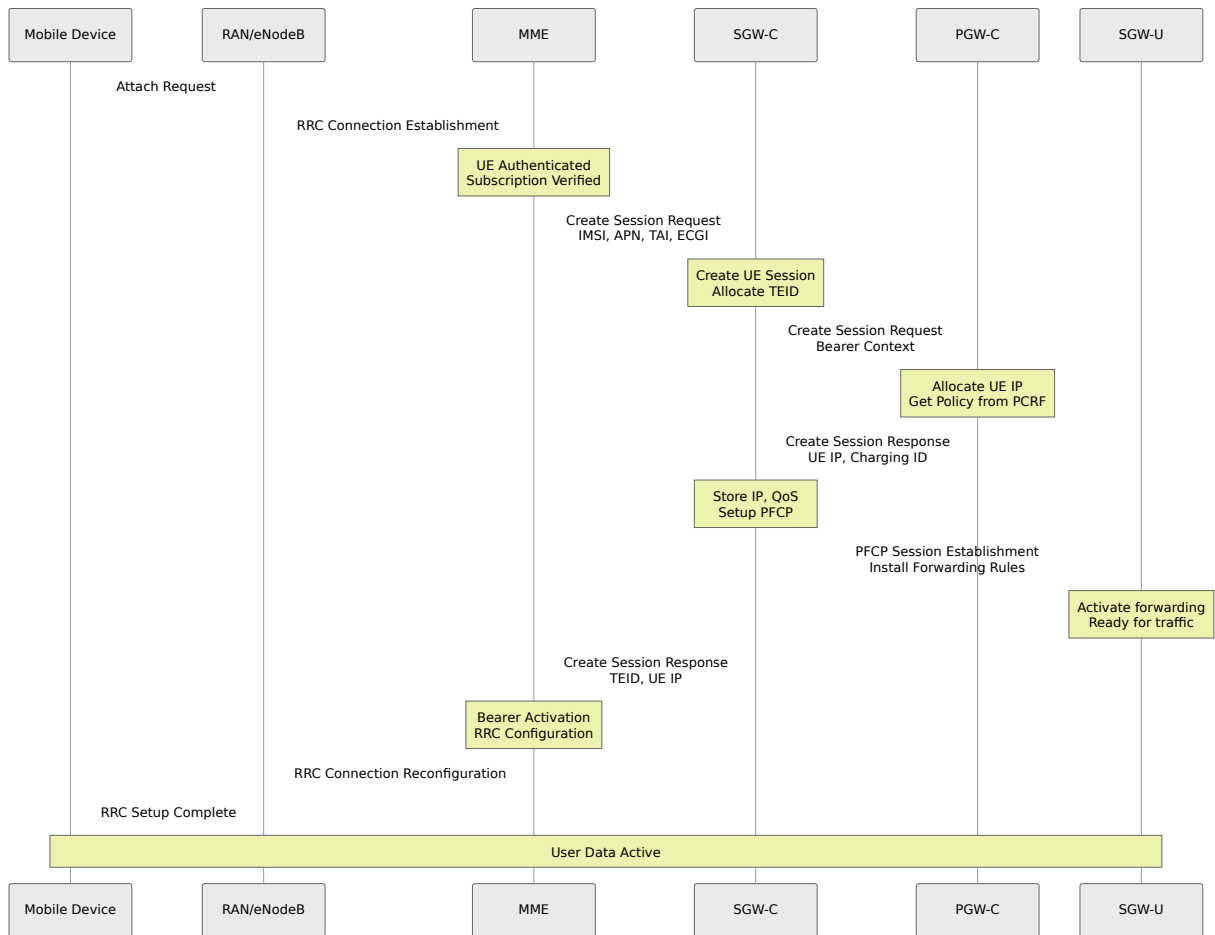
□□□□□□□□□□ IMSI□□□□□□□□□□□□□□□□ PDN □□□

□□□□

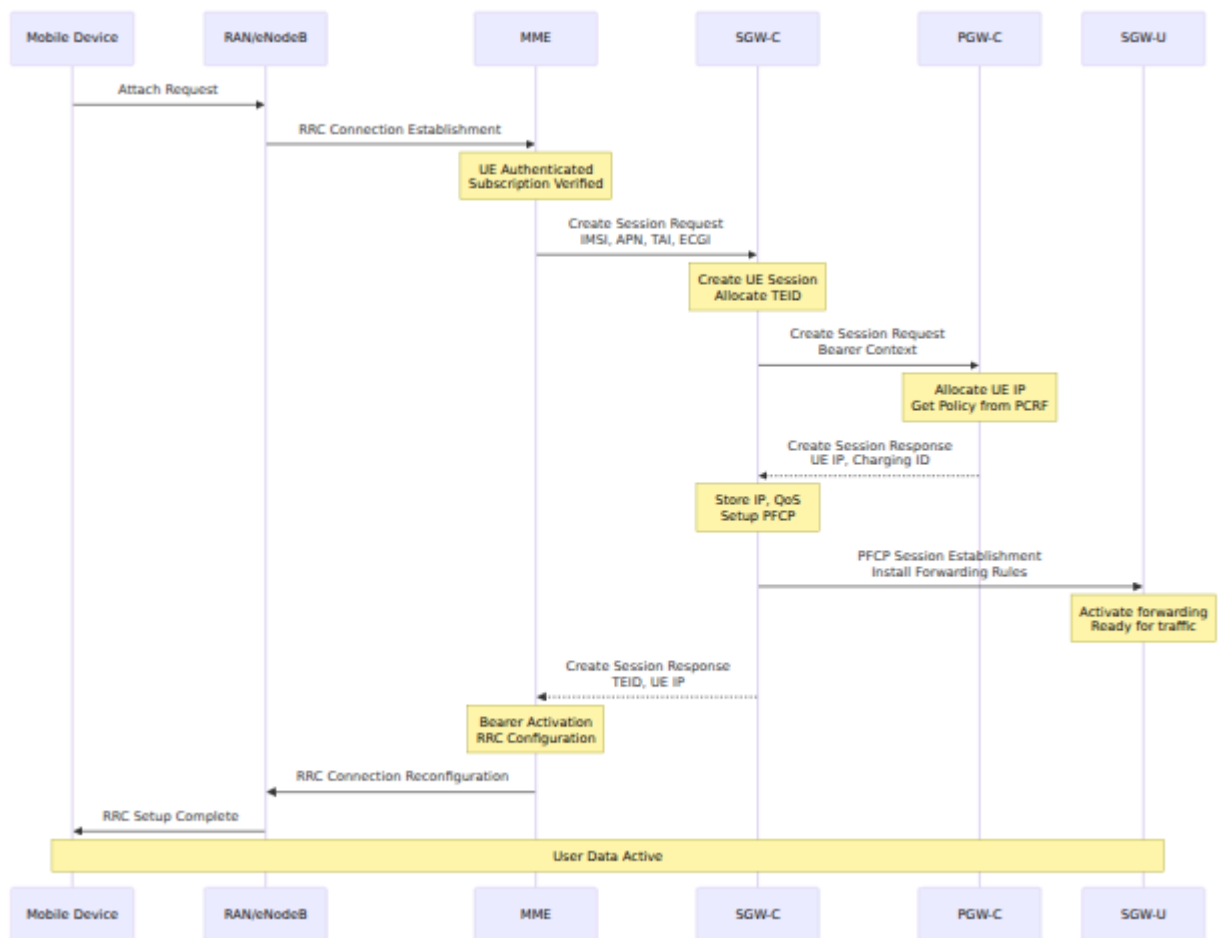


□□□□□□

□□□□□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

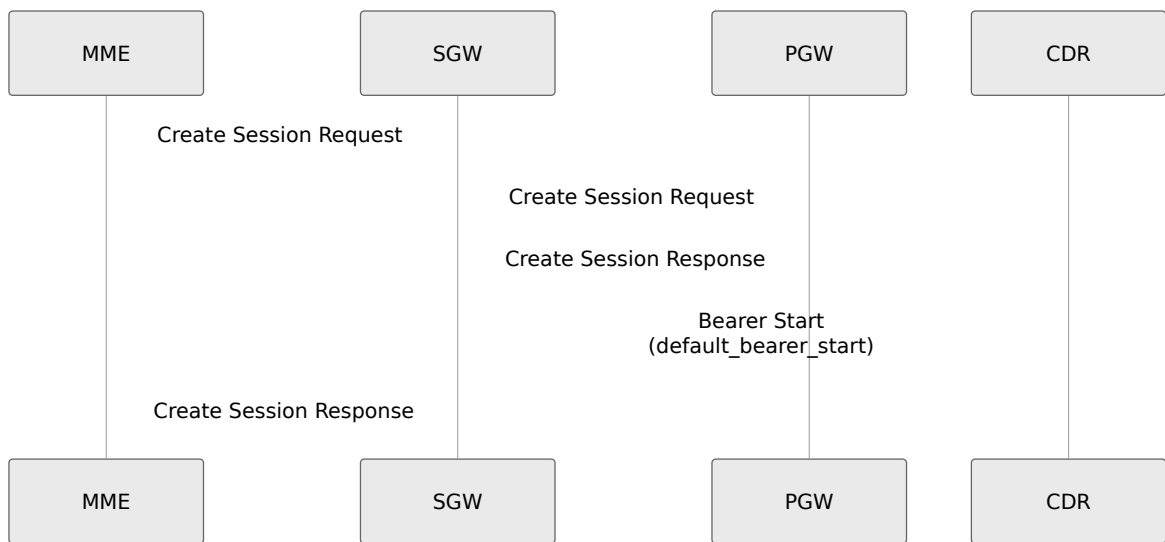
PDN Context

PDN Context

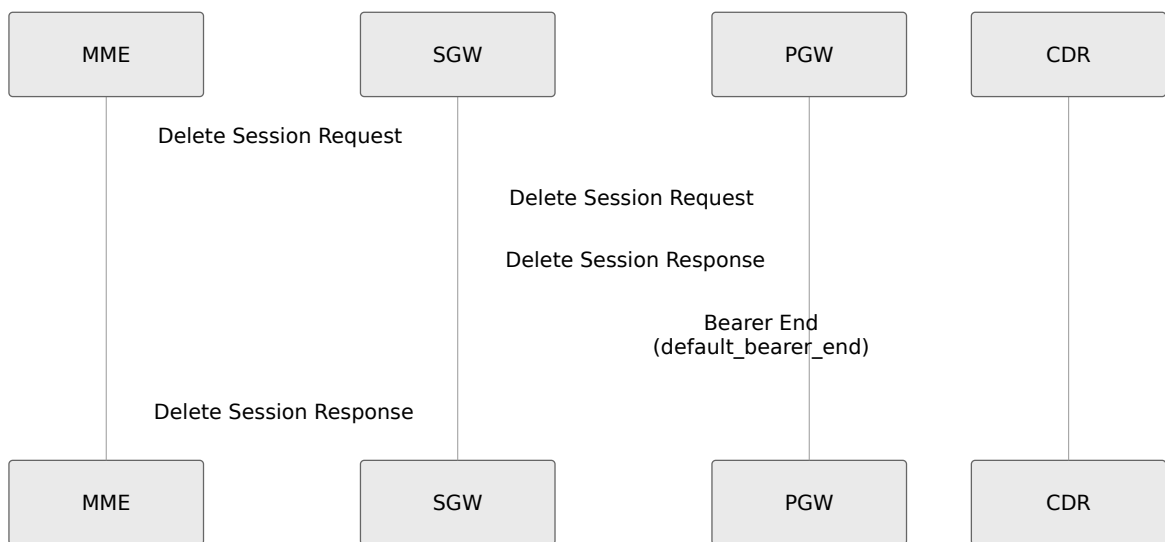
PDN Context Parameters

- **QoS:** QCI 9 (Best Effort)
- **APN:** PDN APN
- **IP:** IP address
- **IP Type:** PDN IP type (IPv4, IPv6, Dual Stack)

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)

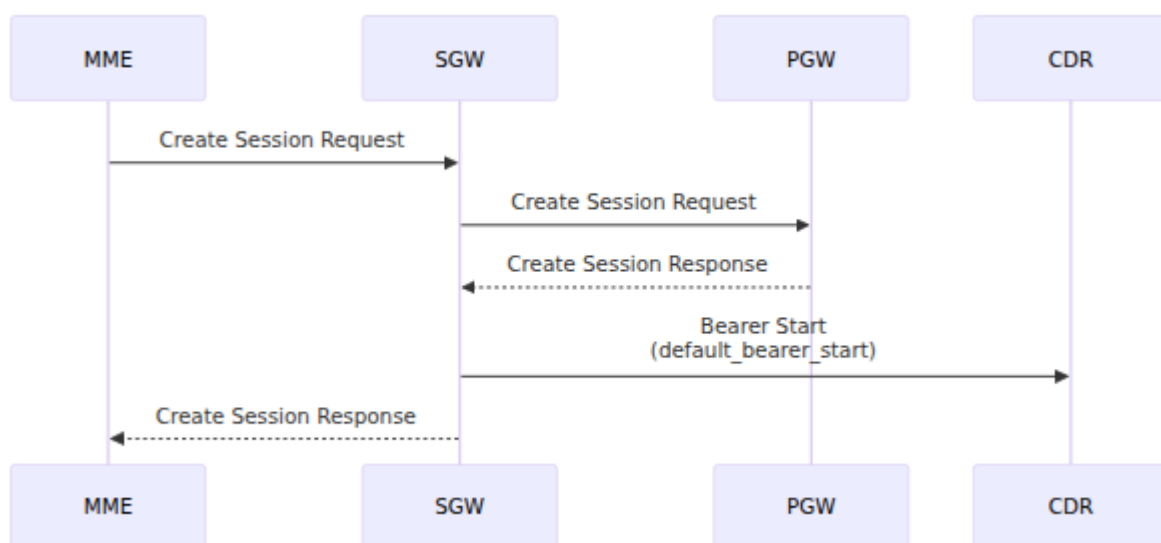
↓

MME deactivates bearer on RAN

↓

Delete Bearer Response back through SGW to PGW

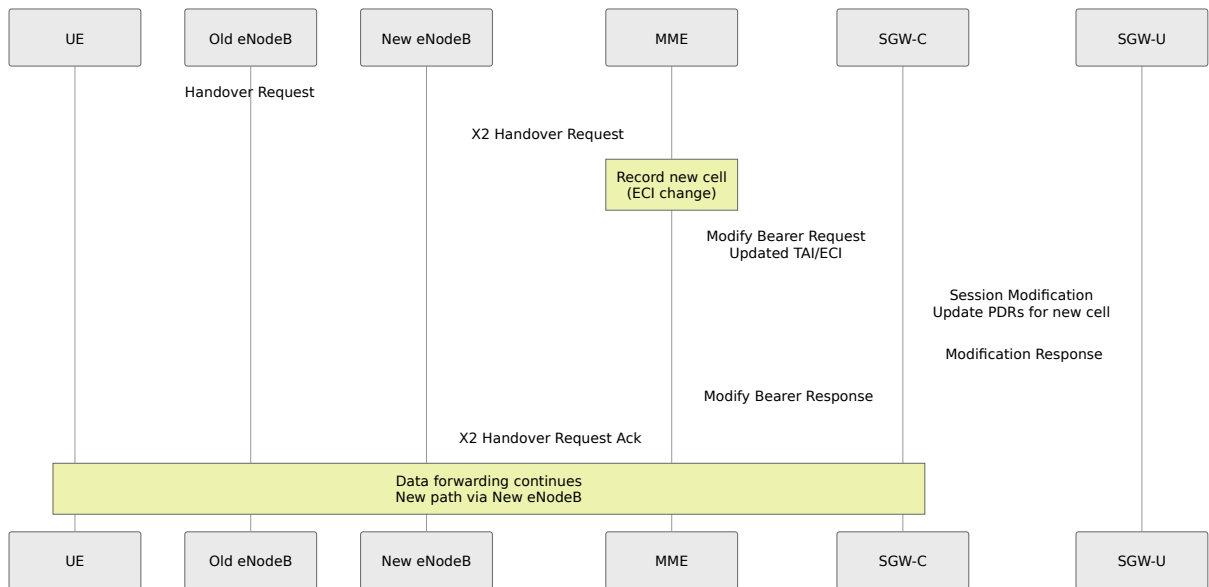
□□ **QoS** □□



Handover

Handover from MME to SGW

Handover from UE to MME

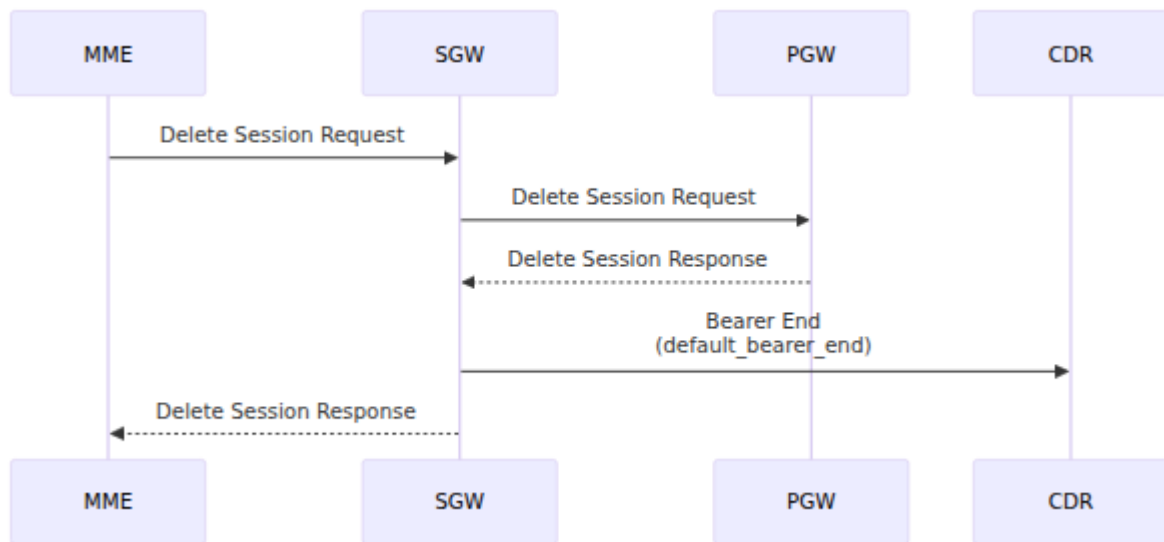


Handover from MME to SGW

- Handover from MME to SGW
- TEID change
- Handover from MME to SGW
- CDR (Charging Data Record) Charging ID

Handover from MME to SGW

Handover from UE to MME



CDR

- CDR “”
- Charging ID
-
- SGW-U

TAU

SGW TAU:

```

UE
↓
MME TAU Accept
↓
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 查看网络指标

网络指标

网络指标

```
# 查看网络指标
curl -s http://10.0.0.40:42068/metrics | grep active_ue_sessions

# 查看网络指标
curl -s http://10.0.0.40:42068/metrics | grep active_bearers

# 查看 APN 指标
curl -s http://10.0.0.40:42068/metrics | grep sessions_by_apn

# 查看网络指标
curl -s http://10.0.0.40:42068/metrics | grep
s11_inbound_messages_total
```

部署 Prometheus 监控系统

部署 Prometheus

部署 Prometheus

1. 部署 **API** 服务
2. 部署 Prometheus

3. 数据库 数据
4. 数据库 **CDR** 数据

数据库

数据库

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

# 数据库 80% 数据
# 数据库
```

数据库

数据库

数据库 数据

数据库

1. 数据库数据
2. 数据库 S11 数据
3. 数据库 PGW 数据
4. 数据库 Charging ID 数据

数据库

項目	説明
PGW アドレス	PGW S5/S8 アドレス
PGW IP	PGW IP アドレス
APN	PGW APN
SGW-U	SGW-U アドレス
	PGW アドレス

手順

1. 設定を確認する

2.

1. 設定を確認する
2. SGW-U アドレス
3. PGW アドレス
4. 設定を確認する

3.

項目	説明
SGW-U	SGW-U アドレス
PGW	PGW アドレス
	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 文件目录结构 CDR 文件

目录

目录

- 查看目录结构
- 查看目录 70-80% 查看
- 查看目录
- 查看目录

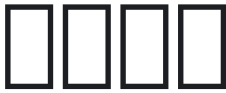
目录

- 查看目录结构
- 查看目录
- 查看目录 SGW-U 查看
- 查看 查看

目录

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

Sxa



PFCP  **SGW-U** 

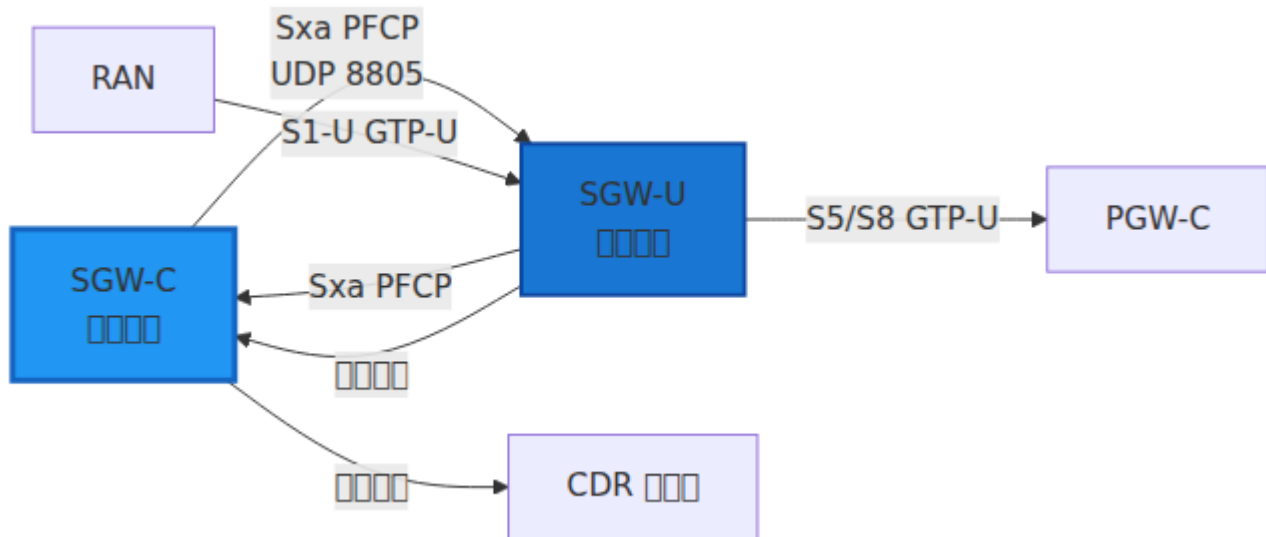
OmniSGW  Omnitouch 



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



Sxa   **PFCP** OmniSGW  SGW-U
 QoS 



SGW-C

- **PFCP v1.0** - 3GPP TS 29.244
- **SEID** - 32비트
- **QoS** - 3GPP TS 23.201
- **CP** - 3GPP TS 23.201
- **UP** - 3GPP TS 23.201

SGW-U

PFCP v1.0

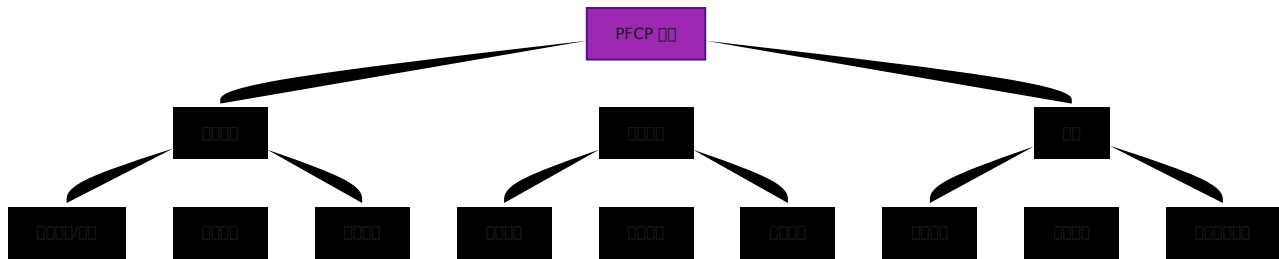
- PFCP v1.0 (3GPP TS 29.244)
- UDP
- 8805
- CP
- UP

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
```

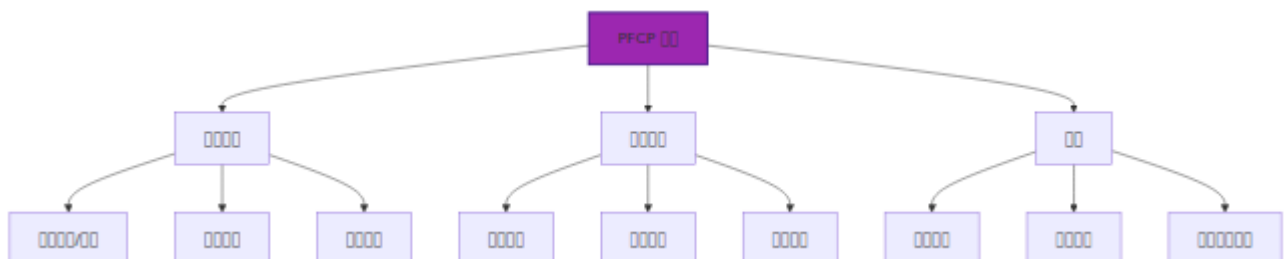
```
# SGW-U
ip route add <sgwu_network>/24 via <gateway_ip> dev eth0
```

```
# PFCP
# " "

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

PFCP

SGW-C SGW-U PFCP

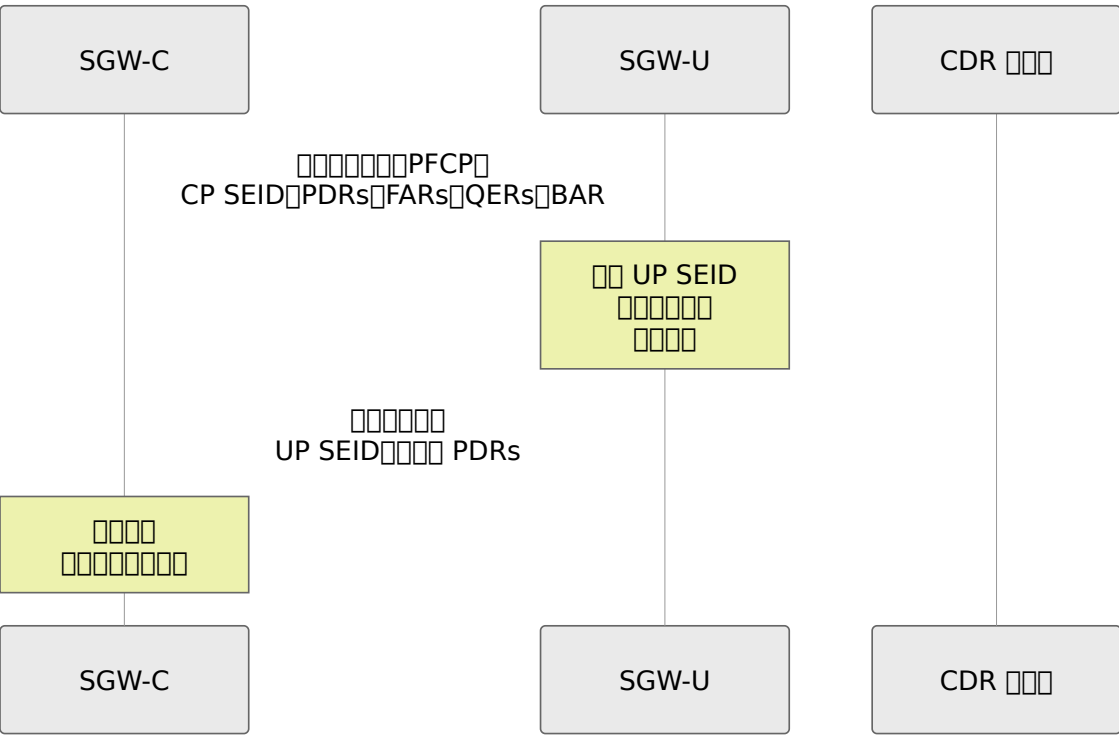


#

네트워크

구성

네트워크는 MME, SGW-C, SGW-U, CDR 등으로 구성



데이터베이스

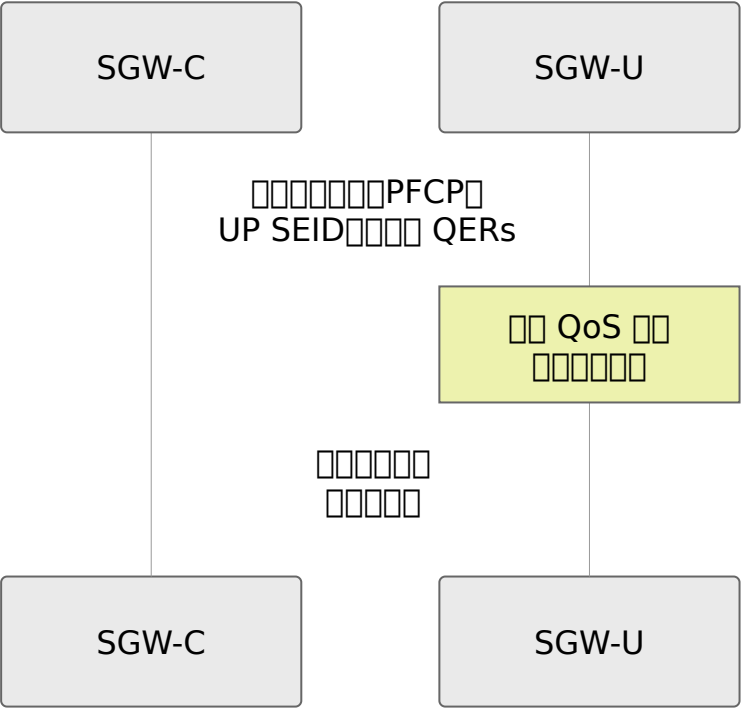
데이터	설명
CP SEID	SGW-C의 고유 식별자
PDRs	패킷 데이터 레코드의 집합
FARs	패킷 데이터 레코드의 집합
QERs	QoS 레코드의 집합
BAR	패킷 데이터 레코드의 집합
UP PDR	UP SEID와 PDRs의 집합

□□□□

[□□□]
↓ (□□□□)
[□□□]
↓ (□□□□)
[□□□□]

□□□□

□□□ □□ MME □□□□□□□□□□QoS □□□□□□

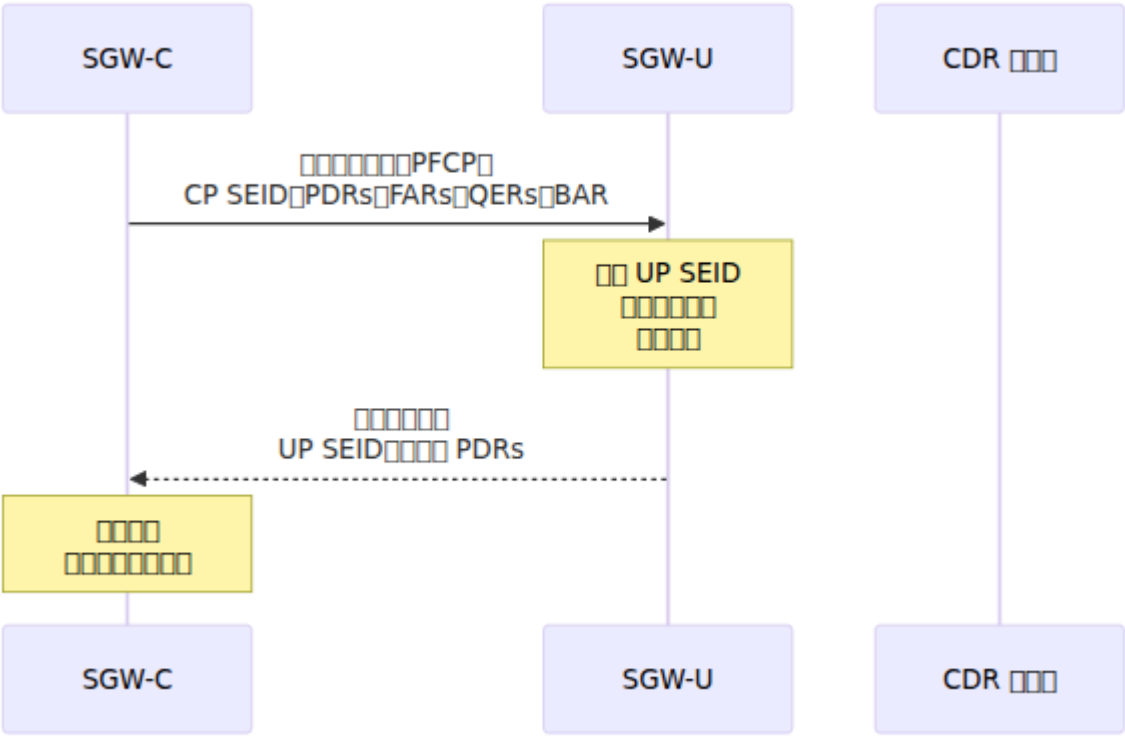


□□□□

項目	項目名	単位
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	項目 PDR/FAR/QER	項目項目

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 - EPC

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

UE - EPC

- UE IP address
- PDN type PGW-U 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 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 000001-90
MBR00000000	000	00000000
GBR00000000	000	0000000
ARP	00	00000000001-150

QoS 00QCI00

QCI	サービス	パラメータ
1	緊急通報専用GBR	MBR: 64 kbps
2	標準データ専用GBR	MBR: 256 kbps
3	標準データ専用GBR	MBR: 50 kbps
4	標準データ専用GBR	GBR: 128 kbps, MBR: 256 kbps
5	IMS 音声	GBR: 100 kbps, MBR: 256 kbps
6	標準データ専用	MBR: 10 Mbps
7	標準データ専用GBR	GBR: 64 kbps, MBR: 384 kbps
8	標準データ専用	MBR: 5 Mbps
9	標準データ専用	MBR: 3 Mbps

QCI 9 - 標準データ専用

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

QCI 1 - 緊急通報専用

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

BAR

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□□□□□

1. □□□□□

- UE → eNodeB → MME
- MME → HSS
- HSS → MME

2. □□□□□

- UE 1000000000000000
- 1000000
- 10000 UE 10000

3. SGW ☐☐☐

- **SGW** **MME**
- **SGW** **SGW**
-

BAR

BAR

T □□□□□□□□□□

 — 

L □□□□□□□□

└─ 00000000 CP 0000

□□□□

□□□□□□

SGW-U □ SGW-C □□□□□□□□□□□□



□□□□□□

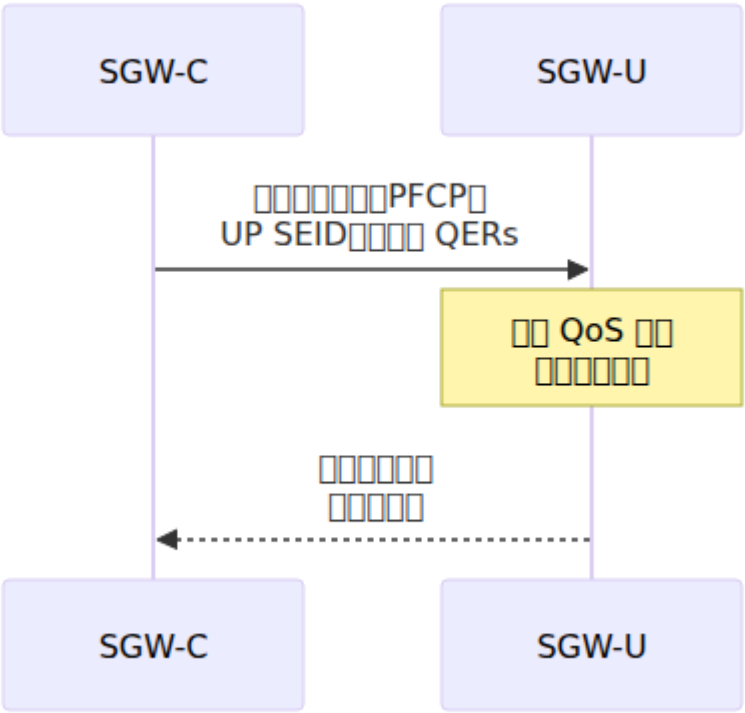
□□□□□□□□□□

□□	□□
□□□	□ 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

실행 PFCP 실행

```
# 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	確認事項

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"
```

確認事項

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

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
```

確認項目

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

確認項目

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 SGW-U 確認コマンド

確認

- 確認コマンド 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 □□□□□□
-