

# MTAS SIP Upstream Overload Control Management Guide

MTAS

---

USER GUIDE

**Copyright**

© Ericsson AB 2016, 2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

**Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

**Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Prerequisites	1
<b>2</b>	<b>Overview</b>	<b>3</b>
2.1	Content of OC Parameters	3
2.2	SIP Upstream Overload Control in MTAS	4
2.3	Limitations in Supported Functionality	5
2.4	SIP Message Examples for Overload Control Negotiation	5
2.5	System Utilization	6
2.6	OC Calculation Algorithm	6
<b>3</b>	<b>SIP Upstream Overload Control Configuration</b>	<b>9</b>
3.1	SIP Upstream OC Administrative State Configuration	9
3.2	SIP Upstream OC Parameter Configuration	9
3.3	SIP Upstream OC Resource Configuration	11
<b>4</b>	<b>Performance Management</b>	<b>13</b>
<b>5</b>	<b>Fault Management</b>	<b>15</b>





# 1 Introduction

This document describes how to configure the SIP Upstream Overload Control (OC) function in the MTAS.

## 1.1 Prerequisites

This section describes the prerequisites that must be fulfilled. It is assumed that the user of this document is familiar with the Operation and Maintenance (O&M) area, in general.

### 1.1.1 Licenses

No License required for SIP Upstream Overload Control.

### 1.1.2 Documents

Before starting any procedure in this document, ensure that the following documents are available:

- *Managed Object Model (MOM)*
- *Ericsson Command-Line Interface User Guide*

### 1.1.3 Conditions

Before starting any of the procedures described in this document, ensure that the following condition is met:

- An Ericsson Command-Line Interface (ECLI) session in Exec mode is in progress.





## 2 Overview

The SIP Upstream Overload Control in MTAS implements the reporting or server role of the RFC 7339 “SIP Overload Control”, which describes the exchange of the overload information between the SIP servers and the SIP clients, so that the SIP clients can reduce the volume of traffic sent towards the overloaded SIP servers, avoiding congestion collapse and increasing useful throughput.

### 2.1 Content of OC Parameters

OC parameters are added by reacting node (CSCF) in the topmost `Via` header of the `SIP INVITE`, which indicates that the OC is supported.

OC parameters added by CSCF are shown in Example 1.

```
Via: SIP/2.0/TLS p1.example.net
    ;branch=z9hG4bK2d4790.3;
    ;oc;oc-algo="loss";
```

*Example 1 OC Parameters Added By CSCF*

When feature is enabled and OC parameters are in the request, MTAS adds OC parameters to the topmost `Via` header of the response.

OC parameters added by MTAS are shown in Example 2.

```
Via: SIP/2.0/TLS p1.example.net
    ;branch=z9hG4bK2d4790.3;received=192.0.2.111
    ;oc=20;oc-algo="loss";oc-validity=1500
    ;oc-seq=1456387313.187
```

*Example 2 OC Parameters Added By MTAS*

When the feature is disabled and OC parameters are in the request, MTAS adds OC parameters to the topmost `Via` header of the response.

OC parameters added by the MTAS are shown in Example 3.

```
Via: SIP/2.0/TLS p1.example.net
    ;branch=z9hG4bK2d4790.3;received=192.0.2.111
    ;oc;oc-algo="loss"
```

*Example 3 OC Parameters Added By MTAS*

Details of each parameter are described in Table 1



Table 1 OC Parameters of RFC 7339

OC Parameter	Content
oc	<p>A value to indicate the level of traffic reduction to apply, that is, <code>oc=0</code> if no traffic reduction is required (because the reporting node is not overloaded, for example, <code>oc=20</code> when overload control active and requesting 20% traffic reduction).</p> <p>The value is calculated by MTAS based on current system overload status.</p>
oc-algo	<p>It is used to indicate the supported traffic reduction algorithms, where <code>loss</code> currently is the only supported one.</p>
oc-validity	<p>It indicates for how long duration in milliseconds the sent <code>oc</code> parameter can be used by the reacting or client node. The default value can be configured by CM attribute <code>mtasSipOcValidity</code>.</p>
oc-seq	<p>It is used to differentiate two <code>oc</code> parameter values generated by an overload control algorithm at two different instants in time.</p> <p>MTAS generated <code>oc-seq</code> is based on “epoch time” and have the following format: [epoch seconds].[epoch milliseconds]</p>

## 2.2 SIP Upstream Overload Control in MTAS

MTAS uses OC parameters to report its overload status towards the reacting node (CSCF).

The concept of overload in this document is not the same as the legacy load regulation. This means that when monitored resource usages reach the configured value, MTAS treat this as overload state and ask reacting node to reduce traffic to avoid call rejection by load regulation.

If MTAS operates in normal condition, thus, the node is not overloaded then the same OC parameters with `oc=0`; `oc-algo=loss`; `oc-validity=0`; `oc-seq=xxx` are reported in each response. In this case, the `oc-seq` here is not changed until the system enters the overload status.

If MTAS uses SIP upstream overload control, the `oc` value changes between 1-100 and reported in a configurable interval. The `oc` value is calculated from the current monitored resource utilization level and from the previous `oc` status. The `oc-validity` can be configured by the operator and the `oc-seq` indicates the time stamp when the `oc` value has been calculated.





## 2.3 Limitations in Supported Functionality

MTAS does not report the OC status in the following cases to avoid system complexity and to decrease the latency in overload condition:

- The system load regulation rejects an out-of-dialog request in local overload condition.
- The transaction layer rejects the malformed or faulty requests, for example, if the Max-Forwards header value is less than 1.
- The early Priority check rejects the request if the Resource-Priority Header (RPH) is invalid.

## 2.4 SIP Message Examples for Overload Control Negotiation

Consider a SIP client, P1, which is sending requests to a downstream SIP server, P2.

The following snippets of SIP messages demonstrate how the overload control parameters work.

### **INVITE Request from P1, a SIP Client (for Example, CSCF)**

```
INVITE sips:user@example.com SIP/2.0
Via: SIP/2.0/TLS p1.example.net
    ;branch=z9hG4bK2d4790.1
    ;oc; oc-algo="loss"
```

P1 supports Overload control Algorithm LOSS.

### **INVITE Response from P2, a SIP Server (for Example, MTAS)**

```
SIP/2.0 100 Trying
Via: SIP/2.0/TLS p1.example.net
    ;branch=z9hG4bK2d4790.1
    ;oc=0;oc-algo="loss"
    ;oc-validity=0
    ;oc-seq=201602011634.45
```

P2 supports Overload control Algorithm LOSS, and does not require any overload control by P1 since `oc=0` and `oc-validity=0`.

When the P2 starts to experience overload, it sends the following SIP message indicating that P1 is to decrease the messages arriving to P2 by 20% for 1.5 seconds.



### INVITE Response from P2, a SIP Server (for Example, MTAS)

```
SIP/2.0 180 Ringing
Via: SIP/2.0/TLS p1.example.net
    ;branch=z9hG4bK2d4790.3
    ;oc=20;oc-algo="loss"
    ;oc-validity=1500
    ;oc-seq=201602011635.45
```

After some time, the overload condition at P2 subsides. It then changes the parameter values in the response it sends to P1 to allow P1 to send all messages destined to P2.

### INVITE Response from P2, a SIP Server (for Example, MTAS)

```
SIP/2.0 182 Queued
Via: SIP/2.0/TLS p1.example.net
    ;branch=z9hG4bK2d4790.4
    ;oc=0;oc-algo="loss"
    ;oc-validity=0
    ;oc-seq=201602011636.45
```

## 2.5 System Utilization

For MTAS there are several resource types available on system level, see Section 3.3 on page 11. A user can configure which resource to be monitored by CM `mtasSipOcResource`. The CPU utilization is always considered no matter whether it is configured or not.

However, for the sake of simplicity, the Sip upstream overload control feature needs a single parameter which indicates the node overall utilization level. The term of system utilization has been introduced for that purpose, which indicates the normalized utilization level of the most critical resource (the utilization value which is closest to the load regulation limit).

The normalized here means, the system utilization equals to  $100 * \text{utilization\_level} / \text{limit\_value}$ .

The SIP Upstream Overload control implements a mechanism which periodically (set by CM `mtasSipOcRegulationInterval`) collects the current aggregated utilization values of the monitored resources and calculates the actual system utilization.

## 2.6 OC Calculation Algorithm

The OC value is dynamically recalculated based on the system utilization level in each regulation period, (configured by CM `mtasSipOcRegulationInterval`, default 1s).



The OC value is adjusted according to the following rules:

- At the initial phase, the OC value is 0, indicating no restriction (no SIP server overload).
- The following are the specific OC increase and decrease rules:
  - The OC value is increased in intervals not shorter than OC regulation timer if the system utilization is greater than overload onset (OO) threshold.
  - The OC value remains unchanged if the system utilization is between overload onset (OO) threshold and overload abatement (OA) threshold.
  - The OC value is decreased in intervals not shorter than OC regulation timer if the system utilization is below the overload abatement (OA).

Figure 1 shows the OC increase and decrease rules.

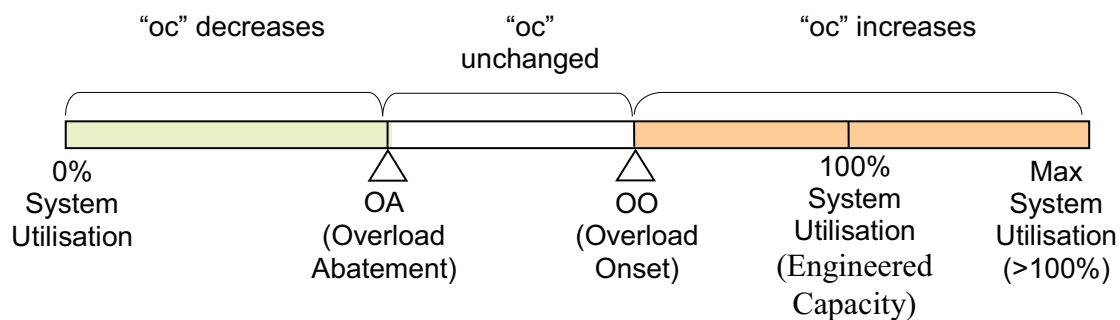


Figure 1 OC Management Rules





## 3 SIP Upstream Overload Control Configuration

An overview of the MO structure related to the SIP Upstream Overload Control is shown in Figure 2.

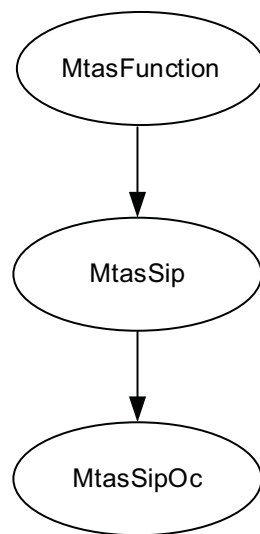


Figure 2 MO Structure Related to SIP Upstream Overload Control

For definition of configurable MOs and attributes related to the SIP Upstream Overload Control, refer to *Managed Object Model (MOM)*.

### 3.1 SIP Upstream OC Administrative State Configuration

The SIP upstream overload control is enabled by setting the `mtasSipOcAdministrativeState` attribute in the `MtasSipOc` MO to 1 (Unlocked).

If the `mtasSipOcAdministrativeState` is set to 0 (Locked), the SIP upstream overload control feature becomes inactive.

### 3.2 SIP Upstream OC Parameter Configuration

The configuration of system utilization thresholds, increase or decrease OC steps, default OC validity and OC Regulation Interval are shown in Table 2.

**Table 2 SIP Upstream OC Parameter Configuration**

Attribute	Activity
<code>mtasSipOcOnset</code>	The attribute designates the threshold which triggers the increase of the <code>oc</code> value. MTAS increments the reported <code>oc</code> value periodically while the system utilization level is above this threshold.
<code>mtasSipOcAbatement</code>	The attribute designates the threshold which triggers the decrease of the <code>oc</code> value. MTAS decrements the reported <code>oc</code> value periodically while the system utilization level is below this threshold.
<code>mtasSipOcDefIncrStep</code>	The attribute indicates the default step value by which the <code>oc</code> is increased.
<code>mtasSipOcDefDecrStep</code>	The attribute indicates the default step value by which the <code>oc</code> is decreased.
<code>mtasSipOcValidity</code>	The attribute indicates the OC validity in milliseconds which set as <code>oc-validity</code> in Via header.
<code>mtasSipOcRegulationInterval</code>	The attribute indicates the OC regulation period in milliseconds.

The SIP upstream overload control feature can be fine-tuned and adapted to the traffic behavior of the target network via these parameters. A few examples about their recommended usage are provided as follows:

### Overload Onset, Overload Abatement

The `mtasSipOcOnset` (OO) and `mtasSipOcAbatement` (OA) indicate the threshold to increase or decrease OC value.

For example, if the CPU is monitored and the load regulation limit is 85, the OO 85 here means when system utilization is above 85, the OC value start to increase; and the OA 75 here means when system utilization below 75, the OC value start to decrease.

In general, the default values of OO=85 and OA=75 must work properly but they can have to be adjusted to the nature of the traffic in the target network.

### Default OC Increase/Decrease Steps

`mtasSipOcDefIncrStep` and `mtasSipOcDefDecrStep` indicate how much the default step of the OC value should increase/decrease. Default values are 12 and 8. The mechanism that calculates the OC is not only based on the default oc increase/decrease step values. It is also based on the distance



between the system utilization and the OO, OA thresholds, and the previous OC value.

However, these default values are only guidelines and the optimal values are to be determined by the measurements in the target network.

### **OC Regulation Interval**

Current MTAS platforms support update resource utilization levels every 1 second (set the CM to 1000), therefore, the OC regulation interval cannot be shorter than the default value of 1 second.

It takes time for the reacting node to update traffic based on the OC value, and the update traffic load does not affect the utilization level immediately, so this duration provides enough time to learn the effect of the previously reported OC value.

If the traffic delays and the modifications of the server offered traffic volume within the client or both take longer time (>250–300 ms), then the OC regulation interval is to be prolonged to avoid the oscillations.

### **OC Validity**

The `oc-validity` is to be 1.5 times longer than the OC regulation interval by default.

If there is a risk in the client side that the new oc value cannot be received and evaluated within this interval, then the `oc-validity` period is to be prolonged for scheduling a proper restriction level towards the server until the next oc value is processed.

## **3.3 SIP Upstream OC Resource Configuration**

The supported resources are:



```
VS.LPM.LoadReg.CPUCoreUsage.LT
VS.LPM.LoadReg.CPUCoreUsage.ST
VS.LPM.LoadReg.CPUMaxUsage.LT
VS.LPM.LoadReg.CPUMaxUsage.ST
VS.LPM.LoadReg.CPUUsage.LT
VS.LPM.LoadReg.CPUUsage.ST
VS.LPM.LoadReg.HeapUsage.LT
VS.LPM.LoadReg.HeapUsage.ST
VS.LPM.LoadReg.MemUsage.LT
VS.LPM.LoadReg.MemUsage.ST
VS.LPM.LoadReg.MultiMMapUsage.LT
VS.LPM.LoadReg.MultiMMapUsage.ST
VS.LPM.LoadReg.TipcUsage.In.LT
VS.LPM.LoadReg.TipcUsage.In.ST
VS.LPM.LoadReg.TipcUsage.Out.LT
VS.LPM.LoadReg.TipcUsage.Out.ST
```

The `VS.LPM.LoadReg.CPUUsage.ST` is always monitored even if it is removed from the configuration. The operator can override only its utilization limit value used by the OC calculation algorithm – although it is not recommended.

The attribute `mtasSipOcResource` allows the operator to configure additional system resources and their utilization limits which are considered in overload situation.

The attribute `mtasSipOcResource` allows the operator to configure additional system resources and optionally their utilization limits which are considered in overload situation, for example:

- The empty configuration means that only the `VS.LPM.LoadReg.CPUUsage.ST` is considered in the OC calculation and the limit value 85 is used.
- The configuration of “`VS.LPM.LoadReg.CPUUsage.ST&85`”, “`VS.LPM.LoadReg.MemUsage.ST&95`”, “`VS.LPM.LoadReg.MultiMMapUsage.ST&85`” specifies that the CPU load, the Memory consumption, and map usage are to be regarded using the utilization limit values of 85, 95 and 85.

**Note:** Be cautious by overriding the default values.

The configuration of the monitored resources significantly affects the behavior of the SIP upstream overload control feature.





## 4 Performance Management

For measurements related to the SIP upstream overload control, refer to *Managed Object Model (MOM)*.





## 5 Fault Management

There is no alarm for SIP upstream overload control.