

# Load and Overload Management


- Feature Summary and Revision History, on page 1
- Feature Description, on page 2
- SBA Interface Overload Control, on page 3
- GTP-C Load and Overload Control, on page 8
- N4 Interface Load and Overload Control , on page 16
- Configuring Load and Overload Control on SMF, on page 20
- Node Overload, on page 28
- Monitoring and Troubleshooting, on page 29


# Feature Summary and Revision History

## Summary Data

***Table 1: Summary Data***

| | |
|---|---|
| Applicable Products or Functional Area | SMF |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled - Configuration Required |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

## Revision History

***Table 2: Revision History***

| Revision Details | Release |
|---|---|
| Added support for GTPC peer overload control | 2021.02.3 |
| Added support for GTPC load and overload control | 2021.02.0 |

| Revision Details | Release |
|---|---|
| Added support for message priority configuration. | 2020.04.0 |
| First introduced. | 2020.03.0 |

# Feature Description

*Table 3: Feature History*

| Feature Name | Release Information | Description |
|---|---|---|
| Load and Overload Control over SBI, GTP-C, and N4 Interfaces | 2024.04.1 | This feature handles the load and overload control mechanism for GTP-C, N4, and SBA Interfaces.<br>This feature improves the network robustness by considering the load and overload status of self and the peer nodes.<br>**Commands Enhanced:**<br>• **profile overload-exclude** *overload_exclude_profile_name* **message-priority [ n4 \| n7 \| n10 \| n11 \| n16 \| n40 \| s5 ] upto** *message_priority*<br>• **profile overload-exclude** *overload_exclude_profile_name* **procedure-list [ session-delete \| new-call \| xnho \| modify \| chf-reauth \| inter-rat-ho \| intra-rat-ho \| imexit-nw \| imexit-ue \| usar ]**<br>**Default Settings**: Disabled—Configuration Required to enable |

The SMF provides mechanisms to manage the overload and congestion that occur on the SMF and Service-based Architecture (SBA). The SMF receives ingress messages at a rate higher than the engineered capacity. The internal queues on the SMF may experience a higher utilization level than the configured level. This scenario may occur on the SBA servers, directly or indirectly, due to overloaded traffic from the network or from the SMF.

**Important** The PGW-C term used in this chapter denote the EPS interworking functionality supported by SMF and must not be assumed as a standalone P-GW that is used in the LTE network.

# SBA Interface Overload Control

## Feature Description

An interface handles only a specified number of incoming requests. When the incoming requests exceed the specified numbers, the interface overloads. For example, an interface is overloaded when:

- A network element failure exists that causes large number of re-attaches
- Multiple users perform location update or transition from idle to active mode frequently

Overloading causes the interface to either drop the requests or delay processing the request. The overall network performance degrades due to the overloading at the interface. This scenario can lead to node congestion, failure, or collapse which in turn causes load increase on the other nodes.

The SMF measures different resources and defines the load based on those measurements. Also, the SMF updates the NRF about the load. Currently, the SMF applies overload protection on inbound messages. The external nodes throttle towards the SMF to come out of a congestion when overload protection is applied on the inbound interface (SBA Interface).



**Note** The scope of this feature is only on overload due to inbound requests on SBA interface.

## How it Works

The SMF protects inbound requests from overloading at Endpoint and Application levels.

- Endpoint Level—The protection is based on the HTTP request method without taking the message type into account.
- Application Level—The protection is based on the message type.

### Message Priority

The SMF applies the overload protection on the incoming request messages after evaluating the resources availability to process the request and the message priority. The high priority messages get the lower preference to throttle, and low-priority messages get higher preference. An overloaded NF applies the message prioritization schemes on the incoming messages during an overloaded condition. In such conditions, the NF excludes the messages of the highest priority from the overload protection mechanism.

Once you configure message priority, SMF starts classifying the messages based on their priority. This configuration is optional. If you chose not to use this configuration, SMF applies the overload protection technique without considering the message priority.

## Overload Protection at Endpoint

For endpoints, the SMF offers overload protection at both the endpoint and client levels. The SMF defines the overload threshold limits for the inbound request messages. Based on the threshold range, the SMF can

reject the inbound request messages. The SMF sends back an HTTP response with the configured status to the request initiator.

The following are the overload threshold limits defined in the SMF:

- **Low** – When this threshold is met, only the POST method (with generic URI contributing to resource allocation) is rejected.
- **High** – All messages are rejected with the configured (reject) statuses when this threshold is met.
- **Critical** – All messages are rejected with the configured (reject) statuses when this threshold is met.

# Configuring Overload Protection

This section describes the configuration procedures involved in configuring the overload protection for inbound request messages.

## Configuring Overload Protection at Endpoint Level

Use the following configuration to configure overload protection at endpoint level.

```
config
   instance instance-id gr_instance_id
      endpoint sbi
         overload-control threshold threshold_limit threshold_range action
action_status action_code range
         commit
         end
```

**NOTES**:

- **overload-control**: Specify the overload control at endpoint level.
- **threshold** : Specify the threshold limit and range.
- *threshold_limit*: Specify the threshold limit. *threshold_limit* must be one of the following:
  - *low*: Specify the low threshold limit for overload protection.
  - *high*: Specify the high threshold limit for overload protection.
  - *critical*: Specify the critical threshold limit for overload protection.
- *thershold_range*: Specify the threshold range. *threshold_range* must be an integer in the range of 10 – 100000.
- **action** : Specify the action to be taken for the threshold limit.
- *action_status*: Specify the action for the threshold limit. *action_status* must be:
  - **reject**: Reject the inbound messages if the specified threshold range is met.
- *action_code*: Specify the action status code. *action_code* must be:
  - **reject-code**: Specify the reject status code.

- *range*: Specify the range of the action code. *range* must be an integer in the range of 100 – 600.

The following is an example configuration:

```
overload-control threshold low 500 action reject reject-code 501
overload-control threshold critical 10000 action reject reject-code 329
```

## Configuring Overload Protection at Client Level

Use the following sample configuration to configure overload protection at client level.

```
config
   instance instance-id gr_instance_id
      endpoint sbi
        overload-control client threshold threshold_limit threshold_range action
 action_status action_code range
        commit
        end
```

**NOTES**:

- **overload-control client**: Specify the overload control at client level.
- **threshold** : Specify the threshold limit and range.
- *threshold_limit*: Specify the threshold limit. *threshold_limit* must be one of the following:
  - *low*: Specify the low threshold limit for overload protection.
  - *high*: Specify the high threshold limit for overload protection.
  - *critical*: Specify the critical threshold limit for overload protection.
- *thershold_range*: Specify the threshold range. *thershold_range* must be an integer in the range of 10 – 100000.
- **action** : Specify the action to be taken for the threshold limit.
- *action_status*: Specify the action for the threshold limit. *action_status* must be:
  - **reject**: Reject the inbound messages if the specified threshold range is met.
- *action_code*: Specify the action status code. *action_code* must be:
  - **reject-code**: Specify the reject status code.
- *range*: Specify the range of the action code. *range* must be an integer in the range of 100 – 600.

The following is an example configuration:

```
overload-control client threshold low 50 action reject reject-code 329
overload-control client threshold critical 20000 action reject reject-code
501
```

## Verifying the Overload Protection Configuration

Use the **show running-config** command to view the overload protection configuration in the SMF Ops Center. The following is a sample output of the **show running-config** command.

```
[cluster1/data] example# show running-config
instance instance-id 1
 endpoint sbi
  overload-control threshold low 5000 action reject reject-code 555
  overload-control threshold high 7000 action reject reject-code 329
  overload-control threshold critical 10000 action reject reject-code 503
  overload-control client threshold low 750 action reject reject-code 329
  overload-control client threshold high 500 action reject reject-code 329
  overload-control client threshold critical 1000 action reject reject-code 503
  interface n11
  overload-control threshold low 4000 action reject reject-code 555
  overload-control threshold high 6000 action reject reject-code 329
  overload-control threshold critical 7000 action reject reject-code 503
  overload-control client threshold low 500 action reject reject-code 329
  overload-control client threshold high 700 action reject reject-code 329
  overload-control client threshold critical 800 action reject reject-code 503
 exit
exit
```

## Configuring the Message Priority

Use the following configuration to configure message priority for the inbound request messages.

**config**
**overload-control threshold** *threshold_limit threshold_range* **action reject reject-code** *range* **exclude message-priority** *priority_value*
**end**

**NOTES:**

- **overload-control** – Specify the overload control at endpoint level.
- **threshold***threshold_limit* – Specify the threshold limit and range.
- – Specify the threshold limit. *threshold_limit* must be one of the following:
  - low – Specify the low threshold limit for overload protection.
  - high – Specify the high threshold limit for overload protection.
  - critical– Specify the critical threshold limit for overload protection.
- *thershold_range* – Specify the threshold range. *thershold_range* must be an integer in the range of 10–100000.
- **action** – Specify the action to be taken for the threshold limit.
- *action_status* – Specify the action for the threshold limit.*action_status* must be:
  - **reject** – Rejects the inbound messages if the specified threshold range is met.
- **exclude message-priority** – Excludes the messages from the overload protection mechanism depending on the assigned priority.
- *priority_value* – Specifies the priority value.

The following is an example configuration:

```
overload-control threshold low 1000 action reject reject-code 100 exclude
message-priority 8

overload-control threshold high 2000 action reject reject-code 100 exclude
message-priority 5
```

If the priority value is 8, then the messages received with priority 8 or higher are not throttled. This applies even when the system threshold is lower than the priority value. The 3GPP defined message priority is 0–31 as per *3GPP TS 29.500 version 15.4.0*.

# Self-Overload Handling for SBA Interface Messages

SMF fetches load factor information of the system periodically from app-infra. SMF then compares this load factor with the thresholds configured in the overload profile. Based on the load status, SMF triggers the load or overload control mechanism toward SBA interface messages.

According to the configured threshold values, there are three possible load and overload states:

| Load or Overload State | SMF Action |
|---|---|
| Normal | System is under normal load condition. Load Control applied as per configuration. |
| Overloaded | System is overloaded. Overload control applied as per configuration. |
| Self-protection | The load status crossed the threshold of overload. Need to self-protect. SMF applies message throttling or exclusion as per configuration. |

## Self-Load Information Communication to NRF

In addition to the load factor calculation, SMF sends the load information toward NRF when SMF detects a change in load factor derived by OAM pod. SMF calculates and shares the load information even if the load configuration is not enabled.

SMF sends the **NF Update Patch API** using the HTTP Patch request to perform the partial NF profile update. This partial update is used to add, delete, or replace any of the services offered by the NF Instance.

NF Update Patch contains the following two parameters:

- **load:** It indicates the current load percentage of the NF. Its value ranges between 0-100.
- **loadTimeStamp:** It indicates the point in time in which the latest load information was generated at the NF instance.

## Self-Overload Throttling and Exclusions of SBI Messages

SMF performs self-overload throttling and exclusion only for incoming SBI messages. In a self-protection mode, SMF throttles all the incoming SBI messages unless the subscriber is configured under the overload exclude profile.

To configure the overload exclude profile for SBI interfaces, see Create Exclude Profile section.

When SMF is in an overload condition and it needs to reject the HTTP requests, it rejects the incoming SBI messages with an HTTPS status code **503 Service Unavailable**. Upon receiving this code, the receiving NF

compares the rejected and timed-out traffic with the accepted traffic. Based on this analysis, the receiving NF diverts or throttles the traffic sent toward SMF.

### Backoff Timer for SBI Messages

In the self-protection state, if the backoff timer is configured and SMF rejects the **N11 SM Context Create** and **N11smcontextModify** messages, then SMF includes the configured backoff interval in DNN profile level in N1 container with below N1msgtypes:

- PDU session establishment reject
- PDU session modification reject

To configure the backoff timer for PDU session establishment and modification messages, see Enabling Message-level Back-off Timer section.

Upon throttling the N11, N16, N40, N10 response messages, the response messages include one of the following cause codes:

- DNN_CONGESTION
- NF_CONGESTION
- Notify Failure
- Retrieve Failure

**Note** The N11N1N2MessageTransferFailNotificationReq message is not throttled in any situation.

## Limitations

Following are the known limitations of load and overload support for SBA interface messages:

- All incoming SBI requests or clearsub requests that initiates PDU session release and flow release shall not be throttled.
- Self-protection handing is not supported for Gx, Gy, Gz, Radius, Diameter (AAA) messages.
- Currently, SMF does not support advertising of load or overload information to peers over SBA interfaces.

# GTP-C Load and Overload Control

## Feature Description

**Important** The GTP-C Load and Overload Control is an optional feature.

The SMF uses the system load information to determine the operating status of the resources of the GTP-C entity. This information, when sent to the GTP-C peers, helps to balance the session load adaptively across entities supporting the same function based on their effective load.

A GTP-C overload occurs when the number of incoming requests exceeds the maximum request throughput supported by the receiving GTP-C entity. The GTP-C is over UDP transport, and it relies on the retransmissions of unacknowledged requests. When a GTP-C entity experiences overload (or severe overload), the number of unacknowledged GTP-C messages exponentially increase leading to a node congestion or collapse. An overload or a node failure leads to an increase of the load on the other nodes in the network.

Overload of the core network nodes in the network results in service degradation. Improved load distribution over the network helps in addressing the overload issue.

Overload conditions can occur in various network scenarios. The following are some examples of GTP-C signaling-based scenarios which lead to GTP-C overload:

- A traffic flood resulting from the failure of a network element, inducing a signaling spike.
- A traffic flood resulting from many users performing TAU or RAU or from frequent transitions between idle and connected modes.
- An exceptional event locally generating a traffic spike, for example, many calls (and dedicated bearers) being set up almost simultaneously.
- Frequent RAT reselection due to scattered non-3GPP (for example, Wi-Fi) coverage or a massive mobility between a 3GPP and non-3GPP coverage. This operation may potentially cause frequent or massive intersystem change activities.

GTP-C overload may result in any of the following service impacts:

- Emergency call drops
- Loss of PDN connectivity (IMS, Internet, and so on) and associated services.
- Loss of ability to set up and release radio and core network bearers necessary to support services, for example, GBR bearers.
- Loss of ability to report the change in—
  - User information, for example, location information for emergency services and lawful intercept
  - RAT or QoS
- Billing errors which result in loss of revenue.

GTP-C Load and Overload Control is a standards-driven feature. For standards compliance information, see the Standards Compliance, on page 14 section in this feature chapter.

GTP-C Load Control and Overload Control are complimentary concepts which can be supported and activated independently on the network.

This feature works both in a standalone deployment of SMF and an integrated deployment with cnSGWc.



**Note** This feature works only when the SMF interworks with PGW-C (that is, the EPS network). The term "SMF" used in this chapter denotes the combination of both SMF and PGW-C.

# GTP-C Load Control

This feature enables cnSGWc and PGW-C to gather and send Load Control Information (LCI) to GTP-C peers (for example, MME via cnSGWc, and ePDG). In broad terms, GTP-C load control denotes a preventive action and GTP-C overload control indicates a corrective action.

The advantages of enabling GTP-C Load Control are as follows:

- Load control allows better balancing of the session load; this mechanism prevents the GTP-C overload scenario.
- LCI helps to balance the session load adaptively across entities supporting the same function according to their effective load.
- Load control does not trigger overload mitigation actions even if the GTP-C entity reports a high load.

# GTP-C Overload Control

This feature enables cnSGWc and PGW-C to gather and send Overload Control Information (OCI) to GTP-C peers (for example, MME via cnSGWc, and ePDG). A GTP-C entity is in overload when it operates over its signaling capacity, which results in diminished performance.

The advantages of enabling GTP-C Overload Control are as follows:

- Avoids overloading of GTP-C entity
- Improves load distribution on SMF and cnSGWc which in turn reduces the occurrence of SMF overload.
- Aims at shedding the incoming traffic as much as possible when an overload has occurred

# Message Throttling

### Ingress Messages

GTP-C entity uses traffic reduction metric information in the Overload Control Information (OCI) for message throttling. To mitigate overload scenario, the GTP-C entity reduces the ingress message flow towards the overloaded peer based on the metric information.

When a node is in self-protection mode, the SMF rejects the ingress GTP-C messages based on the message throttling exclude configuration. For details on the exclude profile configuration, see the Create Exclude Profile, on page 21 section.

### Egress Messages

To mitigate the GTP-C overload scenario, the SMF controls the egress message flow towards the overloaded GTP-C peer based on the information received within the OCI.

The SMF rejects the egress messages towards the GTP-C peers based on the exclude profile configuration. Exclusion profile contains the DNN list, 5QI list, ARP list, and priority corresponding to the messages to be excluded from throttling.

Peer overload control for GTP-C interface can be configured through the **profile overload** *profile-name* **peer-level interface gtpc action throttle** command.



**Important** Message throttling applies only to the initial messages. The SMF does not throttle the triggered request or response messages as it might result in retransmission of the corresponding request message.

For throttling, the SMF uses the loss algorithm as specified in 3GPP 29.274.

Message groups are formed based on the category of procedures mentioned in 3GPP 29.274, section 12.3.9.3.2. The following are the peer overload groups for message throttling.

- Group 1 corresponds to update of existing resources. This group includes the Update Bearer Request message.
- Group 2 corresponds to creation of new resources. This group includes the Create Bearer Request message.

Message groups allow the user to configure, in percentage, how many number of messages SMF is expected to generate in each message group. The default value for both the groups are 50%. The default value 50% means that, out of 100 outgoing messages, 50 messages are update bearer requests (group 1) and 50 messages are create bearer requests (group 2).

If the peer is overloaded and the overload reduction matrix is 30%, then the SMF throttles 30 create bearer request messages and sends all the remaining messages.

If the peer is overloaded and the overload reduction matrix is 70%, the SMF throttles 50 create bearer request messages and 20 update bearer request messages and sends the remaining 30 update bearer messages.

## Overloaded Peer Detection

The SMF determines whether or not the GTP-C peer entity is overloaded based on the received Overload Control Information (OCI) IE information in any of the following GTP-C messages.

- Create Session Request
- Create Bearer Response
- Modify Bearer Request
- Update Bearer Response
- Delete Session Request
- Delete Bearer Response
- Modify Bearer Command
- Delete Bearer Command
- Bearer Resource Command

Note that all the GTP-C messages include the OCIs of cnSGWc, MME or S4-SGSN, and TWAN or ePDG except for Delete Bearer Command and the Bearer Resource Command messages.

The SMF receives OCI that corresponds to multiple GTP-C entities in a single message (for example, the OCI of cnSGWc and MME or S4-SGSN). The SMF service pod parses and stores all such OCI IEs received in a single message.

The SMF considers the GTP-C peer as overloaded when one of the following conditions is met.

- the validity period of the OCI expires

- the OCI is received as 0

In the case of geo redundancy (GR), it is expected that fresh cache records are built by the new instance with time based on OCI received from new messages.

When the primary cache pod is inactive, the secondary cache pod becomes active and serves all the cache requests.

# How it Works

This section describes the detailed working mechanism of this feature.

1. The SMF fetches the system load periodically from the Application infrastructure. For details on load calculation, see the Node Overload, on page 28 section in this chapter.
2. The SMF identifies the current node overload state based on thresholds configured in the Overload Profile, and the system load value.
3. The SMF applies the overload control mechanism on the incoming request messages based on the node overload states.

| Node Overload State | Definition | Criteria | Overload Control Action |
|---|---|---|---|
| Normal | The system is not under any overloaded condition. | Load < Minimum tolerance | Applies GTP-C Load Control |
| Overloaded | The system is overloaded. | Minimum tolerance <= Load < Maximum tolerance | Applies GTP-C Overload Control |
| Self-protection | The system has reached the extreme limits of overload. | Load >= Maximum tolerance | Applies Message Throttling |

### GTP-C Load Control Mechanism

The SMF communicates the LCI to the GTP-C peers (for example, MME or ePDG) upon meeting the following conditions:

- If the feature is enabled through the **profile load** *load_profile_name* **interface gtpc action advertise** command
- If the load profile and overload profile are associated with the SMF profile
- If the LCI is never sent to the peer
- Periodically as per the configuration **profile load** *load_profile_name* **advertise interval** *lci_broadcast_interval*
- If the difference between current load value and last indicated load value is greater than the configured change factor **profile load** *load_profile_name* **advertise change-factor** *load_value_change_factor*

**Note** The SMF exchanges LCI through GTP-C request and response messages without triggering extra signaling.

The SMF includes the LCI in the following messages:

- Create Session Response
- Create Bearer Request
- Modify Bearer Response
- Delete Bearer Request
- Delete Session Response
- Update Bearer Request

For message formats and LCI IE details, see the 3GPP TS 29.274 specification, version 15.4.0.

### GTP-C Overload Control Mechanism

The SMF calculates and sends the overload metric based on the load value and the overload reduction-metric configuration. The SMF then communicates the OCI to the GTP-C peers upon meeting the following conditions:

- If the feature is enabled through the **profile overload** *overload_profile_name* **node-level interface gtpc action advertise** command
- If the OCI is never sent to the peer
- Periodically as per the configuration **profile overload** *overload_profile_name* **node-level advertise interval** *oci_broadcast_interval*
- If the difference between current reduction-metric and last indicated reduction-metric is greater than the configured change factor **profile overload** *overload_profile_name* **node-level advertise change-factor** *overload_value_change_factor*
- If the validity timer expires and the SMF is still in overloaded state

**Note** The SMF exchanges OCI through GTP-C request and response messages without triggering extra signaling.

The SMF includes the OCI in the following messages:

- Create Session Response
- Create Bearer Request
- Modify Bearer Response
- Delete Bearer Request
- Delete Session Response
- Modify Bearer Failure Indication
- Update Bearer Request

- Delete Bearer Failure Indication

For message formats and OCI IE details, see the *3GPP TS 29.274 specification, version 15.8.0.*

#### Message Throttling

In the self-protection mode, the SMF rejects the ingress GTP-C messages with failure cause set to "GTP-C Entity Congestion" as per the self-protection exclusion configuration.

## Standards Compliance

The GTP-C Load and Overload Control feature complies with the following standards:

- *3GPP TS 29.807, version 12.0.0*
- *3GPP TS 29.274, version 15.8.0*

## Limitations

The GTP-C Load and Overload Control feature has the following limitation:

- Allows configuration of only one load profile and one overload profile

# OAM Support for GTP-C Load and Overload Control

This section describes operations, administration, and maintenance information for this feature.

## Bulk Statistics Support

The SMF maintains the following metrics as part of this feature.

- **node_lci_metric**

  Description: This counter indicates the current load (LCI) value at the node level that is, SMF with PGW-C.

  Metrics Type: Gauge

  Labels:

  - app_name
  - cluster
  - data_center
  - instance_id

- **node_oci_metric**

  Description: This counter indicates the current overload (OCI) value at the node level.

  Metrics Type: Gauge

  Labels:

  - app_name

- cluster
- data_center
- instance_id

- **node_overload_status**

  Description: This counter indicates the current overloaded status at the node level.

  - 0 - Normal
  - 1 - OverLoaded
  - 2 - SelfProtection

  Metrics Type: Gauge

  Labels:

  - app_name
  - cluster
  - data_center
  - instance_id

- **smf_inc_msg_throttling_stats**

  Description: This counter provides the number of incoming messages throttled on each interface in self-protection mode.

  Metrics Type: Counter

  Labels:

  - app_name
  - cluster
  - data_center
  - instance_id
  - interface
  - message_type
  - cause

- **smf_og_msg_throttling_stats**

  Description: This counter provides the number of outgoing messages throttled on each interface when peer entity is overloaded.

  Metrics Type: Counter

  Labels:

  - app_name

- cluster
- data_center
- gr_instance_id
- instance_id
- interface
- message_type
- service_name
- throttled_target_peer_type
- cause

# N4 Interface Load and Overload Control

SMF handles the incoming or outgoing N4 messages based on the load and overload status of the associated UPF node. SMF performs the following functions to control the load and overload scenarios over the N4 interface:

- **Self-Overload Throttling and Exclusion of N4 Messages**: In a self-protection state, SMF throttles or excludes the incoming or outgoing messages from throttling as per the configurations.
- **Peer Load and Overload Handling for N4 Messages**: The N4 outgoing messages are processed or throttled based on the load or overload conditions of the peer.

## Load and Overload Reporting for N4 Interface

SMF takes the following actions based on the profile associated under SMF profile:

| Profile Associated | Action Taken |
|---|---|
| Load Profile | SMF periodically fetches the load factor from cache POD. SMF sends the CP Function LOAD toward the UPF. |
| Overload Profile | SMF periodically fetches the overload parameters (such as overload reduction value and overload state of the system) from cache POD. SMF sends the CP FunctionOVRL toward the UPF. |

## Self-Overload Throttling and Exclusion of N4 Messages

SMF performs self-overload throttling and exclusion of N4 messages for the following types of messages:

- Incoming Messages
- Outgoing Messages

- Session Report DLDR

### Self-Overload Throttling and Exclusion for Incoming N4 Messages

In the self-protection state, as a default behavior, SMF throttles all the incoming N4 messages with the following congestion cause code:

| Rejection Cause Code | Description |
|---|---|
| PFCP_ENTITY_IN_CONGESTION | SMF returns this cause code when a PFCP entity detects a node level congestion and performs the overload control. This cause code does not allow the request to be processed. |

This feature also allows SMF to exclude the N4 messages from throttling based on configured message priority under the overload exclude profile. If the message priority is configured for the N4 interface, the messages are processed even in overload scenarios.

If the overload exclude profile is configured, in the self-protection mode, SMF compares all the incoming N4 messages with the configured parameters (such as DNN, 5QI, ARP, Message Priority, and Category of Message). In case of any match, the messages are excluded from throttling on the SMF.

To configure the overload exclude profile for N4 messages, see the Create Exclude Profile section.

### Self-Overload Throttling and Exclusion for Outgoing N4 Messages

In the self-overload condition, the outgoing messages on the N4 interface from SMF are throttled indirectly as throttling is applied on the trigger messages at SBI and GTPC interfaces. SMF processes the messages that are excluded from throttling at the triggering NFs.

### Self-Overload Throttling for DLDR Messages

*Figure 1: Call Flow for Handling N4 DLDR during SMF Overload*

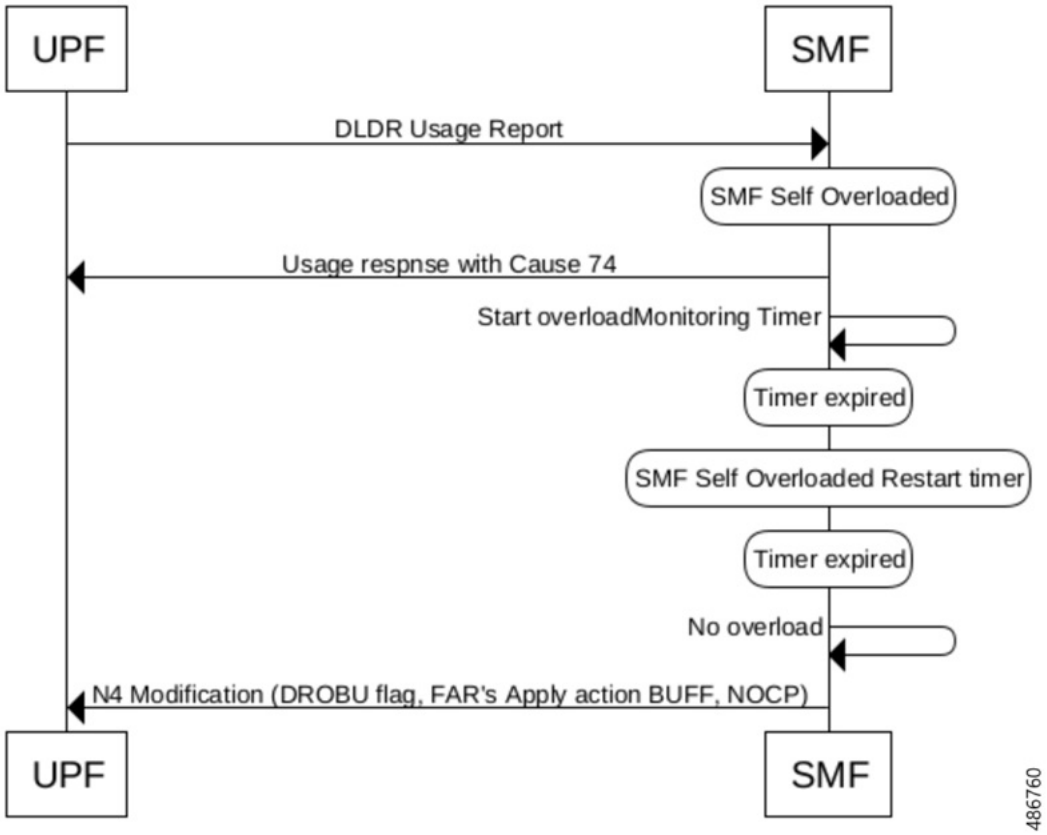



*Table 4: Call Flow Description for Handling N4 DLDR during SMF Overload*

| Step | Description |
|---|---|
| 1. | SMF receives DLDR usage report from UPF. |
| 2. | If SMF is in an overloaded (self-protection) state, it throttles the incoming DLDR message and sends a Usage Response message with the **Congestion Cause Code 74 PFCP_ENTITY_IN_CONGESTION** toward the UPF. |
| 3. | SMF then starts the overload monitoring timer. |
| 4. | After the timer expires, if the SMF is still in an overloaded state, SMF restarts the overload monitoring timer. |
| 5. | After the timer expires, if the SMF is not in an overloaded state, the SMF sends a N4 Modification message toward the UPF. This message contains a DROBU flag and sends apply action as BUFF, NOCP for downlink FARs.<br>Before the timer expires, if the SMF moves out of idle mode, then overload monitoring timer stops and N4 modification is not sent with DROBU flag. |

# Peer Load and Overload Handling for N4 Interface

The Control Plane (CP) informs the User Plane (UP) about the load or overload profile associated with the SMF. Based on this information, UP decides to send load or overload information toward CP peer.

SMF parses all the incoming N4 messages and N4 responses for LCI and OCI IEs. Whenever SMF receives LCI IE or OCI IE from a peer node, SMF learns about the load and overload parameter of that node. SMF considers a peer node as overloaded, if it receives OCI IE in any of the N4 messages.

The load value received from the UPFs is used to control the session load from CP. This load information is used to select UPF to balance the sessions across UPFs. The overload information is used for UPF selection and for throttling the messages toward the UPF.

SMF handles peer load and overload for the following two message types:

- Outgoing messages
- Incoming messages

### Peer Load and Overload Handling for Outgoing N4 Messages

If a peer is selected for session establishment or session modification, throttling will be applied based on the OCI Reduction Factor conveyed by the peer/UPF node in OCI IE. The outgoing N4 messages are not throttled by the peer node if the sessions meet the exclusion criteria based on the peer exclusion profile.

The exclusion profile may contain the DNN List, 5QI List, ARP List, and Message Priority parameters corresponding to the messages to be excluded from throttling.

**Note** N4 Establishment Requests and N4 Modification request are throttled, but N4 Delete Requests are not throttled.

### UPF Selection based on Peer Load

SMF selects the UPFs based on the LCI value received from UPFs. As per the load value received, the capacity of UPFs is altered. This leads to distribution of sessions based on load factor across the UPFs. The UPF selection process is independently applied at each service pod.

**Note** The UPF selection process is independently applied at each service pod.

### User Plane Initiated Session Termination

During self-protection mode, if no improvement in the load condition at the UP is observed, the UP starts the Session Termination Request toward SMF/PGW-C. This request is sent through a Session Report Request message indicating that UPF is in Overload condition. The SMF/PGW-C starts releasing the corresponding sessions based on this session report.

The Self-Protection Termination Request (SPTER) is a message bit that is set from the User Plane toward the control plane for initiating self-protection-based termination. SMF/PGW-C releases the session with Disc-Reason as **userplane_requested_termination**.

The SPTER message bit setting depends upon whether the EPFAR functionality is negotiated between SMF and UPF.

For more details on EPFAR functionality, see the *UCC 5G UPF Configuration and Administration Guide, Release 2024.04*.

## N4 Failure Handling during Peer in Congestion

When the peer UPF node is in the overload condition and the outgoing N4 messages are dropped due to peer overload reduction received on OCI, the N4 Failure Handling mechanism is initiated. To clear the session, SMF allows to add a cause code **peer-overload-reduction**.

To configure this cause code under failure Handling Profile, see Enable UPF failure handlingsection.

When SMF receives the cause code **74- PFCP ENTITY IN CONGESTION** from UPF and in if the cause code 74 **action terminate** is configured in the N4 failure handling profile, the SMF deletes the session.

# Configuring Load and Overload Control on SMF

Configuring the Load and Overload Control on SMF involves the following steps:

1. Create Load Profile, on page 20
2. Create Exclude Profile, on page 21
3. Create Overload Profile, on page 22
4. Associate Load and Overload Profiles, on page 26
5. Configure Load Factor

**Note** Currently, SMF does not support advertising of load or overload information to peers over SBA nterfaces.

# Create Load Profile

Use the following sample configuration to create the load profile. This profile defines the parameters that are required to calculate the load of SMF.

```
config
   profile load load_profile_name
      load-calc-frequency load_calculation_interval
      load-fetch-frequency load_fetching_time
     advertise [ interval lci_broadcast_interval | change-factor lci_change_factor
 ]
      interface gtpc action advertise
      end
```

**NOTES**:

- **profile load** *load_profile_name*: Specify the load profile name. *load_profile_name* must be an alphanumeric string.

  Use the load profile for system load calculation and LCI broadcast.

- **load-calc-frequency** *load_calculation_interval*: Specify the system load calculation interval in seconds.

*load_calculation_interval* must be an integer in the range of 5-3600. Default value is 10 seconds.

- **load-fetch-frequency** *load_fetching_time*: Specify the time interval at which service pod fetches load from cache pod.

  *load_fetching_time* must be an integer in the range of 5-3600. Default value is 10 seconds.

- **advertise interval** *lci_broadcast_interval*: Specify the periodic interval for sending LCI to the GTP-C peers.

  *lci_broadcast_interval* must be an integer in the range of 0-3600. Value 0 indicates that the LCI is sent in all the messages. Default value is 300 seconds.

- **advertise change-factor** *lci_change_factor*: Specify the minimum change between current LCI and last indicated LCI, after which the advertising occurs.

  *lci_change_factor* must be an integer in the range of 1-20. Default value is 5.

- **interface gtpc action advertise**: Specify to enable LCI publish or broadcast on GTP-C interface. By default, this option is disabled.

## Verify Load Profile Configuration

Use the following command to view the load control profile configuration settings.

**show running-config**

The following is an example of the **show running-config** command output.

```
#show running-config
.
.
.
profile load loadprofile
load-calc-frequency  10
load-fetch-frequency 10
advertise interval 300
advertise change-factor 5
interface gtpc
  action advertise
exit
exit
```

# Create Exclude Profile

Use the following sample configuration to create the exclude profile for use during self-protection state. This profile determines the session-related messages that should be excluded from throttling decisions.

```
config
   profile overload-exclude overload_exclude_profile_name
      arp-list list_of_arps
      dnn-list list_of_dnns
      message-priority { n4 | n7 | n10 | n11 | n16 | n40 | s5 | any }
upto  message_priority
      procedure-list [ session-delete | new-call | xnho | modify |
chf-reauth | inter-rat-ho | intra-rat-ho | imexit-nw | imexit-ue | usar
]
```

```
qi5-list list_of_qos_identifiers
end
```

**NOTES**:

- **profile overload-exclude** *overload_exclude_profile_name*: Specify the exclude profile name. *overload_exclude_profile_name* must be an alphanumeric string.

  You can configure multiple exclude profiles with this command. Be sure to reference the exclude profile name in the Overload Control configuration.

- **arp-list** *list_of_arps*: Specify the list of Allocation and Retention Priorities (ARPs) that must be excluded from throttling decisions.

  *list_of_arps* must be an integer in the range of 1-15.

  You can configure a maximum of eight entries.

- **dnn-list** *list_of_dnns*: Specify the list of DNNs that must be excluded from throttling decisions.

  You can configure a maximum of three entries.

- **message-priority [ n4 | n7 | n10 | n11 | n16 | n40 | s5 ] upto** *message_priority*: Specify the interface from which the messages need to be excluded from throttling. The command **upto** specifies the range of message priority. The message priority ranges between (0-31) for SBI interface and (0-15) for GTPC and PFCP interfaces.

- **procedure-list [ session-delete | new-call | xnho | modify | chf-reauth | inter-rat-ho | intra-rat-ho | imexit-nw | imexit-ue | usar ]**: Specify the procedure to be performed for exclusion.

- **qi5-list** *list_of_qos_identifiers*: Specify the 5G QoS Identifiers that must be excluded from throttling decisions.

  *list_of_qos_identifiers* must be an integer in the range of 1-15.

## Verify Exclude Profile Configuration

Use the following command to view the exclude profile configuration settings.

```
show running-config
```

The following is an example of the **show running-config** command output.

```
#show running-config
.
.
.
profile overload-exclude excludeProfile
dnn-list       [ starent.com.mnc456.mcc123.gprs ]
qi5-list        [ 1 2 ]
arp-list       [ 1 2 ]
procedure-list [ session-delete ]
message-priority s5
  upto 1
exit
exit
```

# Create Overload Profile

The overload profile determines the various conditions for overload control and throttling decisions.

To create the overload profile, use the following sample configuration:

```
config
   profile overload overload_profile_name
      overload-exclude-profile self-protection overload_exclude_profile_name
      node-level
         tolerance minimum min_percentage maximum max_percentage
         reduction-metric minimum min_percentage maximum max_percentage
         advertise [ interval oci_broadcast_interval | change-factor
oci_change_factor | validity-period oci_validity_period ]
         interface [ gtpc | any] overloaded-action [ advertise ]
      peer-level interface gtpc
         message-prioritization group1 weight group2 weight
         interface gtpc
            action throttle
         end
```

**NOTES**:

- **profile overload** *overload_profile_name*: Specify the overload profile name. *overload_profile_name* must be an alphanumeric string.

**Important** You can configure only one overload profile with this command. Create exclude profile before configuring overload profile.

- **overload-exclude-profile self-protection** *overload_exclude_profile_name*: Specify the exclude profile name that is configured for use during overload self-protection mode.

  *overload_exclude_profile_name* must be an alphanumeric string.

- **node-level**: Specify to apply the configuration only for the overloaded SMF node.

- **tolerance minimum** *min_percentage* **maximum** *max_percentage*: Specify the minimum and maximum percentage of the system load tolerance. *min_percentage* and *max_percentage* must be an integer in the range of 1-100.

  *min_percentage*: This value is the tolerance level below which the system is considered to be in Normal state. Default value is 80.

  *max_percentage*: This value is the tolerance level above which the system is considered to be in Self-protection state. Default value is 95.

  If the value is between the configured minimum and maximum tolerance values, then the system is in Overloaded state.

- **reduction-metric minimum** *min_percentage* **maximum** *max_percentage*: Specify the minimum and maximum percentage of the traffic reduction factor. *min_percentage* and *max_percentage* must be an integer in the range of 1-100.

  *min_percentage*: This value is the percentage of traffic reduction in tandem with minimum tolerance configuration. Default value is 10.

  *max_percentage*: This value is the percentage of traffic reduction in tandem with maximum tolerance configuration . Default value is 100.

- **advertise interval** *oci_broadcast_interval*: Specify the periodic interval for sending OCI to the GTP-C peers.

  *oci_broadcast_interval* must be an integer in the range of 0-3600. Value 0 indicates that the OCI is sent in all the messages. Default value is 300 seconds.

- **advertise change-factor** *lci_change_factor*: Specify the minimum change between current OCI and last indicated OCI, after which the OCI advertising occurs.

  *oci_change_factor* must be an integer in the range of 1-20. Default value is 5.

- **advertise validity-period** *oci_validity_period*: Specify the validity period of the advertised OCI value.

  *oci_validity_period* must be an integer in the range of 1-3600. Default value is 600 seconds.

- **interface [ gtpc | any] overloaded-action [ advertise ]** : Specify the action on different interfaces, when the node is overloaded. The overloaded action for these interfaces is to advertise.

- **peer-level interface gtpc**: Specify to apply the configuration only for the overloaded peer over GTPC interface.

- **message-prioritization group1** *weight* **group2** *weight*: Specify the ratio in which the messages need to be throttled for both the message groups. Each group contains a predefined set of messages from every SMF interface.

  *weight* must be an integer in the range of 1-100. The default value is 50.

- **interface gtpc action throttle** : Enables the throttling action over S6, S8, and S2b interfaces.

**Note**

- In an overloaded state, when SMF rejects the ingress messages, it sends the OCI value in responses.
- The SMF identifies and deletes any expired peer OCI value and its associated entry in the SMF. Upon receiving a new OCI value from a peer, the SMF uses this new value regardless of whether it has the same sequence number as the previous OCI.

## Verify Overload Profile Configuration

To view the overload control profile configuration settings, use the following command:

**show running-config**

The following is an example of the **show running-config** command output.

```
#show running-config
.
.
.
profile overload overloadprofile
overload-exclude-profile self-protection excludeProfile
node-level tolerance minimum 80
node-level tolerance maximum 95
node-level reduction-metric minimum 10
node-level reduction-metric maximum 80
node-level advertise interval 300
node-level advertise change-factor 5
node-level advertise validity-period 600
node-level interface gtpc
```

```
  overloaded-action [ advertise ]
exit
exit
```

To view the overload information of all the peers, use the following command:

**show overload-info peer all**

The following is an example of the **show overload-info peer all** command output.

```
[smf] smf# show overload-info peer all
                              OVERLOAD
                             CONTROL     OVERLOAD
PEER                         SEQUENCE    REDUCTION                          INSTANCE
TYPE  INTERFACE  PEER IP     NUMBER      METRIC     PERIOD OF VALIDITY      ID
-------------------------------------------------------------------------------------
SGW   S5         10.1.45.151  1632888403  50         2024-11-08 11:46:03 UTC  1
MME   S5         10.1.2.30    1632888403  60         2024-11-08 11:46:03 UTC  1
```

This command displays the overload information of all the peers.

To view the overload information of a specific peer, use the following command:

**show overload-info peer all** *peer-type*

The following is an example of the **show overload-info peer all SGW** command output.

```
[smf] smf# show overload-info peer all SGW
                             OVERLOAD
                             CONTROL     OVERLOAD
PEER                         SEQUENCE    REDUCTION
TYPE  INTERFACE  PEER IP     NUMBER      METRIC     PERIOD OF VALIDITY
-------------------------------------------------------------------------
SGW   S5         209.165.201.1    1632888445     5        2021-12-01 00:04:02 UTC
```

This command displays the overload information of S-GW.

To view the overload information of peers at an interface level, use the following command:

**show overload-info peer all interface S5**

The following is an example of the **show overload-info peer all interface S5** command output.

```
[smf] smf# show overload-info peer all interface S5
                             OVERLOAD
                             CONTROL     OVERLOAD
PEER                         SEQUENCE    REDUCTION
TYPE  INTERFACE  PEER IP     NUMBER      METRIC     PERIOD OF VALIDITY
-------------------------------------------------------------------------
SGW   S5         209.165.201.1    1632888445     5        2021-12-01 00:04:02 UTC
MME   S5         209.165.201.2    1632888445     6        2021-12-01 05:04:02 UTC
```

This command displays the overload information of all the peers at S5 interface.

To view the overload information by IP address of peer, use the following command:

**show overload-info peer all peerIP** *ip_address*

The following is an example of the **show overload-info peer all peerIP 209.165.201.2** command output.

```
[smf] smf# show overload-info peer all peerIP 209.165.201.2
                             OVERLOAD
                             CONTROL     OVERLOAD
PEER                         SEQUENCE    REDUCTION
TYPE  INTERFACE  PEER IP     NUMBER      METRIC     PERIOD OF VALIDITY
-------------------------------------------------------------------------
MME   S5         209.165.201.2    1632888445     6        2021-12-01 05:04:02 UTC
```

# Associate Load and Overload Profiles

Use the following sample configuration to associate the load control profile and overload profile with the SMF service profile.

```
config
   profile smf smf_profile_name
      load-profile load_profile_name
      overload-profile overload_profile_name
      end
```

**NOTES**:

- **profile smf** *smf_profile_name*: Specify the existing SMF service profile name.

  *smf_profile_name* must be an alphanumeric string.

- **load-profile** *load_profile_name*: Specify the load profile name to associate with the SMF service profile.

  *load_profile_name* must be an alphanumeric string.

- **overload-profile** *overload_profile_name*: Specify the overload profile name to associate with the SMF service profile.

  *overload_profile_name* must be an alphanumeric string.

- Linking of the overload profile with SMF profile works only when the load profile is linked.

## Verify Load and Overload Profile Association

Use the following command to view the association of load and overload profiles with the SMF service profile.

```
show running-config
```

The following is an example of the **show running-config** command output.

```
#show running-config
.
.
.
profile smf smf1
<……..>
load-profile  loadprofile
overload-profile overloadprofile
<……..>
exit
```

# Configure Load Factor

This configuration allows the users to configure the calculation of load factor at specific time intervals and exclusion of service pods.

### Procedure

**Step 1** Configure load factor frequency calculation using the command **load factor**

**Example:**

```
[smf] smf(config)# load factor
```

**Step 2** Define the load factor calculation frequency or the service pods to exclude using the command **[ no ] load factor { calc-frequency** *calc_frequency_time* **| exclude-pods** *exclude_pod_name***}**

**Example:**

```
[smf] smf(config)# load factor calc-frequency 15
[smf] smf(config)# load factor exclude-pods bgpspeaker-pod
```

- The default value of the CLI **calc-frequency** is 30.
- The CLI **exclude-pods** excludes following service pods:
  - **bfdmgr**
  - **bgpspeaker-pod**
  - **cache-pod**
  - **edr-monitor**
  - **georeplication-pod**
- Both the CLIs **calc-frequency** and **exclude-pods** are backward-compatible.

**Step 3** Exit the global configuration mode using the command **exit**.

**Example:**

```
[smf] smf(config)#exit
```

## Configuration Verification

The show command **show resources** displays the load factor configuration parameters.

```
[smf] smf# show resources
Thu Oct  10 09:32:52.900 UTC+00:00
                                TOTAL   USED    DISK
                                NODE    POD     USAGE   GO              GC
                        CPU     MEMORY  MEMORY  IN      ROUTINES  GC    PAUSE  LOAD
POD INSTANCE            USAGE   IN MB   IN MB   KBPS    COUNT     COUNT IN NS  FACTOR
-------------------------------------------------------------------------------------
cache-pod-1             3       32096   99      0       438       672   99     0.438
cache-pod-2             2       32097   87      0       438       671   87     0.438
diameter-ep-gx-client-0 2       32096   92      0       217       670   92     0.19999881
diameter-ep-gy-client-0 2       32096   83      0       173       670   83     0.18888578
dns-proxy-0             0       32096   64      0       86        672   64     0.39150715
gtpc-ep1-0              2       32096   112     0       293       672   112    0.24416667
gtpp-ep-0               2       32096   109     0       208       669   109    0.22245248
gtpp-ep-1               2       32097   110     0       202       669   110    0.22777763
```

# Node Overload

The node overload refers to the resource utilization data of all the SMF pods in the NF deployment. The SMF periodically gathers the current resource utilization data for these pods. The default frequency to read the resource utilization data is 5 seconds. The SMF monitors the CPU, memory utilization, go-routines, and stores the average values for the current, last 5 minutes and 15 minutes for the pods.

### Pod Level Load Factor

The maximum values against the current values for CPU, memory utilization and go-routines for a pod are used to calculate its load factor. The GOMAXPROCS environment variable is used to calculate the capacity of a pod. The maximum value per core is defined with constant values, which is used to derive the capacity of CPU, memory and go-routines.

An example of the maximum value per core is show below.

MAX_CPU_PERCENTAGE_PER_CORE = 100

MAX_MEMORY_PER_CORE = 4 GB

MAX_GO_ROUTINE_PER_CORE = 10,000

The **NewApplicationWithOptions** is used to get the maximum values. If the values are not provided by the application, then the default values are used.

The load factor for a pod is calculated as follows:

- CPU load factor = Current load percentage / Maximum load percentage at pod x 100
- Memory load factor = Current memory usage / Maximum memory at pod x 100
- Go-routine load factor = Go-routine count / Maximum Go-routine count at pod x 100

The maximum value from the CPU, memory and go-routines load factors is considered as the final load factor.

### Self-NF Load Factor from an OAM Pod

The OAM pod periodically gathers the load factor data from each SMF pod and updates the cache pod. The OAM pod also receives the session load factor from the CDL and updates the cache pod at the same time.

The system APIs provide the load factor data based on the following logic:

- **Pod level load factor** - If an application queries the load factor for a pod to get its resource utilization data in the SMF, then the response contains the maximum load factor for all the pod type categories in that cluster.
- **System level load factor** - If an application queries the load factor at the system level, then the response contains the maximum load factor for all the pods in that cluster along with the session load factor data.
- **Load factor based on a category** - If an application queries the load factor for a specific type of service like, smf-service, smf-rest-ep, and so on, then the following conditions are met:
  - **Active-Active deployment** - The query response contains the average value of the load factors.
  - **Active-Standby deployment** - The query response contains the maximum value of the load factors.

A system level capacity to handle the number of sessions is configured in the SMF. The load factor for each session is calculated in the OAM pod as the Current session count / Maximum number of sessions.

### Maximum Sessions

A datastore configuration is used to include the session load factor for supported namespaces. The values must be set from the application while registering the session database. If the value is not set, then the default 1,000,000 is used to calculate the session load factor.

### Application level ConfigMap Support for OAM

The OAM infra chart mounts the configmaps from the OAM application in the following ways:

- **Infra-OAM**
  - Update template to add volumes for configuration maps from render.yaml.
  - Update template to mount volumes from render.yaml by using volumeMounts.
- **Application-OAM**
  - Add configuration map with the same in application configuration chart.
  - Provide values from Values.yaml or from CLI for the configuration map.

# Monitoring and Troubleshooting

This section provides information regarding bulk statistics available to monitor and troubleshoot this feature.

## Statistics

The following statistics are available in support of Overload Control.

| Bulk Statistics | Statistics Type | Description |
|---|---|---|
| endpoint_overload_status | Gauge | Contains Endpoint-Name, Interface-Name and Overload-Level as labels. Once any level(low/high/critical) is hit, the gauge value will be set to 1. In normal condition the value is set to 0. |
| endpoint_client_overload_status | Gauge | Contains Endpoint-Name, Interface-Name, peer-host name and Overload-Level as labels. Once any level(low/high/critical) is hit, the gauge value will be set to 1. In normal condition the value is set to 0. |
| endpoint_pending_request | Gauge | Display current outstanding request for an endpoint. It contains Endpoint name and Interface Name as label. |
| endpoint_client_pending_request | Gauge | Display current outstanding request for a peer connected with an endpoint. It contains Endpoint name, Interface Name and peer host address connected to the endpoint as label. |

| Bulk Statistics | Statistics Type | Description |
|---|---|---|
| endpoint_overload_exclude | Counter | Display the messages with their priority details that were excluded from the overload control mechanism. The metric is incremented for every message, which bypasses the overload control mechanism. |

As part of load and overload handling for SBA, N4, and GTPC interfaces, the following metrics are enhanced:

- **Self-Overload Condition:** When SMF is in the overload condition, the statistics **smf_inc_msg_throttling_stats** gets incremented for the number of dropped incoming messages. The following new messages are added as part of the label **message_type**:
  - **SBI Interfaces:**
    - N7SmPolicyUpdateNotifyReq
    - N7SmPolicyUpdateSuccess
    - N7SmPolicyTerminateNotifyReq
    - N11SmContextCreateReq
    - N11SmContextRetrieveReq
    - N11SmContextStatusNotifyReq
    - N16PduSessionHsmfUpdateReq
    - N16PduSessionHsmfUpdateReqClient
    - N16PduSessionVsmfUpdateReq
    - N16PduSessionVsmfUpdateReqClient
    - N16VsmfPduSessionCreateReq
    - N16HsmfSmContextRetrieveReq
    - N10UpdateNotifyReq
    - N10PcscfRestorationNotifyReq
  - **GTPC Interfaces:**
    - S5S8CreateSessReq
    - S5S8ModifyBearerCmd
    - S5S8ModifyBearerReq
    - S5S8CreateBearerRsp
    - S5S8UpdateBearerRsp
  - **N4 Interfaces:**
    - N4SessionEstablishmentReq
    - N4SessionModificationReq

- **Peer Overload Condition:** When the peer node is in overload state, the statistics **smf_og_msg_throttling_stats** gets incremented for the number of dropped outgoing messages.