# Troubleshoot of EGTP Path Failures

## Contents


**Introduction**

**Overview**

- Possible Reasons for EGTP Path Failures
- Logs Required

**Troubleshooting Commands**

**Scenario/Reasons in Brief**

- Reachability issue - Network Connectivity Issues
- Restart Counter Values Changes
- Huge Incoming Traffic Request - Network Congestion

**Solution**

**Workaround**

**Configuration Changes**

**Debugging Logs**


## Introduction

This document describes how to troubleshoot EGTP Path failure issues.

## Overview

Evolved GPRS Tunneling Protocol (EGTP) path failures refer to issues with the communication path between the GTP nodes in a mobile network. GTP is a protocol used in the transport of user data and signaling messages between different network elements.

### Possible Reasons for EGTP Path Failures

1. Reachability issue - Network Connectivity Issues

2. Restart counter values changes

3. Huge incoming traffic request - Network Congestion

4. Configuration issue in terms of DSCP/QOS and so on

5. No subscribers/sessions on the EGTPC link

### Logs Required

1. SSD/syslogs around problematic time covering the timeframe at least two hours before the issue started till the current time.

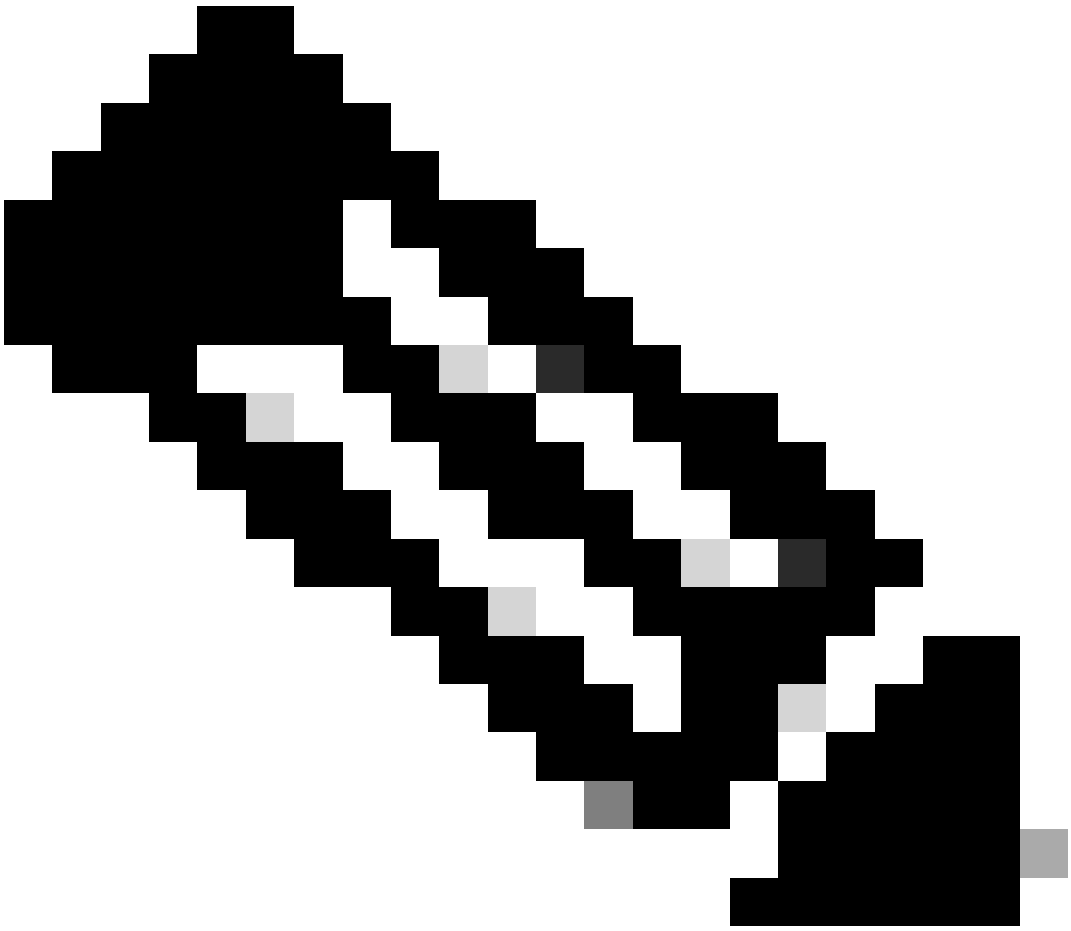
2. Reachability confirmation with logs, that is, ping and traceroute for the path for which path failures are seen.

3. Configuration check between problematic and non-problematic nodes.

4. Need to confirm if any sudden increase in traffic or any increase in rejection on the same path.

5. Bulkstats during problematic times covering the timeframe at least 2-3 days prior to the issue.

---



> **Note**: Depending on the type of issue, the logs mentioned earlier can be required. Not all the logs are required every time.

---

# Troubleshooting Commands

<#root>

```
show egtpc peers interface

show egtpc peers path-failure-history

show egtpc statistics path-failure-reasons

show egtp-service all
```

```
show egtpc sessions

show egtpc statistics

egtpc test echo gtp-version 2 src-address <source node IP address>  peer-address <remote node IP addres
```

**For more details related to above command refer doc as mentioned below**

**https://www.cisco.com/c/en/us/support/docs/wireless-mobility/gateway-gprs-support-node-ggsn/119246-techn**

SNMP traps:

```
Sun Feb 05 03:00:20 2023 Internal trap notification 1112 (EGTPCPathFail) context s11mme, service s11-mm

Tue Jul 09 18:41:36 2019 Internal trap notification 1112 (EGTPCPathFail) context pgw, service s5-s8-sgw
```

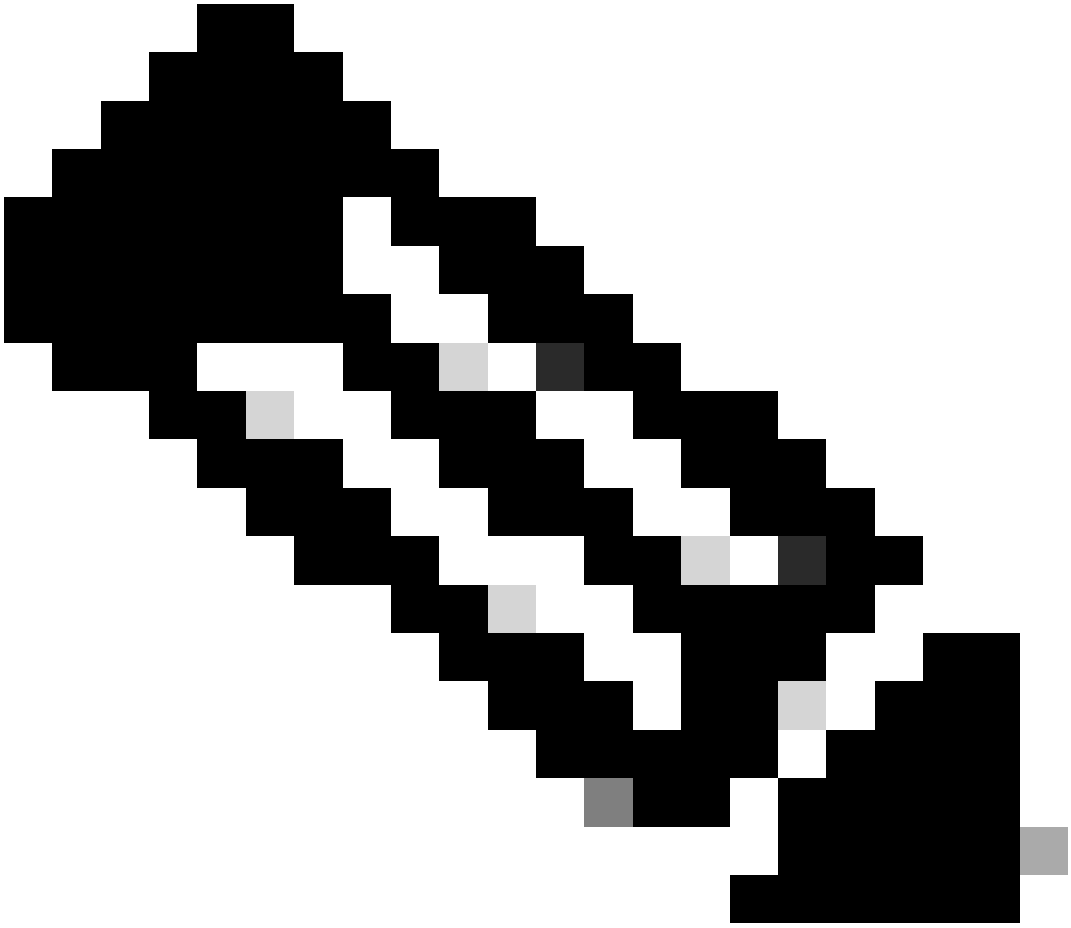
# Scenario/Reasons in Brief

## Reachability issue - Network Connectivity Issues

Reachability issues occur when an issue in the route path can be at the router end or firewall between SGSN/MME and SPGW/GGSN.

```
ping <destination IP>

traceroute <destination IP> src <source IP>
```



**Note**: Both the commands to check reachability must be checked from the content where the EGTP service is running.

## Restart Counter Values Changes

EGTP path maintains the restart counters at both ends of the path between SGSN/MME and GGSN/SPGW.


Refer to the https://www.cisco.com/c/en/us/support/docs/wireless/asr-5000-series/200026-ASR-5000-Series-Troubleshooting-GTPC-and.html link in order to understand this type of issue in detail.

## Huge Incoming Traffic Request - Network Congestion

Whenever there are sudden high-traffic transactions, there is a chance of EGTP Tx and Rx packet drops. Basic checks to confirm this scenario:

1. You must check if there is any high CPU utilization for egtpinmgr.

```
Mar 25 14:30:48 10.224.240.132 evlogd: [local-60sec48.142] [resmgr 14907 debug] [6/0/10088 <rmmgr:60> _
Mar 25 14:31:05 10.224.240.132 evlogd: [local-60sec5.707] [resmgr 14907 debug] [6/0/10088 <rmmgr:60> _r
```

2. Check if the echo request/response is failing (command shared earlier).

3. Can check if there are any packet drops from the demux card.

All EGTP inbound traffic must through the same egtpmgr. If path failures are observed with one node, the volume of inbound traffic probably goes up. And, you can experience a traffic drop at the egtpmgr process level. Even the co-located process must proceed through the same egtpmgr queue and get impacted.

Here is the step to check packet loss that must be taken with multiple iterations

<#root>

```
debug shell card <> cpu 0

cat /proc/net/boxer
```

```
******** card1-cpu0 /proc/net/boxer *******

Wednesday March 25 17:34:54 AST 2020

what                    total_used next    refills     hungry  exhausted  system_rate_kbps      system_cr

bdp_rld           4167990936249KB  094   51064441        292          1  3557021/65000000 7825602KB/7934

what                  bhn local           remote                  ver         rx    rx_drop         tx

total cpu 34          *   *               *                         *    3274522         59         60

total cpu 35          *   *               *                         *    6330639         46        121

total cpu 46          *   *               *                         *    5076520         27      15524
```

```
total cpu 47         *   *                *                              * 4163101019      83922  133540922
```

4. Must capture egtpinmgr CPU profiler output if you see high CPU for egtpinmgr.

If all of the above conditions are valid, then you can check for the mentioned possible solution.

# Solution

1. Increase in EGTP echo timeout - If 5 seconds does not help, you can try 15 or 25. You can discuss this with your AS team to tune this.

2. Decrease peer-salvation timeout - The lower the timeout value, the lesser the number of inactive peers, so, you can change the time value with this command:

```
gtpc peer-salvation min-peers 2000 timeout 24
```

3. overload protection - overload protection optimization can be done based on the traffic trend because without knowing the exact incoming traffic rate before egpinmgr hits the problem, it is difficult to tune this. Also, wrong tuning can cause additional signaling traffic due to silent drops.

So, for overload protection optimization, you can collect some packet drops from the demux card for egtpinmgr and CPU profiler outputs as mentioned earlier.

4. No subscribers/sessions on the EGTPC link - when there are no sessions over a specific tunnel, the GTP echo functionality is stopped. If there is zero/no connected subscriber, GTPC echo must not be sent.

Here are the errors that you see when echo functionality is stopped:

```
2019-Jul-26+08:41:51.261 [egtpmgr 143047 debug] [1/0/4626 <egtpinmgr:2> egtpmgr_pm.c:798] [context: EPC
2019-Jul-26+08:41:51.261 [egtpmgr 143048 debug] [1/0/4626 <egtpinmgr:2> egtpmgr_pm.c:818] [context: EPC
```

# Workaround

You can try to restart the egtpinmgr task in order to recover. But, restarting the egtpinmgr can bring a short-term impact, unnoticeable to the end-user, while the NPU flows are reinstalled in the new task.

This operation must take less than 1 second to complete.

1. Disable the path failure detection:

```
    egtp-service S5-PGW
```

```
        no gtpc path-failure detection-policy
```

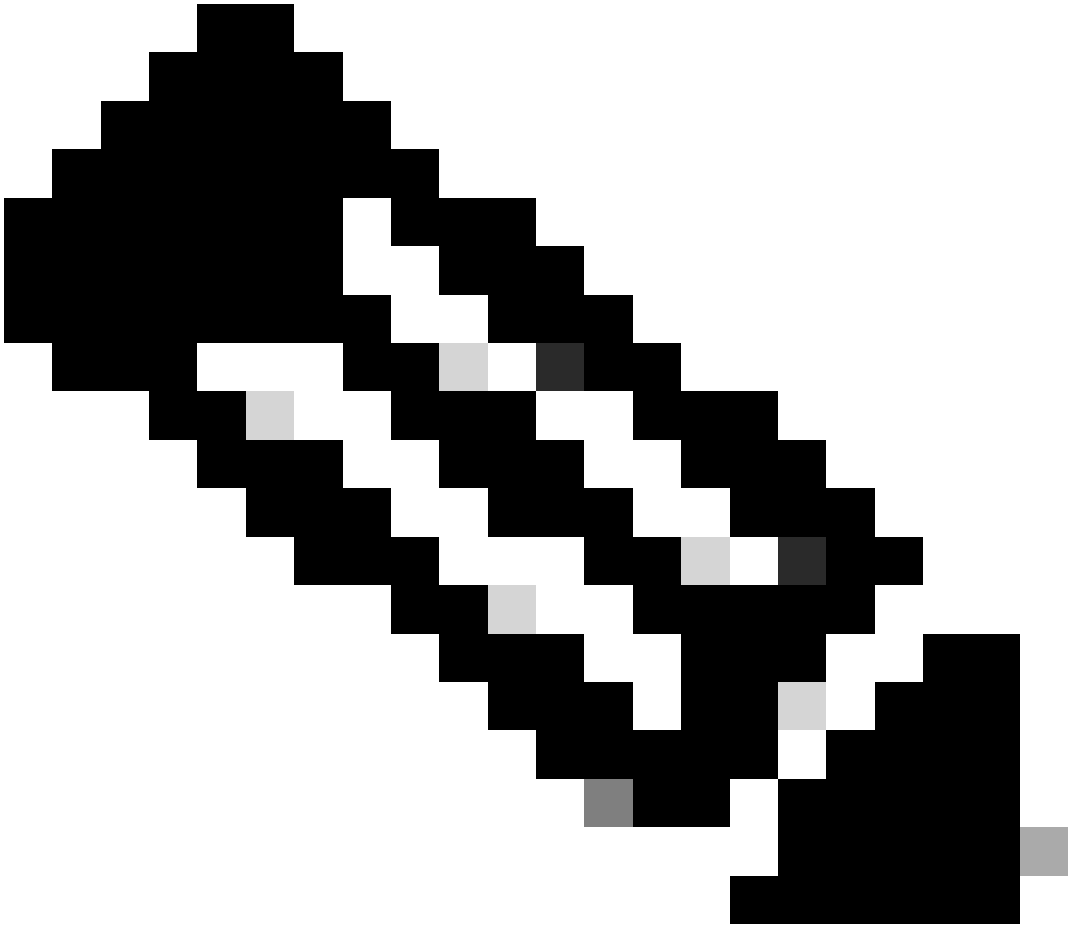
2. Kill egtpinmgr task:

```
task kill facility egtpinmgr  all
```

3. Enable the path failure detection:

```
    egtp-service S5-PGW
        gtpc path-failure detection-policy
```

---



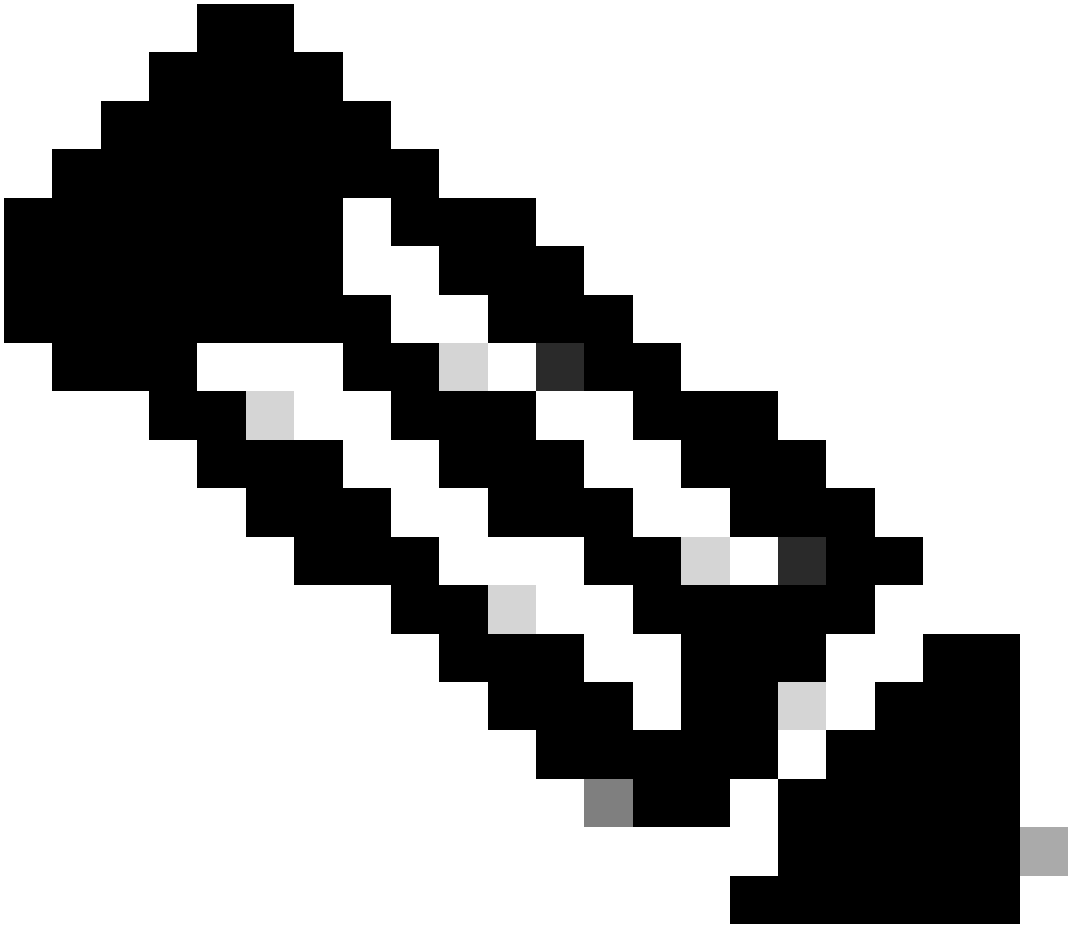
**Note**: This workaround must be implemented only in MW as it can cause some impact.

---

## Configuration Changes

Configuration in terms of DSCP/QOS/EGTP IP path/service mapping can be checked.

---



> **Note**: These are the major reasons contributing to EGTP path failures but in case none of the scenarios are found you can collect some traces and debugging logs further.

---

## Debugging Logs

(If required)

```
logging filter active facility egtpc level<critical/error/debug>
logging filter active facility egtpmgr level<critical/error/debug>
logging filter active facility egtpinmgr level<critical/error/debug>
```