Troubleshoot KPI Degradation - 4G ASR

Contents

Introduction

This document describes how to troubleshoot 4G Attach Success Rate (ASR) Key Performance Indicators (KPIs) degradation.

Possible Scenarios

4G ASR degradation can be caused by multiple factors:

1. Network Problems
2. Call Flow-Specific Issue
3. Node-Specific Issues
4. Configuration Issues
5. RAN End Issues

Logs Required for Initial Analysis

1. KPI trend graphs highlighting the degradation.
2. KPI formula used for measurement.
3. Raw bulkstat counters and cause code trends since the issue started.
4. Two instances of Show Support Details (SSD) captured at a 30-minute interval during the problematic time.
5. Syslogs collected from two hours before the degradation until the current time.
6. Capture these logs:
    - **Mon-sub/pro traces**
    - Logging monitor msid <imsi>

Troubleshooting Sequence

1. Identify the ASR formula:

```
1-((emm-msgtx-decode-failure+emm-msgtx-attach-rej-gw-reject+emm-msgtx-attach-rej-activation-reject+emm-m
```

> **Caution**: Formula varies based on the Customers way of measuring the KPIs.

---

2. Based on the formula, there are multiple counters used to calculate ASR, so from the bulkstats, you need to check the KPI trend of each counter.

3. KPI trend to be compared with non-problematic timelines and problematic timelines.

4. Once the problematic bulkstat counter is identified from the KPI formula, you need to check how this counter is defined based on flow and try to establish a pattern.

5. Also, collect the disconnect reasons from the node with multiple iterations with time intervals of 3 to 5 mins.

You can find the delta of disconnect reasons from two SSDs collected at different timestamps. The disconnect reason that increases rapidly from the delta disconnects can be attributed to the cause of KPI degradation. In addition, the description of all the disconnects is available in Ciscoâ€™s Statistics and Counters Reference; https://www.cisco.com/c/en/us/td/docs/wireless/asr_5000/21-23/Stat-Count-Reference/21-23-show-command-output/m_showsession.html.

```
show session disconnect-reasons verbose
```

Here is an example of troubleshooting steps to address a degradation scenario caused by an increase in the Disconnect Reason "MME-HSS-User-Unknown". Refer to https://www.cisco.com/c/en/us/support/docs/wireless/mme-mobility-management-entity/214633-troubleshoot-4g-asr-kpi-degradation-due.html.

6. Check egtp statistics based on the type of node.

```
--- SGW end -----

show egtpc statistics interface sgw-ingress path-failure-reasons
show egtpc statistics interface sgw-ingress summary
show egtpc statistics interface sgw-ingress verbose
show egtpc statistics interface sgw-ingress sessmgr-only

show egtpc statistics interface sgw-egress path-failure-reasons
show egtpc statistics interface sgw-egress summary
show egtpc statistics interface sgw-egress verbose
show egtpc statistics interface sgw-egress sessmgr-only

---- PGW end -----

show egtpc statistics interface pgw-ingress path-failure-reasons
show egtpc statistics interface sgw-ingress summary
show egtpc statistics interface sgw-ingress verbose
show egtpc statistics interface sgw-ingress sessmgr-only

--- MME end -----

show egtpc statistics interface mme path-failure-reasons
show egtpc statistics interface mme summary
show egtpc statistics interface mme verbose
show egtpc statistics interface mme sessmgr-only
```

7. To further analyze and troubleshoot the KPI degradation, capture mon-sub/mon pro call traces and consider using external tools to obtain Wireshark traces. These traces help identify the specific call flow that causes the problem.

Commands to capture Mon sub-traces are as follows:

```
monitor subscriber imsi <IMSI number>  ---------- verosity level +++++,A, S, X, Y, 19. 26, 33, 34, 35

More options can be enabled depending on the protocol or call flow we need to capture specifically
```

8. In cases where capturing traces like mon-sub is not possible due to a minimal percentage of KPI degradation, capture system-level debug logs. Also, capture debug logs for sessmgr and egptc, and if the suspected issue involves entities like HSS/RAN, capture debug logs for s1-ap/diameter based on the specific problem.

```
logging filter active facility sessmgr level debug
logging filter active facility egtpc level debug
logging filter active facility diameter level debug ----- depending on scenario
logging filter active facility s1-ap evel debug     ----- depending on scenario

logging active ----------------- to enable
no logging active ------------- to disable

Note :: Debugging logs can increase CPU utilization so need to keep a watch while executing debugging lo
```

9. Once you get any clue from debuglogs, then you can also capture corefile for that particular event where you see error logs:

```
logging enable-debug facility sessmgr instance <instance-ID> eventid 11176 line-number 3219 collect-core

For example  :: consider we are getting below error log in debug logs which we suspect can be a cause of
and we don;t have any call trace

[egtpc 141027 info] [15/0/6045 <sessmgr:93> _handler_func.c:10068] [context: MME01, contextID: 6] [softw

So in this error event

facility :: sessmgr
event ID = 141027
line number = 10068
```

These are the various steps to troubleshoot this issue.