Data Charging Data Record (CDR) Format

Offline Charging for SGW-C

OmniSGW by Omnitouch Network Services

Table of Contents

Overview

The **Data CDR (Charging Data Record)** format provides offline charging capabilities for the Serving Gateway Control Plane (SGW-C). CDRs are generated to record bearer session events, data usage, and subscriber information for billing and analytics purposes.

This common format is compatible with PGW-C CDRs, ensuring consistency in charging records across the EPC infrastructure.

Key Features

- **CSV-based format** - Simple, human-readable comma-separated values
- **Event-based recording** - Captures bearer start, update, and end events
- **Volume metering** - Records uplink and downlink data usage
- **Automatic rotation** - Configurable file rotation based on time intervals
- **3GPP compliant** - Follows 3GPP TS 32.251 (PS domain charging) and TS 32.298 (CDR encoding)

Use Cases

Use Case	Description
Offline Charging	Generate CDRs for postpaid billing
Analytics	Analyze subscriber usage patterns
Audit Trail	Track all bearer session events
Capacity Planning	Monitor network resource utilization
Troubleshooting	Debug session and bearer issues

CDR File Format

File Naming Convention

```
<epoch_timestamp>
```

Example:

```
1726598022
```

The filename is the Unix epoch timestamp (in seconds) of when the file was created.

File Location

Default directory:

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

Configurable via `directory` parameter in the CDR reporter configuration.

File Header

Each CDR file begins with a multi-line header containing metadata:

```
# Data CDR File:
# File Start Time: HH:MM:SS (unix_timestamp)
# File End Time: HH:MM:SS (unix_timestamp)
# Gateway Name: <gateway_name>
#
epoch,imsi,event,charging_id,msisdn,ue_imei,timezone_raw,plmn,tac,eci
```

Header Fields:

- **File Start Time** - When the CDR file was created (human-readable and Unix timestamp)
- **File End Time** - When the file rotation will occur (human-readable and Unix timestamp)
- **Gateway Name** - Identifier for the SGW-C instance
- **Column Headers** - CSV field names for the data records

CDR Fields

Field Summary

Position	Field Name	Type	Description
0	epoch	integer	Event timestamp (Unix epoch seconds)
1	imsi	string	International Mobile Subscriber Identity
2	event	string	CDR event type (e.g., "default_bearer_start")
3	charging_id	integer	Unique charging identifier for the bearer
4	msisdn	string	Mobile Station ISDN Number (phone number)
5	ue_imei	string	International Mobile Equipment Identity
6	timezone_raw	string	UE timezone (reserved, currently empty)
7	plmn	integer	Public Land Mobile Network identifier
8	tac	integer	Tracking Area Code
9	eci	integer	E-UTRAN Cell Identifier
10	sgw_ip	string	SGW-C S5/S8 control plane IP address

Position	Field Name	Type	Description
11	ue_ip	string	UE IP address (IPv4|IPv6 format)
12	pgw_ip	string	PGW-C S5/S8 control plane IP address
13	apn	string	Access Point Name
14	qci	integer	QoS Class Identifier
15	octets_in	integer	Downlink data volume (bytes)
16	octets_out	integer	Uplink data volume (bytes)

CDR Events

Event Types

CDRs are generated for three types of events:

Event Type	Format	Description	When Generated
Bearer Start	`<type>_bearer_start`	Bearer establishment	Create Session Response sent
Bearer Update	`<type>_bearer_update`	Usage reporting during session	Periodic usage reports from user plane
Bearer End	`<type>_bearer_end`	Bearer termination	Delete Session Request/Response

Bearer Types:

- `default` - Default bearer (one per PDN connection)
- `dedicated` - Dedicated bearer (zero or more per PDN connection)

Event Examples

```
default_bearer_start      - Default bearer established
default_bearer_update     - Default bearer usage update
default_bearer_end        - Default bearer terminated
dedicated_bearer_start    - Dedicated bearer established
dedicated_bearer_update   - Dedicated bearer usage update
dedicated_bearer_end      - Dedicated bearer terminated
```

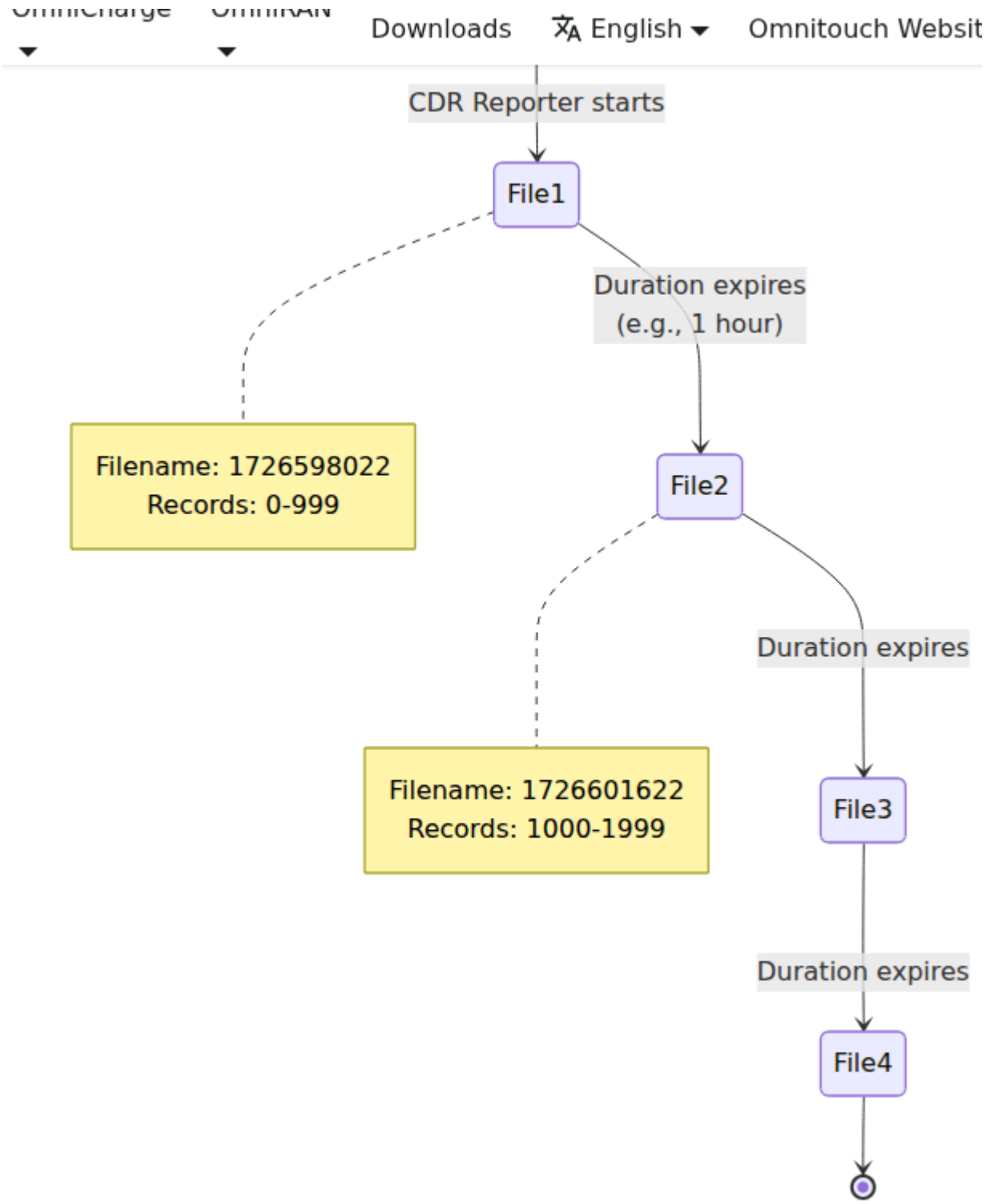
File Structure

Example CDR File

```
# Data CDR File:
# File Start Time: 18:53:42 (1726598022)
# File End Time: 19:53:42 (1726601622)
# Gateway Name: 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
```

File Rotation

CDR files are automatically rotated based on the configured duration:



Rotation Process:

1. Close current CDR file
2. Create new file with current timestamp
3. Write header to new file
4. Continue recording CDRs to new file

Configuration

Configuration Parameters

Parameter	Type	Description	Default	Recommended
`gateway_name`	string	SGW-C instance identifier	-	Use hostname or instance ID
`duration`	integer	File rotation interval (ms)	-	3600000 (1 hour)
`directory`	string	CDR output directory path	-	`/var/log/sgw_c/cdrs`

Configuration Examples

Production:

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

Development:

- **gateway_name:** "sgw-c-dev"
- **duration:** 300,000 ms (5 minute rotation for testing)
- **directory:** "/tmp/sgw_c_cdrs"

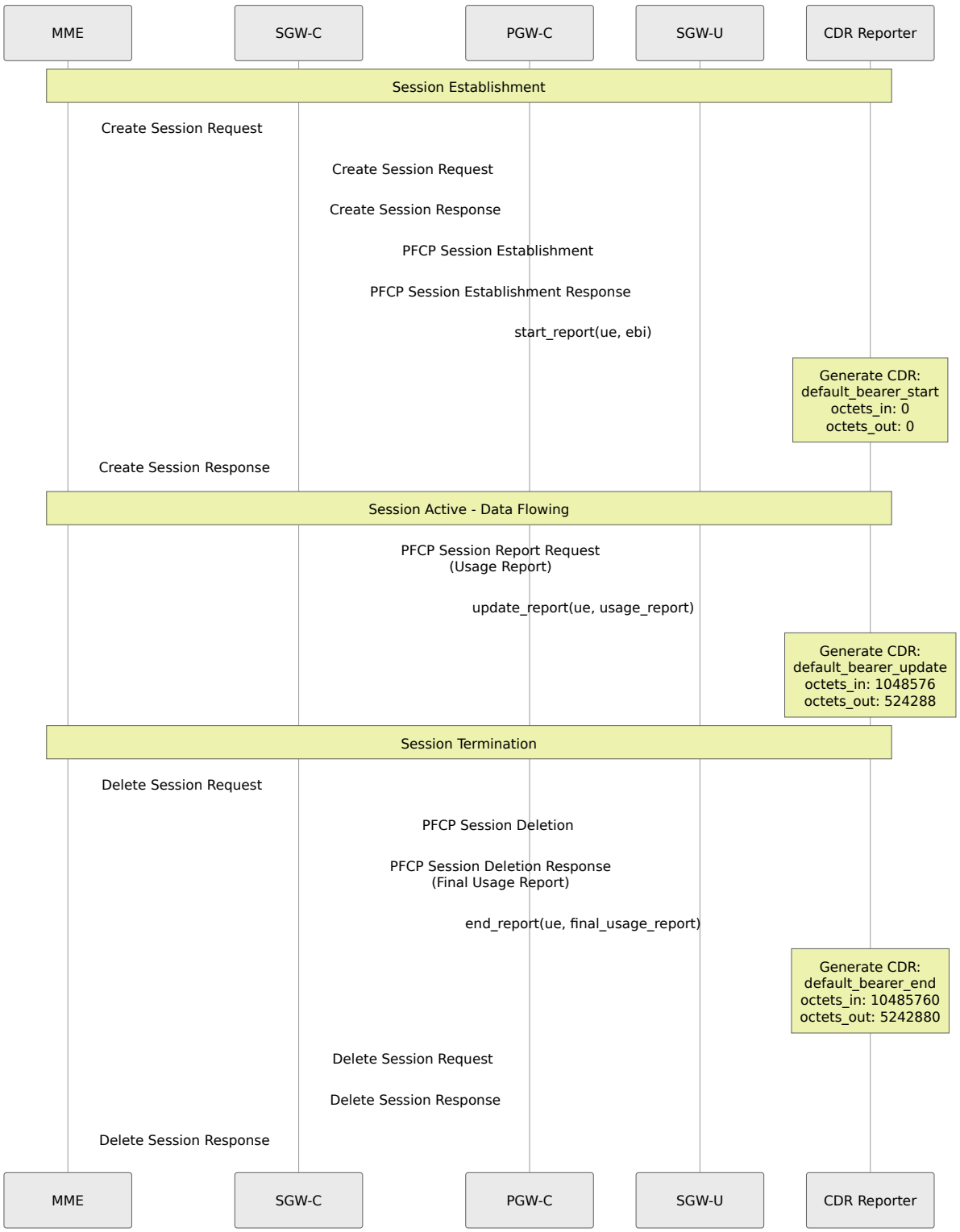
High-Volume:

- **gateway_name:** "sgw-c-prod-heavy"
- **duration:** 1,800,000 ms (30 minute rotation)
- **directory:** "/mnt/fast-storage/cdrs"

CDR Generation Flow

Bearer Lifecycle CDR Events

SGW-C CDR Generation:



CDR Generation Events

1. Bearer Start:

- **When:** Create Session Response is sent
- **Purpose:** Records bearer establishment with zero usage

- **octets_in:** 0
- **octets_out:** 0

2. Bearer Update:

- **When:** PFCP Session Report Request received from user plane
- **Purpose:** Records incremental data usage
- **octets_in:** Cumulative downlink bytes since bearer start
- **octets_out:** Cumulative uplink bytes since bearer start

3. Bearer End:

- **When:** PFCP Session Deletion Response received (with final usage)
- **Purpose:** Records final data usage before session termination
- **octets_in:** Final total downlink bytes
- **octets_out:** Final total uplink bytes

Field Details

1. epoch (Timestamp)

Type: Unix epoch timestamp (seconds)

Description: The time when the CDR event occurred

Example:

```
1726598022  → 2025-09-17 18:53:42 UTC
```

---

2. imsi (Subscriber Identity)

Type: String (up to 15 digits)

Format: MCCMNC + MSIN

Description: International Mobile Subscriber Identity uniquely identifying the subscriber

Example:

```
310260123456789
  └──┘ └──┘ └─────────┘
   MCC  MNC     MSIN
   (310)(260) (123456789)
```

Source: UE context, received in Create Session Request

---

3. event (CDR Event Type)

Type: String

Format: `<bearer_type>_bearer_<event>`

Values:

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

Determination:

- If EBI (EPS Bearer ID) equals LBI (Linked Bearer ID): `default`
- If EBI does not equal LBI: `dedicated`

Source: Bearer context (EBI vs LBI comparison)

---

4. charging_id (Charging Identifier)

Type: Unsigned 32-bit integer

Description: Unique identifier for charging correlation across network elements

Example:

```
12345
```

Source: Assigned by PGW-C, received in Create Session Response

Usage:

- Correlates charging events across SGW and PGW
- Used in Diameter Gy/Gz charging interfaces
- Unique per bearer

---

5. msisdn (Phone Number)

Type: String (E.164 format)

Description: Mobile Station ISDN Number (subscriber's phone number)

Format: Country code + national number

Example:

```
15551234567
 └┘└─────────┘
  │     │
  CC  National
  (1) (5551234567)
```

Source: UE context, typically from HSS via MME

---

6. ue_imei (Equipment Identity)

Type: String (15 digits)

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

Description: International Mobile Equipment Identity (device identifier)

Example:

```
123456789012345
└──────┘└────┘└
   TAC   SNR S
```

Source: UE context, received from MME

---

7. timezone_raw (UE Timezone)

Type: String (currently reserved/empty)

Description: Reserved field for UE timezone information

Current Status: Not populated (empty field in CSV)

Future Use: May include timezone offset and daylight saving time flag

Example:

```
,  (empty field)
```

---

8. plmn (Network Identifier)

Type: Integer (legacy format)

Description: Public Land Mobile Network identifier encoded as little-endian hex

Encoding Process:

```
MCC: 505, MNC: 57
  ↓
"50557"
  ↓
Swap pairs: "055570"
  ↓
Hex to decimal: 0x055570 = 349552
```

Example:

```
349552  → MCC: 505, MNC: 57
```

Source: UE location information from MME

Note: This is a legacy encoding format for backward compatibility

---

9. tac (Tracking Area Code)

Type: Unsigned 16-bit integer

Description: Tracking Area Code identifies the tracking area where the UE is located

Range: 0 - 65535

Example:

```
1234
```

Source: UE location information, received from MME in Create Session Request

Usage:

- Identifies mobility management area

- Used for paging and location updates
- Part of TAI (Tracking Area Identity)

---

10. eci (E-UTRAN Cell Identifier)

Type: Unsigned 28-bit integer

Description: E-UTRAN Cell Identifier uniquely identifies the cell serving the UE

Format: eNodeB ID (20 bits) + Cell ID (8 bits)

Range: 0 - 268,435,455

Example:

```
5678
```

Source: UE location information from MME

Usage:

- Identifies specific cell tower and sector
- Used for handover and mobility management
- Granular location information

---

11. sgw_ip (SGW Control Plane IP)

Type: String (IPv4 or IPv6 address)

Description: SGW-C's S5/S8 control plane IP address (F-TEID)

Format: Dotted decimal (IPv4) or colon-hex (IPv6)

Example:

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

Source: Local configuration, assigned to S5/S8 interface

---

12. ue_ip (UE IP Address)

Type: String (IPv4|IPv6 format)

Description: IP address assigned to the UE for the PDN connection

Format: `<ipv4>|<ipv6>`

Examples:

```
172.16.1.100|                      (IPv4 only)
|2001:db8::1                       (IPv6 only)
172.16.1.100|2001:db8::1          (Dual-stack)
```

Source: PDN Address Allocation (PAA) from PGW-C

Notes:

- Empty IPv4: No IPv4 address allocated
- Empty IPv6: No IPv6 address allocated
- Both present: Dual-stack PDN connection

---

13. pgw_ip (PGW Control Plane IP)

Type: String (IPv4 or IPv6 address)

Description: PGW-C's S5/S8 control plane IP address (remote F-TEID)

Format: Dotted decimal (IPv4) or colon-hex (IPv6)

Example:

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

Source: Received in Create Session Response from PGW-C

---

14. apn (Access Point Name)

Type: String (up to 100 characters)

Description: Access Point Name identifying the external network (PDN)

Format: DNS-like label format

Examples:

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

Source: Received in Create Session Request from MME

Usage:

- Determines which external network to connect to
- Drives policy and charging rules
- May determine IP address pool

---

15. qci (QoS Class Identifier)

Type: Unsigned 8-bit integer

Description: QoS Class Identifier defines the bearer's quality of service

Range: 1 - 9 (standardized), 128-254 (operator-specific)

Standardized QCI Values:

QCI	Resource Type	Priority	Packet Delay	Packet Loss	Example Service
1	GBR	2	100 ms	10^-2	Conversational Voice
2	GBR	4	150 ms	10^-3	Conversational Video
3	GBR	3	50 ms	10^-3	Real-time Gaming
4	GBR	5	300 ms	10^-6	Non-conversational Video
5	Non-GBR	1	100 ms	10^-6	IMS Signaling
6	Non-GBR	6	300 ms	10^-6	Video (buffered)
7	Non-GBR	7	100 ms	10^-3	Voice, Video, Gaming
8	Non-GBR	8	300 ms	10^-6	Video (buffered)
9	Non-GBR	9	300 ms	10^-6	Default Bearer

Example:

```
9  → Default bearer (best effort)
```

Source: Bearer QoS parameters from PGW-C

---

16. octets_in (Downlink Volume)

Type: Unsigned 64-bit integer

Description: Number of bytes transmitted in the downlink direction (network → UE)

Units: Bytes

Example:

```
1048576  → 1 MB downlink
```

Source: PFCP Volume Measurement from SGW-U

Notes:

- Cumulative for `update` events
- Final total for `end` events
- Always 0 for `start` events

---

17. octets_out (Uplink Volume)

Type: Unsigned 64-bit integer

Description: Number of bytes transmitted in the uplink direction (UE → network)

Units: Bytes

Example:

```
524288  → 512 KB uplink
```

Source: PFCP Volume Measurement from SGW-U

Notes:

- Cumulative for `update` events
- Final total for `end` events
- Always 0 for `start` events

Examples

Example 1: Basic Session with Single Update

Timeline:

1. Bearer established
2. 5 minutes later: Usage update (10 MB down, 5 MB up)
3. Session terminated

CDR Output:

```
# Data CDR File:
# File Start Time: 10:00:00 (1726570800)
# File End Time: 11:00:00 (1726574400)
# Gateway Name: sgw-c-01
# epoch,imsi,event,charging_id,msisdn,ue_imei,timezone_raw,plmn,tac,e
1726570800,310260111111111,default_bearer_start,10001,15551111111,111
1726571100,310260111111111,default_bearer_update,10001,15551111111,11
1726571400,310260111111111,default_bearer_end,10001,15551111111,11111
```

Example 2: Dual-Stack Session with Multiple Updates

Timeline:

1. Dual-stack bearer established (IPv4 + IPv6)
2. Multiple usage updates
3. Session terminated

CDR Output:

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

Example 3: Session with Dedicated Bearer

Timeline:

1. Default bearer established (QCI 9)
2. Dedicated bearer created for video (QCI 6)
3. Usage updates for both bearers
4. Dedicated bearer deleted
5. Default bearer terminated

CDR Output:

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

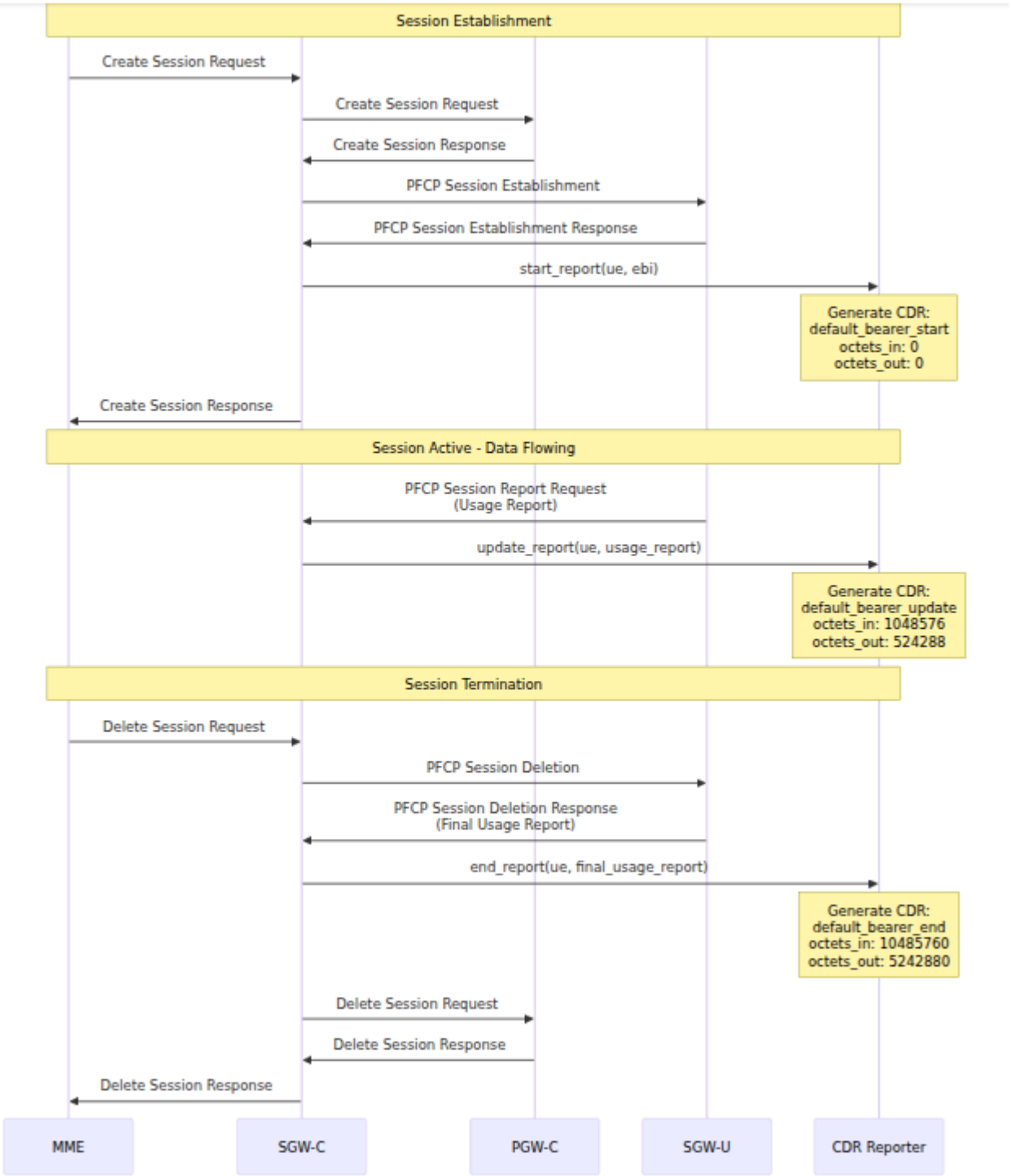
Analysis:

- Default bearer (10003) carries background traffic (10 MB down, 4 MB up)
- Dedicated bearer (10004) carries video traffic (200 MB down, 2 MB up)
- Different QCI values (9 vs 6) reflect different QoS treatment

---

Integration

CDR Processing Pipeline



CDR Collection Methods

1. File-based Collection:

```
# Monitor CDR directory (SGW-C)
inotifywait -m /var/log/sgw_c/cdrs/ -e close_write | while read
path action file; do
    # File rotation completed, process CDR
    process_cdr "$path$file"
done
```

2. Real-time Streaming:

```
# Tail and stream to processing pipeline
tail -F /var/log/sgw_c/cdrs/* | process_cdr_stream
```

Related Documentation

- Session Management - Session lifecycle
- Sxa Interface - Usage reporting from SGW-U
- Monitoring Guide - Metrics and alerting

3GPP References

- TS 32.251 - Packet Switched (PS) domain charging
- TS 29.274 - 3GPP Evolved Packet System (EPS); GTP-C protocol
- TS 29.244 - Interface between CP and UP nodes (PFCP)
- TS 32.298 - CDR encoding

CDR Format - *Offline Charging Records for SGW-C*

Developed by Omnitouch Network Services

Documentation Version: 1.0 **Last Updated:** 2025-12-10

SGW-C Configuration Guide

Complete runtime.exs Reference

OmniSGW by Omnitouch Network Services

Table of Contents

Overview

All OmniSGW runtime configuration is managed through `config/runtime.exs`. This file is loaded at startup and controls:

- Network interface bindings and ports
- Peer connectivity (MME, PGW-C, SGW-U)
- Metrics and monitoring
- CDR generation
- Operational parameters

Configuration is NOT compiled into the binary - changes to `runtime.exs` take effect on restart without recompilation.

View current runtime configuration via the Web UI Configuration page:

Configuration Structure

Basic Structure

```
# config/runtime.exs
import Config

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

Metrics Configuration

Basic Configuration

```
config :sgw_c,
  metrics: %{
    # Metrics exporter HTTP binding
    metrics_bind_address: "127.0.0.40",
    metrics_port: 42068,

    # Metrics poll interval (milliseconds)
    poll_interval_ms: 10000
  }
```

Production Configuration

```
config :sgw_c,
  metrics: %{
    # Bind to management network interface (not public)
    metrics_bind_address: System.get_env("MGT_IP") || "10.0.0.40",
    metrics_port: 42068,

    # Poll frequently for responsive dashboards
    poll_interval_ms: 5000
  }
```

Accessing Metrics

```
# Export metrics in Prometheus format
curl http://10.0.0.40:42068/metrics

# Common metrics:
# - teid_registry_count: Active S11/S5S8 TEIDs
# - seid_registry_count: Active PFCP sessions
# - s11_inbound_messages_total: S11 message count
# - sxa_inbound_messages_total: Sxa message count
```

For detailed metrics reference, Prometheus dashboards, and alerting configuration, see the Monitoring & Metrics Guide.

S11 Interface Configuration

Basic Configuration

```
config :sgw_c,
  s11: %{
    # Local IPv4 address for S11 (MME interface)
    local_ipv4_address: "10.0.0.10",

    # Optional: Local IPv6 address (for dual-stack)
    local_ipv6_address: nil,

    # Optional: Override default port
    local_port: 2123,

    # Message timeout (milliseconds)
    message_timeout_ms: 5000,

    # Retry configuration
    max_retries: 3,
    retry_backoff_ms: 1000
  }
```

Network Interface Selection

```
# Single interface (recommended)
config :sgw_c,
  s11: %{
    local_ipv4_address: "10.0.0.10"  # Single interface for S11
  }

# Dual interface (separate control and user plane networks)
config :sgw_c,
  s11: %{
    local_ipv4_address: "10.0.0.10"    # Control plane network
  },
  sxa: %{
    local_ip_address: "10.1.0.20"      # User plane network
  }
```

Message Timing Configuration

```
config :sgw_c,
  s11: %{
    # For high-latency networks (> 100ms RTT)
    message_timeout_ms: 10000,
    max_retries: 5,
    retry_backoff_ms: 2000,

    # For low-latency networks (< 50ms RTT)
    message_timeout_ms: 3000,
    max_retries: 2,
    retry_backoff_ms: 500
  }
```

S5/S8 Interface Configuration

Basic Configuration

```
config :sgw_c,
  s5s8: %{
    # Local IPv4 address for S5/S8 (PGW interface)
    local_ipv4_address: "10.0.0.15",

    # Optional: Local IPv6 address
    local_ipv6_address: nil,

    # Optional: Override default port
    local_port: 2123,

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

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

PGW Peer Configuration

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

# Redundant PGWs (load-balanced)
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"}
    ]
  }

# Redundant PGWs (active-standby)
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 Interface Configuration

Basic Configuration

```
config :sgw_c,
  sxa: %{
    # Local IP address for Sxa interface
    local_ip_address: "10.0.0.20",

    # Optional: Override default port
    local_port: 8805,

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

    # Heartbeat interval (seconds)
    heartbeat_interval_s: 20,

    # Session timeout (milliseconds)
    session_timeout_ms: 5000,

    # Retries
    max_retries: 3
  }
```

SGW-U Peer Configuration

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

# Redundant SGW-Us
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"
      }
    ]
  }
```

Heartbeat Configuration

```
# Fast detection (aggressive)
config :sgw_c,
  sxa: %{
    heartbeat_interval_s: 10,
    max_retries: 2
  }

# Normal detection (balanced)
config :sgw_c,
  sxa: %{
    heartbeat_interval_s: 20,
    max_retries: 3
  }

# Slow detection (lenient)
config :sgw_c,
  sxa: %{
    heartbeat_interval_s: 40,
    max_retries: 5
  }
```

CDR Configuration

Basic Configuration

```
config :sgw_c,
  cdr: %{
    # Gateway identifier in CDRs
    gateway_name: "sgw-c-prod-01",

    # File rotation interval (milliseconds)
    rotation_interval_ms: 3600000,  # 1 hour

    # Output directory for CDR files
    directory: "/var/log/sgw_c/cdrs"
  }
```

Production Configuration

```
config :sgw_c,
  cdr: %{
    # Use hostname or instance ID from deployment
    gateway_name: System.get_env("HOSTNAME") || "sgw-c-prod-01",

    # Rotate every hour for manageability
    rotation_interval_ms: 3600000,

    # Use fast storage for CDRs
    directory: System.get_env("CDR_DIR") || "/var/log/sgw_c/cdrs"
  }
```

High-Volume Configuration

```
config :sgw_c,
  cdr: %{
    gateway_name: "sgw-c-prod-high-vol",

    # Rotate more frequently to manage file size
    rotation_interval_ms: 1800000,  # 30 minutes

    # Use dedicated fast storage
    directory: "/mnt/fast-ssd/sgw_c/cdrs"
  }
```

Deployment Examples

Single Gateway (Minimal)

```
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"
  }
```

High-Availability Setup (Redundant)

```
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"
  }
```

High-Volume Carrier Grade

```
import Config

# Load all settings from environment (required in production)
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")

# Parse PGW peers from environment (JSON format)
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)

# Parse SGW-U peers from environment
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 minute rotation
    directory: "/mnt/fast-ssd/sgw_c/cdrs"
  }
```

Environment Variables Reference

Required Variables

Variable	Description	Example
`SGW_S11_IP`	S11 interface IP	`10.0.0.10`
`SGW_S5S8_IP`	S5/S8 interface IP	`10.0.0.15`
`SGW_SXA_IP`	Sxa interface IP	`10.0.0.20`
`MGT_IP`	Metrics bind address	`10.0.0.40`

Optional Variables

Variable	Description	Default
`HOSTNAME`	Gateway name for CDRs	System hostname
`PGW_PEERS`	JSON array of PGW peers	`[]`
`SGWU_PEERS`	JSON array of SGW-U peers	`[]`
`CDR_DIR`	CDR output directory	`/var/log/sgw_c/cdrs`

Example Deployment

```
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
```

Verification

Check Configuration at Startup

Monitor startup logs:

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

# Expected output:
# [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
```

Verify Active Configuration

```
# Check metrics are accessible
curl http://10.0.0.40:42068/metrics | head -20

# Verify S11 port listening
netstat -an | grep 2123

# Check S11 peer connectivity in logs
tail -f /var/log/sgw_c/sgw_c.log | grep "S11"
```

Common Configuration Issues

"Address already in use"

Problem: Port binding fails on startup

Solution:

```
# Find process using port
lsof -i :2123

# Kill existing process or use different port
killall sgw_c
# or
config :sgw_c, s11: %{local_port: 2124}
```

"Connection refused" to PGW

Problem: S5/S8 cannot reach PGW-C

Solution:

```
# Verify PGW IP
ping 10.0.0.20

# Check firewall rules
iptables -L | grep 2123

# Test connectivity
nc -u -v 10.0.0.20 2123
```

"Cannot reach SGW-U"

Problem: Sxa association fails

Solution:

```
# Verify SGW-U reachable
ping 10.0.0.30

# Check PFCP port
netstat -an | grep 8805

# Verify PFCP port open
iptables -L | grep 8805
```

Monitoring & Metrics Guide

Prometheus Metrics, Grafana Dashboards, and Alerting

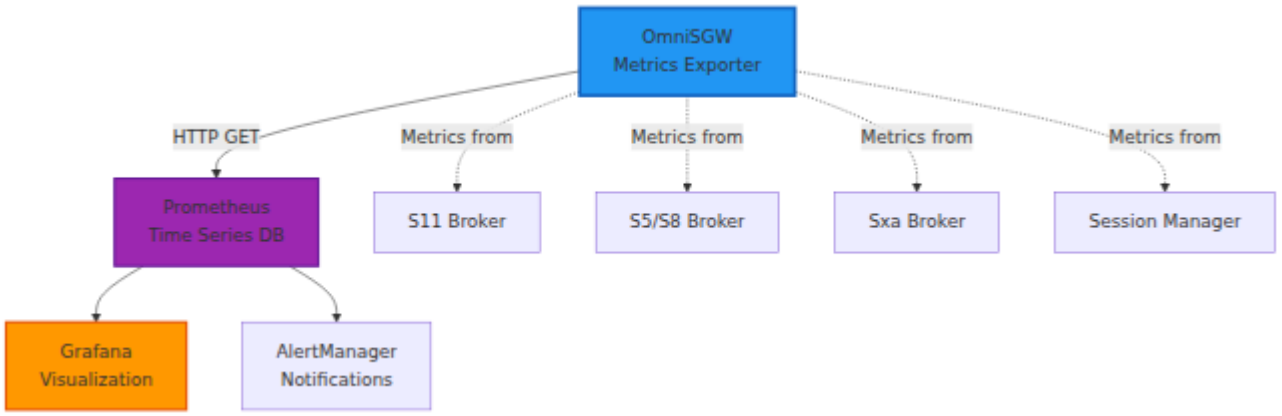
OmniSGW by Omnitouch Network Services

Table of Contents

Overview

OmniSGW exposes Prometheus-compatible metrics for comprehensive monitoring of network operations, session management, and system health.

Metrics Architecture



Metrics Exporter

Accessing Metrics

Metrics are exposed at the configured HTTP endpoint:

```
# Default endpoint (if configured)
curl http://127.0.0.40:42068/metrics

# Export to file
curl http://127.0.0.40:42068/metrics > metrics.txt

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

For metrics endpoint configuration (bind address, port, and poll interval), see the Configuration Guide.

Metrics Format

Metrics are in Prometheus text format:

```
# HELP teid_registry_count Total number of allocated TEIDs
# TYPE teid_registry_count gauge
teid_registry_count 1234

# HELP s11_inbound_messages_total Total number of inbound S11
messages
# 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
```

Available Metrics

Session Management Metrics

Active Sessions:

```
teid_registry_count
├── Description: Active S11/S5S8 TEID allocations
├── Type: Gauge
├── Range: 0 to max licensed capacity
└── Example: 1234 (1234 active sessions)

seid_registry_count
├── Description: Active PFCP sessions (per SGW-U peer)
├── Type: Gauge
├── Labels: peer_ip
└── Example: seid_registry_count{peer_ip="10.0.0.30"} 1234

active_ue_sessions
├── Description: Total active UE sessions
├── Type: Gauge
└── Example: 5000

active_bearers
├── Description: Total active bearers (default + dedicated)
├── Type: Gauge
└── Example: 5500 (1 default + 0.1 dedicated per session)

charging_id_registry_count
├── Description: Active charging IDs
├── Type: Gauge
└── Example: 5000
```

Message Counters

S11 (MME Interface):

```
s11_inbound_messages_total
├── Type: Counter (increasing)
├── Labels: message_type
├── Values:
│   ├── create_session_request
│   ├── delete_session_request
│   ├── modify_bearer_request
│   ├── create_bearer_request
│   ├── delete_bearer_request
│   ├── release_access_bearers_request
│   ├── downlink_data_notification
│   └── echo_request
└── Example:
s11_inbound_messages_total{message_type="create_session_request"}
5432
```

S5/S8 (PGW Interface):

```
s5s8_inbound_messages_total
├── Type: Counter
├── Labels: message_type
├── Values: (same as S11 request types)
└── Example:
s5s8_inbound_messages_total{message_type="create_session_request"}
4500
```

Sxa (SGW-U Interface):

```
sxa_inbound_messages_total
├── Type: Counter
├── Labels: message_type
├── Values:
│   ├── session_establishment_request
│   ├── session_modification_request
│   ├── session_deletion_request
│   ├── session_report_request
│   ├── association_setup_request
│   └── heartbeat_request
└── Example:
sxa_inbound_messages_total{message_type="session_report_request"}
67000
```

Performance Metrics

Message Latency:

```
s11_inbound_duration_seconds
├── Type: Histogram (with buckets)
├── Description: S11 message processing time
├── Percentiles: _count, _sum, _bucket
└── Example: s11_inbound_duration_seconds_bucket{le="0.1"} 5000

s5s8_inbound_duration_seconds
├── Type: Histogram
├── Description: S5/S8 message processing time

sxa_inbound_duration_seconds
├── Type: Histogram
├── Description: Sxa message processing time
```

PFCP Association:

```
pfcp_association_status
├── Type: Gauge
├── Values: 1 (associated) or 0 (not associated)
├── Labels: peer_ip, node_id
└── Example: pfcp_association_status{peer_ip="10.0.0.30"} 1

pfcp_heartbeat_latency_ms
├── Type: Gauge
├── Description: Heartbeat round-trip time
├── Labels: peer_ip
└── Example: pfcp_heartbeat_latency_ms{peer_ip="10.0.0.30"} 15
```

Error Metrics

Protocol Errors:

```
s11_inbound_errors_total
├── Type: Counter
├── Labels: error_type
├── Values:
│   ├── parse_error
│   ├── validation_error
│   ├── timeout
│   └── other
└── Example: s11_inbound_errors_total{error_type="timeout"} 12

s5s8_inbound_errors_total
├── Type: Counter
├── Description: S5/S8 errors

sxa_inbound_errors_total
├── Type: Counter
├── Description: Sxa errors
```

Create Session Failures:

```
create_session_response_cause
├── Type: Counter
├── Labels: cause_code
├── Values: (3GPP cause codes)
├── Examples:
│   ├── cause_code="0": Success
│   ├── cause_code="16": No resources available
│   ├── cause_code="25": Semantics error
│   └── cause_code="49": No matching rule
```

Prometheus Configuration

Installation

```
# Download 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
```

Configuration File

prometheus.yml:

```yaml
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']
```

Starting Prometheus

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

Accessing Prometheus

```
http://localhost:9090
```

Grafana Dashboards

Installation

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

Adding Data Source

1. **Open Grafana:** http://localhost:3000
2. **Configuration → Data Sources**
3. **Add → Prometheus**
4. **URL:** http://prometheus:9090

Dashboard: Session Overview

Panels:

```
Row 1:
├── Active Sessions (Gauge)
├── Active Bearers (Gauge)
├── S11 Messages/sec (Graph)
└── S5/S8 Messages/sec (Graph)

Row 2:
├── Sxa Messages/sec (Graph)
├── S11 Latency p95 (Graph)
├── S5/S8 Latency p95 (Graph)
└── Sxa Latency p95 (Graph)

Row 3:
├── S11 Errors/min (Graph)
├── S5/S8 Errors/min (Graph)
├── Sxa Errors/min (Graph)
└── PFCP Associations (Status)
```

Dashboard: Interface Health

Panels:

```
Row 1:
├── S11 Peer Status (Status)
├── S5/S8 Peer Status (Status)
├── SGW-U Peer Status (Status list)
└── System Load (Gauge)

Row 2:
├── S11 Message Rate (Graph)
├── S5/S8 Message Rate (Graph)
├── Sxa Message Rate (Graph)
└── Error Rate (Graph)

Row 3:
├── Message Latency Histogram (Heatmap)
├── Session Creation Rate (Graph)
├── Session Termination Rate (Graph)
└── Bearer Creation Rate (Graph)
```

Dashboard: Capacity Planning

Panels:

```
Row 1:
├── Sessions vs Capacity (Gauge + Threshold)
├── Bearers vs Capacity (Gauge + Threshold)
├── PFCP Sessions Distribution (Bar Chart)
└── Sessions by APN (Pie Chart)

Row 2:
├── Session Growth Trend (Graph)
├── Bearer Growth Trend (Graph)
├── Peak Session Time (Heatmap)
└── Session Duration Distribution (Histogram)
```

Dashboard Query Examples

Active Sessions:

```
teid_registry_count
```

Session Creation Rate:

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

S11 Latency (95th percentile):

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

Error Rate:

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

PFCP Association Status:

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

Alerting Rules

Alert Rules File

sgw-c-alerts.yml:

```yaml
groups:
  - name: sgw-c-alerts
    interval: 30s
    rules:
      # Session capacity alerts
      - alert: SGWCapacityHigh
        expr: (teid_registry_count / 100000) > 0.8
        for: 5m
        annotations:
          summary: "SGW session capacity above 80%"
          description: "Sessions: {{ $value }} of 100000"

      # Interface health alerts
      - alert: S11PeerDown
        expr: absent(s11_inbound_messages_total) > 0
        for: 2m
        annotations:
          summary: "S11 interface unreachable"

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

      - alert: SGWUAssociationDown
        expr: pfcp_association_status == 0
        for: 1m
        annotations:
          summary: "SGW-U association lost"
          description: "Peer: {{ $labels.peer_ip }}"

      # Message latency alerts
      - alert: S11LatencyHigh
        expr: histogram_quantile(0.95,
rate(s11_inbound_duration_seconds_bucket[5m])) > 1
        for: 5m
        annotations:
          summary: "S11 latency above 1 second"
          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 latency above 1 second"

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

      - alert: SessionEstablishmentFailure
        expr: rate(create_session_response_cause{cause_code!="0"}
[5m]) > 20
        for: 3m
        annotations:
          summary: "High session establishment failure rate"
          description: "{{ $value }} failures/sec"
```

Configuring AlertManager

alertmanager.yml:

```yaml
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'
```

Alert Notification Examples

Slack Integration:

```
 SGW Capacity High
Severity: Warning
Active Sessions: 85,000 / 100,000 (85%)
Time: 2025-12-10 15:30:00 UTC
Action: Monitor for capacity increase
```

Email Integration:

```
Subject: [ALERT] S11 Peer Unreachable

SGW-C S11 interface has not received messages for 2 minutes.
This may indicate:
- MME network connectivity issue
- SGW-C restart required
- S11 port configuration changed

Immediate Action Required: Check S11 status
```

Troubleshooting

Metrics Not Appearing

Problem: Metrics endpoint empty or 404

Diagnosis:

```
# Check metrics endpoint reachable
curl -v http://127.0.0.40:42068/metrics

# Check logs for metrics exporter errors
tail -f /var/log/sgw_c/sgw_c.log | grep -i metric

# Verify configuration
cat config/runtime.exs | grep metrics
```

Solutions:

1. Restart SGW-C process
2. Verify metrics IP/port not blocked by firewall
3. Check bind address configuration
4. Ensure sufficient memory for metrics collection

Missing Metrics for Specific Interface

Problem: S11 metrics show but S5/S8 or Sxa missing

Diagnosis:

1. Check interface is configured
2. Verify interface is active
3. Monitor logs for connection errors

Solution:

- Verify peer reachability
- Check interface binding
- Review configuration

High Memory Usage

Problem: Metrics exporter consuming excessive memory

Diagnosis:

```
# Check process memory
ps aux | grep sgw_c | grep -v grep | awk '{print $6}'

# Monitor growth over time
watch -n 5 'ps aux | grep sgw_c'
```

Solutions:

1. Reduce metrics poll interval
2. Limit number of metric samples
3. Implement metric retention policy
4. Scale to multiple instances

Best Practices

Metrics Collection

- **Scrape Interval:** 15-30 seconds for balance
- **Retention:** 15-30 days of metrics storage
- **Aggregation:** Pre-aggregate high-cardinality metrics
- **Sampling:** Use percentiles for latency, not raw values

Dashboard Design

- **Context:** Include time range, instance, peer info
- **Layering:** Overview → Detail → Debug
- **Alerting:** Visual thresholds on capacity graphs
- **Correlation:** Link related metrics

Alerting Strategy

- **Hierarchy:** Critical → Warning → Info
- **Escalation:** Page on-call for critical alerts
- **Threshold Tuning:** Baseline then +20% for warning
- **Regular Testing:** Test alert paths monthly

Monitoring Guide - *OmniSGW Metrics and Observability*

Developed by Omnitouch Network Services

Documentation Version: 1.0 **Last Updated:** 2025-12-10

S11 Interface Documentation

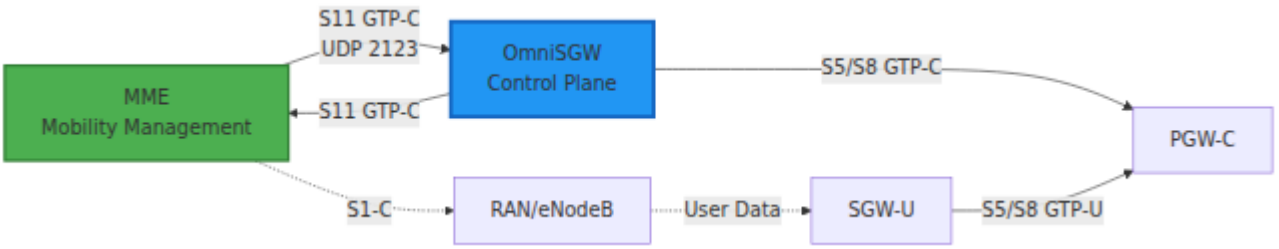
GTP-C Communication with MME

OmniSGW by Omnitouch Network Services

Table of Contents

Overview

The **S11 interface** connects OmniSGW to the MME (Mobility Management Entity) using the **GTP-C v2** (GPRS Tunnelling Protocol - Control plane) protocol. This interface handles all control plane signaling for UE session management, bearer operations, and mobility procedures.



Key Features

- **GTP-C v2 Protocol** - Standards-compliant message signaling
- **TEID-based Routing** - Tunnel endpoint identifiers for session tracking
- **Stateful Session Management** - Maintains UE context across messages
- **Handover Support** - Coordinates inter-MME and intra-MME mobility
- **Bearer Operations** - Create, modify, and delete bearers
- **Downlink Data Notifications** - Paging for suspended sessions

---

Protocol Details

GTP-C Version 2

- **Protocol:** GTP-C v2 (3GPP TS 29.274)
- **Transport:** UDP
- **Port:** 2123 (standard)
- **Interface Type:** Control Plane
- **Direction:** Bidirectional request/response

TEID (Tunnel Endpoint Identifier)

Each session has unique TEIDs for routing messages:

- **Local TEID** - Allocated by OmniSGW for incoming messages from MME
- **Remote TEID** - Allocated by MME for outgoing messages to MME

```
Message Routing:
  MME → SGW: Uses OmniSGW's Local TEID in message header
  SGW → MME: Uses MME's Remote TEID in message header
```

Message Format

All S11 messages follow GTP-C v2 format:

```
GTP-C Header (12-16 bytes)
├── Version (3 bits): 0x2 (GTP-C v2)
├── Piggyback Flag (1 bit)
├── TEID Flag (1 bit): 1 (TEID present)
├── Message Type (8 bits): Identifies message type
├── Message Length (16 bits): Length of message contents
├── TEID (32 bits): Tunnel Endpoint Identifier
├── Sequence Number (24 bits): For request/response matching
└── Spare (8 bits): Always 0

Message Contents (variable)
├── Information Elements (IE)
│   ├── IE Type (8 bits)
│   ├── Length (16 bits)
│   └── Value (variable)
└── ... more IEs
```

Configuration

Basic Configuration

```
# config/runtime.exs
config :sgw_c,
  s11: %{
    # Local IPv4 address for S11 interface
    local_ipv4_address: "10.0.0.10",

    # Optional: Local IPv6 address (for dual-stack)
    local_ipv6_address: nil,

    # Optional: Override default port
    local_port: 2123,

    # Message timeouts
    message_timeout_ms: 5000,

    # Retry configuration
    max_retries: 3,
    retry_backoff_ms: 1000
  }
```

Network Requirements

Firewall Rules:

```
# Allow GTP-C from MME network (inbound)
iptables -A INPUT -p udp --dport 2123 -s <mme_network>/24 -j
ACCEPT

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

Routing:

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

Network Testing:

```
# Test connectivity to MME (use GTP heartbeat)
# Check logs for "S11 Broker connected" message

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

Message Types

Overview of S11 Messages

Session Establishment Messages

Create Session Request (S11)

Direction: MME → OmniSGW

Purpose: Establish new UE session (initial attach or PDN connectivity)

Key Information Elements:

IE Name	Type	Description
IMSI	Binary	International Mobile Subscriber Identity
MSISDN	BCD	Mobile phone number
MEI	Binary	Mobile Equipment Identity
RAT Type	Enum	Radio Access Technology (EUTRAN)
Bearer Context	Grouped	Default bearer configuration
UE Time Zone	DateTime	UE's current timezone
ULI	Grouped	User Location Information (TAI, ECGI)
Serving Network	PLMN	MCC/MNC
APN	String	Access Point Name

Response: Create Session Response

IE Name	Type	Description
Cause	Enum	Result of request (success/failure)
Bearer Context	Grouped	Allocated bearer info with TEID
PDN Address Allocation	Grouped	Allocated IP address from PGW
APN Restriction	Enum	APN restrictions

Session Modification Messages

Modify Bearer Request (S11)

Direction: MME → OmniSGW (request initiated from MME)

Purpose: Modify bearer parameters during active session

Key Information Elements:

IE Name	Type	Description
MEI	Binary	Mobile Equipment Identifier
ULI	Grouped	Updated User Location Information
UE Time Zone	DateTime	Updated timezone
TAI	TAI	Tracking Area Identifier
ECGI	ECGI	E-UTRAN Cell Global Identifier

Response: Modify Bearer Response

IE Name	Type	Description
Cause	Enum	Result of modification
Bearer Context	Grouped	Updated bearer parameters

Bearer Management Messages

Create Bearer Request/Response

Direction: Can be initiated from MME or SGW

Purpose: Activate dedicated bearer for QoS-requiring services

Trigger Scenarios:

- Voice service activation

- Video streaming request
- Game online activation

Delete Bearer Request/Response

Direction: Can be initiated from MME or SGW (via PGW)

Purpose: Deactivate dedicated bearer when no longer needed

Mobility Messages

Release Access Bearers Request/Response

Direction: MME → OmniSGW

Purpose: Suspend all bearers during radio disconnect (paging scenario)

Effects:

- Bearers remain in context but suspended
- User plane forwarding paused
- Data buffering initiated in SGW-U
- UE can resume with service request

Modify Access Bearers Request/Response

Direction: OmniSGW → MME or MME → OmniSGW

Purpose: Update bearer access during handover or recovery

Paging Messages

Downlink Data Notification (S11)

Direction: PGW-C → OmniSGW → MME

Purpose: Notify MME of pending downlink data for suspended UE

Key Information Elements:

IE Name	Type	Description
EBI	Integer	EPS Bearer ID
IMSI	Binary	Subscriber identity

Response: Downlink Data Acknowledge

Session Establishment

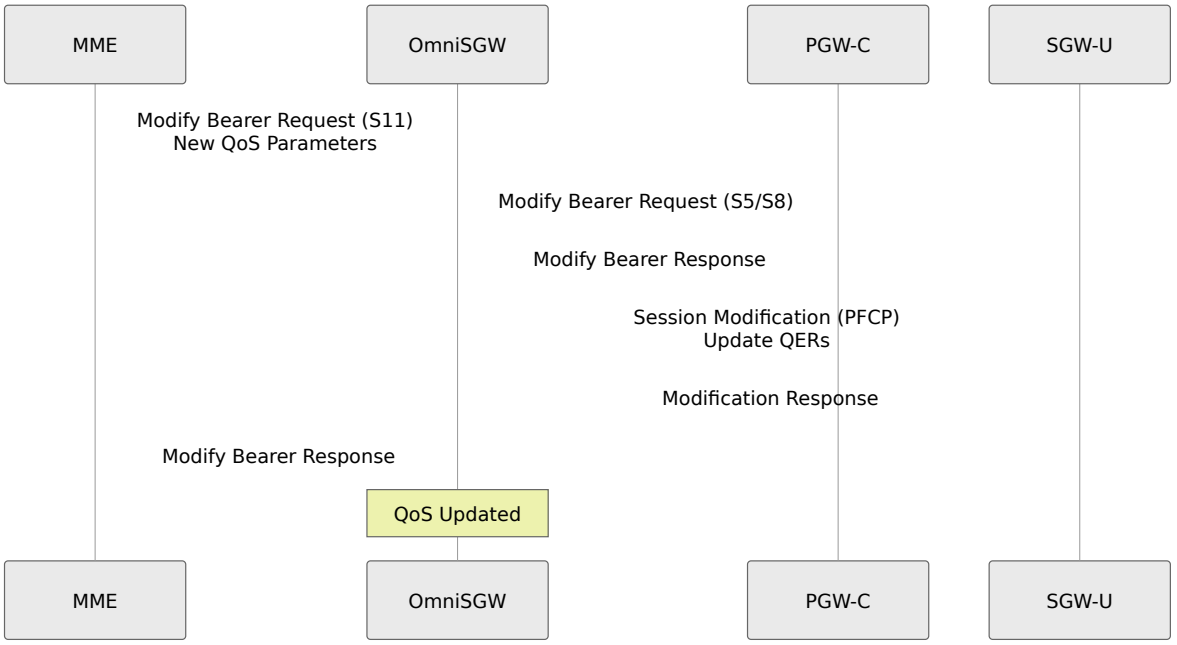
UE Initial Attach Flow

State Transitions:

```
[UE Not Connected]
    ↓ (Attach Request)
[Creating Session to PGW]
    ↓ (PGW responds)
[Establishing User Plane]
    ↓ (PFCP session active)
[Session Active]
```

Session Modification

Bearer QoS Modification



Tracking Area Update (TAU)

TAU without SGW Change:

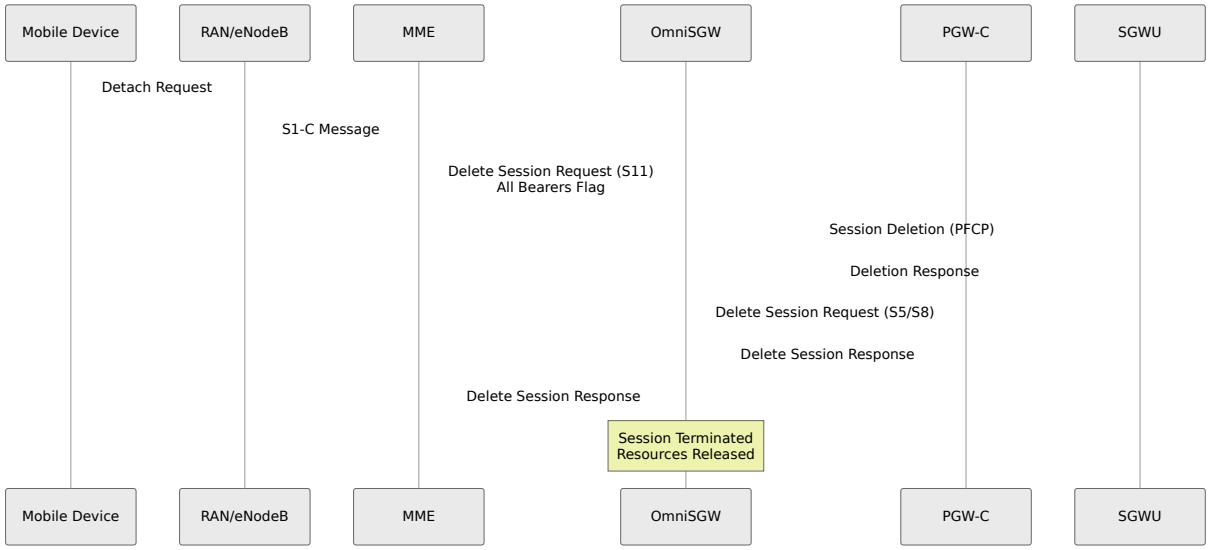
- MME updates UE location
- ULI/TAI sent to SGW via Modify Bearer
- SGW updates local UE context
- No session relocation needed

TAU with SGW Change:

- Old SGW receives Release Access Bearers from MME
- New SGW receives Create Session Request
- Data forwarding from old to new SGW
- After forwarding completes, old SGW releases session

Session Termination

Normal Session Termination



State Transitions:

```
[Session Active]
    ↓ (Delete Session Request)
[Releasing User Plane]
    ↓ (PFCP session deleted)
[Notifying PGW]
    ↓ (PGW session deleted)
[Session Terminated]
```

Network Operations

Message Flow Monitoring

Monitor S11 message activity in real-time:

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

# Output example:
#
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
```

Session Inspection

View active sessions and their S11 state:

```
Web UI → UE Sessions Page

For each session:
  - IMSI and GUTI
  - Current TAI (Tracking Area)
  - Local TEID (for S11)
  - Remote TEID (from MME)
  - Bearer list with QoS parameters
  - Associated PGW-C
```

Handover Monitoring

Track handover activity:

```
# Count modify bearer requests (indicate handovers)
curl -s http://127.0.0.40:42068/metrics | grep
modify_bearer_request_total

# Monitor for handover delays
# Check logs for "TAU with SGW change" messages
```

Troubleshooting

Session Establishment Failures

Problem: Create Session Request rejected

Diagnosis:

1. Check Web UI → UE Sessions for rejection reason
2. Verify metrics: `s11_inbound_errors_total`
3. Check logs for specific cause code

Common Causes & Solutions:

Cause	Reason	Solution
16	No resources available	Check SGW-U capacity, PFCP session count
25	Semantic error in IE	Verify bearer context in request
49	PGW unreachable	Check S5/S8 connectivity to PGW-C
65	APN not supported	Verify APN configuration

Message Routing Issues

Problem: "Message routed to unknown TEID"

Diagnosis:

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

# Verify TEID allocation
# Web UI → UE Sessions → search by IMSI
```

Common Causes:

- Session released but delayed message still arriving
- Duplicate Create Session with different TEID
- Message from different MME instance with same TEID

Handover Issues

Problem: Handover fails or data loss

Diagnosis:

1. Monitor Modify Bearer Request/Response in metrics
2. Check logs for "handover" or "TAU" messages
3. Inspect PFCP session state during handover

Solutions:

- Verify SGW-U active during handover window
- Check data forwarding rules installed
- Monitor Release Access Bearers timing

Performance Issues

Problem: High S11 message latency

Metrics to Check:

```
# Message processing duration
curl -s http://127.0.0.40:42068/metrics | grep
s11_inbound_duration_seconds

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

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

Optimization Steps:

1. Reduce unnecessary Modify Bearer operations
2. Monitor and optimize PFCP session establishment time
3. Scale horizontally with multiple SGW-C instances
4. Check CPU and memory usage during peak load

For comprehensive metrics information, Prometheus setup, and dashboard configuration, see the Monitoring & Metrics Guide.

Best Practices

Configuration

- **Port Binding:** Bind S11 to management network interface for security
- **Timeouts:** Set appropriate message timeouts based on network RTT
- **Retries:** Balance between reliability and network load

Operations

- **Session Limits:** Monitor vs. capacity to prevent overload
- **Peer Monitoring:** Track MME connectivity status
- **Error Tracking:** Alert on sustained S11 error rate increases
- **Graceful Shutdown:** Drain sessions before maintenance

Security

- **Network Isolation:** S11 should be on isolated network segment
- **Access Control:** Restrict S11 port to authorized MME IPs
- **Monitoring:** Alert on unexpected peer connections

Message Reference Summary

Message	Direction	Frequency	Priority
Create Session Request/Response	MME → SGW	Session create	High
Delete Session Request/Response	MME → SGW	Session end	High
Modify Bearer Request/Response	MME ↔ SGW	QoS changes, TAU	Medium
Create Bearer Request/Response	MME ↔ SGW	Bearer activation	Medium
Delete Bearer Request/Response	MME ↔ SGW	Bearer deactivation	Medium
Release Access Bearers Req/Resp	MME → SGW	Paging suspend	High
Modify Access Bearers Req/Resp	MME ↔ SGW	Mobility recovery	High
Downlink Data Notification/Ack	SGW → MME	Data paging	Medium
Echo Request/Response	MME ↔ SGW	Path monitoring	Low

S11 Interface - *MME to SGW-C Control Plane Signaling*

S5/S8 Interface Documentation

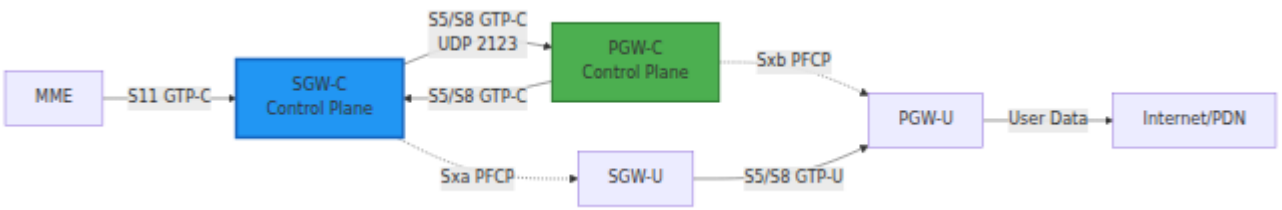
GTP-C Communication with PGW-C

OmniSGW by Omnitouch Network Services

Table of Contents

Overview

The **S5/S8 interface** connects OmniSGW to the PGW-C (Packet Gateway Control Plane) using the **GTP-C v2** (GPRS Tunnelling Protocol - Control plane) protocol. This interface handles PDN session management signaling between the gateways.



Key Features

- **GTP-C v2 Protocol** - Standards-compliant signaling
- **TEID-based Session Routing** - Tunnel endpoint identifiers for tracking
- **PDN Connectivity Management** - Create/modify/delete PDN connections
- **Bearer Management** - Default and dedicated bearer operations
- **Charging ID Exchange** - Coordinated billing across gateways
- **IP Address Allocation** - UE IP provisioning from PGW pools

---

Protocol Details

GTP-C Version 2

- **Protocol:** GTP-C v2 (3GPP TS 29.274)
- **Transport:** UDP
- **Port:** 2123 (standard)
- **Interface Type:** Control Plane
- **Direction:** Bidirectional request/response

TEID (Tunnel Endpoint Identifier)

Each PDN session has unique TEIDs for both directions:

- **SGW TEID** - Allocated by SGW-C for S5/S8 messages from PGW
- **PGW TEID** - Allocated by PGW-C for S5/S8 messages from SGW

```
Message Flow:
  SGW-C → PGW-C: Uses PGW-C's TEID in header
  PGW-C → SGW-C: Uses SGW-C's TEID in header
```

Charging ID

Charging ID is critical for billing coordination:

- **Generated by:** PGW-C during Create Session Response
- **Passed to:** SGW-C for CDR generation
- **Used for:** Correlating offline charges between SGW and PGW CDRs
- **Format:** 32-bit integer, unique per PDN connection

Configuration

Basic Configuration

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

    # Optional: Local IPv6 address
    local_ipv6_address: nil,

    # Optional: Override default port
    local_port: 2123,

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

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

Network Requirements

Firewall Rules:

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

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

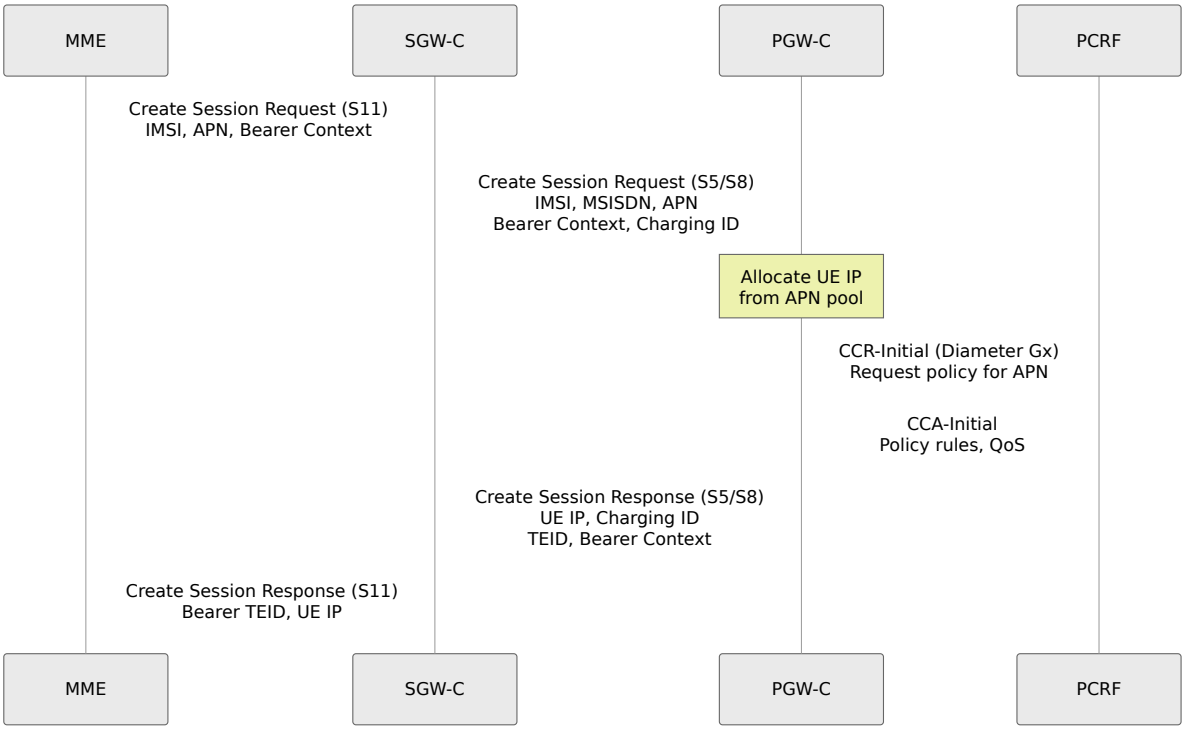
Routing:

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

Session Establishment

Initial PDN Connection Request

When MME requests a PDN connection via S11, SGW-C forwards to PGW-C via S5/S8.



Create Session Request (SGW-C → PGW-C)

Key Information Elements:

IE Name	Source	Description
IMSI	MME	Mobile subscriber identity
MSISDN	MME	Mobile phone number
MEI	MME	Mobile Equipment Identity
Bearer Context	MME	Bearer configuration (QCI, ARP)
APN	MME	Access Point Name (internet, ims, mms)
Serving Network	MME	PLMN code (MCC/MNC)
RAT Type	MME	Radio Access Technology (EUTRAN)
ULI	MME	User Location Information (TAI, ECGI)
Charging ID	SGW	SGW-generated charging reference

Create Session Response (PGW-C → SGW-C)

Key Information Elements:

IE Name	Source	Description
Cause	PGW	Success/failure indication
Bearer Context	PGW	Allocated bearer with TEID
PDN Address Allocation	PGW	Assigned UE IP address
APN Restriction	PGW	Policies for this APN
Charging ID	PGW	PGW-generated charging ID
TEID	PGW	Allocated for S5/S8 tunnel

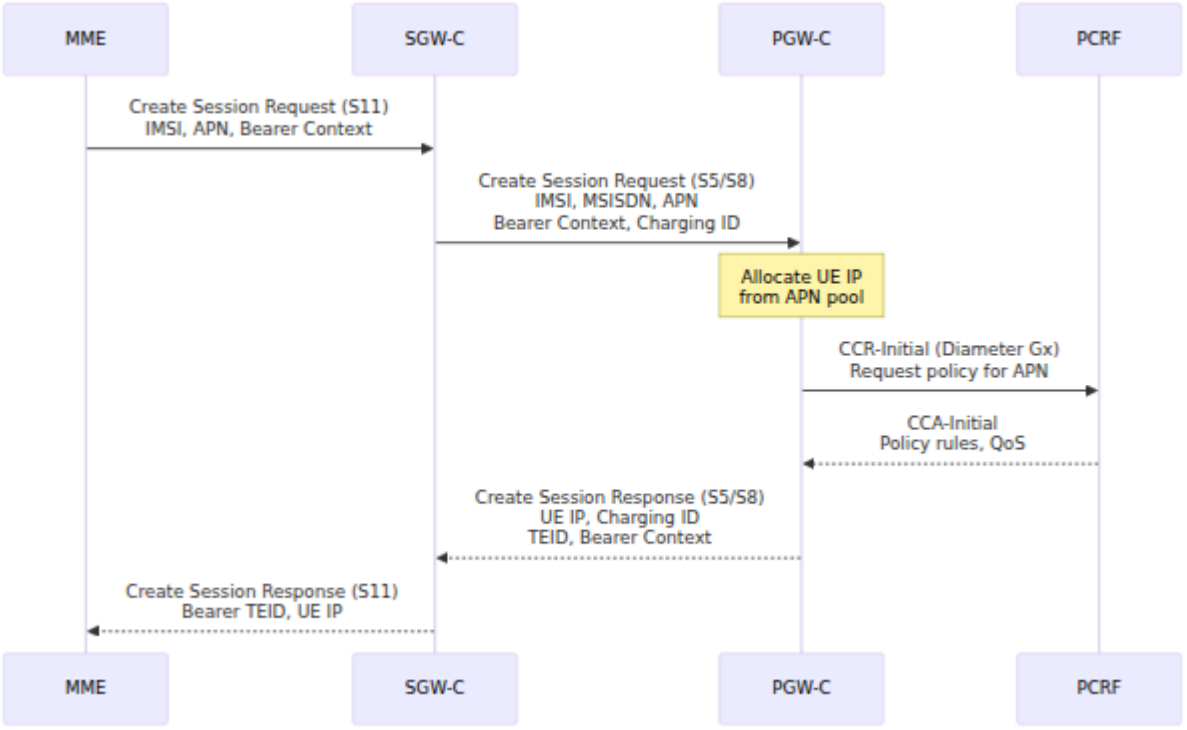
Response Codes

Cause Code	Description	Recovery
0	Request Accepted	Session established
16	No resources available	Reject to MME, user action
25	Semantics error in IE	Check message formatting
49	No matching rule	PGW-C policy mismatch
64	Context not found	Session already exists
65	Semantic error in response	PGW misconfiguration
72	Missing/incorrect mandatory IE	Message incomplete

Session Modification

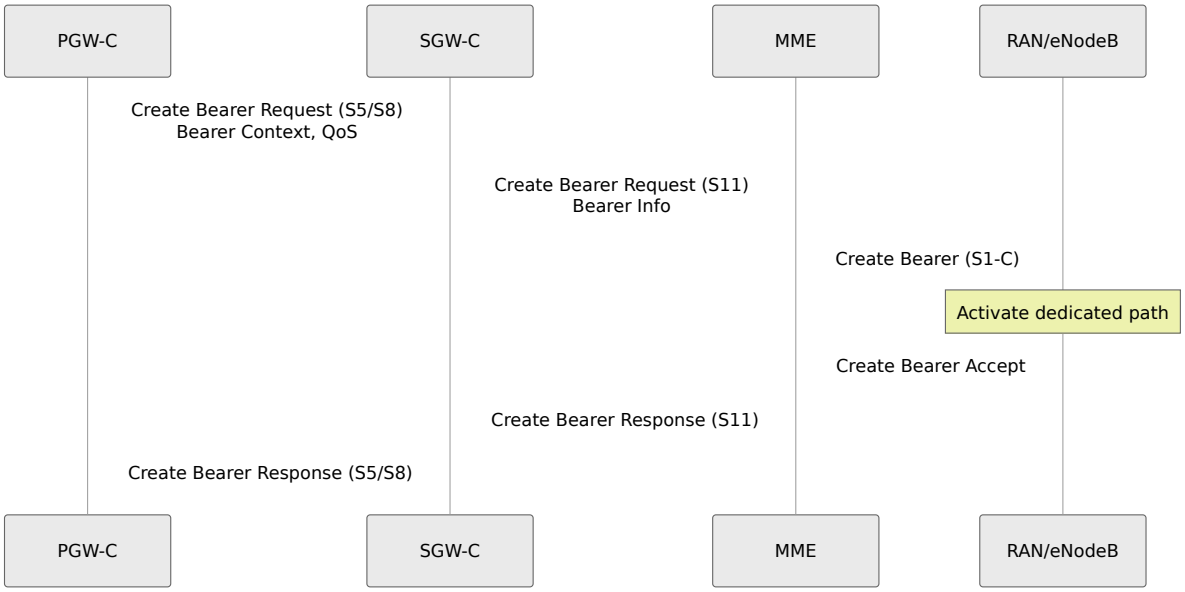
Bearer QoS Modification

When MME requests QoS changes via S11, SGW-C propagates to PGW-C via S5/S8.



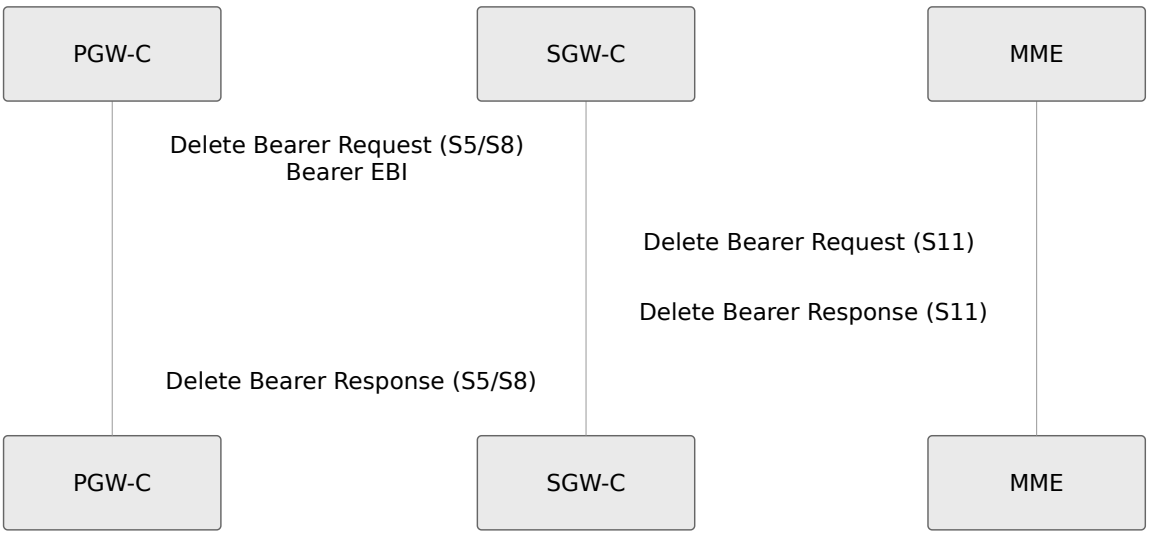
Bearer Creation (Dedicated Bearer)

PGW-C can request dedicated bearer activation via S5/S8:



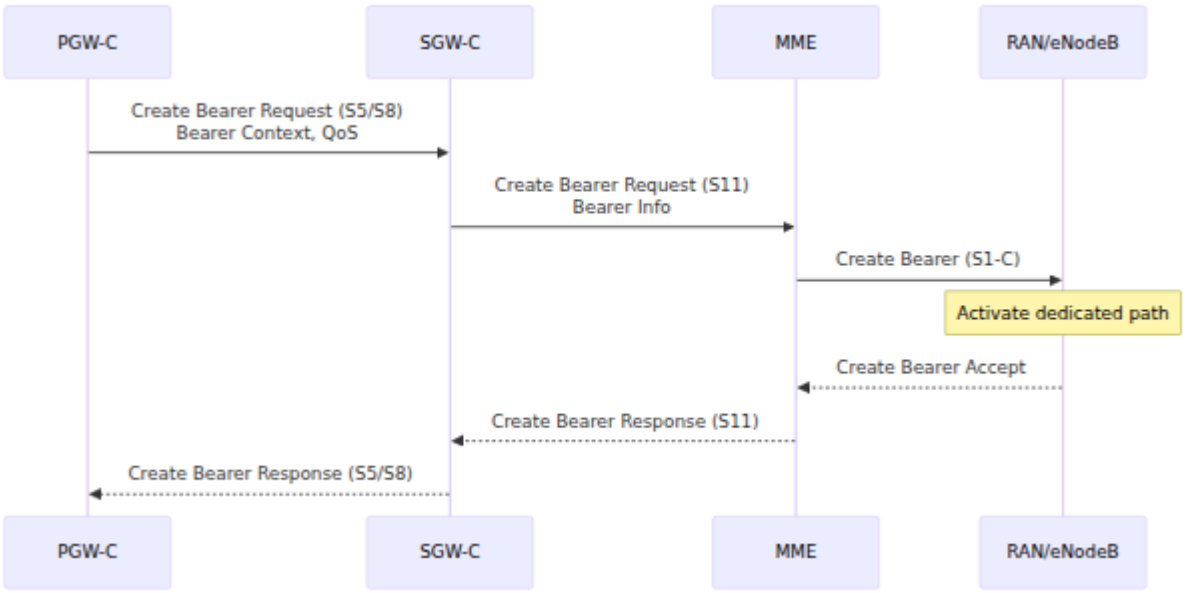
Bearer Deletion (Dedicated Bearer)

When a dedicated bearer is no longer needed:



Session Termination

Normal PDN Disconnection

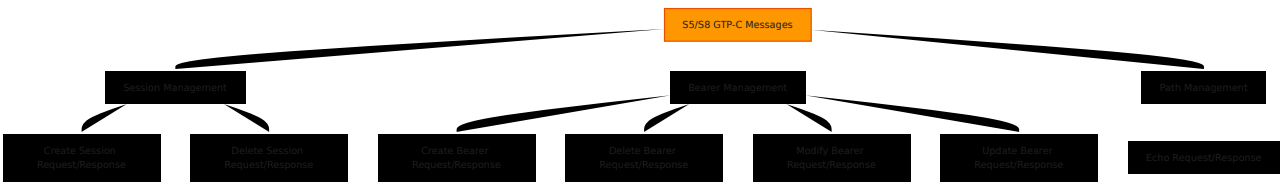


State Transitions:

```
[PDN Connected]
    ↓ (Delete Session Request from MME)
[Releasing PGW Session]
    ↓ (PGW Delete Response received)
[Releasing SGW Resources]
    ↓ (TEID released, CDR logged)
[PDN Disconnected]
```

Message Types

Summary of S5/S8 Messages



Message Details

Create Session Request/Response

- **Triggers:** Initial attach, PDN connectivity request
- **Frequency:** ~1 per PDN connection per UE
- **Direction:** Bidirectional

Delete Session Request/Response

- **Triggers:** Detach, PDN disconnection
- **Frequency:** ~1 per PDN connection termination
- **Direction:** Bidirectional

Modify Bearer Request/Response

- **Triggers:** QoS change, bearer modification
- **Frequency:** Variable (0 to many per session)
- **Direction:** Bidirectional

Create/Delete Bearer Request/Response

- **Triggers:** Dedicated bearer activation/deactivation
- **Frequency:** Variable (0 to many per session)
- **Direction:** Bidirectional

Echo Request/Response

- **Triggers:** Path/peer monitoring
- **Frequency:** Periodic (recommended 1/minute minimum)
- **Direction:** Bidirectional

Network Operations

Peer Monitoring

Monitor PGW-C connectivity:

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

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

# Expected: steady stream of Create/Delete/Modify messages
```

PDN Session Verification

Inspect active PDN connections:

```
Web UI → UE Sessions page
└── For each UE session:
    ├── Associated PGW-C peer
    ├── Charging ID (from PGW)
    ├── UE IP address (from PGW)
    ├── Bearer list with QoS
    └── S5/S8 TEID pair
```

Message Flow Inspection

Track S5/S8 message activity:

```
# Count Create Session operations
curl -s http://127.0.0.40:42068/metrics | grep
create_session_request_total

# Monitor bearer modifications
curl -s http://127.0.0.40:42068/metrics | grep modify_bearer

# Check error rate
curl -s http://127.0.0.40:42068/metrics | grep
s5s8_inbound_errors_total
```

PGW Selection Strategy

If multiple PGW-C peers configured:

```
Selection Logic:
├── Load-balanced: Round-robin across peers
├── Sticky: Same APN always uses same PGW
├── Active-Standby: Failover on peer unavailable
└── Custom: Application-specific logic
```

Monitor distribution:

```
# Sessions per PGW peer
curl -s http://127.0.0.40:42068/metrics | grep session_by_pgw_peer
```

Troubleshooting

Session Establishment Failures

Problem: "Create Session Request rejected by PGW"

Diagnosis:

```
# Check cause code
curl -s http://127.0.0.40:42068/metrics | grep
create_session_response_cause

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

Common Causes & Solutions:

Cause	Reason	Solution
16	No resources	Check PGW capacity, IP pool exhaustion
25	Semantic error	Verify bearer context matches PGW expectations
49	No matching rule	Check APN configuration at PGW
72	Missing IE	Verify MME is sending required fields

Bearer Operation Failures

Problem: "Modify Bearer Request fails"

Diagnosis:

1. Check metrics for modify_bearer error rate
2. Inspect QoS parameters for validity
3. Verify PGW is reachable

Solutions:

- Reduce QoS modification frequency
- Verify QoS values within PGW policy
- Check PGW for PCRF/policy issues

Message Timeout Issues

Problem: "S5/S8 messages timing out"

Metrics:

```
# Message latency
curl -s http://127.0.0.40:42068/metrics | grep
s5s8_inbound_duration_seconds

# Timeout count
curl -s http://127.0.0.40:42068/metrics | grep s5s8_timeout_total
```

Solutions:

- Increase message_timeout_ms if network RTT is high
- Check network congestion
- Verify PGW CPU/memory availability
- Monitor for packet loss

Charging ID Mismatch

Problem: "Charging ID mismatch in CDRs"

Diagnosis:

- Verify PGW is returning valid Charging ID
- Check CDR logs for missing Charging ID
- Compare SGW and PGW CDRs

Solution:

- Ensure PGW sends Charging ID in all responses
- Handle missing Charging ID gracefully in CDR logging

For detailed metrics reference and Prometheus dashboard setup, see the Monitoring & Metrics Guide.

Best Practices

Configuration

- **PGW Redundancy:** Configure multiple PGW-C peers for failover
- **Load Distribution:** Use round-robin for balanced load
- **Timeouts:** Set appropriately based on WAN RTT (typical: 5-10 seconds)
- **Retries:** 2-3 retries with exponential backoff

Operations

- **Peer Health:** Monitor echo response times
- **APN Routing:** Match SGW APN config to PGW APNs
- **Error Tracking:** Alert on sustained S5/S8 error rate
- **Capacity Planning:** Monitor IP pool usage at PGW

Session Management

- **Session Limits:** Track concurrent sessions vs. PGW capacity
- **Bearer Count:** Monitor default + dedicated bearer distribution
- **QoS Validation:** Verify QoS parameters accepted by PGW
- **Charging:** Verify Charging ID received and logged

Integration with Other Interfaces

S11 ↔ S5/S8 Coordination

```
S11 Message Flow (from MME)
    ↓
SGW-C Session Processing
    ↓
S5/S8 Message (to PGW-C)
    ↓
Wait for Response
    ↓
S11 Response (back to MME)
```

S5/S8 ↔ Sxa Coordination

```
S5/S8 Create Session Response (from PGW)
    ↓
Extract Bearer/QoS Info
    ↓
Sxa Session Establishment (to SGW-U)
    ↓
Wait for User Plane Ready
    ↓
Complete S11 Create Session Response
```

Session Management Guide

UE Session Lifecycle and Operations

OmniSGW by Omnitouch Network Services

Table of Contents

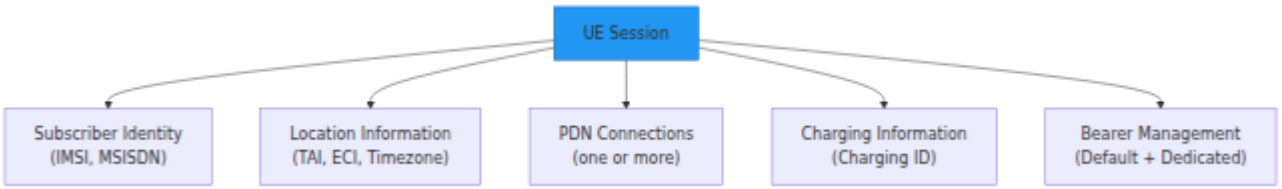
Overview

A UE Session represents an active mobile device connected to the network. SGW-C maintains the session context and coordinates between:

- **MME** - Mobility Management Entity (via S11)
- **PGW-C** - Packet Gateway Control Plane (via S5/S8)
- **SGW-U** - User Plane forwarding (via Sxa)

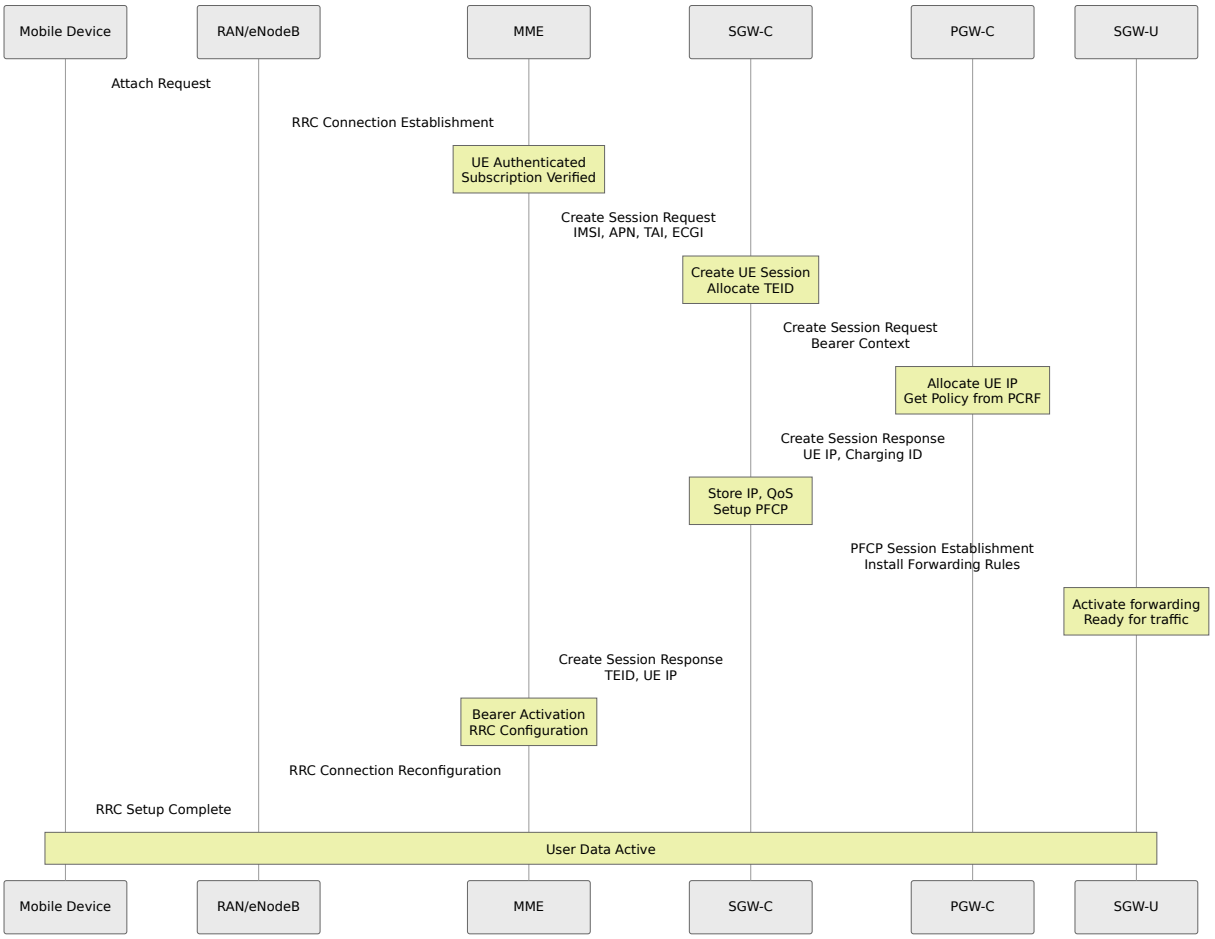
Each session has a unique IMSI (subscriber identity) and may contain one or more PDN connections.

Session Responsibilities

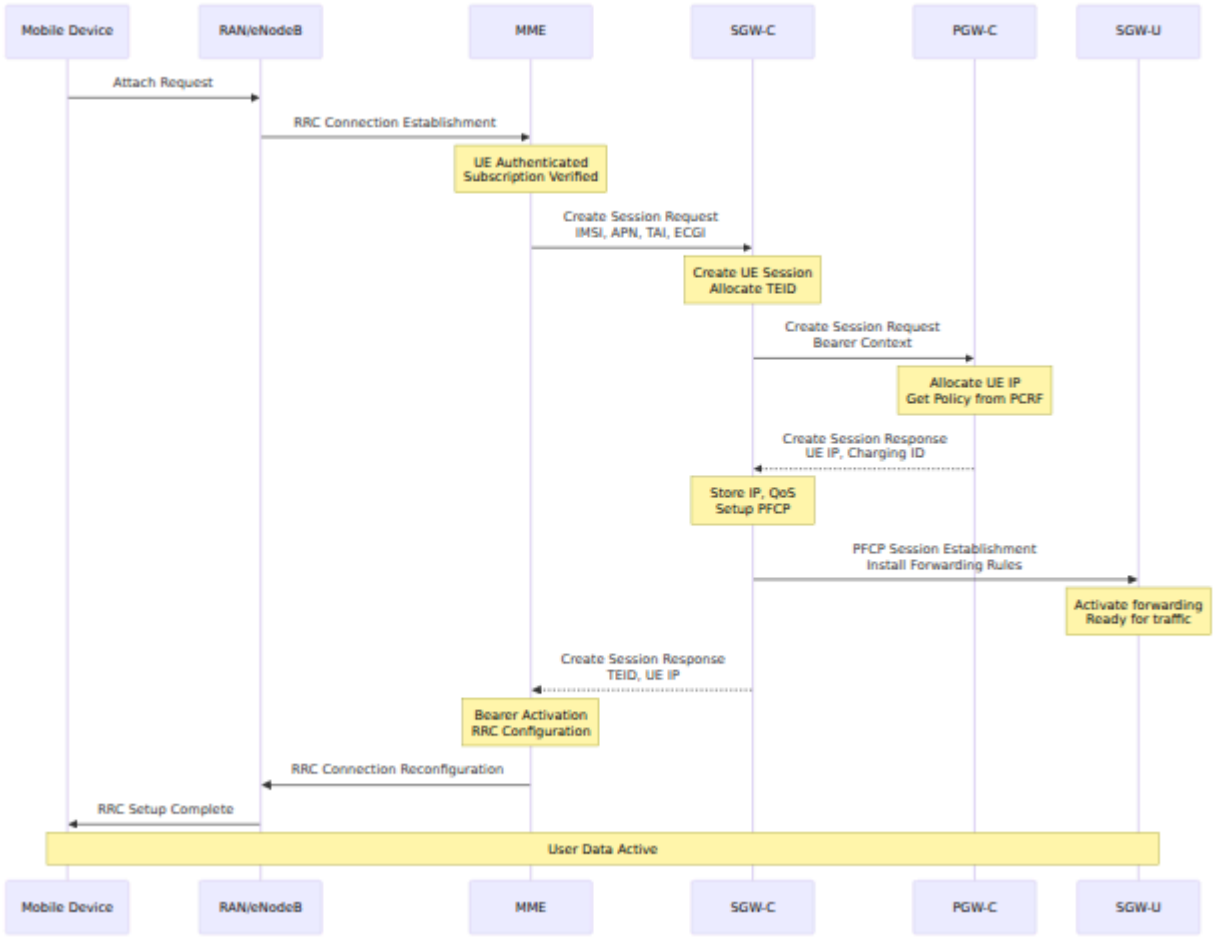


Session Lifecycle

Session Creation (UE Attach)



Session Termination (UE Detach)



Session States

UE Session State Machine

```
[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]
```

Key State Variables

```
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
```

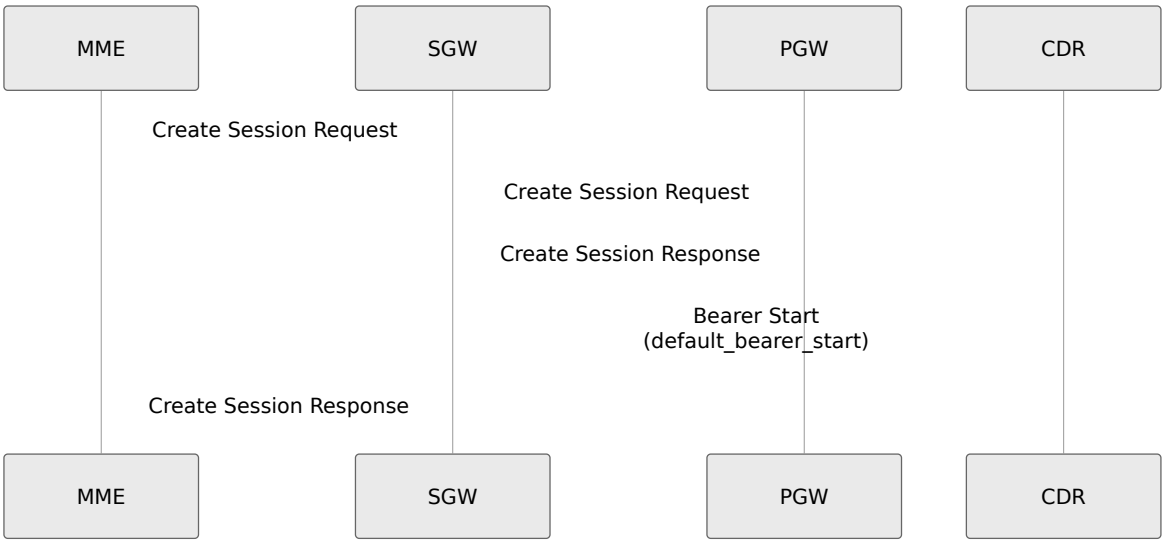
Bearer Operations

Default Bearer

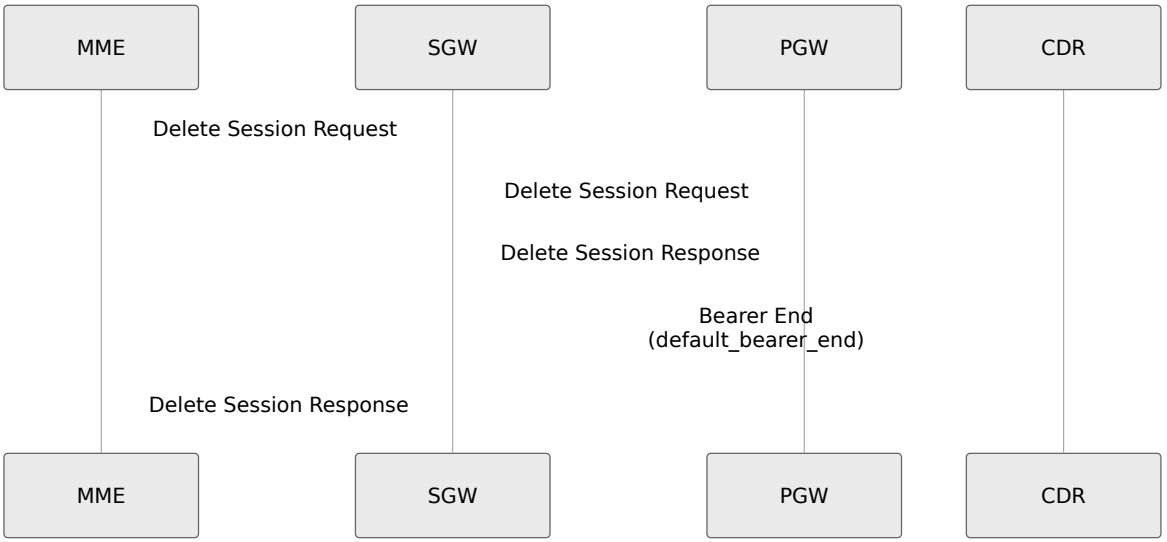
The default bearer is created with every PDN connection:

- **QoS:** Typically QCI 9 (best effort)
- **Lifetime:** Same as PDN connection
- **Traffic:** Carries all traffic not matched by dedicated bearers
- **Mandatory:** Every PDN connection must have default bearer

Bearer Start Event:



Bearer End Event:



Dedicated Bearers

Dedicated bearers provide premium QoS for specific services:

- **Activation:** Requested by application or network policy
- **QoS:** QCI 1-8 (various service types)
- **Lifetime:** Can be shorter than PDN connection
- **Optional:** Zero or more per PDN connection

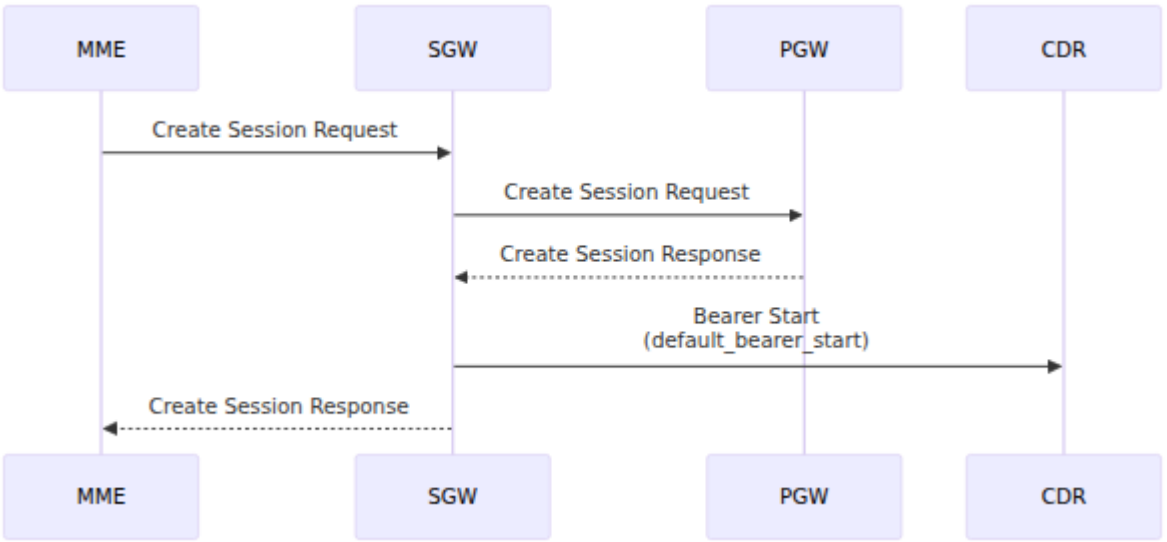
Dedicated Bearer Activation:

```
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
```

Dedicated Bearer Deactivation:

```
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
```

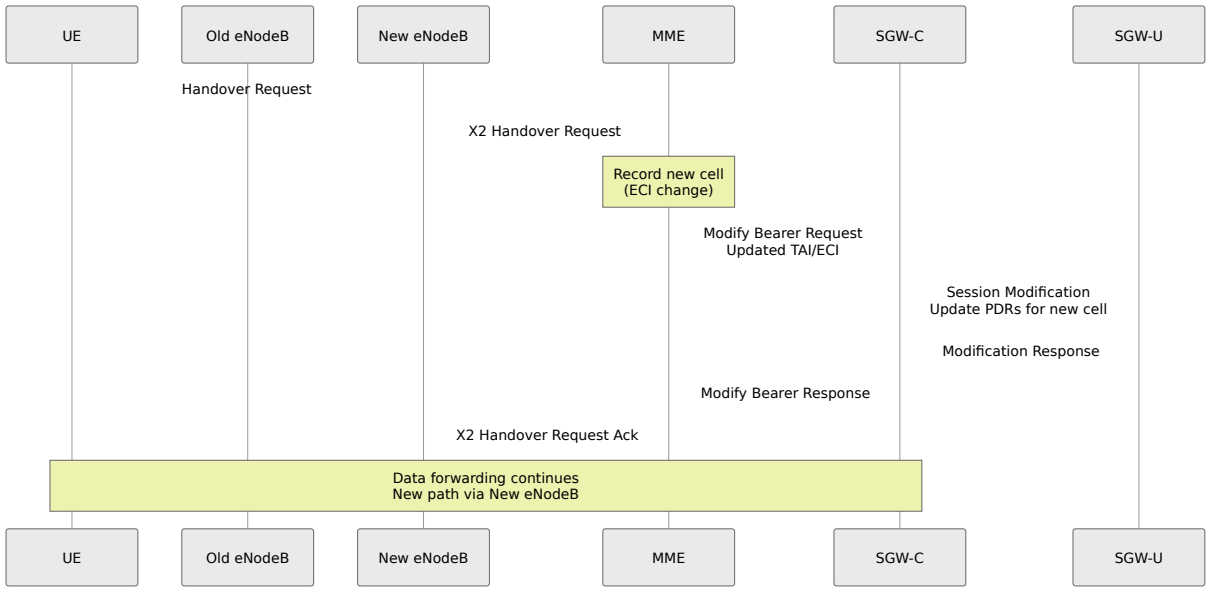
Bearer QoS Classes



Mobility Handling

Intra-MME Handover (No SGW Change)

Scenario: UE moves between cells in same MME area

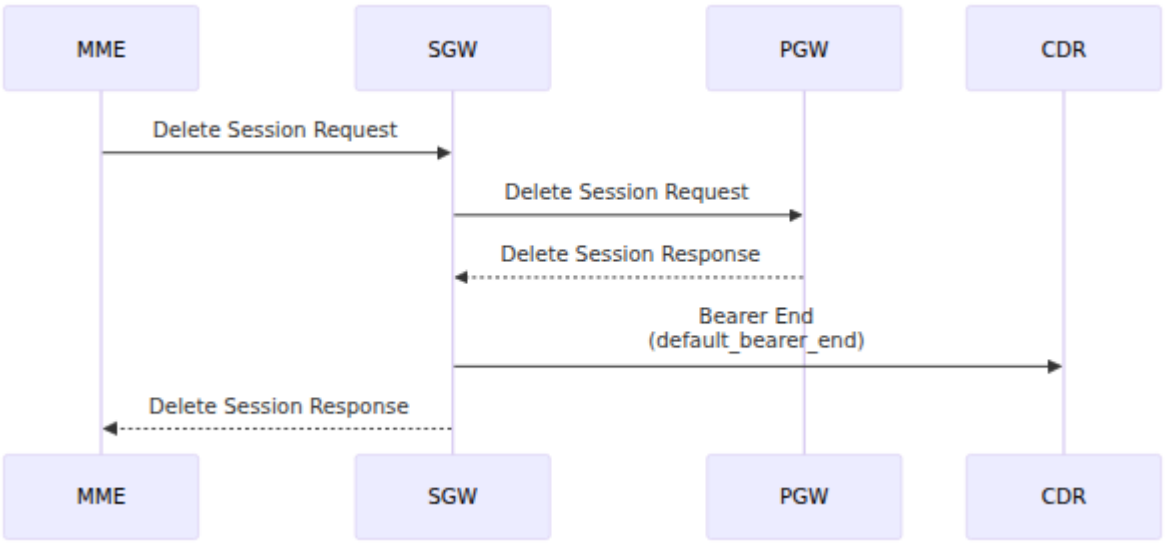


Session Impact:

- Session remains active
- TEID remains same
- Location updated in session
- CDR continues with same Charging ID

Inter-MME Handover (With SGW Change)

Scenario: UE moves to different MME, new SGW required



Session Impact:

- Old session terminates, CDR logged with "handover" indication
- New session created with same Charging ID
- Data forwarding maintains connectivity
- User plane rerouted through new SGW-U

Tracking Area Update (TAU)

TAU without SGW Change:

```
UE updates location
    ↓
MME sends TAU Accept
    ↓
MME updates SGW with new location
    ↓
SGW modifies session (TAI, ECI)
    ↓
No service interruption
```

TAU with SGW Change:

- Similar to inter-MME handover
- Session migrated to new SGW
- CDR coordination across old and new SGW

Handover Procedures

Preparation Phase

Before handover completes:

1. **New SGW-U Selection** - Choose forwarding path
2. **PDR Installation** - Install new forwarding rules
3. **Buffer Activation** - Enable buffering for in-flight packets
4. **Signaling Coordination** - S11/S5/S8 message exchange

Data Forwarding Phase

During handover transition:

- **Buffering in Old SGW-U** - Packets held temporarily
- **Buffering in New SGW-U** - Ready to receive
- **GTP Tunneling** - Data forwarded from old to new path
- **Packet Sequencing** - Maintain order

Completion Phase

After handover completes:

1. **Buffer Flush** - Buffered packets released
2. **Path Cutover** - Traffic switches to new path
3. **Old Path Cleanup** - Release old forwarding rules
4. **Session Updates** - Location and TEID updated

Operational Procedures

Session Inspection

Monitor active sessions via Web UI:

```
1. Open http://<sgw-ip>:<port>/ue_sessions
2. View all active UE sessions
3. Search by IMSI, GUTI, or phone number
4. Click session for detailed view:
   - Location (TAI, ECI)
   - Active bearers and QoS
   - PGW-C association
   - TEID pair information
   - Charging ID
```

The UE Sessions overview displays all active sessions with key identifiers:

Click any session to view comprehensive details including TEIDs, location, bearers, and PDN connections:

See OPERATIONS Guide for Web UI navigation and access instructions.

Metrics Monitoring

Track session metrics:

```
# Count active sessions
curl -s http://10.0.0.40:42068/metrics | grep active_ue_sessions

# Count active bearers
curl -s http://10.0.0.40:42068/metrics | grep active_bearers

# Monitor by APN
curl -s http://10.0.0.40:42068/metrics | grep sessions_by_apn

# Monitor message rate
curl -s http://10.0.0.40:42068/metrics | grep
s11_inbound_messages_total
```

For a complete reference of available metrics, Prometheus dashboards, and alerting setup, see the Monitoring & Metrics Guide.

Graceful Session Termination

To cleanly terminate a session:

1. **Trigger via API:** Request session deletion

2. **Wait for completion:** Monitor session state
3. **Verify cleanup:** Check metrics
4. **Review CDR:** Confirm final record

Session Limits

Monitor against capacity:

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

# Alert at 80% of licensed capacity
# Gracefully handle when reaching limits
```

Troubleshooting

Session Won't Establish

Symptom: Create Session Request fails

Diagnosis:

1. Check metrics for cause code
2. Inspect S11 error logs
3. Verify PGW connectivity
4. Check Charging ID availability

Common Causes:

Cause	Solution
PGW unreachable	Verify S5/S8 network connectivity
No IP available	Check PGW IP pool status
APN not configured	Verify APN at PGW
No SGW-U available	Ensure SGW-U association active
Policy mismatch	Check PGW policy configuration

Session Drops Unexpectedly

Symptom: Active session terminates without Delete Request

Diagnosis:

1. Check logs for error messages
2. Monitor SGW-U heartbeat status
3. Check PGW connectivity
4. Review metrics for error spikes

Common Causes:

Cause	Solution
SGW-U crash	Restart SGW-U, monitor logs
Network disconnect	Check interface status
PGW failure	Failover to backup PGW
Message timeout	Increase timeout, check RTT

Handover Failures

Symptom: Handover loses packets or fails entirely

Diagnosis:

1. Monitor Modify Bearer messages
2. Check PFCP rule updates
3. Verify data forwarding setup
4. Check bearer buffering

Common Causes:

Cause	Solution
Buffering disabled	Enable BAR in PFCP rules
PDR not updated	Verify PFCP modifications sent
Forwarding path broken	Check routing to new SGW-U
Timing too tight	Increase handover timeout

High Message Latency

Symptom: S11/S5S8 message processing slow

Diagnosis:

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

# Check queue depth
curl -s http://10.0.0.40:42068/metrics | \
  grep queue_depth

# Check system load
top -n1 | head -1
```

Solutions:

1. Increase message timeout if network RTT high
2. Load balance across multiple SGW-C instances
3. Monitor and reduce message rate
4. Check for stuck sessions

CDR Generation Issues

Symptom: CDRs missing or incomplete

Diagnosis:

1. Check CDR directory exists
2. Verify write permissions
3. Check disk space
4. Review for generation errors in logs

Solutions:

```
# Monitor CDR generation
tail -f /var/log/sgw_c/cdrs/*

# Check file permissions
ls -la /var/log/sgw_c/cdrs/

# Ensure directory writable
chmod 755 /var/log/sgw_c/cdrs/
```

See CDR Format Guide for complete CDR field reference and integration details.

Best Practices

Session Management

- **Monitor Capacity:** Track vs. licensed limits
- **Alert on Thresholds:** Trigger at 70-80% capacity
- **Graceful Degradation:** Drain before maintenance
- **Health Checks:** Monitor peer connectivity

Handover Operations

- **Fast Failover:** Configure aggressive heartbeat
- **Graceful Handover:** Ensure buffering enabled
- **Path Redundancy:** Multiple SGW-U peers
- **Testing:** Regular handover simulations

Charging Coordination

- **Verify Charging ID:** Ensure PGW allocation
- **CDR Validation:** Compare SGW and PGW CDRs
- **Event Correlation:** Link CDR events across gateways
- **Archival:** Long-term CDR storage

Sxa Interface Documentation

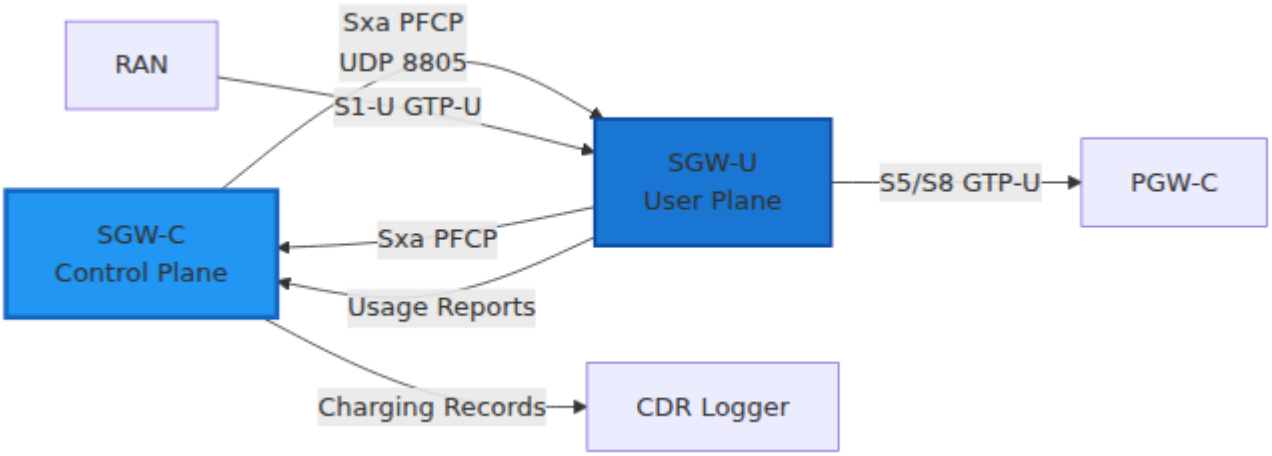
PFCP Communication with SGW-U

OmniSGW by Omnitouch Network Services

Table of Contents

Overview

The **Sxa interface** connects OmniSGW to the SGW-U (Serving Gateway User Plane) using the **PFCP** (Packet Forwarding Control Protocol) protocol. This interface controls the forwarding of user plane packets, QoS enforcement, and usage reporting.



Key Features

- **PFCP v1.0** - Standards-compliant packet forwarding control
- **SEID-based Session Tracking** - Session Endpoint Identifiers for association
- **Packet Detection Rules** - Flexible packet matching for uplink/downlink
- **Forwarding Action Rules** - Control packet routing and encapsulation
- **QoS Enforcement** - Per-bearer bitrate limiting and prioritization
- **Usage Metering** - Volume tracking for charging and analytics
- **Buffering Control** - Automatic buffering during mobility events

---

Protocol Details

PFCP Version 1.0

- **Protocol:** PFCP v1.0 (3GPP TS 29.244)
- **Transport:** UDP
- **Port:** 8805 (standard)
- **Interface Type:** Control Plane
- **Association Model:** CP and UP form persistent association

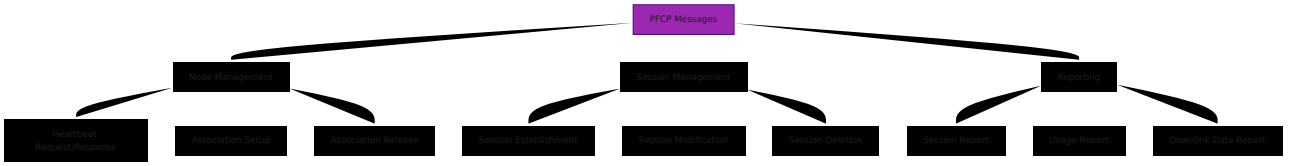
SEID (Session Endpoint Identifier)

Each session has unique SEIDs for tracking:

- **CP SEID** - Allocated by SGW-C, used in Uplink messages to SGW-U
- **UP SEID** - Allocated by SGW-U, used in Downlink messages to SGW-C

```
Message Routing:
  SGW-C → SGW-U: Uses SGW-U's UP SEID
  SGW-U → SGW-C: Uses SGW-C's CP SEID
```

Message Types Overview



Configuration

Basic Configuration

```
# config/runtime.exs
config :sgw_c,
  sxa: %{
    # Local IP address for Sxa interface
    local_ip_address: "10.0.0.20",

    # Optional: Override default port
    local_port: 8805,

    # SGW-U peers to connect to
    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"
      }
    ],

    # Association heartbeat interval (seconds)
    heartbeat_interval_s: 20,

    # Session establishment timeout (milliseconds)
    session_timeout_ms: 5000,

    # Maximum retries for session operations
    max_retries: 3
  }
```

Network Requirements

Firewall Rules:

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

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

Routing:

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

Network Testing:

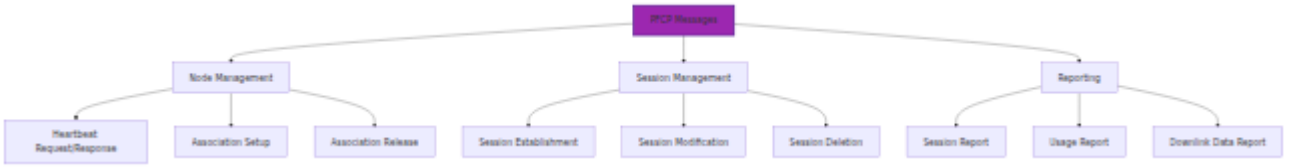
```
# Test PFCP connectivity
# Check logs for "Association Setup Complete" message

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

PFCP Association

Association Lifecycle

Before any sessions can be established, SGW-C and SGW-U must form a PFCP association.



Association State Machine

```
[Disconnected]
    ↓ (Setup Request)
[Associating]
    ↓ (Setup Response OK)
[Associated]
    ↓ (Session Creates)
[Sessions Active]
    ↓ (Heartbeat Failure)
[Re-associating]
    ↓ (Setup OK or Timeout)
[Associated or Disconnected]
```

Recovery Handling

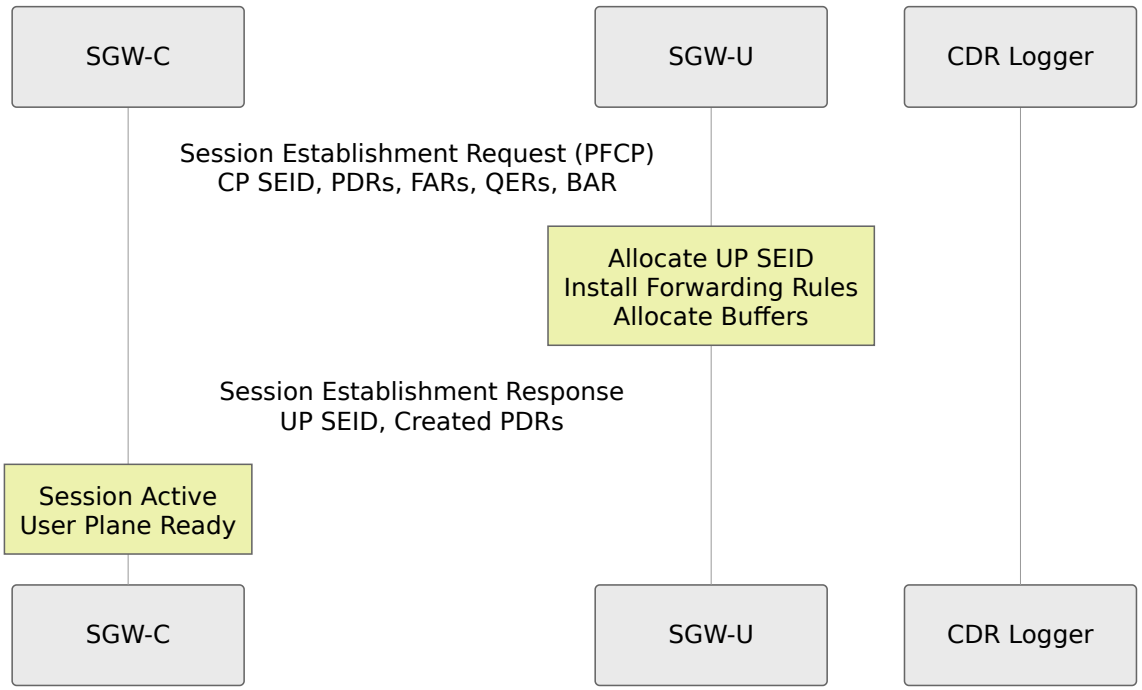
If a PFCP association is lost and recovered:

1. **Recovery Detection:**
    - Heartbeat timeout triggers recovery
    - New Association Setup sent
    - Recovery Timestamp checked
2. **Session Recovery:**
    - Sessions may or may not be recoverable
    - Query SGW-U for existing sessions
    - Re-establish lost sessions if needed
3. **Data Forwarding:**
    - User plane buffering in SGW-U during recovery
    - PDRs remain active until explicitly deleted
    - Minimize packet loss during failover

Session Management

Session Establishment

Trigger: Create Session Request from MME (received on S11)



Information Elements Sent:

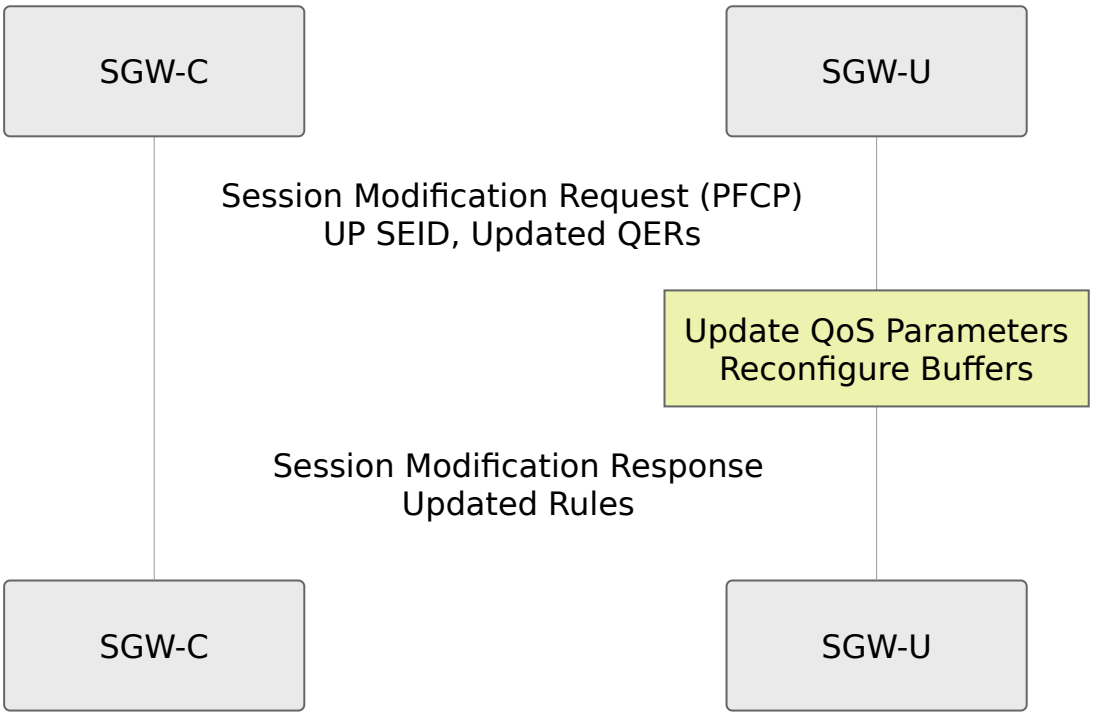
Element	Description
CP SEID	Allocated by SGW-C for this session
PDRs	Packet Detection Rules (see below)
FARs	Forwarding Action Rules
QERs	QoS Enforcement Rules
BAR	Buffering Action Rule for mobility
Create PDR	Rule identifiers for response

Session State:

```
[No Session]
    ↓ (Establishment Request)
[Establishing]
    ↓ (Establishment Response)
[Session Active]
```

Session Modification

Trigger: Modify Bearer Request from MME (QoS change, handover)

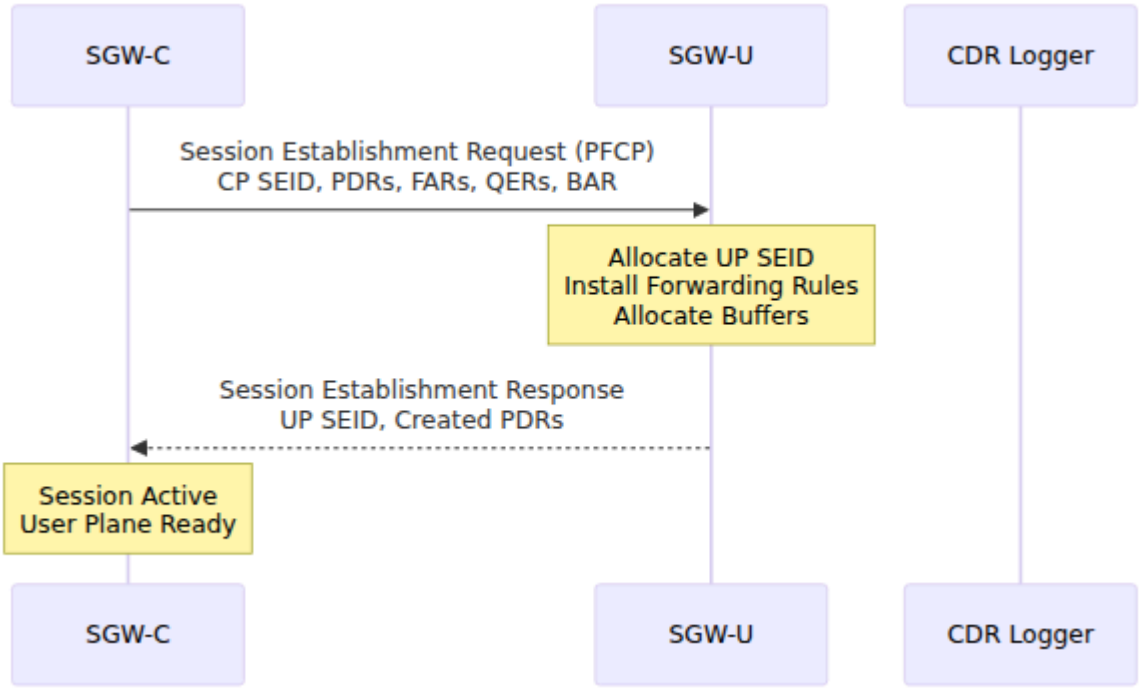


Common Modifications:

Modification	Elements Updated	Reason
QoS Change	QERs	Bearer upgrade/downgrade
Handover	PDRs, FARs	eNodeB change, SGW-U selection
Bearer Add	New PDR, FAR, QER	Dedicated bearer activation
Bearer Delete	Remove PDR, FAR, QER	Bearer deactivation

Session Deletion

Trigger: Delete Session Request from MME (detach)



Session State:

```
[Session Active]
    ↓ (Deletion Request)
[Deleting]
    ↓ (Deletion Response)
[Session Terminated]
```

PFCP Rules

PDR (Packet Detection Rule)

Matches incoming packets to identify traffic flows.

Packet Detection Criteria:

Criterion	Description	Example
Source Interface	Where packet arrives from	Access (S1-U), Core (S5/S8)
Source IP Address	UE IP address (for Access)	10.45.0.50
Destination IP Address	External network IP (for Core)	8.8.8.8
Protocol Type	IP protocol number	TCP (6), UDP (17)
Source Port	Port matching	1024-65535
Destination Port	Port matching	80 (HTTP), 443 (HTTPS)
TEID	GTP-U tunnel identifier	For downlink packets

PDR Structure:

```
PDR Structure:
├── PDR ID (unique within session)
├── Precedence (priority for overlapping rules)
├── Packet Detection Criteria
│   ├── Source Interface
│   ├── Network Instance (APN)
│   └── UE IP Address / Destination IP
├── FAR ID (which forwarding rule to apply)
├── QER ID (which QoS rule to apply)
└── Usage Report Trigger
```

Example Use Case - Default Bearer:

- Detects: All packets from/to UE IP
- Action: Forward through PDN (PGW-U direction)
- QoS: Applied per bearer

Example Use Case - Dedicated Bearer:

- Detects: Packets matching specific flow (port range, protocol)
- Action: Forward on dedicated path
- QoS: Premium rates (GBR)

FAR (Forwarding Action Rule)

Specifies how to handle matched packets.

Forwarding Actions:

Action	Description	Use Case
Forward	Send packet to destination network	Normal forwarding
Buffer	Store packet temporarily	During mobility/paging
Discard	Drop packet	Policy enforcement, firewall
Duplicate	Send packet to multiple destinations	Lawful intercept

Encapsulation Options:

- **GTP-U** - Add GTP-U tunnel header (S1-U, S5/S8)
- **Ethernet** - Add Ethernet header (for direct interconnect)
- **IPv4** - Plain IPv4 forwarding (for internet breakout)
- **IPv6** - Plain IPv6 forwarding

Example - UE to Internet:

```
PDR Match: Source Interface = Access, UE IP = 10.45.0.50
FAR Action:
  - Forward = Yes
  - Outer Header Encap = None (direct internet)
  - Forwarding Parameters = Internet gateway
```

QER (QoS Enforcement Rule)

Enforces bitrate limits per bearer.

QoS Parameters:

Parameter	Type	Description
QCI	Integer	QoS Class Identifier (1-9)
MBR (Max Bitrate)	Bitrate	Maximum allowed rate
GBR (Guaranteed Bitrate)	Bitrate	Minimum guaranteed rate
ARP	Integer	Allocation & Retention Priority (1-15)

QoS Classes (QCI):

QCI	Service Type	Bitrate Examples
1	Voice (GBR)	MBR: 64 kbps
2	Video Call (GBR)	MBR: 256 kbps
3	Real-time Gaming (GBR)	MBR: 50 kbps
4	Non-GBR	GBR: 128 kbps, MBR: 256 kbps
5	IMS Signaling	GBR: 100 kbps, MBR: 256 kbps
6	Video Streaming	MBR: 10 Mbps
7	Voice with Video (GBR)	GBR: 64 kbps, MBR: 384 kbps
8	Web Browsing	MBR: 5 Mbps
9	Email	MBR: 3 Mbps

Example - Default Bearer (QCI 9):

```
QCI: 9 (Best Effort)
MBR: 100 Mbps (site-dependent)
GBR: None (non-GBR)
ARP: 15 (lowest priority)
```

Example - Dedicated Voice Bearer (QCI 1):

```
QCI: 1 (Voice)
MBR: 128 kbps (uplink + downlink)
GBR: 64 kbps (guaranteed)
ARP: 1 (highest priority)
```

BAR (Buffering Action Rule)

Controls packet buffering during mobility events.

Buffering Scenarios:

1. **Handover Buffering:**
    - UE hands off between eNodeBs
    - Packets buffered while moving
    - Released when handover completes
2. **Paging Buffering:**
    - UE in idle (suspended bearers)
    - Downlink data arrives
    - Buffered until UE reactivates
3. **SGW Relocation:**
    - During inter-MME handover with SGW change
    - Old SGW buffers and forwards to new SGW
    - Ordered delivery maintained

BAR Configuration:

```
BAR Settings:
├── Buffer Timeout: How long to hold packets
├── Packet Count Threshold: Max buffered packets
└── Downlink Data Report Trigger
    └── Send notification to CP when data arrives
```

Usage Reporting

Usage Report Messages

SGW-U sends usage reports to SGW-C for charging and analytics.

Usage Report Triggers

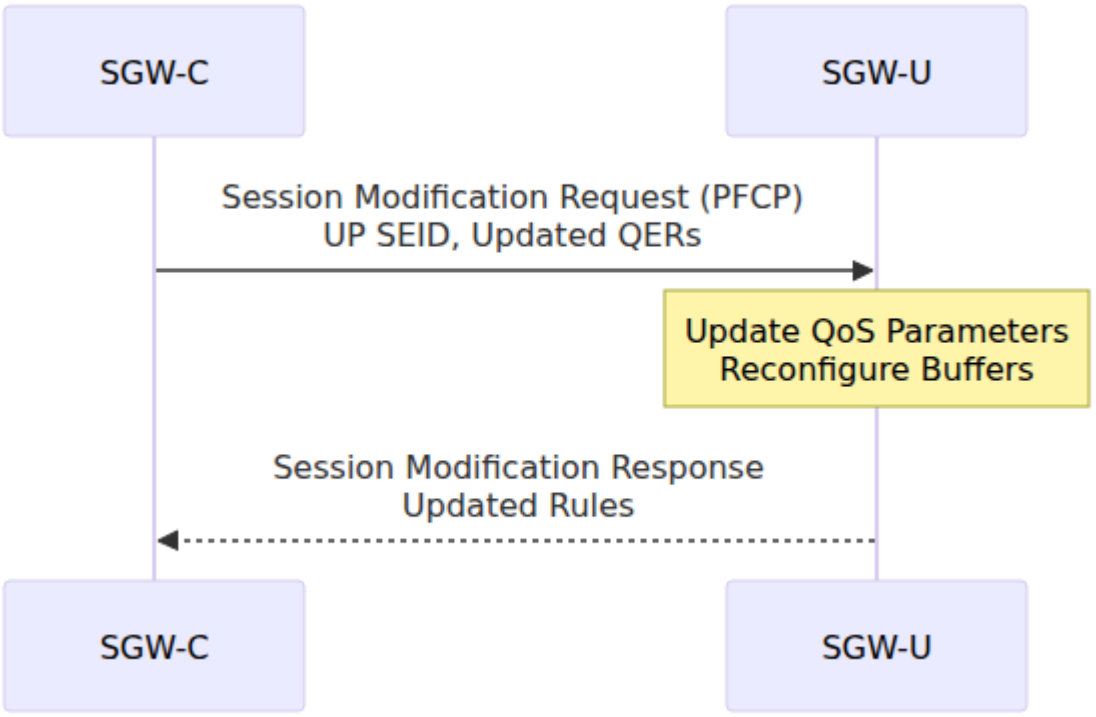
Reports are sent when:

Trigger	Condition
Time Period	Periodic report every N seconds
Volume Threshold	After N octets forwarded
Duration Threshold	After N seconds of forwarding
Session End	When session is deleted
Modification	When rules are updated
Immediate Report	Requested in modification message

Usage Report Fields

```
Usage Report:
├── Usage Report Trigger: What caused this report
├── UR-SEQN: Sequence number for ordering
├── Usage Information Per Bearer:
│   ├── EBI: Bearer identifier
│   ├── Volume Measurement
│   │   ├── UL: Uplink octets
│   │   ├── DL: Downlink octets
│   │   ├── Total: Total octets
│   │   └── Packets: Total packets
│   ├── Duration Measurement: Seconds active
│   └── Time of First/Last Packet: Timestamps
└── Query UR: Request immediate report
```

CDR Generation Flow



Network Operations

PFCP Association Monitoring

Monitor active PFCP associations:

```
# Check association status
curl -s http://127.0.0.40:42068/metrics | grep pfcp_association

# Expected output:
# pfcp_association_status{peer_ip="10.0.0.30"} 1  (associated)
# pfcp_association_status{peer_ip="10.0.0.31"} 1  (associated)

# Web UI → SGW-U Status page
# Shows all peers with "Associated" status and recovery info
```

Session Metrics

Monitor active PFCP sessions:

```
# Count active sessions
curl -s http://127.0.0.40:42068/metrics | grep seid_registry_count

# Monitor per-SGW-U distribution
curl -s http://127.0.0.40:42068/metrics | grep seid_by_peer

# Usage rate (octets/sec)
curl -s http://127.0.0.40:42068/metrics | grep usage_octets_rate
```

Message Flow Monitoring

Track PFCP message activity:

```
# Monitor all PFCP messages
watch -n 1 'curl -s http://127.0.0.40:42068/metrics | grep sxa_inbour

# Example output:
#
sxa_inbound_messages_total{message_type="session_establishment_respor
5432
#
sxa_inbound_messages_total{message_type="session_modification_respons
12100
# sxa_inbound_messages_total{message_type="session_report_request"} 0
```

Rule Installation Verification

Check if rules are properly installed in SGW-U:

```
# Monitor session establishment successes/failures
curl -s http://127.0.0.40:42068/metrics | grep
sxa_session_establishment

# Check for PDR installation issues
curl -s http://127.0.0.40:42068/metrics | grep pdr_installation

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

Troubleshooting

Association Failures

Problem: "Association Setup Failed"

Diagnosis:

1. Check network connectivity: `ping <sgwu_ip>`
2. Verify port: `netstat -an | grep 8805`

3. Check logs for error details

Common Causes & Solutions:

Cause	Symptom	Solution
Network unreachable	Timeout on setup	Verify routing to SGW-U
Port blocked	Connection refused	Check firewall rules
SGW-U down	No response	Restart SGW-U process
Node ID mismatch	Setup rejected	Verify configuration

Session Establishment Failures

Problem: "Session Establishment Failed"

Diagnosis:

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

# Check logs for specific error
tail -f /var/log/sgw_c/sgw_c.log | grep "Session Establishment"
```

Common Causes:

Cause	Error Message	Solution
SGW-U no resources	"Insufficient resources"	Check SGW-U capacity
Invalid PDR	"Mandatory IE missing"	Verify rule definitions
SEID conflict	"SEID already exists"	Check session duplication
Timeout	"Session establishment timeout"	Increase timeout or check SGW-U

Usage Report Issues

Problem: "Usage Reports Missing"

Diagnosis:

```
# Check report count
curl -s http://127.0.0.40:42068/metrics | grep
session_report_request_total

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

Solutions:

- Verify SGW-U heartbeat is active
- Check Session Report trigger configuration
- Ensure CDR directory permissions are correct
- Monitor for SGW-U buffer overflow

Performance Issues

Problem: High PFCP message latency

Metrics to Check:

```
# Message processing duration
curl -s http://127.0.0.40:42068/metrics | grep
sxa_inbound_duration_seconds

# Per-peer session load
curl -s http://127.0.0.40:42068/metrics | grep seid_by_peer

# Queue depth
curl -s http://127.0.0.40:42068/metrics | grep pfcp_queue_depth
```

Optimization Steps:

1. Load balance across multiple SGW-U peers
2. Increase heartbeat timeout if network is lossy
3. Monitor and reduce rule complexity
4. Scale horizontally with additional SGW-C instances

For complete metrics reference, dashboard configuration, and alerting setup, see the Monitoring & Metrics Guide.

Best Practices

Configuration

- **Heartbeat Interval:** Set to 20-30 seconds for reliable detection
- **Session Timeout:** 5-10 seconds based on network RTT
- **Maximum Retries:** 2-3 for balance between reliability and latency
- **Peer Selection:** Distribute load across all SGW-U peers

Operations

- **Peer Redundancy:** Configure multiple SGW-U instances for failover
- **Graceful Reload:** Support in-service software upgrades

- **Session Draining:** Migrate sessions before maintenance
- **Monitoring:** Track association recovery frequency

Troubleshooting

- **Keep Logs:** Retain PFCP message traces for debugging
- **Correlation:** Link S11 messages to PFCP session operations
- **Baseline Metrics:** Establish normal performance baseline
- **Test Failure Scenarios:** Practice SGW-U failover procedures

---

OmniSGW Operations Guide

OmniSGW - Serving Gateway (SGW)

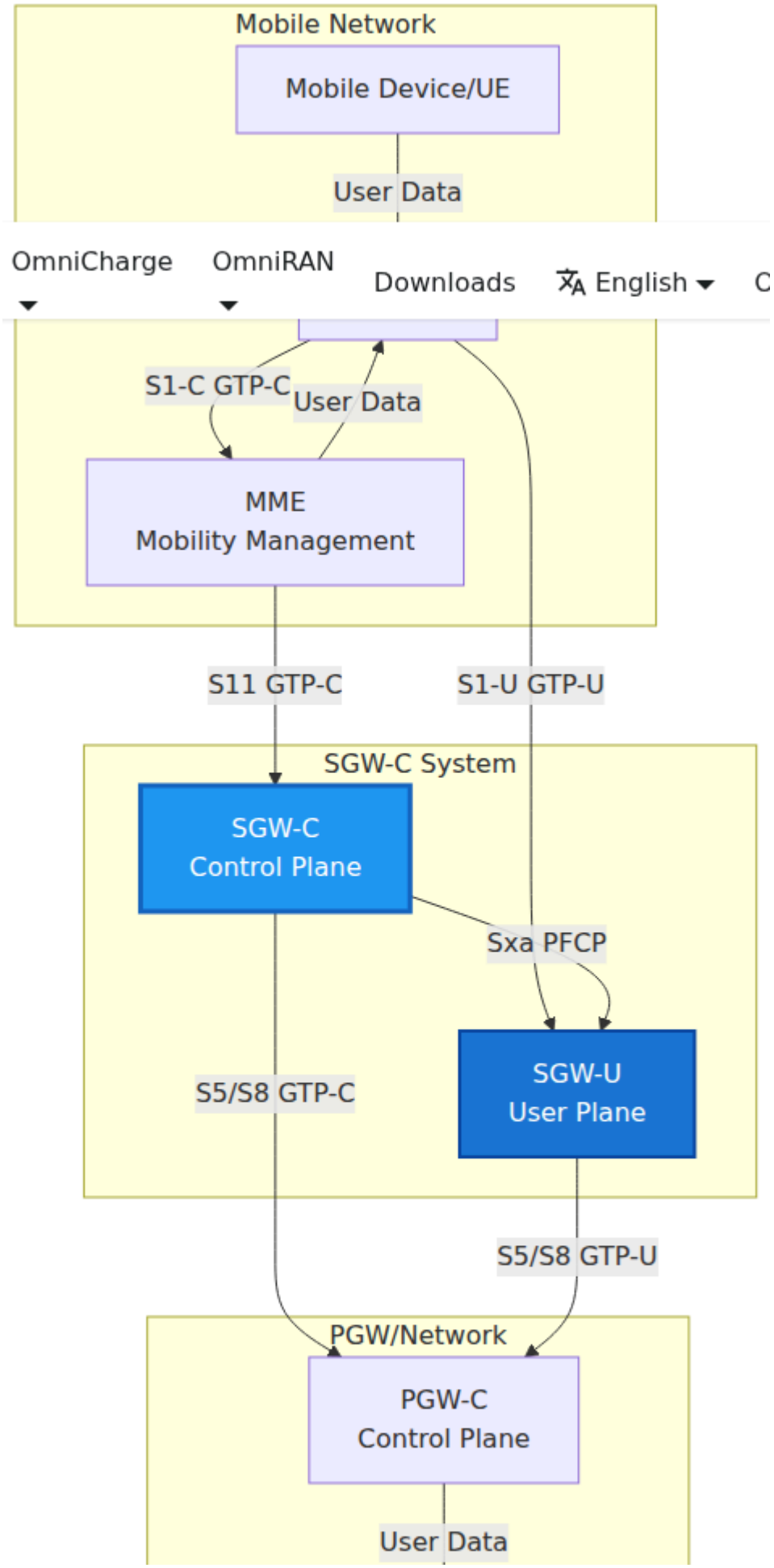
by Omnitouch Network Services

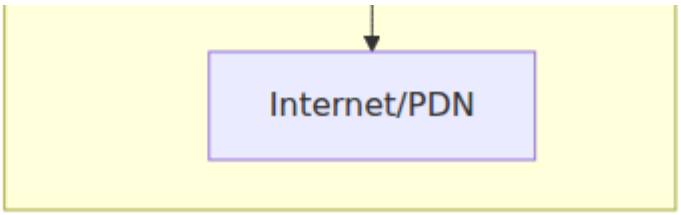
Table of Contents

Overview

OmniSGW is a high-performance Serving Gateway (SGW) implementation for 3GPP LTE Evolved Packet Core (EPC) networks, developed by Omnitouch Network Services. It manages the functions for UE mobility and bearer management, including:

- **Session Management** - Creating, modifying, and terminating UE (User Equipment) data sessions
- **Mobility Coordination** - Handling handovers between eNodeBs with data forwarding
- **Bearer Management** - Creating and modifying dedicated bearers for different QoS requirements
- **Charging Information** - Tracking session events for offline charging
- **User Plane Coordination** - Controlling the SGW-U (User Plane) for packet forwarding


Mobile Network
Mobile Device/UE
User Data
OmniCharge
OmniRAN
Downloads
English
S1-C GTP-C
User Data
MME
Mobility Management
S11 GTP-C
S1-U GTP-U
SGW-C System
SGW-C
Control Plane
Sxa PFCP
SGW-U
User Plane
S5/S8 GTP-C
S5/S8 GTP-U
PGW/Network
PGW-C
Control Plane
User Data



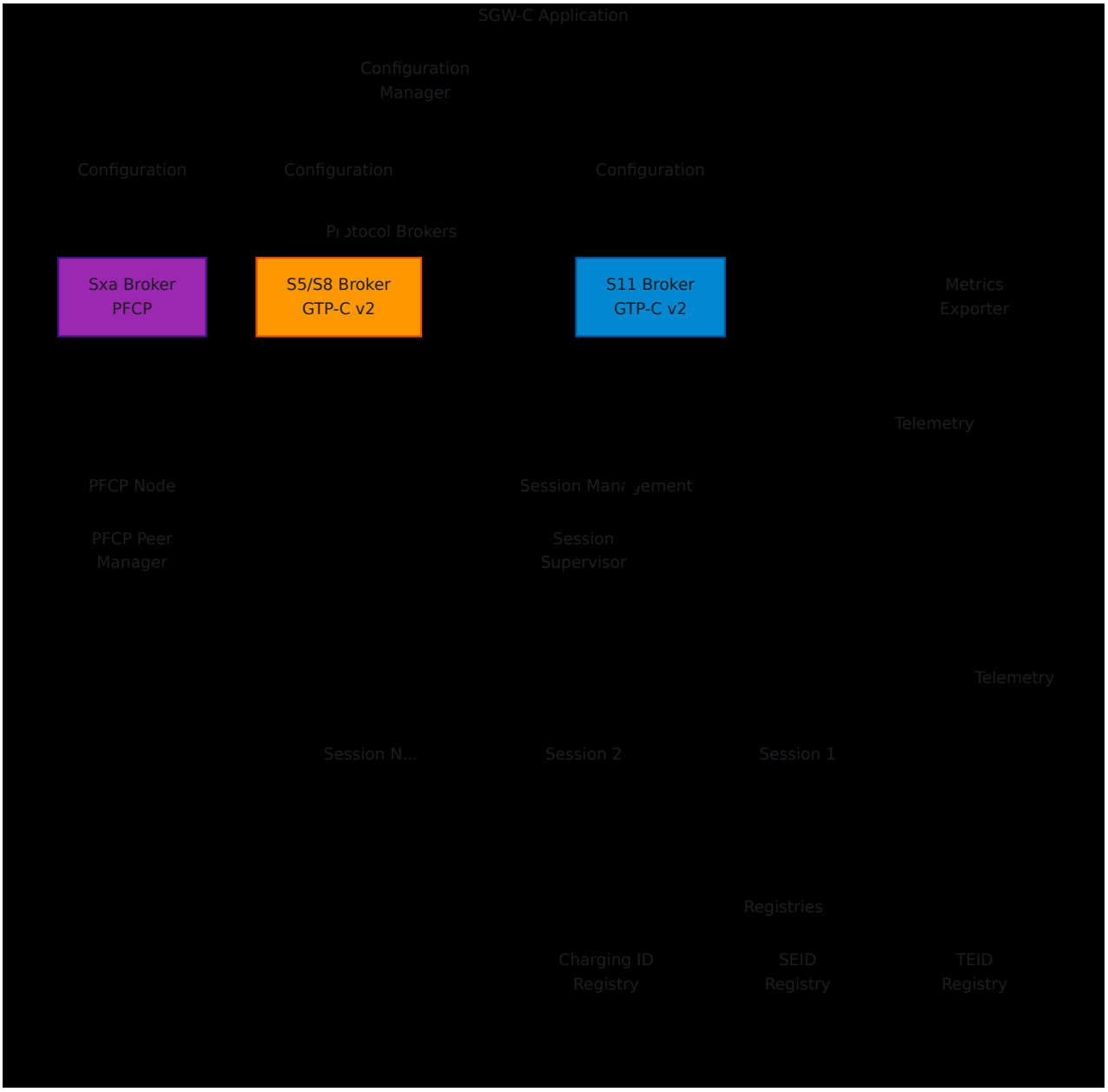
What SGW-C Does

- **Accepts session requests** from MME via S11 interface (GTP-C)
- **Coordinates with PGW-C** for PDN connectivity via S5/S8 interface (GTP-C)
- **Manages bearer lifecycle** including creation, modification, and deletion
- **Programs forwarding rules** in SGW-U via Sxa interface (PFCP)
- **Handles UE mobility** by managing handovers between eNodeBs
- **Provides downlink data paging** for suspended sessions
- **Tracks charging information** for offline billing systems

---

Architecture

Component Overview



Process Architecture

SGW-C is built on Elixir/OTP and uses a supervised process architecture:

- **Application Supervisor** - Top-level supervisor managing all components
- **Protocol Brokers** - Handle incoming/outgoing protocol messages (S11, S5/S8, Sxa)

- **Session Processes** - One GenServer per active UE session
- **Registries** - Track allocated resources (TEIDs, SEIDs, Charging IDs, etc.)
- **PFCP Node Manager** - Maintains PFCP associations with SGW-U peers

Each component is supervised and will automatically restart on failure, ensuring system reliability.

Real-time system health metrics can be monitored via the Web UI Application page:

Network Interfaces

SGW-C implements three primary 3GPP interfaces:

S11 Interface (GTP-C v2)

Purpose: Control plane signaling between MME and SGW-C

Protocol: GTP-C Version 2 over UDP

Key Messages:

- Create Session Request/Response
- Delete Session Request/Response
- Modify Bearer Request/Response

- Create Bearer Request/Response
- Delete Bearer Request/Response
- Downlink Data Notification/Acknowledge

Configuration: See S11 Interface Documentation

Sxa Interface (PFCP)

Purpose: Control plane signaling between SGW-C and SGW-U

Protocol: PFCP (Packet Forwarding Control Protocol) over UDP

Key Messages:

- Association Setup Request/Response
- Session Establishment Request/Response
- Session Modification Request/Response
- Session Deletion Request/Response
- Session Report Request/Response
- Heartbeat Request/Response

Configuration: See PFCP/Sxa Interface Documentation

S5/S8 Interface (GTP-C v2)

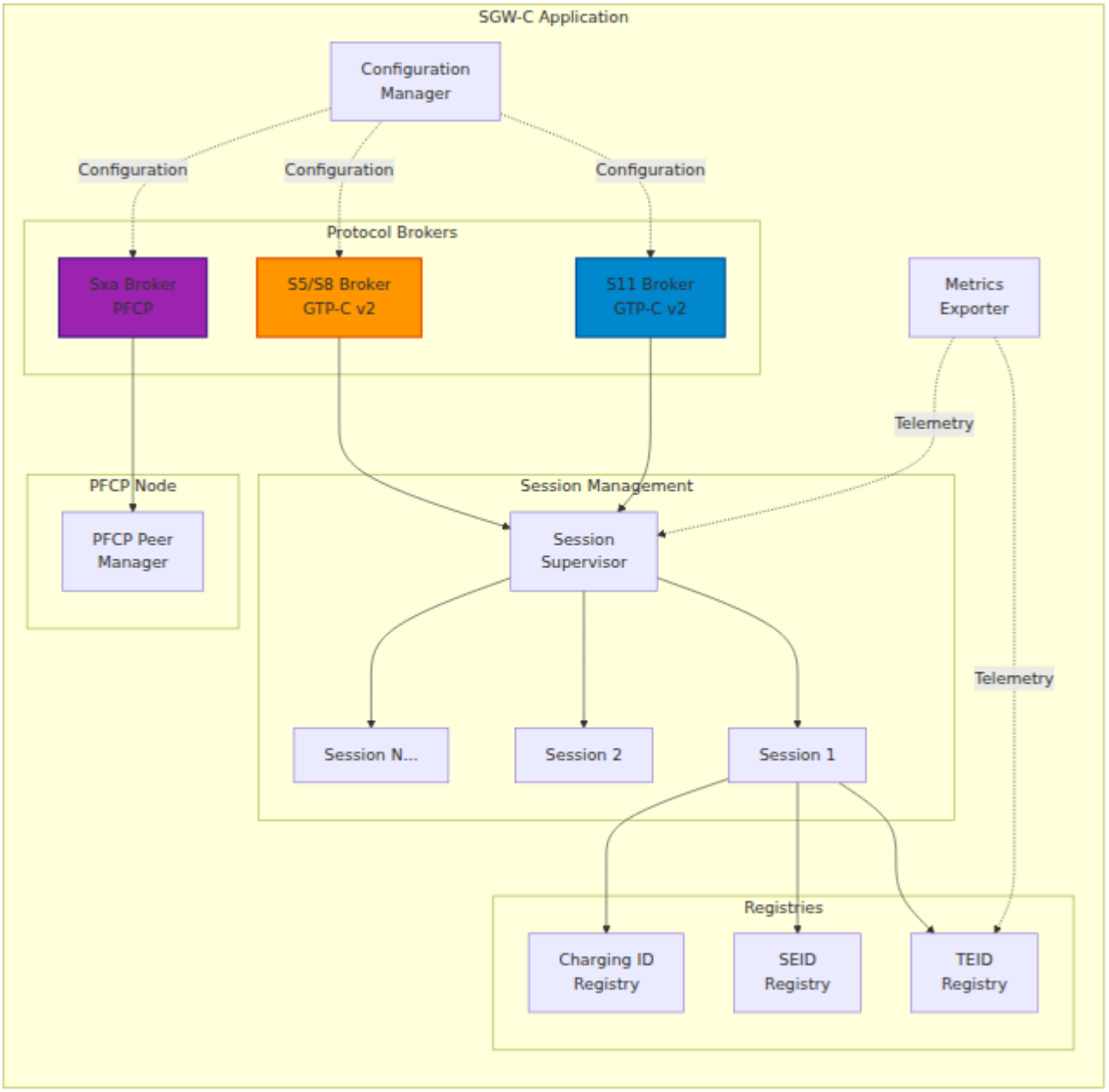
Purpose: Control plane signaling between SGW-C and PGW-C for PDN connectivity

Protocol: GTP-C Version 2 over UDP

Key Messages:

- Create Session Request/Response
- Delete Session Request/Response
- Modify Bearer Request/Response
- Create Bearer Request/Response
- Delete Bearer Request/Response

Configuration: See S5/S8 Interface Documentation



Key Concepts

UE Session

A UE Session represents an active mobile device connected to the network. Each session manages:

- **IMSI** (International Mobile Subscriber Identity) - Unique subscriber identifier
- **GUTI** (Globally Unique Temporary Identifier) - Temporary UE identifier from MME
- **MSISDN** - Mobile phone number
- **TAI** (Tracking Area Identifier) - Current location area
- **Session TEIDs** - Tunnel endpoint identifiers for S11 and S5/S8
- **Active Bearers** - List of associated data bearers

PDN Connection

A PDN (Packet Data Network) Connection represents a UE's data connection through a specific PGW-C. Each session has:

- **APN** (Access Point Name) - Identifies the external network
- **Charging ID** - Unique identifier for billing across SGW and PGW
- **TEID** (Tunnel Endpoint ID) - S5/S8 interface tunnel identifier
- **SEID** (Session Endpoint ID) - Sxa interface session identifier
- **Default Bearer** - Created with every PDN connection
- **Dedicated Bearers** - Additional bearers for specific QoS needs

Bearer Context

A bearer represents a traffic flow with specific QoS characteristics:

- **Default Bearer** - Created with every PDN connection for best-effort traffic
- **Dedicated Bearers** - Additional bearers for specific service requirements (voice, video, etc.)
- **EBI** (EPS Bearer ID) - Unique identifier for each bearer within a session
- **QoS Parameters** - QCI (QoS Class Identifier), ARP (Allocation & Retention Priority), bitrates (MBR, GBR)

PFCP Rules

The SGW-C programs the SGW-U with packet processing rules:

- **PDR** (Packet Detection Rule) - Matches packets (uplink/downlink)
- **FAR** (Forwarding Action Rule) - Specifies forwarding behavior
- **QER** (QoS Enforcement Rule) - Enforces bitrate limits
- **BAR** (Buffering Action Rule) - Controls packet buffering during handovers

See Sxa Interface Documentation for details.

Mobility & Handover

SGW-C supports UE mobility across eNodeBs:

- **Intra-MME Handover** - Handover within same MME (no SGW change)
- **Inter-MME Handover** - Handover between MMEs with SGW relocation
- **Data Forwarding** - Buffering and forwarding data during handover
- **Tracking Area Update** - UE re-registration when moving between areas

---

Getting Started

Prerequisites

- Elixir ~1.16
- Erlang/OTP 26+
- Network connectivity to MME, SGW-U, and PGW-C
- Understanding of LTE EPC architecture

Verifying Operation

Check the logs for successful startup:

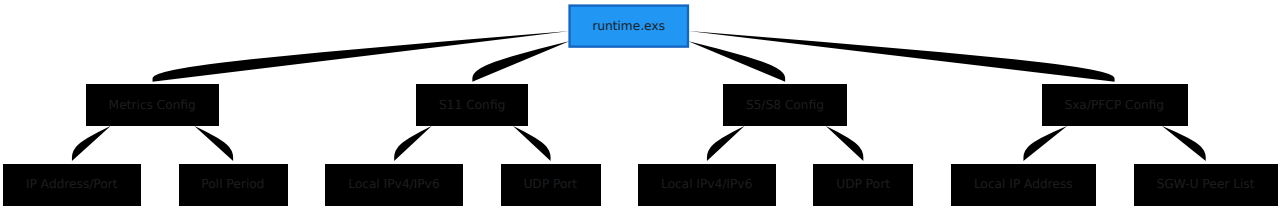
```
[info] Starting OmniSGW...
[info] Starting Metrics Exporter on 127.0.0.40:42068
[info] Starting S11 Broker on 127.0.0.10
[info] Starting S5/S8 Broker on 127.0.0.15
[info] Starting Sxa Broker on 127.0.0.20
[info] Starting PFCP Node Manager
[info] OmniSGW successfully started
```

Access metrics at `http://127.0.0.40:42068/metrics` (configured address).

Configuration

All runtime configuration is defined in `config/runtime.exs`. The configuration is structured into several sections:

Configuration Overview



Quick Configuration Reference

Section	Purpose	Documentation
metrics	Prometheus metrics exporter	Monitoring Guide
s11	GTP-C interface to MME	S11 Config
s5s8	GTP-C interface to PGW-C	S5/S8 Config
sxa	PFCP interface to SGW-U	Sxa Config

See the Complete Configuration Guide for detailed information.

Web UI - Real-Time Operations Dashboard

OmniSGW includes a built-in **Web UI** for real-time monitoring and operations, providing instant visibility into system status without needing command-line tools or metrics queries.

Accessing the Web UI

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

Available Pages:

Page	URL	Purpose	Refresh Rate
UE Sessions	`/ue_sessions`	View all active UE sessions and bearers	2 seconds
PFCP Sessions	`/pfcp_sessions`	View PFCP sessions with SGW-U	2 seconds
SGW-U Status	`/sgwu_status`	Monitor PFCP peer associations	2 seconds
Logs	`/logs`	Real-time log streaming	Live

Key Features

Real-Time Updates:

- All pages auto-refresh (no manual reload needed)
- Live data streaming from OmniSGW processes
- Color-coded status indicators (green/red)

Search & Filter:

- Search sessions by IMSI, GUTI, phone number
- Instant filtering without page reload

Expandable Details:

- Click any row to see complete session details
- Inspect all active bearers and QoS parameters
- View peer configuration and capabilities

No Authentication Required (Internal Use):

- Direct access from management network
- Designed for NOC/operations team use
- Bind to management IP only for security

Operational Workflows

Session Troubleshooting:

```
1. User reports connectivity issue
2. Open UE Sessions page
3. Search by IMSI or phone number
4. Verify session exists and has correct:
   - Tracking Area
   - Active bearers and their QoS
   - Tunnel endpoints established
   - Correct PGW-C association
5. If no session found → Check logs for rejection reason
```

System Health Check:

```
1. Open SGW-U Status page → Verify all SGW-U peers "Associated"
2. Open UE Sessions → Check active session count vs. capacity
3. Monitor bearer distribution across APNs
```

Capacity Monitoring:

- Glance at UE Sessions count
- Compare to licensed/expected capacity
- Identify peak usage times
- Monitor distribution by service type

Web UI vs. Metrics

Use Web UI for:

- Individual session and bearer details
- Real-time peer status
- Quick health checks
- Troubleshooting specific users
- Verifying configuration

Use Prometheus Metrics for:

- Historical trends
- Alerting and notifications
- Capacity planning graphs
- Performance analysis
- Long-term monitoring

Best Practice: Use both together - Web UI for immediate operations, Prometheus for trends and alerts.

Monitoring & Metrics

In addition to the Web UI, OmniSGW exposes Prometheus-compatible metrics for monitoring:

Available Metrics

- **Session Metrics**
  - `teid_registry_count` - Active S11/S5S8 TEIDs
  - `seid_registry_count` - Active PFCP sessions
  - `charging_id_registry_count` - Active charging IDs
  - `active_ue_sessions` - Total active UE sessions
  - `active_bearers` - Total active bearers across all sessions
- **Message Metrics**
  - `s11_inbound_messages_total` - GTP-C messages received on S11
  - `s5s8_inbound_messages_total` - GTP-C messages received on S5/S8
  - `sxa_inbound_messages_total` - PFCP messages received
  - Message handling duration distributions
- **Error Metrics**
  - `s11_inbound_errors_total` - S11 protocol errors
  - `s5s8_inbound_errors_total` - S5/S8 protocol errors
  - `sxa_inbound_errors_total` - Sxa protocol errors

Accessing Metrics

Metrics are exposed via HTTP at the configured endpoint:

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

See Monitoring & Metrics Guide for dashboard setup and alerting.

Detailed Documentation

This section provides a comprehensive overview of all OmniSGW documentation. Documents are organized by topic and use case.

Documentation Structure

```
OmniSGW Documentation
├── OPERATIONS.md (This Guide)
│
└── docs/
    ├── Configuration & Setup
    │   ├── configuration.md          Complete runtime.exs
reference
    │
    ├── Network Interfaces
    │   ├── sxa-interface.md          Sxa/PFCP (SGW-U
communication)
    │   ├── s11-interface.md          S11 (MME communication)
    │   └── s5s8-interface.md         S5/S8 (PGW-C communication)
    │
    └── Operations
        ├── session-management.md     UE session lifecycle
        ├── bearer-management.md      Bearer operations
        ├── cdr-format.md             Offline charging records
        └── monitoring.md             Prometheus metrics &
alerting
```

Documentation by Topic

Getting Started

Document	Description	Purpose
OPERATIONS.md	Main operations guide (this document)	Overview and quick start

⚙ Configuration

Document	Description
configuration.md	Complete runtime.exs configuration reference

□ Network Interfaces

Document	Description
sxa-interface.md	PFCP/Sxa interface to SGW-U
s11-interface.md	GTP-C S11 interface to MME
s5s8-interface.md	GTP-C S5/S8 interface to PGW-C

□ Operations & Monitoring

Document	Description
session-management.md	UE session lifecycle and operations
bearer-management.md	Bearer creation, modification, deletion
cdr-format.md	Offline charging data record format
monitoring.md	Prometheus metrics, Grafana dashboards, alerting

Reading Paths

For Network Operators

1. OPERATIONS.md - Overview (this document)
2. configuration.md - Setup

3. monitoring.md - Monitoring
4. session-management.md - Day-to-day operations

For Network Engineers

1. OPERATIONS.md - Architecture overview (this document)
2. sxa-interface.md - User plane control
3. s11-interface.md - Mobile management
4. s5s8-interface.md - PDN connectivity
5. session-management.md - Session lifecycle
6. bearer-management.md - Bearer operations

For Configuration & Deployment

1. configuration.md - Complete reference
2. monitoring.md - Set up monitoring

Additional Resources

3GPP Specifications

Spec	Title
TS 29.274	GTP-C v2 (S11 and S5/S8 interfaces)
TS 29.244	PFCP (Sxa interface)
TS 32.251	Packet Switched domain charging
TS 32.298	CDR encoding
TS 23.401	EPC architecture