



Overload Management

- [Feature Summary and Revision History, on page 1](#)
- [Feature Description, on page 2](#)
- [SBA Interface Overload Control, on page 2](#)
- [GTP-C Load and Overload Control, on page 7](#)
- [Node Overload, on page 18](#)

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 load and overload control	2021.02.0
Added support for message priority configuration.	2020.04.0
First introduced.	2020.03.0

Feature Description



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.

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 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.

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.
- *threshold_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.
- *threshold_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 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.
- **threshold_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** – 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*.

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.

Bulk Statistics	Statistics Type	Description
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.
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.

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 12](#) 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 13](#) section.

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 18](#) 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

Configuring GTP-C Load and Overload Control Feature

This section describes how to configure the GTP-C Load and Overload Control feature.

Configuring the GTP-C Load and Overload Control feature involves the following steps:

1. [Create Load Profile, on page 12](#)
2. [Create Exclude Profile, on page 13](#)
3. [Create Overload Profile, on page 14](#)
4. [Associate Load and Overload Profiles, on page 16](#)

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 s5 upto message_priority
    procedure-list session-delete
    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 s5 upto** *message_priority*: Specify the message priority up to which has to be excluded from throttling decisions.
message_priority must be an integer in the range of 0-15.
- **procedure-list session-delete** : Specify the session deletion procedures that must be excluded from throttling decisions.
- **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 overloaded-action advertise

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 overloaded-action advertise**: Specify to enable OCI publish or broadcast on GTP-C interface when the node is overloaded. By default, this option is disabled.

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
```

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
```

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

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.