GTPv2 and Sx Messages Retransmission and Timeout Handling

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	cnSGW-C
Applicable Platform(s)	SMI
Feature Default Setting	Enabled - Always-on
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
First introduced.	2021.01.0

Feature Description

cnSGW-C supports the handling of retransmission and timeout messages. It sends the following:

- Messages over a GTPC interface towards MME and PGW
- Sx messages are also sent towards the User Plane (UP).

Note For handling the retransmission and timeout functionalities, you must add the retransmission configuration (N3/T3) for the interface (S5e, S11, and Sxa).

How it Works

This section describes how this feature works.

Inbound Message Retransmission

At each N4 and GTP endpoint, there’s a set of queues for incoming and outgoing traffic. Each queue has a dispatcher thread running that pulls the message from the queue. It dispatches the message to the application for further processing.

Each dispatcher references a retransmission cache to check if the incoming request is already in service. It further performs the following actions:

- If it’s a retry request, the dispatcher drops the incoming request.
- If the retransmission cache reaches the threshold for outstanding requests, the incoming request is dropped.

Each dispatcher has a separate retransmission cache. This cache is also updated with the response of the request sent. It’s for the retransmission request received after the response is sent.

Call Flows

This section describes the key call flows for this feature.

The following are the outbound message retransmission call flows.

GTPCv2 (GTPC-EP or Service) Call Flow

This section describes the GTPCv2 (GTPC-EP or Service) call flow.

Figure 1: GTPCv2 (GTPC-EP or Service) Call Flow

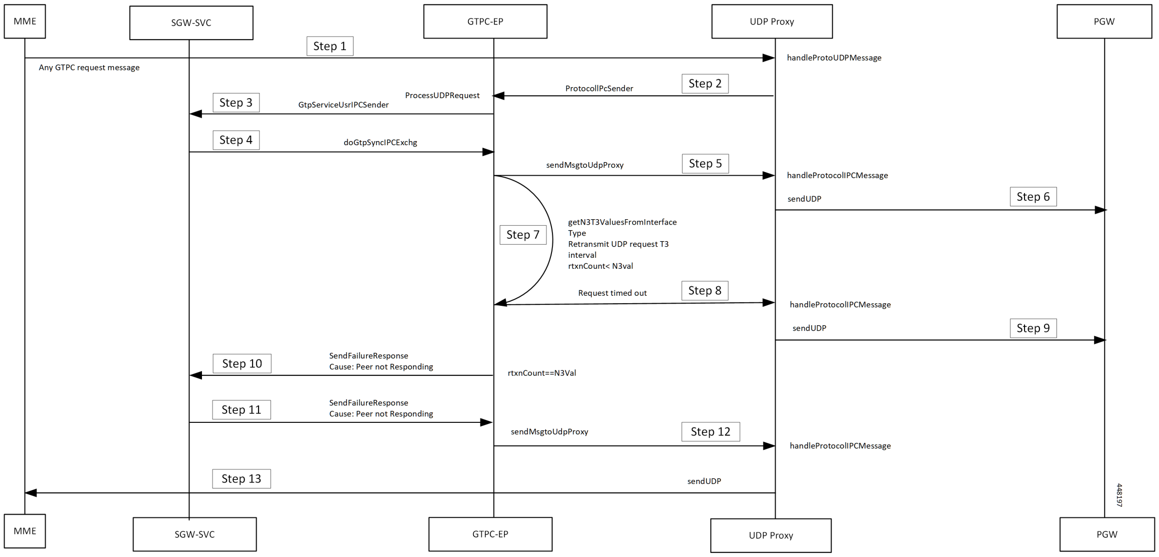

Table 3: GTPCv2 (GTPC-EP or Service) Call Flow Description

Step	Description
1	The MME sends the GTPC-EP message request to the UDP Proxy.
2	The UDP Proxy processes **handleProtoUDPMessage** and send **ProtocolIPCSender** to the GTPC-EP.
3	The GTPC-EP processes **ProcessUDPRequest** and sends **GtpServiceUsrIPCSender** to the SGW-SVC.
4	The SGW-SVC processes and sends **doGtpSyncIPCExchg** to the GTPC-EP.
5	The GTPC-EP sends **sendMsgtoUdpProxy** to the UDP Proxy.
6	The UDP Proxy processes **handleProtocolIPCMessage** and **sendUDP** messages to the PGW.
7	The GTPC-EP processes the following requests: • It gets the N3T3 value from the interface (S5e for the S5 interface). • It retransmits the UDP request when the T3 interval expires and the **retransmit_count** value is less than the N3 value.
8	The GTPC-EP sends the request timed out message to the UDP Proxy.
9	The UDP Proxy processes **handleProtocolIPCMessage** and **sendUDP** messages to the PGW.
10	The SGW-SVC receives the Timeout failure response when the **retransmit_count** is equal to the N3, and no response is received.
11	The SGW-SVC sends **SendFailureResponse** to the GTPC-EP, with the cause as Peer not Responding.

Step	Description
12	The GTPC-EP sends **sendMsgtoUdpProxy** to the UDP Proxy.
13	The UDP Proxy processes **handleProtocolIPCMessage** and **sendUDP** messages to the MME.

Sxa (smf-proto or Service) Call Flow

This section describes the call flow for the Sxa (smf-proto or service).

Figure 2: Sxa (smf-proto or Service) Call Flow

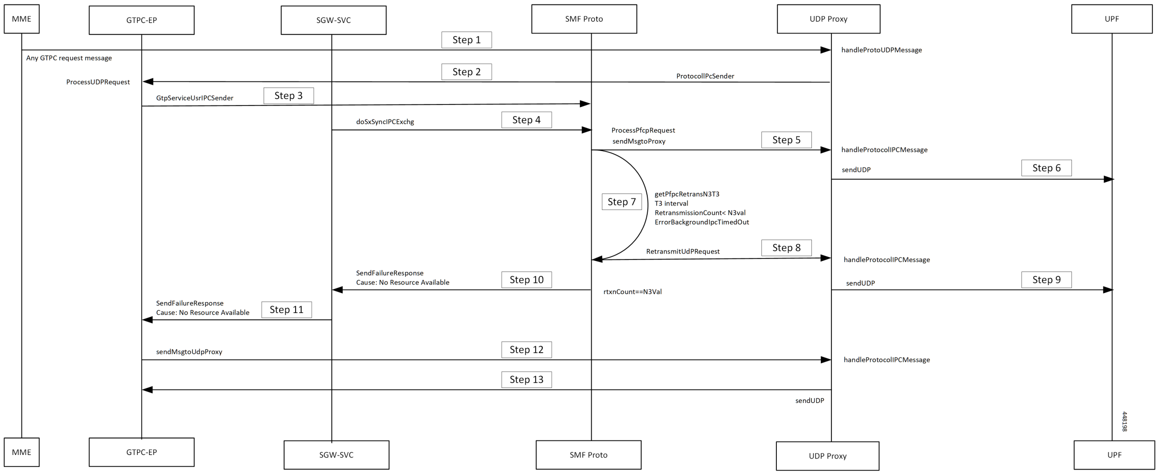

Table 4: Sxa (smf-proto or Service) Call Flow Description

Step	Description
1	The MME sends the GTPC-EP message request to the UDP Proxy.
2	The UDP Proxy processes **handleProtoUDPMessage** and sends **ProtocolIPCSender** to the GTPC-EP.
3	The GTPC-EP processes **ProcessUDPRequest** and sends **GtpServiceUsrIPCSender** to the SGW-SVC.
4	The SGW-SVC processes and sends **doGtpSyncIPCExchg** to the GTPC-EP.
5	The GTPC-EP sends **sendMsgtoUdpProxy** to the UDP Proxy.
6	The UDP Proxy processes **handleProtocolIPCMessage** and **sendUDP** messages to the PGW.
7	The GTPC-EP processes the following requests: • It gets the N3T3 values from the interface (S5e for the S5 interface). • It retransmits the UDP request, when the T3 interval expires, and the **retransmit_count** value is less than the value of N3.
8	The GTPC-EP sends the request timed out message to the UDP Proxy.

Step	Description
9	The UDP Proxy processes **handleProtocolIPCMessage** and **sendUDP** messages to the PGW.
10	The SGW-SVC receives the Timeout failure response when the **retransmit_count** is equal to the value of N3, and no response is received.
11	The SGW-SVC sends **SendFailureResponse** to the GTPC-EP, with the cause as Peer not Responding.
12	The GTPC-EP sends **sendMsgtoUdpProxy** to the UDP Proxy.
13	The UDP Proxy processes **handleProtocolIPCMessage** and **sendUDP** messages to the MME

Configuring the Dispatcher

This section describes the dispatcher configuration for GTPC-EP (GTP) and SMF_PROTO (PFCP) endpoint.

```
config
  instance instance-id instance_id
    endpoint endpoint_name
      interface interface_name
        dispatcher dispatcher_values
        end
```

NOTES:

- **instance instance-id** *instance_id*—Specify the instance ID.
- **endpoint** *endpoint_name*—Specify the endpoint name.
- **interface** *interface_name*—Specify the interface name.
- **dispatcher** *dispatcher_values*—Configure the dispatcher values.
  - **cache** *true | false*—Disable the retransmission cache support for dispatcher queue. The default value is false.

    Must be one of the following:

    - *true*—Disable the retransmission cache support for dispatcher queue, set the **cache** to true.
    - *false*—Enable the retransmission cache support for dispatcher queue, set the **cache** to false.
  - **capacity** *capacity_value*—Specify the queue size for each dispatcher queue. The default value is 5000.
  - **count** *value*—Specify the number of supported dispatcher queues for the interface or the endpoint.
  - **expiry** *expiry_duration*—Specify the duration for which the cache entry with response is held in the cache. The default value is 60 seconds.
  - **nonresponsive** *nonresponsive_duration*—Specify the duration for which the cache entry without response is held in the cache.

- **outbound** *true* | *false*—Disable dispatcher queue support for outgoing messages. The default value is true. When set to false, the queue support is enabled for outgoing messages.

  It means by default, the queue support is enabled for the outgoing messages. Must be one of the following:

  - *true*—Disable dispatcher queue support for outgoing messages, set the **outbound** to true.
  - *false*—Enable dispatcher queue support for outgoing messages, set the **outbound** to false.

- **rate-limit** *rate_limit*—Specify the rate limit for each queue.
- **threshold** *threshold*—Specify the outstanding limit for non-responsive cache entries. When the threshold is reached, the incoming requests are dropped. It must be an integer. The default value is 30000 milliseconds.

Configuration Example

The following is an example configuration.

```
config
   instance instance-id 1
   endpoint protocol
   interface n4
      dispatcher
         count 5
         outbound true
      threshold 5000
      end
```

Configuration Verification

To verify the configuration:

```
show running-config instance instance-id 1 endpoint gtp
instance instance-id 1
endpoint gtp
replicas 1
interface s5e
sla response 7000
dispatcher
count 1
capacity 1000
outbound true
threshold 10000
expiry 40000
nonresponsive 20000
exit
vip-ip 209.165.201.25
exit
interface s11
sla response 7000
dispatcher
count 1
capacity 1000
outbound true
```

```
threshold 10000
expiry 40000
nonresponsive 20000
exit
vip-ip 209.165.201.2
exit
exit
```

```
show running-config instance instance-id 1 endpoint pfcp
instance instance-id 1
endpoint pfcp
replicas 1
interface sxa
dispatcher
count 1
capacity 1000
outbound true
threshold 10000
expiry 40000
nonresponsive 20000
exit
heartbeat
interval 0
retransmission-timeout 3
max-retransmissions 5
exit
retransmission timeout 5 max-retry 1
exit
interface n4
heartbeat
interval 0
retransmission-timeout 3
max-retransmissions 5
```