



# GTPv2 Load/Overload Support

- [Feature Summary and Revision History, on page 1](#)
- [Feature Description, on page 1](#)
- [Configuring the GTPv2 Load and Overload Feature, on page 3](#)
- [GTPv2 Load and Overload OAM Support, on page 9](#)

## Feature Summary and Revision History

### Summary Data

**Table 1: Summary Data**

Applicable Product(s) or Functional Area	cnSGW-C
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

### Revision History

**Table 2: Revision History**

Revision Details	Release
First introduced.	2021.02.0

## Feature Description

The following are the details for the load control and the overload control features.

## Load Control

The load control enables a GTPC entity, such as SGW or PGW, to send its load information to a GTPC peer, such as MME, ePDG, and TWAN. This information is used to balance the session load across all the nodes supporting the same function, such as SGW cluster, as per their effective loads. The load information reflects the operational status of the GTPC entity resources.

cnSGW-C load control behavior is as follows:

- Activate or deactivate the load control support in cnSGW-C, using the CLI.
- When the load control feature is activated, cnSGW-C signals its load control information to the MME for the following reasons:
  - Optimum GW selection procedures
  - Enhanced load balancing across cnSGW-C in the network
- The calculation of the load control information is based on the deployment scenarios.
- The applicable GTPC request or the response message contains piggybacked load control information.
- cnSGW-C includes only the single instance of LCI (Load Control Information) IE as per the SGW load control information. cnSGW-C sends the LCI IE received from the PGW to the MME, along with its LCI information.
- The frequency of load control information inclusion at SGW is based on the deployment scenario. The SGW ensures the propagation of the new or the updated load control information to the target receivers is within the acceptable delay. This acceptable delay helps in achieving the effective load balancing act in the network.

## Overload Control

The overload control enables a GTPC entity to reduce gracefully its own incoming signaling load by instructing its GTPC peers to send the reduced traffic. The GTPC entity reaches overload, when it operates above the signaling capacity. This overload results in a diminished performance, resulting in to impacts on the incoming and the outgoing traffic handling. The GTPC node uses the load information to reduce, or throttle, or reduce and throttle, the amount of GTPC signaling traffic between these nodes.

cnSGW-C overload control behavior is as follows:

- Activate or deactivate overload control support in cnSGW-C using the CLI.
- When the overload control feature is activated, cnSGW-C signals its overload control information to the MME or the PGW. This helps in controlling the GTPC signaling traffic towards itself.
- SGW supports the handling of the overload control information in all the applicable messages.
- The applicable GTPC request or the response message contains piggybacked overload control information.
- cnSGW-C includes only the single instance of OCI (Overload Control Information) IE as per the SGW overload control information. cnSGW-C sends the OCI IE received from the PGW to the MME, along with its OCI information.
- The calculation of the overload control information is based on the deployment scenario.
- cnSGW-C rejects with the cause as GTPC entity congestion, when the SGW is in self-protection mode.

- SGW doesn't store the MME or the PGW overload control information.
- SGW doesn't perform throttling towards the MME and the PGW.



**Note** The load control and the overload control are optional features.

## Configuring the GTPv2 Load and Overload Feature

This section describes how to configure the GTPv2 load or overload conditions.

Configuring this feature involves the following steps:

- [Configuring the Load Profile](#): This section describes how to configure the load profile and the parameters required to calculate the load of cnSGW-C.
- [Configuring the Exclude Profile, on page 4](#): This section describes how to make an exclusion and configure the exclude profile in overload conditions.
  - This profile determines the session-related messages to exclude from the throttling decisions.
  - Both self-protection and peer overload control, use this configuration.
- [Configuring the Overload Condition Profile, on page 5](#): This section describes how to configure the profile in overload conditions.
  - The profile determines the various conditions for overload control and the resulting throttling decisions.
  - It supports only one overload profile.
  - The load profile supports overload profile functionality.
- [Configuring the Maximum Session Count, on page 6](#): This section describes how to configure the maximum session count that contributes to the session percent load factor in LCI/OCI calculation.
- [Configuring the Overload-Profile and the SGW-Profile Association](#): The association of the Overload-Profile and the SGW-Profile, can be configured.

## Configuring the Load Profile

To configure this feature use the following configuration:

```
config
  profile load profile_name
    load-calc-frequency load_calc_frequency_value
    load-fetch-frequency load_fetch_frequency_value
    advertise
    interval interval_value
    change-factor change_factor_value
  exit
```

```

interface gtpc
    action advertise
end

```

**NOTES:**

- **profile load** *profile\_name*—Specify the load profile name.
- **load-calc-frequency** *load\_calc\_frequency\_value*—Specify the system load calculation time in seconds. Must be an integer in the range of 5-3600. The default value is 10 seconds.
- **load-fetch-frequency** *load\_fetch\_frequency\_value*—Specify the time interval in seconds at which protocol pods fetch load from the cache POD. Must be an integer in the range of 5-3600. The default value is 10 seconds.
- **advertise interval** *interval\_value*—Specify the time interval of sending LCI to the peers in seconds. Must be an integer in the range of 0-3600. The default value is 300 seconds.
- **advertise change-factor** *change\_factor\_value*—Specify the LCI value to corresponding peers, if the difference between the current load value and the last indicated load value is greater than the change-factor. Must be an integer in the range of 1-20. The default value is five.
- **interface gtpc action advertise**—Enables LCI publishing on the GTPC interface.

## Configuration Example

The following is an example configuration.

```

config
profile load pl
load-calc-frequency 30
load-fetch-frequency 60
advertise
interval 300
change-factor 1
exit
interface gtpc
action advertise
end

```

## Configuring the Exclude Profile

To configure this feature use the following configuration:

```

config
    profile overload-exclude overload_exclude_profile_name
        dnn-list dnn_list
        arp-list arp_list
    end

```

**NOTES:**

- **profile overload-exclude** *overload\_exclude\_profile\_name*—Specify the name of the exclude profile. You can configure multiple exclude profiles.
- **dnn-list** *dnn\_list*—Specify the list of DNNs that needs to be excluded from throttling decisions. The maximum limit is three.

- **arp-list** *arp\_list*—Specify the list of 5G allocation and retention priorities and exclusion of throttling decision messages. Must be an integer in the range of 1-15. The maximum limit is eight.

## Configuration Example

The following is an example configuration.

```
config
profile overload-exclude ol-excl-profl
dnn-list emergency-dnn1 wps-dnn2
arp-list 1 2 3
end
```

## Configuring the Overload Condition Profile

To configure this feature use the following configuration:

```
config
  profile overload overload_profile_name
    overload-exclude-profile self-protection self_protection_profile_name
    node-level
    tolerance
      minimum min_percentage
      maximum max_percentage
    reduction-metric
      minimum min_percentage
      maximum max_percentage
    interface gtpc
      overloaded-action advertise
      advertise
        interval interval_value
        change-factor change_factor_value
        validity-period validity_period_value
      end
    end
```

### NOTES:

- **profile overload** *overload\_profile\_name*—Specify the overload profile name.
- **overload-exclude-profile self-protection** *self\_protection\_profile\_name*—(This is an optional configuration) Exclude messages from throttling decisions in self-protection condition.
- **tolerance minimum** *min\_percentage* **maximum** *max\_percentage*—Specify the system overload limits. Refer the following scenarios:
  - When the system load is less than *min\_percentage*, the system is in a normal state.
  - When the system load is in between *min\_percentage* and *max\_percentage*, the system is in an overloaded state. In this scenario, the node overload control action is triggered.
  - When the system load is greater than *max\_percentage*, the system is in a self-protection state. In this scenario, the self-protection action is triggered.
  - *max\_percentage* must be an integer in the range of 1-100. The default value is 95.
  - *min\_percentage* must be an integer in the range of 1-100. The default value is 80.

- **reduction-metric minimum** *min\_percentage* **maximum** *max\_percentage*—Specify the reduction metric limits. Refer the following scenarios:

- Both percentage values, *min\_percentage* and *max\_percentage* work along with the **tolerance** configuration.
- The percentage value *max\_percentage* must be an integer in the range of 1-100. The default value is 100.
- The percentage value *min\_percentage* must be an integer in the range of 1-100. The default value is 10.

**Example:** Send 10 percent OCI to peer nodes, when the load is 80 percent, and 30 percent, when the load is 95 percent, during the following conditions:

- **tolerance** *min\_percentage* is 80 and *max\_percentage* is 95.
- **reduction-metric** *min\_percentage* is 10 and *max\_percentage* is 30.
- **interface gtpc overloaded-action advertise**—Configures the action on GTPC interface when a node gets overloaded. GTPC includes S5/S8/S11/S2b interfaces. Certain actions apply only to specific interfaces.
- **advertise interval** *interval\_value*—Specify the periodicity of sending LCI to the peers in seconds. Must be an integer in the range of 0-3600. The default value is 300 seconds.
- **advertise change-factor** *change\_factor\_value*—Specify the change-factor value. GTPC sends the LCI to corresponding peers, if the difference between the current load value and the lastly indicated load value is greater than the change-factor value. Must be an integer in the range of 1-20. The default value is five.
- **advertise validity-period** *validity\_period\_value*—Specify the validity period of the advertised OCI value in seconds. Must be an integer in the range of 1-3600. The default value is 600 seconds.

## Configuring the Maximum Session Count

To configure this feature use the following configuration:

```
config
  profile converged-core profile_name
    max-session-count max_session_count_value
  end
```

### NOTES:

- **profile converged-core** *profile\_name*—Specify the name of the converged core profile.
- **max-session-count** *max\_session\_count\_value*—Specify the maximum number of sessions supported. Must be an integer in the range of 1-12000000.

## Configuration Example

The following is an example configuration.

```
config
profile converged-core convergedCoreProfile
```

```
max-session-count 12000000
exit
```

## Configuring the Overload-Profile and the SGW-Profile Association

The association of the Overload-Profile and the SGW-Profile, can be configured.

To configure this feature use the following configuration:

```
config
  profile overload overload_profile_name
    overload-exclude-profile self-protection self_protection_profile_name
  node-level
    tolerance
      minimum min_percentage
      maximum max_percentage
    reduction-metric
      minimum min_percentage
      maximum max_percentage
      advertise
      interval interval_value
      change-factor
    exit
  interface gtpc
    overloaded-action [ advertise ]
    exit
  exit
exit
profile load load_name
load-calc-frequency load_calc_frequency_value
load-fetch-frequency load_fetch_frequency_value
advertise
interval interval_value
change-factor change_factor_value
exit
interface gtpc
action advertise
exit
exit
profile sgw sgw_name
load-profile profile_name
overload-profile overload_profile_name
end
```

### NOTES:

- **overload** *overload\_name*—Specify the overload protection profile name. Must be a string.
- **overload-exclude-profile**—Excludes profiles for overload scenarios.
- **self-protection** *overload\_value*—Specify the profile to be excluded for self-protection. Must be a string.
- **tolerance minimum** *min\_percentage*—Specify the minimum tolerance level below which the system is in a normal state. Must be an integer in the range of 1-100. The default value is 80.

- **tolerance maximum** *max\_percentage*—Specify the maximum tolerance level above which the system is in a self-protection state. Must be an integer in the range of 1-100. The default value is 95.
- **reduction-metric minimum** *min\_percentage*—Specify the percentage of reduction along with minimum tolerance-level for configuration. Must be an integer in the range of 1-100. The default value is 10.
- **reduction-metric maximum** *max\_percentage*—Specify the percentage of reduction along with maximum tolerance-level for configuration. Must be an integer in the range of 1-100. The default value is 100.
- **interval** *interval\_value*—Specify the advertising interval in seconds. Must be an integer in the range of 0-3600. The default value is 300 seconds.
- **validity** *validity\_value*—Specify the validity period of the advertised OCI value in seconds. Must be an integer in the range of 1-3600. The default value is 600 seconds.
- **change-factor** *change\_factor\_value*—Specify the minimum change between current OCI and last indicated OCI, after which the advertising should happen. Must be an integer in the range of 1-20. The default value is five.
- **profile load** *load\_name*—Specify the name of the load profile. Must be a string.
- **load-calc-frequency** *load\_calc\_frequency\_value*—Specify the system load calculation interval in seconds. Must be an integer in the range of 5-3600. The default value is 10 seconds.
- **load-fetch-frequency** *load\_fetch\_frequency\_value*—Specify the time interval in seconds at which the service pods fetch load from the cache pod. Must be an integer in the range of 5-3600. The default value is 10 seconds.
- **load-profile** *profile\_name*—Specify the name of the load profile.
- **overload-profile** *overload\_profile\_name*—Specify the name of the overload profile.

## Configuration Example

The following is an example configuration.

```
config
profile overload op
overload-exclude-profile self-protection <overload-exclude-profile-name>
node-level
tolerance minimum 5
tolerance maximum 50
reduction-metric minimum 50
reduction-metric maximum 100
advertise
interval 0
change-factor 1
exit
interface gtpc
overloaded-action [ advertise ]
exit
exit
exit
profile load lp
load-calc-frequency 120
load-fetch-frequency 15
advertise
interval 0
change-factor 1
exit
```



```

interface gtpc
action advertise
exit
exit
profile sgw <sgw_name>
load-profile <profile_name>
overload-profile <overload_profile_name>
end

```

## Configuration Verification

To verify the configuration:

```

show running-config profile
profile sgw sgw1
load lpl
overload opl
end

```

# GTPv2 Load and Overload OAM Support

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

## Bulk Statistics

### Normal

```

node_overload_status{app_name="smf", cluster="cn", data_center="cn",
instance_id="0", service_name="oam-pod"} 0

```

### Overload

```

node_overload_status{app_name="smf", cluster="cn", data_center="cn",
instance_id="0", service_name="oam-pod"} 1

```

### Self-Protection

```

node_overload_status{app_name="smf", cluster="cn", data_center="cn",
instance_id="0", service_name="oam-pod"} 2

```

### SGW Service Statistics

```

sgw_service_stats{app_name="smf", cluster="cn", data_center="cn",
fail_reason="gtp_entity_in_congestion", instance_id="0",
interface="interface_sgw_ingress", reject_cause="entity_in_congestion",
service_name="sgw-service", sgw_procedure_type="initial_attach",
status="rejected", sub_fail_reason=""}

```

**LCI/OCI Metric Values**

```
node_lci_metric{app_name="SGW", cluster="cn", component="oam-pod",  
data_center="DC", namespace="cn", instance_id="0", service_name="oam-pod"}
```

```
node_oci_metric{app_name="SGW", cluster="cn", component="oam-pod",  
data_center="DC", namespace="cn", instance_id="0", service_name="oam-pod"}
```