



## RTLLI Management for 2G M2M Devices

---

- [Feature Description, on page 1](#)
- [How It Works, on page 1](#)
- [Configuring RTLLI Management, on page 1](#)
- [Monitoring and Troubleshooting, on page 2](#)

### Feature Description

Fixed Random TLLI (RTLLI) Management for 2G M2M devices is intended to expand the operator's control of TLLI (temporary logical link identifier) in the following scenario:

When multiple M2M devices attempt PS Attaches, with the same fixed RTLLI coming from different NSEIs (network service entity identifier), the SGSN cannot distinguish between the devices. The SGSN functions *as if* the first device bearing an RTLLI is no longer attached and begins to communicate with the next device using that same RTLLI. With multiple M2M devices attempting attaches - all with the same RTLLI - the result is TLLI collision and dropped calls.

### How It Works

This feature deals with Attach problems due to simultaneous IMSI attaches, all with the same fixed RTLLI.

Beginning with Release 16.3, it became possible to configure the SGSN to discard/drop Attach Request messages received from an MS with an RTLLI already in use on the SGSN by adding validation of the NSEI. Attach gets processed if the attach is coming from a different NSEI. This functionality is disabled by default.

Beginning with Release 19.3, to further reduce jumbling of authentication vectors across subscribers, the Fixed Random TLLI Handling mechanism extends the functionality noted above. A new verification table has been added to the GbMgr. The table maintains a list of TLLI + NSEI and if an incoming Attach Request includes a TLLI + NSEI already on the table then the call is dropped. This functionality is disabled by default.

### Configuring RTLLI Management

No new commands or keywords have been added to the command line interface (CLI) in support of Fixed Random TLLI Management. Enabling / disabling this mechanism is integrated into existing CLI.

For information about the commands, parameters and parameter values, please check your *Command Line Interface Reference* manual for each of the commands listed below.



### Important

The following configurations should be performed during system boot up. It is not advisable to enable/disable this TLLI management functionality during runtime.

### Verifying Both the RTLLI and the NSEI

To enable the SGSN to handle Attach Requests with the same fixed RTLLI by verifying both the RTLLI and the NSEI, use the following configuration:

```
config
sgsn-global
gmm-message attach-with-tlli-in-use discard-message only-on-same-nsei

old-tlli invalidate tlli hex_value
old-tlli hold-time time
end
```

Notes:

- **only-on-same-nsei** - This keyword is required to enable this new verification mechanism.

### Verifying Only the RTLLI

To enable the SGSN to handle Attach Requests with the same fixed RTLLI by verifying only the RTLLI, use the following configuration:

```
config
sgsn-global
gmm-message attach-with-tlli-in-use discard-message
old-tlli invalidate tlli hex_value
old-tlli hold-time time
end
```

Notes:

- **only-on-same-nsei** - Do not include this keyword to disable this new verification mechanism. The system defaults to the verification mechanism provided with Release 16.3 (see *How It Works*).

### Verifying Configuration

To verify if the functionality is enabled or disabled, use the following commands from the Exec mode:

```
show configuration | grep gmm-mess
show configuration | grep old-
show configuration verbose | grep old-
```

## Monitoring and Troubleshooting

This section provides information for monitoring and/or troubleshooting the RTLLI Management functionality.

To see the statistics of attach drops that are due to same-RTLLI collisions, execute the show commands listed below. When you are looking at the generated statistics, consider the following:

- If the generated counter values are not increasing then collisions are not occurring.
- If the generated counter values are increasing then it means collisions are occurring and attaches were dropped.

### Configured to Verify Both RTLLI and NSEI

If `gmm-message attach-with-tlli-in-use discard-message only-on-same-nsei` is configured then the following show command can give the drop count of attaches caused by same RTLLI and NSEI:

```
show gbmgr all parser statistics all | grep use
```

```
IMSI Key: 1487 P-TMSI Key: 0 attach with tlli in use : 592 <-- drops from existing table
with RTLLI+NSEI
Add P-TMSI Key: 0 attach drop tlli in use(pre tlli check): 297 <-- drops from new table
with RTLLI
```

```
IMSI Key : 1190 P-TMSI Key : 594 attach with tlli in use : 395
Add P-TMSI Key : 0 attach drop tlli in use(pre tlli check) : 198
```

### Configured to Verify Only RTLLI

If "`gmm-message attach-with-tlli-in-use discard-message`" is configured then the following show command can give the drop count of attaches caused by same RTLLI:

```
show gbmgr all parser statistics all | grep use
```

```
IMSI Key: 1487 P-TMSI Key: 0 attach with tlli in use : 592 <-- drops from existing table
with RTLLI
Add P-TMSI Key: 0 attach drop tlli in use(pre tlli check): 297 <-- drops from new table
with RTLLI
```

```
IMSI Key : 1190 P-TMSI Key : 594 attach with tlli in use : 395
Add P-TMSI Key : 0 attach drop tlli in use(pre tlli check) : 198
```

### Verify Attach Rejects due to Same RTLLI

The following show command generates SessMgr counters that track the Attach Rejects due to same RTLLI collision:

```
show gmm sm stats | grep Same random tlli collision
```

```
Same random tlli collision: 10
```

Beginning with Release 19.3.5, the '`sgsn-implicit-detach(237)`' session disconnect reason pegs when the 2G-SGSN rejects the Attach Request due to same RTLLI collision.

Beginning in Release 19.4, the following show command identifies the two bulk statistics the SGSN uses to track the number of times the SGSN rejects Attach Requests or Combined Attach Requests due to same RTLLI collision.

```
show bulkstats variables sgsn | grep colli
%2G-simple-attach-rej-randomtlli-collision%          Int32    0    Counter
%2G-combined-attach-rej-randomtlli-collision%       Int32    0    Counter
```

