

Contents

[Introduction](#)

[Background Information](#)

[GGSN Behavior](#)

[Cause Code 192 Error](#)

[Example Scenarios](#)

Introduction

This document describes the behavior of the Gateway General Packet Radio Service (GPRS) Supporting Node (GGSN) when the Serving GPRS Supporting Node (SGSN) does not respond to the GPRS Tunneling Protocol (GTP) echo request that is sent from the GGSN.

Background Information

You might experience high Packet Data Protocol (PDP) activation failures in the GGSN during a period of time when the SGSN does not respond to the GTP echo requests. Here are some questions that might arise in this scenario:

1. Do the *Create PDP* or *Update PDP* requests from the SGSN arrive at the GGSN?
2. When the GTP echo requests fail from the GGSN to the SGSNs, how should the GGSN behave if the *Update PDP* context that is sent from the GGSN does not receive a response?
3. How does a GGSN fail a PDP if it does not receive a GTP echo response or a response for the non-echo request messages that arrive from an SGSN for that PDP?
4. How does a lack in GTP echo/non-echo responses directly affect the PDP activation failures?

GGSN Behavior

If the messages do not arrive at the GGSN, then the SGSN triggers a path failure alarm and silently drops them. Additionally, if there is no echo response received for the echo request that is initiated by the GGSN, it indicates that the peer is down, so the GGSN locally clears the calls that are related to that peer.

In the **show support details** command output, or the **show gtpc statistics verbose** command output, you can view the *GGSN Req Timeout* counters:

```
#show gtpc statistics verbose
```

```
SGSN Restart: Timeout:  
Create PDP Req: 5 GTPC Echo Timeout: 149160  
Update PDP Req: 0 GTPU Echo Timeout: 0  
Echo Response: 312 GGSN Req Timeout: 24182
```

Path Management Messages:

Echo Request RX: 34006780 Echo Response TX: 34006780

Echo Request TX: 29603851 Echo Response RX: 29537123

If you investigate the echo request messages that are transferred from the GGSN to the SGSN, it appears that the GGSN does not receive the echo responses. You must ensure that the messages are not dropped due to routing issues on the network or that the SGSN is not available.

The most common problem is the control path failures, which cause a large number of the roaming SGSNs to become unreachable.

If there is any GTP control message (such as an *update PDP context request*) from the GGSN that does not receive a response after all of the attempts are exhausted, the GGSN thinks that the peer is unreachable and tears down only that particular session reports the cause as a *Path Failure*. The PDP context is deleted on the GGSN, but the SGSN is not notified. This count is identified with these statistics:

```
SGSN Restart:                               Timeout:
Create PDP Req: 5 GTPC Echo Timeout: 149160
Update PDP Req: 0 GTPU Echo Timeout: 0
Echo Response: 312 GGSN Req Timeout:           24182
```

```
Update PDP Context Denied:
No Resources: 500 No Memory: 0
System Failure: 0 Non-existent:                 55460
```

The GGSN now tears down the PDP context session and never notifies the SGSN or the User Equipment (UE). The SGSN or UE might trigger an *update PDP context request*, and the GGSN might reject it with a *Cause Code 192* (non-existent).

Here is a section that is taken from **TS 29.060**:

- *If a Gprs Supporting Node(GSN) receives a Gprs Tunneling Protocol-Control plane(GTP-C) message requesting action related to a PDP context that the sending node believes is in existence, but that is not recognized by the receiving node, the receiving node shall send back to the source of the message, a response with the appropriate cause value (either "Non-existent" or "Context not found"). The Tunnel Endpoint Identifier used in the response message shall be set to all zeroes.*
- *If the SGSN receives an Update PDP Context Response with a Cause value "Non-existent", it shall delete the PDP Context*

Cause Code 192 Error

A Cause Code 192 (or *non-existent*) is an error that is sent by the GSNs on the Gn interface. It is populated in the *Cause of GTP messages* information element.

These are the GTP messages that can have a Cause Code 192 error:

- Update_PDP_Context_Response
- Delete_PDP_Context_Response

Note: The Tunnel End Identifier (TEID) that is used in the message that contains this error will be zero. Refer to **TS 29.060** for further details.

This error can appear in the aforementioned messages when it is sent by a GSN and it does not

have a context that corresponds to the one that is sent by the other GSN. The GSNs delete the PDP context when this error is received.

Example Scenarios

This section describes four scenarios in which a Cause Code 192 error can occur.

- **Scenario 1** – A GTP-C path failure occurs between the GSNs.
- **Scenario 2** – An echo request/response failure occurs between the GSNs.
- **Scenario 3** – There is a GTP Version 1 (GTPv1) to GTP Version 0 (GTPv0) handoff issue that causes the error. Here is a sample call-flow for this scenario:

A create PDP context request with GTPv1 is established.

The GTPv1-to-GTPv0 handoff occurs.

The call on the GGSN is now at GTPv0.

The GGSN receives the *update PDP context request* with a non-zero header TEID and rejects it due to the error (non-existent). **Note:** The SGSN should have forgotten the TEID, as call moved to GTPv0 (only flow-labels exist for GTPv0, not TEIDs). This indicates that the SGSN held on to the GTPv1 call even after the handoff to GTPv0.

- **Scenario 4** – The out-of-sync TEID effect is multiplied. Here is an example:

The UE1 establishes a PDP context; the SGSN allots the Control-TEID-1 (C-TEID-1) as its control TEID towards the GGSN on the *sgsn-UE1-ctxt* context. The C-TEID for all of the messages on the GGSN that head towards the SGSN have C-TEID-1.

A signaling message (non-echo) times-out on the SGSN, and the SGSN cleans up that *sgsn-UE1-ctxt* context locally. It also informs the Radio Network Controller (RNC) to clean up. It does not inform the GGSN, as it treats the GGSN as down. Now there is no PDP context for UE1 on the SGSN, and the PDP context for the same UE1 exists on the GGSN with C-TEID-1. The C-TEID-1 goes back to the tail of the free list.

The UE2 then wants to establish a PDP context to the same APN and passes through the same SGSN and GGSN. On the SGSN, the TEID is allocated and an *sgsn-UE2-ctxt* context is sent to the GGSN. If the number of free TEIDs is low, then the recently freed TEID is reallocated to the new PDP context. In this case, C-TEID-1 is reallocated to UE2.

On the GGSN, there are two contexts with C-TEID-1 as the Gn C-TEID. The GGSN does not check whether there is a TEID already present for the same. The GGSN then initiates a Delete PDP Context (DPC) for UE1 towards the SGSN.

On the SGSN, the C-TEID-1 is found, along with the context for it, which is *sgsn-UE2-Ctxt*. An attempt is made in order to delete that context and respond to the GGSN.

If there are GGSN-initiated requests (update/delete PDP) for the other contexts, the SGSN responds with a *Context not found* cause.

The GGSN drops that DPC response for UE2 because it never sent a DPC request for UE2.

There is now a second context on the GGSN that does not correspond to any context on the SGSN.

If the same C-TEID-1 is assigned to another UE, the problem repeats and compounds the issue.

Here is a section that is taken from **TS 29.060**:

Echo Response

The message shall be sent as a response to a received Echo Request.

The GSN that receives an Echo Response from a peer GSN shall compare the Restart Counter value received with the previous Restart Counter value stored for that peer GSN. If no previous value was stored, the Restart Counter value received in the Echo Response shall be stored for the peer GSN.

The value of a Restart Counter previously stored for a peer GSN may differ from the Restart Counter value received in the Echo Response from that peer GSN. In this case, the GSN that sent the Echo Response shall be considered as restarted by the GSN that received the Echo Response. The new Restart Counter value received shall be stored by the receiving entity, replacing the value previously stored for the sending GSN.

If the sending GSN is a GGSN and the receiving GSN is an SGSN, the SGSN shall consider all PDP contexts using the GGSN as inactive. For further actions of the SGSN refer to 3rd Generation Partnership Project(3GPP) Technical Specifications(TS) 23.007 [3].

If the sending GSN is an SGSN and the receiving GSN is a GGSN, the GGSN shall consider all PDP contexts using the SGSN as inactive. For further actions of the GGSN refer to **3GPP TS 23.007 [3]**.

Here is a section that is taken from **3GPP TS 23.007 V8.0**:

Restoration of data in the SGSN

Restart of the SGSN

After an SGSN restart, the SGSN deletes all Mobility Management(MM), PDP, Multimedia Broadcast Multicast Services (

The GGSN performs a polling function (echo request and echo response) towards the SGSNs with which the GGSN is in contact. The SGSN Restart counter shall be included in the echo response. If the value received in the GGSN differs from the one stored for that SGSN, the GGSN will consider that the SGSN has restarted (see 3GPP TS 29.060). The GGSN Restart counters shall be updated in the SGSN to the value received in the first echo message coming from each GGSN after the SGSN has restarted.

When the GGSN detects a restart in an SGSN with which it has PDP context(s) activated, it shall delete all these PDP context(s) . Also, the new value of the SGSN Restart counter received in the echo response from the SGSN restarted shall be updated in the GGSN.