

Collision Handling on the P-GW/SAEGW/S-GW

- Feature Description, on page 1
- How It Works, on page 1
- Configuring Collision Handling, on page 4
- Monitoring the Collision Handling Feature, on page 4

Feature Description

GTPv2 message collisions occur in the network when a node is expecting a particular procedure message from a peer node but instead receives a different procedure message from the peer. GTP procedure collisions are quite common in the network; especially with dynamic Policy and Charging Control, the chances of collisions happening in the network are very high.

These collisions are tracked by statistics and processed based on a pre-defined action for each message collision type. These statistics assist operators in debugging network issues.

Important If the SAEGW is configured as a pure P-GW or a pure S-GW, operators will see the respective collision statistics if they occur.

Relationships to Other Features

- This feature is a part of the base software license for the P-GW/SAEGW/S-GW. No additional license is required.
- A P-GW, S-GW, or SAEGW service must be configured to view GTPv2 collision statistics.

How It Works

Collision Handling

As GTPv2 message collisions occur, they are processed by the P-GW, SAEGW, and S-GW. They are also tracked by statistics and with information on how the collision was handled.

Specifically, the output of the **show egtpc statistics** verbose command has been enhanced to provide information on GTPv2 message collision tracking and handling at the S-GW and P-GW ingress interfaces, The information available includes:

- **Interface**: The interface on which the collision occurred: SGW (S4/S11), SGW (S5) and P-GW (S8).
- **Old Proc (Msg Type)**: Indicates the ongoing procedure at eGTP-C when a new message arrived at the interface which caused the collision. The Msg Type in brackets specifies which message triggered this ongoing procedure. Note that the Old Proc are per 3GPP TS 23.401.
- **New Proc (Msg Type)**: The new procedure and message type. Note that the New Proc are per 3GPP TS 23.401.
- **Action**: The pre-defined action taken to handle the collision. The action can be one of:
  - **No Collision Detected**
  - **Suspend Old**: Suspend processing of the original (old) message, process the new message, then resume old message handling.
  - **Abort Old**: Abort the original message handling and processes the new message.
  - **Reject New**: Reject the new message, and process the original (old) message.
  - **Silent Drop New**: Drop the new incoming message, and the process the old message.
  - **Parallel Hndl**: Handle both the original (old) and new messages in parallel.
  - **Buffer New**: Buffer the new message and process it once the original (old) message has been processed.
  - **Counter**: The number of times each collision type has occurred.



Important The *Message Collision Statistics* section of the command output appears only if any of the collision statistics have a counter total that is greater than zero.

Sample output:

```
Message Collision Statistics
   Interface      Old Proc (Msg Type)       New Proc (Msg Type)    Action    Counter
    SGW(S5)   NW Init Bearer Create (95)  NW Init PDN Delete (99)  Abort Old    1
```

In this instance, the output states that at the S-GW egress interface (S5) a Bearer creation procedure is going on due to a CREATE BEARER REQUEST(95) message from the P-GW. Before its response comes to the S-GW from the MME, a new procedure PDN Delete is triggered due to a DELETE BEARER REQUEST(99) message from the P-GW.

The action that is carried out due to this collision at the eGTP-C layer is to abort (Abort Old) the Bearer Creation procedure and carry on normally with the Initiate PDN Delete procedure. The Counter total of 1 indicates that this collision happened once.

Example Collision Handling Scenarios

This section describes several collision handling scenarios for the S-GW and P-GW.

The S-GW processes additional collisions at the S-GW ingress interface for:

1. Create Bearer Request or Update Bearer Request messages with Inter-MME/Inter-RAT Modify Bearer Request messages (with and without a ULI change).
2. Downlink Data Notification (DDN) message with Create Bearer Request or Update Bearer Request.

The S-GW behavior to handle these collision scenarios are as follows:

1. A CBReq and MBReq [(Inter MME/Inter RAT (with or without ULI change)] collision at the S-GW ingress interface results in the messages being handled in parallel. The CBReq will wait for a Create Bearer Response (CBRsp) from the peer. Additionally, an MBReq is sent in parallel to the P-GW.
2. An UBReq and MBReq [(Inter MME/Inter RAT (with or without a ULI change)] collision at the SGW ingress interface is handled with a suspend and resume procedure. The UBReq would be suspended and the MBReq would be processed. Once the MBRsp is sent to the peer from the SGW ingress interface, the UBReq procedure is resumed.
3. Create Bearer Request (CBR) or Update Bearer Request (UBR) with Downlink Data Notification (DDN) messages are handled parallel.

As a result, no S-GW initiated Cause Code message 110 (Temporarily rejected due to handover procedure in progress) will be seen as a part of such collisions. Collisions will be handled in parallel.

The following GTP-C example collision handling scenarios may also be seen on the P-GW:

DBCmd/MBreq Collision Handling:

The P-GW enables operators to configure the behavior of the P-GW for collision handling of the Delete Bearer command (DBcmd) message when the Modify Bearer Request (MBreq) message for the default bearer is pending at the P-GW.

There are three CLI-controlled options to handle the collision between the DBCmd and MBReq messages:

- Queue the DBcmd message when the MBreq message is pending. The advantage of this option is that the DBCmd message is not lost for most of the collisions. It will remain on the P-GW until the MBRsp is sent out.
- Drop the DBCmd message when the MBreq message is pending. Note that with this option the S-GW must retry the DBCmd.
- Use pre-StarOS 19.0 behavior: abort the MBreq message and handle the DBcmd message. The advantage of this option is that it provides backward compatibility if the operator wants to retain pre-StarOS 19.0 functionality.

The CLI command **collision handling** provides more flexibility in configuring the handling of the DBCmd message and MBReq message collision scenario. Also refer to Configuring DBcmd Message Behavior, on page 4 in this document for instructions on how to configure the behavior for this collision handling scenario.

MBReq/CBreq Parallel Processing; Handling CBRsp:

The P-GW/S-GW has handles the following example collision scenario:

The node queues the CBRsp message and feeds the CBRsp message to the P-GW/S-GW session manager when the MBRsp is sent out. As a result, operators will see no retransmission of CBRsp messages from the MME.

Handling UBrsp when Transaction is Suspended:

The P-GW/S-GW handles the following example collision scenario:

When the P-GW/S-GW receives an UBRsp message, then the P-GW/S-GW handles the UBRsp message for the suspended transaction. As a result, The UBRsp message will be buffered until the MBRsp message is sent out.

Limitations

There are no known limitations to the collision handling feature on the P-GW/SAEGW/S-GW.

Standards Compliance

Specifications and standards do not specify any hard rules for collision handling cases.

Configuring Collision Handling

Operators can use the Command Line Interface (CLI) to configure the behavior of the P-GW for handling the following GTPv2 message collision:

- DBcmd Message when the MBreq Message for the Default Bearer is pending at the P-GW

Important Configuration via the CLI is **not** required for all other P-GW and S-GW collision handling scenarios.

Configuring DBcmd Message Behavior

Use the following example to configure the collision handling behavior for the Delete Bearer command message when the Modify Bearer Request message for the Default Bearer is pending at the P-GW.

```
configure
   context  context_name
      egtp-service  egtp_service_name
         collision-handling dbcmd-over-mbreq  { drop | queue }
         { default | no } collision-handling dbcmd-over-mbreq
         end
```

Notes:

- **drop**: Drop the DBcmd message when the MBreq message is pending.
- **queue**: Queue the DBcmd message when the MBreq is message is pending.
- The default behavior is to abort the MBReq message and handle the DBcmd message.

Verifying the Configuration

To verify the DBcmd Message when the MBreq Message for the Default Bearer is pending at the P-GW configuration, use the following command in Exec Mode:

```
show egtpc service all
      Collision handling: DBcmd when MBreq pending: <Queue DBcmd>, <Drop DBcmd>, or <Abort
 MBreq and handle Dbcmd>
```

Monitoring the Collision Handling Feature

This section describes how to monitor the collision handling feature.

Collision Handling Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of the collision handling on the P-GW/SAEGW/S-GW feature.

show configuration

The output of this command indicates if collision handling for the DBcmd message when the MBreq message is pending is enabled or disabled.

- collision-handling dbcmd-over-mbreq queue
- no collision-handling dbcmd-over-mbreq queue

show egtp-service all

The output of this command indicates how the P-GW is configured to handle the DBcmd Message when the MBreq Message for the Default Bearer is pending at the P-GW.

- Collision handling:
  - DBcmd when MBreq pending: <Queue DBcmd>, <Drop DBcmd>, or <Abort MBreq and handle Dbcmd>

show egtp statistics verbose

The output of this command has been enhanced to provide detailed information for all supported GTPv2 message collisions at the P-GW/S-GW ingress interface, including:

- The interface on which the collision occurred.
- The ongoing procedure at eGTP-C when a new message arrived at the interface which caused the collision. The Msg Type in brackets specifies which message triggered this ongoing procedure.
- The new procedure and message type.
- The pre-defined action taken to handle the collision.
- The number of times each collision type has occurred.



Important The *Message Collision Statistics* section of the command output appears only if any of the collision statistics have a counter total that is greater than zero.