

# Oracle® Communications Diameter Signaling Router vSTP SS7 Security Guide



Release 9.2.0.0.0

G42863-01

September 2025

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2020, 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

<b>1</b>	<b>Introduction</b>	
1.1	Overview	1
1.2	References	1
<b>2</b>	<b>vSTP Security Overview</b>	
2.1	MTP Level Security	1
2.1.1	MTP Screening	1
2.1.2	SCTP Firewall Support	1
2.1.2.1	Feature Description	1
2.1.2.2	Firewall Manager Process	2
2.1.2.3	Feature Configurations	2
2.1.2.4	SCTP Firewall Alarms and Measurements	3
2.1.2.5	Troubleshooting	3
2.1.2.6	Dependencies	3
2.2	SCCP Level Security	3
2.2.1	GTT Routing	4
2.2.1.1	Flexible Linkset Optional Based Routing (FLOBR)	4
2.2.2	GTT Action Feature	12
2.2.2.1	GTT Throttle Action	13
2.2.2.2	GTT SCPVAL Action	20
2.2.3	Support for Applying GTT Action from intermediate GTT Sets	27
2.2.4	SFAPP ATIGTT Set	27
2.2.4.1	MMI Managed Objects for ATIGTT Set	27
2.2.5	MTP Based GTT with Screening Action	32
2.2.5.1	MTP Based GTT Feature Configuration	32
2.2.5.2	Dependencies	34
2.2.5.3	Troubleshooting	34
2.2.6	GTT Action Set Test Mode	34
2.2.6.1	Feature Configurations	34
2.2.6.2	GTT Action Set Test Mode Alarms and Measurements	36
2.2.6.3	Troubleshooting	36
2.2.6.4	Dependencies	36
2.3	TCAP Level Security	37

2.3.1	TCAP Opcode Based Routing (TOBR)	37
2.3.1.1	Feature Configurations	38
2.3.1.2	Troubleshooting	44
2.3.1.3	Dependencies	45
2.3.2	vSTP Multi Component Message Security	45
2.3.2.1	Feature Description	45
2.3.2.2	Feature Configurations	46
2.3.2.3	Troubleshooting	49
2.3.2.4	Dependencies	49
2.3.3	vSTP TCAP Decoding	49
2.3.3.1	Feature Description	49
2.3.3.2	TCAP Decoding Error Scenario	49
2.3.3.3	Feature Configuration	50
2.3.3.4	Troubleshooting	51
2.3.3.5	Dependencies	51
2.4	MAP Level Security	51
2.4.1	Map Based Routing (MBR)	51
2.4.1.1	Feature Configurations	52
2.4.1.2	Troubleshooting	58
2.4.1.3	Dependencies	58
2.4.2	Stateful Application Feature	58
2.4.2.1	Supported MAP Operations	59
2.4.2.2	VLR Validation	59
2.4.2.3	Velocity Check	62
2.4.2.4	Stateful Security Dynamic Learning	69
2.4.2.5	SFAPP Configurations	76
2.4.2.6	Dependencies	91
2.4.2.7	Troubleshooting	91
2.4.3	Support for CAT2 SS7 Security	92
2.4.3.1	Feature Overview	92
2.4.3.2	Feature Configurations	95
2.4.3.3	Troubleshooting CAT2 SS7 Security	99
2.4.3.4	Dependencies	99
2.4.4	vSTP SMS Home Router	100
2.4.4.1	Feature Description	100
2.4.4.2	vSTP Architecture	104
2.4.4.3	Feature Configurations	106
2.4.4.4	Home SMS Router Alarms and Measurements	113
2.4.4.5	Troubleshooting	115
2.4.4.6	Dependencies	115

## 3 GSMA Categorization

---

3.1	Supported Message Categories	1
3.1.1	Category 1	1
3.1.2	Category 2	3
3.1.3	Category 3	4

## 4 Security Logging and Visualization

---

4.1	Feature Description	1
4.1.1	Overview	1
4.1.2	Supported Operation Codes	1
4.2	Feature Configuration	3
4.2.1	MMI Managed Objects for Security Logging	3
4.2.2	GUI Configuration	5
4.3	Alarms and Measurement	7
4.4	Troubleshooting	8
4.5	Dependencies	8

## A Subscriber Information Disclosure

---

## B Network Information Disclosure

---

## C Subscriber Traffic Interception

---

## D Fraud

---

D.1	Illegitimate Redirection of Terminating or Originating Calls	D-1
D.2	USSD Request Manipulation	D-1
D.3	SMS Message Manipulation or Spoofing	D-1
D.4	Subscriber Profile Modification or Spoofing	D-1
D.5	Denial of Service	D-2

## E Manually Adding SGSN Field in UDR

---

# Preface

- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Conventions](#)

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

## Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

# My Oracle Support

My Oracle Support (<https://support.oracle.com>) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <http://www.oracle.com/us/support/contact/index.html>. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request.
2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
  - For Technical issues such as creating a new Service Request (SR), select **1**.
  - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

# What's New in This Guide

This section introduces the documentation updates for Release 9.2.0.0.0.

## Release 9.2.0.0.0 - G42863-01, September 2025

- The following updates are made in the [Feature Overview](#) section:
  - Added "Handling Multiple Countries with Shared MCCMNC Using TADIGIT Codes in CAT2IMSI Validation".
  - Added a [note](#) about SCPVAL to provide information about validations.
- Added the following features in the [SCCP Level Security](#) section:
  - [Support for Applying GTT Action from intermediate GTT Sets](#)
  - [SFAPP ATIGTT Set](#)
    - \* [MMI Managed Objects for ATIGTT Set](#)

# 1

## Introduction

### 1.1 Overview

The Virtual Signaling Transfer Point (vSTP) application is a secure and reliable signaling platform that provides **SS7**-focused signal transfer point (**STP**) and signaling gateway (**SG**) services that help manage intelligent routing, screening services, number portability, equipment identity register, and integrated performance/service management.

This document describes the security considerations and provides an overview of Virtual Signaling Transfer Point (vSTP) configurations to counter potential SS7 attacks. The security features in vSTP provide an additional set of capabilities to monitor, throttle, and validate messages.

### 1.2 References

- Virtual Signaling Transfer Point (vSTP) User's Guide
- Mobile Number Portability (MNP) User's Guide
- Equipment Identity Register (EIR)
- TIF User Guide

# 2

## vSTP Security Overview

This chapter describes the overview of vSTP security with detailed configurations. The vSTP security features are described as per the following security levels:

### 2.1 MTP Level Security

This section describes the MTP Level security features of vSTP:

#### 2.1.1 MTP Screening

The MTP Screening feature provides a mechanism to screen the incoming calls based on the MTP Screening rules configurations. The MTP screening rule is an entity to configure all the screening rules for a Screen Set.

For more details on MTP Screen Rules configurations, refer to *vSTP User's Guide*.

#### 2.1.2 SCTP Firewall Support

vSTP achieves network security by utilizing the Linux firewall provided by Oracle Linux distribution that serves as the platform for vSTP software.

vSTP configures firewall rules in the Linux firewall on each server to allow only essential network traffic. The vSTP software is composed of various components providing unique services, and each component is responsible to configure the firewall rules to allow the network traffic destined to and originated from the provided services. While platform services are internal and not configurable in customer deployments, the signaling services are completely configurable at runtime.

##### Note

By default, the SCTP Firewall feature remains enabled.

##### 2.1.2.1 Feature Description

The SCTP Firewall feature brings the flexibility and capability in vSTP to dynamically determine and customize the linux firewall on each vSTP-MP server. It allows only the essential network traffic, pertaining to active signaling configurations. The in-bound signaling traffic is accepted by the vSTP application over the configured and enabled connections only. By monitoring the active configuration, this feature determines, which configured connections are enabled. It then configures the Linux Firewall on the vSTP-MP servers to allow the signaling network traffic for those connections only and completely deny the non-signaling network traffic (non-signaling traffic is traffic from internal services i.e. SSH, FTP, HTTP, HTTPS, etc.), thus providing added security to the signaling networks.

## 2.1.2.2 Firewall Manager Process

The FMP (Firewall Manager Process) is added to the vSTP MP software. This process manages the Linux firewall (local to the vSTP-MP) by keeping it in sync with the active signaling connection configuration. The signaling firewall is administered (Enabled or Disabled) from SOAM server via the supported user interfaces and will take effect on all the vSTP-MPs in the signaling node. When a connection is enabled or disabled, a configuration change notification is sent by comcol's data change monitor to FMP, which updates the Linux Firewall rules to allow or disallow the network traffic pertaining to the connection in context.

In addition, the FMP also periodically audits the firewall configuration against the active configuration and automatically corrects any mismatch it finds in the firewall rules against the enabled connection quadruples.

### 2.1.2.2.1 Feature Implementation

The firewall rules are implemented using the IPsets and IPTables. The functionality managed by each table is as follows:

- IPsets are used to set up, maintain and inspect the set of IPs in the Linux kernel. Depending on the type of the set, an IP set may store IP(v4/v6) addresses, (TCP/UDP) port numbers, IP and MAC address pairs, IP address and port number pairs and so on. IPSet are the collection of IP addresses. The format of IPSet is: `hash: ip, port`
  - The `hash:ip, port` set type uses a hash to store IP address and port number pairs.
  - The port number is interpreted together with protocol(default TCP) and zero protocol number cannot be used.

These IPsets are utilized by IPTables to either ALLOW or RESTRICT IP's to/from the network. IPsets used in vSTP Firewall Enhancement are:

- `dsrIPv4conns` – Stores active connection attributes/connection quadruple from VstpConnections table (local IPv4 address, transport protocol, local port number and remote IPv4 address).
- `dsrIPv4ServicePorts` – Stores all the configured protocol and port numbers from VstpConnectionNode table which is dynamically updated.
- IPTables matches and targets referring to sets create references, which protect the given sets in the kernel. A set cannot be destroyed while there is a single reference pointing to it.

### 2.1.2.2.2 Auditing

The SCTP Firewall feature provides an Audit Manager to perform periodic auditing of the configured IPTables and IPsets by matching their contents with the configured data in the DB tables. The audit is performed only in case when the Firewall admin state is in **Enabled** state.

## 2.1.2.3 Feature Configurations

This section provides procedures to perform the vSTP SCTP Firewall functionality.

The vSTP SCTP Firewall is configured using the vSTP managed objects and vSTP GUI. The MMI API contains details about the URI, an example, and the parameters available for each managed object.

## 2.1.2.4 SCTP Firewall Alarms and Measurements

### Alarms and Events

The following table lists the alarms or events specific to the SCTP Firewall functionality for vSTP:

Event ID	Event Name
25607	Signaling Firewall is administratively Disabled
25608	Abnormal vSTP-MP Firewall
25601	DSR Signaling Firewall configuration inconsistency detected
25609	Firewall Configuration Error encountered

## 2.1.2.5 Troubleshooting

When vSTP SCTP Firewall feature is On, the events specific to this feature are generated. For more information, see [SCTP Firewall Alarms and Measurements](#).

## 2.1.2.6 Dependencies

The SCTP Firewall feature for vSTP has no dependency on any other vSTP operation.

## 2.2 SCCP Level Security

This section describes the Signaling Connection Control Part (SCCP) of of the SS7 protocol.

The Global Title Translation (GTT) feature is designed for SCCP. The GTT feature uses Global Title Address (GTA) information to determine the destination of the MSU. The Translation Type (TT) indicates which GTT table is used to determine the routing to a particular service database. Each GTT table includes the Point Code (PC) of the node containing the service database, the SubSystem Number (SSN) identifying the service database on that node, and a Routing Indicator (RI). The RI determines if further GTTs are required. GTA and TT are contained in the Called Party Address (CdPA) field of the MSU.

The GTT feature changes the destination PC and the origination PC in the routing label. The GTA information is not altered.

Depending on how the GTT data is configured, the GTT may also change the RI, SSN, or the TT in the CdPA. The gray shaded areas in the following tables show the message fields affected by GTT.

**Figure 2-1 ANSI MSU (ANSI Message Signal Unit)**

BSN FSN LI	SIO			SIF				
	XX NIC	XX PRI	XXXX SI	Routing Label			CGPA	CDPA Length
				DPC	OPC	SLS	Length Address Indicator	Address Indicator
				NCM NC NI	NCM NC NI	xx	(x x xxxx x x)	(x RI xxxx xx)
							Subsystem Point Code	Subsystem Point Code
							(NCM NC NI)	(NCM NC NI)
							Address	Address
							(Translation Type)	(Translation Type)
							(Digits)	(Digits)

**Figure 2-2 ITU-I MSU (ITU International Message Signal Unit)**

BSN FSN LI	SIO			SIF				
	xx	xx	xxxx	Routing Label			CGPA	CDPA Length
	NIC	PRI	SI	DPC	OPC	SLS	Length Address Indicator (x x xxxx x x)	Address Indicator (x RI xxxx xx)
				NCM AREA ZONE	ID AREA ZONE	xx	Subsystem Point Code (ID AREA ZONE)	Subsystem Point Code (ID AREA ZONE)
								Address (Translation Type) (Digits)

**Figure 2-3 14-Bit ITU-N MSU (14-Bit ITU National Message Signal Unit)**

BSN FSN LI	SIO			SIF				
	xx	xx	xxxx	Routing Label			CGPA	CDPA Length
	NIC	PRI	SI	DPC	OPC	SLS	Length Address Indicator (x x xxxx x x)	Address Indicator (x RI xxxx xx)
				NPC	NPC	xx	Subsystem Point Code (NPC)	Subsystem Point Code (NPC)
								Address (Translation Type) (Digits)

**Figure 2-4 24-Bit ITU-N MSU (24-Bit ITU National Message Signal Unit)**

BSN FSN LI	SIO			SIF				
	xx	xx	xxxx	Routing Label			CGPA	CDPA Length
	NIC	PRI	SI	DPC	OPC	SLS	Length Address Indicator (x x xxxx x x)	Address Indicator (x RI xxxx xx)
				MSA SSA SP	MSA SSA SP	xx	Subsystem Point Code (MSA SSA SP)	Subsystem Point Code (MSA SSA SP)
								Address (Translation Type) (Digits)

## 2.2.1 GTT Routing

The Global Title Translation (GTT) feature is designed for the Signaling Connection Control Part (SCCP) of the SS7 protocol. The routing options described in this section allow you to add translations to parameters, code, and components for additional flexibility in routing a message.

### 2.2.1.1 Flexible Linkset Optional Based Routing (FLOBR)

FLOBR supports Linkset based routing and Flexible routing.

- Linkset based routing routes GTT traffic based on the incoming linkset
- Flexible routing routes GTT traffic based on parameters such as MTP, SCCP, and TCAP in a flexible order on a per translation basis

With the FLOBR feature, you can change the default CdPA GTTSET to point to any GTT set type and find the translation.

FLOBR works based on the following rules:

1. When GTT mode is FLOBR CDPA, CDPA fields in the MSU are used for GTT selector search and the GTT set is taken from the CDPA GTT SET Name configured in the selector entry.
2. When GTT mode is FLOBR CGPA, CGPA fields in the MSU are used for GTT selector search and the GTT set is taken from the CGPA GTT SET Name configured in the selector entry.
3. When GTT hierarchy is FLOBR CDPA and FLOBR CGPA, GTT selectors are searched as defined in 1. If no selector match is found or CDPA GTTSET is not provisioned, GTT selectors are searched as defined in 2.
4. When GTT hierarchy is FLOBR CGPA and FLOBR CDPA, GTT selectors are searched as defined in 2. If no selector match is found or CGPA GTTSET is not provisioned, GTT selectors are searched as defined in 1.
5. If GTT selectors are not found as specified in 1, 2, 3 or 4, then vSTP considers this a translation failure.
6. You can provision a fallback option for each translation in FLOBR to tell it how to route an MSU under the following conditions:
  - Routing when a search fails
  - Routing when the same GTT set name is referred to more than once
  - Limiting the number of database searches to seven (7)
7. When a fallback option is set to No, the GTT fails and the MSU is discarded.
8. When a fallback option is set to Yes, the GTT performs based on the last matched entry.

### 2.2.1.1.1 Feature Configurations

This section provides procedures to perform the FLOBR functionality.

The FLOBR is configured using the vSTP managed objects and vSTP GUI. The MMI API contains details about the URI, an example, and the parameters available for each managed object.

#### 2.2.1.1.1.1 MMI Managed Objects for FLOBR

MMI information associated with FLOBR can be configured from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* gets opened, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for FLOBR:

Managed Object Name	Supported Actions
Linkset	GET, PUT, POST, DELETE
GTT Sets	GET, PUT, POST, DELETE
GTT Selector	GET, PUT, POST, DELETE
GTA	GET, PUT, POST, DELETE

**linksets**

**POST**

Create a file with following content to set value for the `gttmode` parameter. File name could be anything, for example option name can be used as filename:

```
{
    "asNotification": false,
    "enableBroadcastException": false,
    "gttmode": "Fcd",
    "ituTransferRestricted": false,
    "linkTransactionsPerSecond": 10000,
    "localSignalingPointName": "VSTP1SP",
    "name": "V1V2LS",
    "remoteSignalingPointName": "VSTP2SP",
    "routingContext": 0,
    "type": "M2pa"
}
```

Execute the following command on Active SOAM to update the data:

```
/vstp/linksets -v POST -r /<Absolute path>/<File Name>
```

Execute the following command to display the content:

```
vstp/gttsets
{
    "configurationLevel": "1",
    "domain": "Itu",
    "gttSetType": "Skbcsm",
    "name": "set1"
}
```

Sample Output:

```
{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}
```

## GET

Execute the following command to display the content:

```
vstp/linksets
```

Sample Output:

```
{
    "data": [
        {
            "asNotification": false,
            "configurationLevel": "0",
            "enableBroadcastException": false,
```

```

        "gttmode": "Fcd",
        "ituTransferRestricted": false,
        "linkTransactionsPerSecond": 10000,
        "localSignalingPointName": "VSTP1SP",
        "name": "V1V2LS",
        "remoteSignalingPointName": "VSTP2SP",
        "routingContext": 0,
        "type": "M2pa"
    }
],
"links": {},
"messages": [],
"status": true
}

```

**gttsets****POST**

Create a file with following content to set value for the `gttSetType` parameter. File name could be anything, for example option name can be used as filename:

```

{
    "configurationLevel": "5",
    "domain": "Itu",
    "gttSetType": "Cdgta",
    "name": "SET1"
}

```

Execute the following command on Active SOAM to create the data:

```
/vstp/gttsets -v POST -r <filename>.json
```

**Sample Output:**

```

{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}

```

**GET**

Execute the following command to display the content:

```
/vstp/gttsets
```

**Sample Output:**

```

{
    "data": [
        {

```

```

        "configurationLevel": "5",
        "domain": "Itu",
        "gttSetType": "Cdgt",
        "name": "SET1"
    }
],
"links": {},
"messages": [],
"status": true

```

### gttselectors

#### POST

Create a file with following content. File name could be anything, for example option name can be used as filename:

```

{
    "domain": "Itui",
    "globalTitleIndicator": "TtOnly",
    "cdgttsn": "SET1",
    "name": "Sel2",
    "translationType": 155,
    "linksetName" : "V1V2LS"
}

```

Execute the following command on Active SOAM to update the data:

```
/vstp/gttselectors -v POST -r <filename>.json
```

#### Sample Output:

```

{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}

```

#### GET

Execute the following command to display the content:

```
/vstp/gttselectors
```

#### Sample Output:

```

{{
    "data": [
        {
            "cdgttsn": "SET1",
            "configurationLevel": "56",
            "domain": "Itui",
            "globalTitleIndicator": "TtOnly",

```

```

        "linksetName": "V1V2LS",
        "name": "Sel2",
        "translationType": 155
    }
],
"links": {},
"messages": [],
"status": true
}

```

### globaltitleaddresses

#### POST

Create a file with following content. File name could be anything, for example option name can be used as filename:

```

{
    "ccgt": false,
    "cgpcaction": "Dflt",
    "endAddress": "11109194412880",
    "fallback": "Sysdflt",
    "gttSetName": "SET1",
    "optsn": "SET401",
    "mapSetId": 0,
    "routingIndicator": "Ssn",
    "rspName": "MEAT3SP",
    "startAddress": "01209197411063",
    "subsystem": 18,
    "translateIndicator": "Dpcssn"
}

```

Execute the following command on Active SOAM to update the data:

```
/vstp/globaltitleaddresses -v POST -r <filename>.json
```

#### Sample Output:

```

{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}

```

#### GET

Execute the following command to display the content:

```
vstp/globaltitleaddresses
```

## Sample Output:

```

{
  "data": [
    {
      "ccgt": false,
      "cgpcaction": "Dflt",
      "configurationLevel": "54",
      "defmapvr": "V3",
      "endAddress": "11109194412880",
      "fallback": "Sysdflt",
      "gttSetName": "SET1",
      "optsn": "SET401",
      "mapSetId": 0,
      "mrnSetId": 0,
      "routingIndicator": "Ssn",
      "rspName": "MEAT3SP",
      "startAddress": "01209197411063",
      "subsystem": 18,
      "translateIndicator": "Dpcssn",
      "uniqueIdentifier": "3fe8634e-0aee-462a-9b46-cf842b739571"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}

```

## 2.2.1.1.1.2 GUI Configurations for FLOBR

The FLOBR functionality can be configured from Active System OAM (SOAM). Select **VSTP**, and then **Configuration** page.

- The following parameter on the **Link Set** page used to perform the configurations:
  - `GTT Mode`: The GTT Mode hierarchy for this link set.
- The following parameter on the **GTT Set** page are used to perform the configurations:
  - `Gtt Set Type`: The parameter defines the type of GTT Set.
- The following parameter on the **SCCP GTT Selectors** page are used to perform the configurations:
  - `CdPA GTT Set Name`: The CdPA GTT set name (`/vstp/gttsets`) associated with this GTT Selector.
  - `CgPA GTT Set Name`: The CgPA GTT set name (`/vstp/gttsets`) associated with this GTT Selector.
  - `Linkset Name`: Linkset name (`/vstp/linksets`) associated with this GTT Selector.
- The following parameter on the **Global Title Addresses** page are used to perform the configurations:
  - `Application Context Name`: Application context name. This parameter specifies the ITU TCAP `acn` field in the incoming MSU.
  - `TCAP Opcode`: The TCAP opcode field in the incoming MSU.
  - `Package Type`: The ANSI and ITU TCAP package type.

- TCAP Opcode: Operation Code Tag.

For more information, see GUI Configurations section in *vSTP User Guide*.

### 2.2.1.1.1.3 FLOBR Alarms and Measurements

#### Alarms/Events

The following table lists the event specific to the FLOBR functionality for vSTP:

**Table 2-1 Alarms/Events**

Alarm/Event ID	Name
70281	GTT FLOBR Duplicate Set Name Failed
70282	GTT FLOBR Duplicate Set Name Warning
70283	GTT FLOBR Max Search Depth Failed
70284	GTT FLOBR Max Search Depth Warning
70279	GTT MBR Duplicate Set Type Failed
70280	GTT MBR Duplicate Set Type Warning
70285	MBR Decoding Failed

For more details related to alarms and events, refer to *DSR Alarms and KPI Guide*.

#### Measuremet

The following table lists the measurements specific to the FLOBR functionality for vSTP:

**Table 2-2 Measurements**

Measurements ID	Measurements Name
21465	VstpMSUCgpaFlexiGTT
21466	VstpMSUCdpaFlexiGTT
21467	VstpCgpaGTTNoSelectorMatch
21468	VstpCdpaGTTNoSelectorMatch
21469	VstpCgpaGTTFail
21470	VstpCdpaGTTFail

For more details related to measurements, refer to *DSR Measurement Reference Guide*.

### 2.2.1.1.2 Troubleshooting

- If selector match gets failed, then the `VstpCdpaGTTNoSelectorMatch` or `VstpCgpaGTTNoSelectorMatch` measurements are incremented as per GTT mode.
- If Global Title Translation fails, then `VstpCdpaGTTFail` or `VstpCgpaGTTFail` measurements are incremented as per GTT mode.
- If GTT is successful then `VstpMSUCdpaFlexiGTT` or `VstpMSUCgpaFlexiGTT` measurements are incremented as per GTT mode.
- If any of the above statement fails as per scenarios then verify the configuration.

In case of the error scenarios, the vSTP measurements are pegged. For information related to XUDT UDT Conversion measurements, see [TCAP Opcode Based Routing Alarms and Measurements](#).

### 2.2.1.1.3 Dependencies

The FLOBR functionality for vSTP has no dependency on any other vSTP operation.

## 2.2.2 GTT Action Feature

The Global Title Translation (GTT) action feature performs additional actions on the incoming/translated Message Signaling Unit (MSU) coming from the GTT. Configure GTT Action, GTT Action Set, and GTA Managed Object (MO) to use these as optional features.

vSTP supports the following types of GTT actions:

- Discard
- UDTS
- TCAP Error
- Forward
- Duplicate
- SFAPP
- SFTHROT
- SCPVAL

#### **Discard**

The Discard GTT action discards incoming MSU.

#### **UDTS**

The Unit Data Service (UDTS) GTT action marks the MSU as discarded and an error response is sent back with an udts error code.

#### **TCAP Error**

The Transaction Capabilities Application Part (TCAP) Error GTT action marks the MSU as discarded and an error response is sent back with an tcap error code.

#### **Forward**

The Forward GTT action forwards the incoming/translated MSU to a specified point code per configuration. The MSU does not forward to translated point code.

If the Forward GTT action fails, then default actions are performed per configuration:

- Fallback means forward the MSU to translated point code
- Discard an incoming MSU
- Send a UDTS response with an udts error code per configuration
- Send a TCAP error response with an tcap error code per configuration

#### **Duplicate**

The Duplicate action sends a copy of incoming/translated MSU to a specified point code per configuration. The MSU does sent to translated as well as duplicate point code.

## SFAPP

The Stateful Application (SFAPP) action validates the messages coming in for a subscriber by validating them against the Visitor Location Register (VLR).

## SFTHROT

The GTT Throttle action is part of SS7 security firewall. It provides support for Egress throttling of GTT messages in vSTP. For more details, see [GTT Throttle Action](#).

## SCPVAL

The SCPVAL GTT action along with relevant parameters performs the validation on MAP parameters by comparing the SCCP and MAP digits. For more details see, [GTT SCPVAL Action](#).

### 2.2.2.1 GTT Throttle Action

The GTT Throttle is a GTT Action that performs the Egress throttling of GTT messages in vSTP. This action is part of SS7 security firewall. GTT Throttle feature provides the support for Egress throttling of individual messages and group of GTT messages in vSTP.

This functionality can be achieved by the following actions:

- Enabling the `SFTHROT` action for Group Throttling
- Enabling the `Indv_Throt` action for individual Throttling

#### Group Throttling

The Group GTT Throttle Feature provides Egress Throttling of GTT messages with `SFTHROT` GTT Action. For each GTT Action, user provides threshold as the maximum number of MSUs hitting the GTT action per second. The SMS framework to accumulates the total number of MSU count per `SFTHROT` action.

When an MSU hits a GTT action of the type `SFTHROT`, the MSU count of that action is updated. SMS framework accumulates the total number of messages per `SFTHROT` action on MP Leader and sends cumulative count to all MPs across site. If the cumulative count of messages crosses the provisioned threshold, MPs starts throttling that messages. Any MSU hitting the GTT action gets discarded and the MSU count of that messages is not increased due to throttling, due to which cumulative value decreases in next sliding window. Once the cumulative value is lower than the configured threshold, the messages are allowed.

#### Note

Group Throttling supports 1000 groups.

#### Individual Throttling

The Individual GTT Throttling Feature provides Egress Throttling of GTT message with `Indv_Throt` GTT Action. For each `Indv_Throt` GTT Action, user provides threshold as maximum number of MSUs hitting the GTT action per second per GTA. The SMS framework is used to accumulate the total number of MSU count of `Indv_Throt` action.

When an MSU hits a GTT action of the type `Indv_Throt`, the MSU count of that GTA gets updated. The SMS framework accumulates the total number of messages on MP Leader and sends cumulative count to all MPs across site. If the cumulative count of GTA crosses the

provisioned threshold, MPs start throttling that GTA. Any MSU hitting that GTA gets discarded and the MSU count of that message is not increased due to throttling, due to which cumulative value decreases in next sliding window. Once the cumulative value is lower than the configured threshold, the messages are allowed.

**Note**

Individual throttling supports 100K GTA.

### 2.2.2.1.1 Workflow for GTT Throttle Action

The GTT Throttle action works based on the following rules:

1. When an MSU hits a GTT action of the type SFTHROT, the MSU count of that action gets updated. For Indv\_Throt, the MSU count of the GTA is updated.

**Note**

The Shared Metric Service (SMS) framework is used to accumulate the total number of MSU count per SFTHROT action.

2. The MSU count is updated only on the Message Processor (MP) on which the message is received for that action. On the other hand, the Threshold configuration for SFTHROT action is across the MPs.

**Note**

For each GTT Action, user provisions a threshold value that is the maximum number of MSUs hitting the GTT action per second.

3. Two sysmetrics are registered. The first is for MSU count per MP and second one for cumulative MSU count across the site.
4. Aggregation of the MSU count from all the MPs is done by the MP Leader. There is only one MP leader across the site. It performs the aggregation of MSU counts. Rest of the MPs across the site are known as followers.
5. Whenever a message comes to any MP, it will increment the sysmetric count of that MP known as local sysmetric count. All the follower MPs will send the local sysmetric count data to the MP Leader to get the aggregated value of that action.
6. The MP Leader receives the data from all the other MPs including its own local sysmetric count. It will do the aggregation and broadcast the cumulative count to all the MPs.
7. The SMS framework is used to send local sysmetric count to MP leader and receive the aggregated sysmetric count from it. The aggregation of the count is taken care by SMS framework hence, any degradation in SMS service will impact the feature.
8. When GTT message is received for SFTHROT/Indv\_Throt action, then the aggregated sysmetric count is compared with the configured threshold value for that action:
  - If the aggregated sysmetric count value is lesser than the configured threshold value, then the message is allowed and the local sysmetric count value is increased by 1.
  - If the aggregated sysmetric count value is more than the configured threshold value, then the local sysmetric count value does not get increased due to throttling. The GTT

message is discarded, discard measurement is pegged for that action, and an alarm is raised.

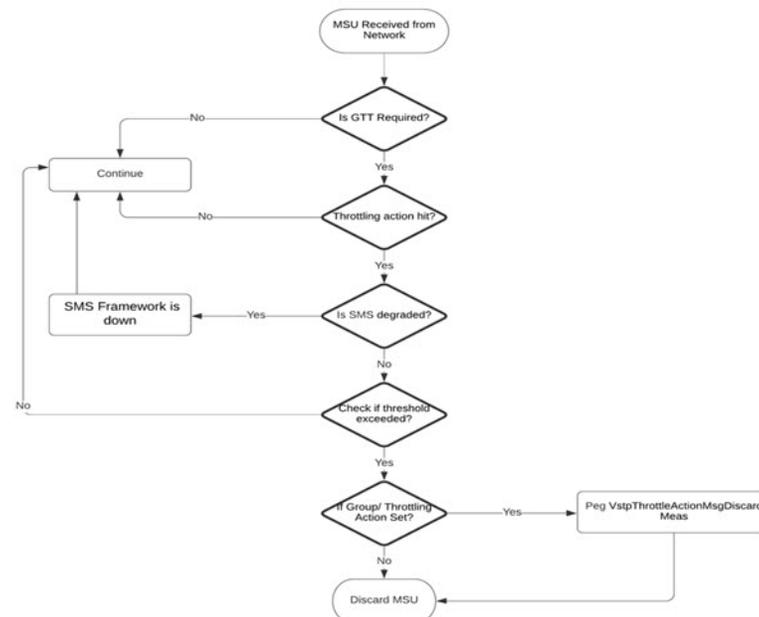
- a. The alarm gets cleared once the aggregated symmetric count drops below 90% of the configured threshold value.
- b. As there is no local symmetric is pegged, the aggregated count will be decreased in next sliding window. Convergence time is 2 sec.
- c. Once the cumulative value drops below the configured threshold, it will allow the GTT messages for that action and the local symmetric count will be increased.

### Note

For GTT Throttle action, an error margin of +2% to -2% of the provisioned threshold value must be considered. The error margin depends on the cloud infrastructure load & burst pattern of incoming traffic.

The following figure shows the process flow for GTT Throttle action:

**Figure 2-5 Process Flow of GTT Throttle Action**



### 2.2.2.1.2 MMI Managed Objects for GTT Throttle Action

MMI information associated with GTT Throttle action is accessed from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* displays, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for GTT Throttle action.

**Table 2-3 GTT Throttle Action Managed Objects and Supported Operations**

Managed Object Name	Supported Operations
gttactions	Insert, Update, Delete
gttactionsets	Insert, Update, Delete

**gttactions - Insert, Update, Delete**

- **For SFTHROT Action Type (Group Throttling)**

Create a file with the following content:

```
$ cat gttaction.txt
{
    "act": "Sfthrot",
    "actid": "GA1",
    "defactid": "fallback",
    "threshold": 99
}
```

**Note**

The **threshold** is mandatory parameter for SFTHROT action type. Range is **1 to 4294967295**. Modification is allowed for threshold.

Execute the following command on an active SOAM to insert:

```
/vstp/gttactions -v POST -r /<Absolute path>/<Filename>
```

Example output:

```
/vstp/gttactions -v POST -r gttaction.txt
{
    "act": "Sfthrot",
    "actid": "GA1",
    "defactid": "fallback",
    "taIndex": 0,
    "threshold": 36
}
```

- **For INDV\_THROT Action Type (Individual Throttling)**

Create a file with the following content:

```
$ cat gttaction.txt
{
    "act": "Indvthrot",
    "actid": "indv1",
    "defactid": "fallback",
    "gtaLength": 10,
```

```
    "threshold": 35
  }
```

### Note

- vSTP supports a total of 100K GTA while provisioning GTA entries with action set having action type as INDV\_THROT. This is the combined limit for all the Indv\_Throt actions.
- The **threshold** and **gtaLength** are mandatory parameters for INDV\_THROT action type. Range is **1 to 4294967295**. Modification is allowed for threshold.
- If no action type is selected, then the **threshold** value remains **1**, by default.

Execute the following command on an active SOAM to insert:

```
/vstp/gttactions -v POST -r /<Absolute path>/<Filename>
```

Example output:

```
/vstp/gttactions -v POST -r gttaction.txt {
    "act": "Indvthrot",
    "actid": "indv1",
    "defactid": "fallback",
    "gtaLength": 10,
    "taIndex": 65535,
    "threshold": 3
}
```

### Note

vSTP supports a total of 100K GTA while provisioning GTA entries with action set having action type as INDV\_THROT. This is the combined limit for all the Indv\_Throt actions.

## gttactionsets - Insert, Update, Delete

Create a file with the following content:

```
{
    "actsn": "actset1",
    "actid1": "Act1"
}
```

**Note**

- Maximum one SFTHROT action is allowed to be provisioned per VstpGTTActionSet entry.
- Maximum one INDV\_THROT action is allowed to be provisioned per VstpGTTActionSet entry.
- If both INDV\_THROT and SFTHROT actions are selected, INDV\_THROT action gets the first preference and it is followed by the SFTHROT action. If INDV\_THROT action is not selected, then SFTHROT gets the first preference.

Execute the following command on an active SOAM to insert:

```
/vstp/gttactionsets -v POST -r /<Absolute path>/<File Name>
```

Example output:

```
/vstp/gttactionsets -v POST -r /tmp/ActSet1
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}
```

Execute the following command on an active SOAM to update:

```
/vstp/gttactionsets -v PUT -r /<Absolute path>/<File Name>
```

Example output:

```
/vstp/gttactionsets -v PUT -r /tmp/actset1
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}
```

Execute this command on an active SOAM to delete:

```
/vstp/gttactionsets/<Set Name> -v DELETE
```

Example output:

```
/vstp/gttactionsets/Set1 -v DELETE
No output returned by URI: https://localhost/mmi/dsr/v3.0/vstp/gttactionsets/
Set1? for 'DELETE' operation
```

Execute the following command to display:

```
/vstp/gttactionsets
```

Example output:

```
/vstp/gttactionsets
{
  "data": [
    {
      "actsn": "actset1",
      "actid1": "Act1"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

### 2.2.2.1.3 GTT Throttle Measurements

#### Measurements

The following table lists the measurements specific to GTT Throttle action:

Measurement ID	Measurement Name
21723	VstpThrottleActionMsgDiscard
	VstpThrottleActionMsgReceived

For more details related to measurements, refer to Measurement Reference document.

### 2.2.2.1.4 Dependencies

The GTT Throttle action support for vSTP has no dependency on any other vSTP operation.

#### Points To Consider

The following points must be configured while using the GTT Throttling:

- There is error margin of the provisioned threshold value depending on the cloud infrastructure load & burst pattern of incoming traffic.
- The error margin for Individual Throttling is greater than the Group Throttling as Aggregation timeout for `Indv_Throt` is 200 ms whereas, the Aggregation timeout for Group Throttling is 10 ms.

### 2.2.2.1.5 Troubleshooting

In case of error scenario, check the incoming traffic. The incoming traffic must be 100% or above the provisioned threshold value for respective actid with SFTHROT action.

### 2.2.2.2 GTT SCPVAL Action

The SCCP MAP Validation (SCPVAL) is a GTT Action that performs validation check on the of vSTP map parameters. This action is part of SS7 security firewall.

For example, in vSTP few of the map parameters must be same as either SCCP CdPA or CgPA. The GTT SCPVAL action do this validation check with a comparison between SCCP parameters and the map digits.

#### Note

The SCPVAL action is applicable only for the following messages coming to the vSTP:

- MO-FSM (MAP version 2 or 3)
- MT-FSM (MAP version 3)

The SCPVAL action has the following set of parameters to execute the functionality:

Parameter Name	Description	Value
SPRM	Define the SCCP parameter value. It is a mandatory parameter.	The value can be either of the following: <ul style="list-style-type: none"> <li>• CGGTA</li> <li>• CDGTA</li> </ul>
TPRM	Define the TCAP parameter value. It is a mandatory parameter.	The value can be either of the following: <ul style="list-style-type: none"> <li>• SMRPOA</li> <li>• SMRPDA</li> </ul>
NDGT	Specifies the number of digits that needs to be matched between SCCP parameter and MAP parameter. This is an optional parameter.	The value can be either of the following: <ul style="list-style-type: none"> <li>• Any digit between 1-21</li> <li>• All</li> </ul> Default value: All

#### Note

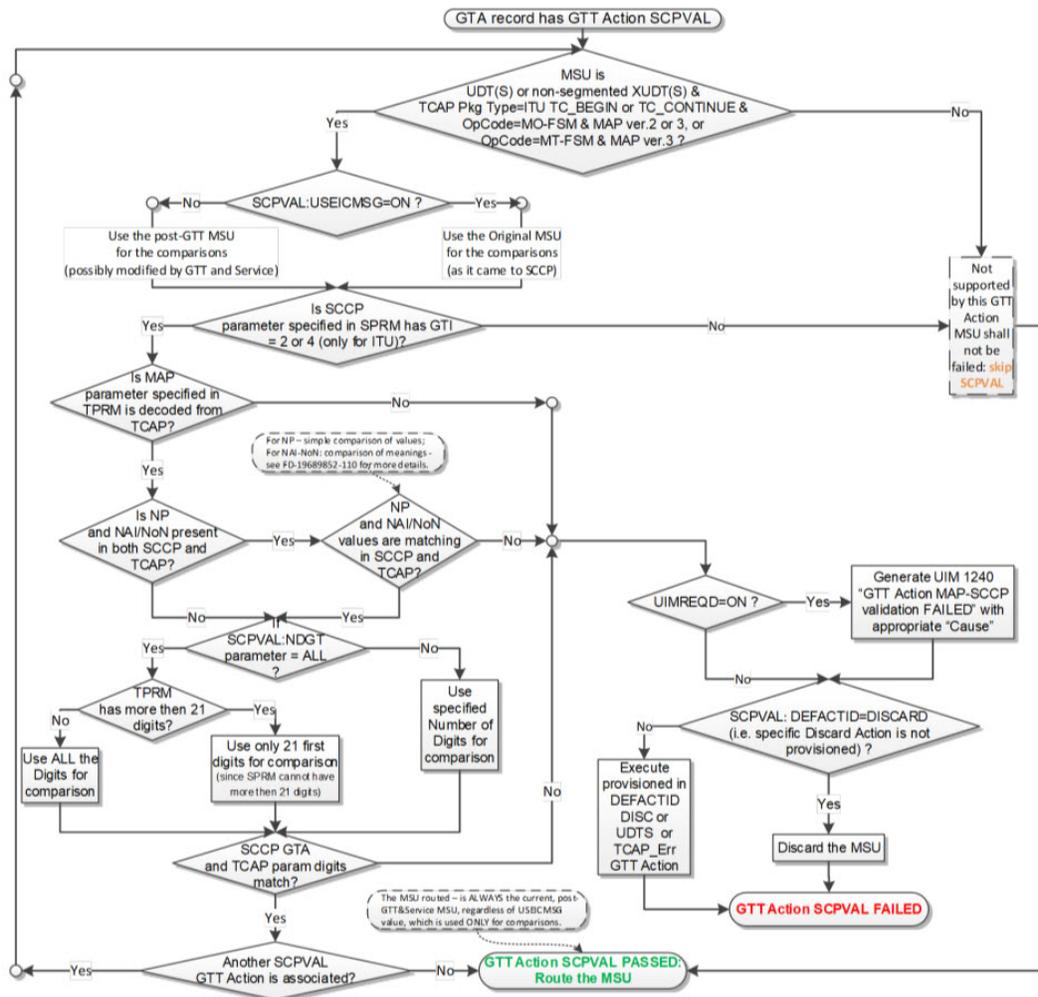
If GTTACTN is SCPVAL (Ir21ToTCAP) and NDGT equals ALL, the HLR GTA will be derived from the CAT2GTA table. If NDGT contains a specific number, the MGT will be retrieved from the CAT2GTA table.

Parameter Name	Description	Value
USEICMSG	Specifies whether to retrieve the data for comparison from the original message or from the post-GTT message.	The value can be either of the following: <b>OFF</b> : Use original message as received by the SCCP. <b>ON</b> : Use post-GTT message that is, after possible GTT translation/modification data has been applied.
UIMREQD	Specifies if an event has be generated in case of GTT action failure.	The value can be either of the following: <ul style="list-style-type: none"> <li>• <b>ON</b>: Event to be generated</li> <li>• <b>OFF</b>: No event to be generated</li> </ul>
DEFACTID	Defines the default action that is performed when SCPVAL GTT action fails.	String value

### 2.2.2.2.1 Workflow for SCPVAL Action

The following flowchart describes the implementation of MAP SCCP validation:

Figure 2-6 SCCP MAP Validation Flowchart



### 2.2.2.2.2 MMI Managed Objects for SCPVAL

MMI information associated with SCPVAL action is accessed from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* displays, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for SCPVAL action.

Table 2-4 vSTP SCPVAL Managed Objects and Supported Operations

Managed Object Name	Supported Operations
gttactions	Insert, Update, Delete
gttactionsets	Insert, Update, Delete

### gttactions - Insert, Update, Delete

Create a file with the following content:

```
{
  "act": "Scpval",
  "actid": "Act1",
  "defactid": "fallback",
  "ndgt": "2",
  "sprm": "Cdgta",
  "tprm": "Smpda",
  "uimreqd": "true"
  "useicmsg": "true"
}
```

Execute the following command on an active SOAM to insert:

```
/vstp/gttactions -v POST -r /<Absolute path>/<File Name>
```

Example output:

```
/vstp/gttactions -v POST -r /tmp/GttAct1
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}
```

Execute the following command on an active SOAM to update:

```
/vstp/gttactions -v PUT -r /<Absolute path>/<File Name>
```

Example output:

```
/vstp/gttactions -v PUT -r /tmp/GttAct1
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}
```

Execute this command on an active SOAM to delete:

```
/vstp/gttactions/<Rule Name> -v DELETE
```

Example output:

```
/vstp/gttactions/Act1 -v DELETE
No output returned by URI: https://localhost/mmi/dsr/v3.0/vstp/gttactions/
Act1? for 'DELETE' operation
```

Execute the following command to display:

```
/vstp/gttactions
```

Example output:

```
/vstp/gttactions
{
  "data": [
    {
      "act": "Scpval",
      "actid": "Act1",
      "defactid": "fallback",
      "ndgt": "2",
      "sprm": "Cdgta",
      "tprm": "Smrpda",
      "uimreqd": true,
      "useicmsg": true
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

### gttactionsets - Insert, Update, Delete

Create a file with the following content:

```
{
  "actsn": "actset1",
  "actid1": "Act1"
}
```

Execute the following command on an active SOAM to insert:

```
/vstp/gttactionsets -v POST -r /<Absolute path>/<File Name>
```

Example output:

```
/vstp/gttactionsets -v POST -r /tmp/ActSet1
{
  "data": true,
  "links": {},
}
```

```
"messages": [],  
"status": true  
}
```

Execute the following command on an active SOAM to update:

```
/vstp/gttactionsets -v PUT -r /<Absolute path>/<File Name>
```

Example output:

```
/vstp/gttactionsets -v PUT -r /tmp/actset1  
{  
  "data": true,  
  "links": {},  
  "messages": [],  
  "status": true  
}
```

Execute this command on an active SOAM to delete:

```
/vstp/gttactionsets/<Set Name> -v DELETE
```

Example output:

```
/vstp/gttactionsets/Set1 -v DELETE  
No output returned by URI: https://localhost/mmi/dsr/v3.0/vstp/gttactionsets/  
Set1? for 'DELETE' operation
```

Execute the following command to display:

```
/vstp/gttactionsets
```

Example output:

```
/vstp/gttactionsets  
{  
  "data": [  
    {  
      "actsn": "actset1",  
      "actid1": "Act1"  
    }  
  ],  
  "links": {},  
  "messages": [],  
  "status": true  
}
```

### 2.2.2.2.3 SCPVAL Alarms and Measurements

#### Alarms and Events

The following table lists the Alarms and Events specific to SCPVAL:

Alarm/ Event ID	Name
70278	GTT Action Failed

For more details related to Alarms and Events, refer to Alarms and KPIs Reference document.

#### Measurements

The following table lists the measurements specific to SCPVAL:

Measurement ID	Measurement Name
21776	VstpCdpaGttActScpvalTotal
21777	VstpCdpaGttActScpvalDiscard
21778	VstpCdpaGttActScpvalNotApplied
21779	VstpCgpaGttActScpvalTotal
21780	VstpCgpaGttActScpvalDiscard
21781	VstpCgpaGttActScpvalNotApplied

For more details related to measurements, refer to Measurement Reference document.

### 2.2.2.2.4 Dependencies

The SCPVAL action has no dependency on any other vSTP operation.

### 2.2.2.2.5 Troubleshooting

The following are the troubleshooting scenarios for SCPVAL action:

- If an incoming MSU successfully passes SCPVAL CdPA GTT action, then the `VstpCdpaGttActScpvalTotal` measurement will be pegged on a per Linkset basis.
- If validation was not applied by SCPVAL CdPA GTT action on an incoming message, `VstpCdpaGttActScpvalNotApplied` will be pegged on a per Linkset basis.
- If incoming MSU is discarded by SCPVAL CdPA GTT action, then `VstpCdpaGttActScpvalDiscard` measurement will be pegged on a per Linkset basis.
- If validation was not applied by SCPVAL CgPA GTT action on an incoming message, `VstpCgpaGttActScpvalNotApplied` will be pegged on a per Linkset basis .
- If an incoming MSU successfully passes SCPVAL CgPA GTT action , then `VstpCgpaGttActScpvalTotal` measurement will be pegged on a per Linkset basis.
- If incoming MSU is discarded by SCPVAL CgPA GTT action, then `VstpCgpaGttActScpvalDiscard` measurement will be pegged on a per Linkset basis.
- When anyone of the GTT Action (i.e. DUPLICATE, FORWARD, TCAP ERROR, SCPVAL) fails and UIMREQD is set to ON, then event 70278 GTT Action Failed will be generated. It contains error cause with SCCP and TCAP details, GTT Action set name and linkset ID.

- If any of the above statement fails as per given scenarios, then verify the configuration. In case the issue persists, contact Oracle Support.

## 2.2.3 Support for Applying GTT Action from intermediate GTT Sets

The Intermediate GTT Action Application feature enhances the flexibility of Global Title Translation (GTT) actions in your network. Previously, certain actions were restricted to the last GTT set only. With this feature, you can now apply GTT actions from intermediate GTT sets for all actions.

This functionality can be performed by the following actions:

- Action in penultimate set should not be conflicting with actions in ultimate sets. If conflicting, then only consider actions in ultimate GTTSETS.
- penultimate GTTSET of any type can be considered for performing actions not just for opcode GTTSET.
- If penultimate GTTSET of the search path also have the same Action Set ID in ultimate GTTSET then the action shall be performed in the order of the ultimate GTTSET .

### Enabling

To enable this feature, the following parameter must be enabled `enableIntermediateGttact` in the **SccpOptions** MO.

### Limitations

- Only one action is allowed in the penultimate GTT Action Set. If more then one actions are configured, then do not consider this gtt action set for performing action.

## 2.2.4 SFAPP ATIGTT Set

The SFAPP ATIGttset feature is designed to optimize the routing of ATI (Any Time Interrogation) messages in the network. When enabled, it ensures that ATI messages are directly routed to their corresponding destinations without the need for a selector lookup. This feature is useful only when the GTT Action (Global Title Translation Action) is set to SFAPP (Specific Forwarding Application).

Perform the following steps to enable SFAPP ATIGttset feature:

1. Verify if the **GTTACT HLR** address is set to "USECDPA" or "TCAPPARAM".
2. Configure `atigttssetId` parameter to the GTT Action MO (Managed Object)..

### Note

If the `atigttssetId` is not configured, the feature remains inactive, and the system operates as it did in previous releases.

### 2.2.4.1 MMI Managed Objects for ATIGTT Set

MMI information associated with ATIGtt Set feature is accessed from DSR NOAM or SOAM from **Main Menu**, and then MMI API Guide. Once the MMI API Guide is opened, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for ATIGtt Set feature:

Table 2-5 MMI Managed Objects for ATlgtt Set

Managed Object Name	Operations Supported	URI
VstpSccpOptions	GET/ PUT	mmiclient.py /vstp/sccpoptions
VstpGTTAction	GET/ POST/ PUT/ DELETE	mmiclient.py /vstp/gttactions

**VstpSccpOptions - PUT**

```
mmiclient.py /vstp/sccpoptions -PUT -r test.json
test.json -
{
  "allowedFirstSegLen": 0,
  "alwMsgDuringRsmblyErr": false,
  "cgpaGttOnResponseUdts": "On",
  "classlseq": "Disabled",
  "dfltfallback": false,
  "dfltgttmode": "Cd",
  "enableIntermediateGttact": "On",
  "isSegXUDTfeatureEnable": false,
  "mobrscpopc": "mtp",
  "mtprggt": "Off",
  "mtprgttfallback": "Mtproute",
  "reassemblyTimerDurationAnsi": 5000,
  "reassemblyTimerDurationItu": 10000,
  "segmentedMSULength": 200,
  "sendVstpGenUdtsOnOpc": "Off",
  "smsDelivery": "Off",
  "smsOrigination": "Off",
  "smsTermination": "Off",
  "tcapErrorDiscard": "Off",
  "tgtt0": "None",
  "tgtt1": "None",
  "tgttudtkey": "Mtp",
  "tgttxudtkey": "Mtp",
  "travelVelocity": 700
}
```

**Sample output**

```
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}
```

**VstpSccpOptions - GET**

```
mmiclient.py /vstp/sccpoptions
Sample output:
```

```
"inpsnail-cdpanail": "None",
```

```
"inpsnai2-cdpanai2": "None",
"inpsnai3-cdpanai3": "None",
"inpsnai4-cdpanai4": "None",
"inpsnai5-cdpanai5": "None",
"inpsprestype": "continue",
"intlunknai": "No",
"maplyrrtg_actss": "Off",
"maplyrrtg_authfailrpt": "Off",
"maplyrrtg_dactss": "Off",
"maplyrrtg_intss": "Off",
"maplyrrtg_procunstrqt": "Off",
"maplyrrtg_purgmobss": "Off",
"maplyrrtg_rdyform": "Off",
"maplyrrtg_regss": "Off",
"maplyrrtg_rstdata": "Off",
"maplyrrtg_sriloc": "Off",
"mnpcrp": "Off",
"mnpnpdbunavl": "dnotfound",
"mosmsaclen": 0,
"mosmsbpartychk": "Off",
"mosmsbpartygttset": "None",
"mosmsdefrn": "None",
"mosmsdigmat": "exact",
"mosmsfwd": "No",
"mosmsgta": "None",
"mosmsgttidig": "sccpcdpa",
"mosmsnai": "intl",
"mosmssa": "No",
"mosmsspfill": "Off",
"mosmstcapseg": "Off",
"mosmstype": "sprn",
"msisdntrunc": 0,
"msrndig": "rn",
"msrnlcn": 30,
"msrnnai": 2,
"msrnp": 1,
"mtmmsackn": "ack",
"mtmmsentyn": "None",
"mtmmsgta": "None",
"mtmmslcn": "None",
"mtmmslcn": "None",
"mtmmslcn": "rn",
"mtsmsackn": "ack",
"mtsmschksrc": "No",
"mtsmsdltr": "no",
"mtsmsdltrv": "None",
"mtsmsimsi": "mccrnda",
"mtsmsnakerr": 1,
"mtsmsnnc": "rn",
"mtsmsnp": "Off",
"mtsmslcn": "rn",
"multcc1": "None",
"multcc10": "None",
"multcc2": "None",
"multcc3": "None",
"multcc4": "None",
"multcc5": "None",
```

```

    "multcc6": "None",
    "multcc7": "None",
    "multcc8": "None",
    "multcc9": "None",
    "serverpfx": "None",
    "srfaddr": "None",
    "srfnai": 0,
    "srfnp": 0,
    "sriPrepaid": "On",
    "sridn": "tcap",
    "sridnnotfound": "gtt",
    "srismdn": "sccp",
    "srismgtrtrtg": "Off",
    "srvcrelaymapset": "None"
  }

```

**Sample Output:**

```

{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}

```

**GTT Action - GET**

```
mmiclient.py /vstp/gttactions
```

Sample Output:

```

"data": [
  {
    "act": "Sfapp",
    "actid": "gttact2",
    "atigtmodName": "None",
    "atigtsetName": "gttset1",
    "defactid": "fallback",
    "failactName": "fallback",
    "handlresp": true,
    "hlraddr": "Usecdpa",
    "ndgt": "All",
    "psigtmodName": "None",
    "scfaddr": "123",
    "uimreqd": false
  }
],
  "links": {},
  "messages": [],
  "status": true

```

**GTT Action - POST**

```
mmiclient.py /vstp/gttactions -v POST -r test.json
```

```
test.json
```

```
{
```

```

    "act": "Sfapp",
    "actid": "gttact1",
    "atigtmodName": "None",
    "atigtsetname": "gttset2",
    "defactid": "fallback",
    "failactName": "fallback",
    "handlresp": true,
    "hlraddr": "Usecdpa",
    "ndgt": "All",
    "psigtmodName": "None",
    "scfaddr": "456",
    "uimreqd": false
  }

```

**Sample Output:**

```

{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}

```

**GTT Action - PUT**

```

mmiclient.py /vstp/gttactions -v POST -r test.json
test.json

```

```

{
  "act": "Sfapp",
  "actid": "gttact1",
  "atigtmodName": "None",
  "atigtsetname": "gttset2",
  "defactid": "fallback",
  "failactName": "fallback",
  "handlresp": true,
  "hlraddr": "Usecdpa",
  "ndgt": "All",
  "psigtmodName": "None",
  "scfaddr": "456",
  "uimreqd": false
}

```

**Sample Output:**

```

{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}

```

## 2.2.5 MTP Based GTT with Screening Action

vSTP supports the MTP based GTT with screening actions feature.

This feature provides the capability of performing SCCP services on MTP-routed messages. Therefore, allows the operator to perform GTT and GTT Actions on MTP Routed MSUs, similar to GTT handling for GT Routed MSUs.

### Note

This feature supports the screening based on MTP3 layer parameters only.

### 2.2.5.1 MTP Based GTT Feature Configuration

The MTP based GTT with Screening Action is performed if the service handling results in Fall through to GTT or if **GTT Required** option is **ON** for Service Relayed MSU.

The following system-wide options are used to configure this functionality:

- **MTP Routed GTT**

The MTP Routed GTT (mtprggt) option is used for MTP Routed GTT functionality as follows:

- If option = **OFF**, then GTT shall not be performed on MTP Routed MSUs.
- If option = **Use MTP Point codes**, then GTT shall be performed on MTP Routed MSU, SCCP Portion shall be updated based on translation entry but MSU shall be sent to Original DPC (and not to translated DPC).
- If option = **Full GTT**, then GTT shall be performed on MTP Routed MSU, SCCP Portion as well as MTP Portion shall be updated based on translation results.

- **MTP Routed GTT fallback**

The MTP Routed GTT fallback (mtprgttfallbk) option is used for error handling to be performed in case of GTT failure for MTP routed MSUs. It has the following values:

- If option = **GTT failure**, then MSU will be discarded with appropriate UIM. UDTs will be sent to originator and measurements shall be pegged as done for GT routed messages.
- If option = **Fall back to MTP routing**, then MSU (with translation/modification/routing data from UDR-related service) shall be MTP routed.

The support for the following features is required for the functionality of MTP based GTT:

- **SCCP Stop Action**: provide a means for the operator to specify SCCP Stop Action in the MTP Screening Rules, to allow the MTP processing to fall through to GTT on non-discarded MSUs.
- **XLAT = NONE**: provide a means for the operator to specify GTT Translation Type = NONE.
- **GTT SET = DPC**: A new GTT set, DPC (with set type dpc) shall be supported. The provisioning and behavior of the DPC Translations shall be same as OPC Translations. However, DPC GTT set cannot be used as secondary optional set (i.e. DPC GTT set cannot be assigned to OPCS parameter in translation entry). The DPC GTT set type can be searched only when the GTT hierarchy is FLOBR specific.

### 2.2.5.1.1 MMI Managed Objects for MTP Based GTT

MMI information associated with MTP Based GTT Support is accessed from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* displays, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for vSTP MTP Based GTT feature:

**Table 2-6 vSTP MTP Based GTT Managed Objects and Supported Operations**

Managed Object Name	Supported Operations
sccpoptions	Update
mtpscreeningrules	Insert, Update, Delete

#### sccpoptions - Display

The Signaling Connection Control Part (SCCP) Options are those configuration values that govern the overall SCCP functionality . There is a single instance of this resource, which contains each of the individual options that can be retrieved and set. Because there is no collection of instances, there is no collection GET action. No new SCCP Options resource can be created, so there is no POST action, and the single instance cannot be removed, so there is no DELETE action. The single instance GET is used to retrieve the options, and PUT is used to update one or more values within the set of options. A name for this single, non- deletable instance is neither required nor expected.

Example output for display:

```
{
  "classlseq": "Disabled",
  "dfltfallback": false,
  "dfltgttmode": "Cd",
  "mtprgtt": "Fullgtt",
  "mtprgttfallback": "Gttfail",
  "tgtt0": "None",
  "tgtt1": "None",
  "tgttudtkey": "Mtp",
  "tgtxudtkey": "Mtp"
}
```

#### mtpscreeningrules - Insert, Update, Delete

Create a file with following content. File name could be anything, for example option name can be used as filename:

```
{
  "actionSccp": true,
  "area": "7",
  "nsfi": "Stop",
  "ruleName": "rule5",
}
```

```

        "scrRuleGroupName": "scr5",
        "scrRuleGroupType": "Opc",
        "signalingPointId": "3",
        "zone": "3"
    }

```

### 2.2.5.1.2 MTP Based GTT Alarms and Measurements

#### Alarms and Events

No specific Alarms and Events are generated for MTP based GTT.

#### Measurements

The following table lists the measurements specific to the MTP based GTT feature:

Measurement ID	Measurement Name
21304	VstpRxMSUMtpRoutedSccp

For more details related to measurements, refer to Measurement Reference Guide.

### 2.2.5.2 Dependencies

The MTP based GTT support for vSTP has no dependency on any other vSTP operation.

### 2.2.5.3 Troubleshooting

In case of the error scenarios, the measurements specific to MTP based GTT feature are pegged. For information related to MTP based GTT measurements, see [MTP Based GTT Alarms and Measurements](#).

## 2.2.6 GTT Action Set Test Mode

The service providers across the world undertake security monitoring projects that requires blocking of illegitimate traffic in SS7 networks. It is an umbrella of all the interconnect traffic and needs to be monitored closely before any traffic is blocked by an operator.

Unauthorized traffic can be blocked by applying certain sets of rules. On the other hand, it is equally imperative that legitimate or revenue generating traffic is not discarded by the framework.

vSTP enables the operators to block the unauthorized traffic using the GTT Action Set Test Mode functionality. This feature provides a detection mode to raise an event for the MSUs that encounters that particular GTT action and skips all the other actions included in that set. This helps to identify all the traffic that is discarded, copied, or forwarded once the rule is made active.

In case of an impact on any legitimate traffic, the rules can be changed accordingly.

### 2.2.6.1 Feature Configurations

This section provides procedures to configure the GTT Action Set Test mode functionality.

GTT Action Set Test mode is configured using the vSTP managed objects and vSTP GUI. The MMI API contains details about the URI, an example, and the parameters available for each managed object.

### 2.2.6.1.1 MMI Managed Objects for GTT Action Set Test Mode

MMI information associated with GTT Action Set Test Mode support is accessed from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* gets opened, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for GTT Action Set Test Mode support:

**Table 2-7 GTT Action Set Test Mode support Managed Objects and Supported Operations**

Managed Object Name	Supported Operations
gttactionset	Insert, Update, Delete

#### gttactionset - Insert, Update, Delete

Create a file with following content. File name could be anything, for example option name can be used as filename:

```
{
    "actid1": "set1",
    "actid2": "set2",
    "actsn": "ActSet1",
    "testMode": "On"
}
```

Execute the following command on Active SOAM to insert the action set:

```
/vstp/gttactionsets -v POST -r /<Absolute path>/<File Name>
```

Sample Output:

```
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}
```

Execute the following command to display the GTT Action:

```
/vstp/gttactionsets
```

Sample Output:

```
{
  "data": [
    {
      "actid1": "set1",
      "actid2": "set2",
      "actsn": "ActSet1",
      "testMode": "On"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

### 2.2.6.1.2 GUI Configurations

The GTT Action Set Test Mode can be configured from Active System OAM (SOAM).

On the Active System OAM (SOAM), select **VSTP > Configuration > GTT Action Sets**.

Configure the parameters on the **GTT Action Sets** page.

For more details on **GTT Action Sets** configurations, refer to *Diameter Signaling Router Virtual Signaling Transfer Point User's Guide* .

### 2.2.6.2 GTT Action Set Test Mode Alarms and Measurements

#### Alarms and Events

The following table lists the events specific to the GTT Action Set Test Mode support for vSTP:

Event ID	Event Name
70445	Vstp GTT Action Test Mode On

For more details related to measurements, refer to *Alarms and KPIs Reference* document.

#### Measurements

There are no measurements specific to the GTT Action Set Test Mode functionality.

### 2.2.6.3 Troubleshooting

When GTT Action Set Test Mode feature is On, then the GTT actions associated with the GTT Action Set is not applied. However, it generates the vSTP GTT Action Test Mode On event.

### 2.2.6.4 Dependencies

The GTT Action Set Test Mode feature has no dependency on any other vSTP operation.

The following point must be considered while using this functionality:

- This feature supports displaying the GTT action set that is triggered by the MSU. It does not detail the result of the GTT actions.

## 2.3 TCAP Level Security

This section describes the TCAP Level security features of vSTP:

### 2.3.1 TCAP Opcode Based Routing (TOBR)

TOBR provides vSTP with the ability to route messages based on its operation codes. With the TOBR feature, vSTP considers the following information contained in TCAP portion of messages for performing GTT.

#### ITU Messages

- Message/Package type
- Application context name
- Operation code
- Operation Code Tag

#### ANSI Messages

- Package type
- Operation code family
- Operation code specifier

#### ITU TCAP

- Begin
- Continue
- End
- Abort
- Unidirectional

#### ANSI TCAP

- Unidirectional
- QueryWithPermission
- QueryWithoutPermission
- Response
- ConversationWithPermission
- ConversationWithoutPermission
- Abort

TOBR works based on the following rules:

- If the message or package type is not mentioned in the above sections, vSTP treats it as an unknown message type and does not proceed with the decoding.
- vSTP attempts to decode the TCAP portion of all UDT/UDTS/Unsegmented XUDT/Unsegmented XUDTS queries coming to the SCCP layer for GTT.

- If decoding fails, the message still undergoes GTT using some default values for the TCAP data that denote their absence in the message.
- ACN is used for all supported ITU TCAP messages except ABORT. No attempt to retrieve ACN is made for Abort messages. All other supported messages may have a Dialog portion containing Dialogue Request/Unidirectional Dialogue/Dialogue Response PDU, from which the ACN is retrieved. If no Dialog portion is detected, then ACN is assumed to be NONE.
- TOBR attempts to find Operation Code Tag (Opcode Tag) in all supported ITU TCAP messages except **ABORT**. If messages have opcode tag value anything other than **Local(0x02)** or **Global(0x06)**, then it is considered as **Invalid**.
- If Opcode tag is configured as **Both** in GTA configuration, then vSTP does not filter messages specifically for **Local** or **Global** tags. However, messages with tag value other than **Local** or **Global** are discarded immediately if the **Tcap Error Discard** is **On** in SCCP Options. If this option is **Off** then MSU will fail TOBR GTT and will either get discarded or fallback to previous successful GTA entry as configured by user.
- If a valid MSU does not have an opcode component such as, Neg-Ack MSUs then user must configure Opcode Tag as **Both** and Opcode as **None** along with ACN and PkgType to successfully GTT route MSU to required DPC.
- TOBR attempts to find Operation Code (Opcode) in all supported ITU TCAP messages except **ABORT**. These messages must contain Invoke or Return Result (Last or Not Last) as the first component. If not, Opcode is assumed to be **NONE**.
- TOBR attempts to find the Operation Family and Specifier in all supported ANSI TCAP messages (except ABORT) containing an INVOKE component. For all other messages, Family and Opcode are assumed to be NONE.

### 2.3.1.1 Feature Configurations

This section provides procedures to perform the TOBR functionality.

The TOBR is configured using the vSTP managed objects and vSTP GUI. The MMI API contains details about the URI, an example, and the parameters available for each managed object.

#### 2.3.1.1.1 MMI Managed Objects for TOBR

MMI information associated with TOBR can be configured from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* gets opened, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for TOBR:

Managed Object Name	Supported Actions
Linkset	GET, PUT, POST, DELETE
GTT Sets	GET, PUT, POST, DELETE
GTT Selector	GET, PUT, POST, DELETE
GTA	GET, PUT, POST, DELETE

**linksets**

**POST**

Create a file with following content to set value for the `gttmode` parameter. File name could be anything, for example option name can be used as filename:

```
{
    "asNotification": false,
    "enableBroadcastException": false,
    "gttmode": "Fcd",
    "ituTransferRestricted": false,
    "linkTransactionsPerSecond": 10000,
    "localSignalingPointName": "VSTP1SP",
    "name": "V1V2LS",
    "remoteSignalingPointName": "VSTP2SP",
    "routingContext": 0,
    "type": "M2pa"
}
```

Execute the following command on Active SOAM to update the data:

```
/vstp/linksets -v POST -r /<Absolute path>/<File Name>
```

Execute the following command to display the content:

```
vstp/gttsets
{
    "configurationLevel": "1",
    "domain": "Itu",
    "gttSetType": "Skbcsm",
    "name": "set1"
}
```

Sample Output:

```
{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}
```

## GET

Execute the following command to display the content:

```
vstp/linksets
```

Sample Output:

```
{
    "data": [
        {
            "asNotification": false,
            "configurationLevel": "0",
            "enableBroadcastException": false,
```

```

        "gttmode": "Fcd",
        "ituTransferRestricted": false,
        "linkTransactionsPerSecond": 10000,
        "localSignalingPointName": "VSTP1SP",
        "name": "V1V2LS",
        "remoteSignalingPointName": "VSTP2SP",
        "routingContext": 0,
        "type": "M2pa"
    }
],
"links": {},
"messages": [],
"status": true
}

```

**gttsets****POST**

Create a file with following content to set value for the `gttSetType` parameter. File name could be anything, for example option name can be used as filename:

```

{
    "configurationLevel": "5",
    "domain": "Itu",
    "gttSetType": "Cdgta",
    "name": "SET1"
}

```

Execute the following command on Active SOAM to create the data:

```
/vstp/gttsets -v POST -r <filename>.json
```

**Sample Output:**

```

{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}

```

**GET**

Execute the following command to display the content:

```
/vstp/gttsets
```

**Sample Output:**

```

{
    "data": [
        {

```

```

        "configurationLevel": "5",
        "domain": "Itu",
        "gttSetType": "CdgtA",
        "name": "SET1"
    }
],
"links": {},
"messages": [],
"status": true

```

### gttselectors

#### POST

Create a file with following content. File name could be anything, for example option name can be used as filename:

```

{
    "domain": "Itui",
    "globalTitleIndicator": "TtOnly",
    "cdgttsn": "SET1",
    "name": "Sel2",
    "translationType": 155,
    "linksetName" : "V1V2LS"
}

```

Execute the following command on Active SOAM to update the data:

```
/vstp/gttselectors -v POST -r <filename>.json
```

#### Sample Output:

```

{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}

```

#### GET

Execute the following command to display the content:

```
/vstp/gttselectors
```

#### Sample Output:

```

{{
    "data": [
        {
            "cdgttsn": "SET1",
            "configurationLevel": "56",
            "domain": "Itui",
            "globalTitleIndicator": "TtOnly",

```

```

        "linksetName": "V1V2LS",
        "name": "Sel2",
        "translationType": 155
    }
],
"links": {},
"messages": [],
"status": true
}

```

### globaltitleaddresses

#### POST

Create a file with following content. File name could be anything, for example option name can be used as filename:

```

{
    "ccgt": false,
    "cgpcaction": "Dflt",
    "endAddress": "11109194412880",
    "fallback": "Sysdflt",
    "gttSetName": "SET1",
    "optsn": "SET401",
    "mapSetId": 0,
    "routingIndicator": "Ssn",
    "rspName": "MEAT3SP",
    "startAddress": "01209197411063",
    "subsystem": 18,
    "translateIndicator": "Dpcssn"
}

```

Execute the following command on Active SOAM to update the data:

```
/vstp/globaltitleaddresses -v POST -r <filename>.json
```

#### Sample Output:

```

{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}

```

#### GET

Execute the following command to display the content:

```
vstp/globaltitleaddresses
```

## Sample Output:

```

{
  "data": [
    {
      "ccgt": false,
      "cgpcaction": "Dflt",
      "configurationLevel": "54",
      "defmapvr": "V3",
      "endAddress": "11109194412880",
      "fallback": "Sysdflt",
      "gttSetName": "SET1",
      "optsn": "SET401",
      "mapSetId": 0,
      "mrnSetId": 0,
      "routingIndicator": "Ssn",
      "rspName": "MEAT3SP",
      "startAddress": "01209197411063",
      "subsystem": 18,
      "translateIndicator": "Dpcssn",
      "uniqueIdentifier": "3fe8634e-0aee-462a-9b46-cf842b739571"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}

```

### 2.3.1.1.2 GUI Configurations for TOBR

The TOBR functionality can be configured from Active System OAM (SOAM). Select **VSTP** , and then **Configuration** page.

- The following parameter on the **Link Set** page used to perform the configurations:
  - `GTT Mode`: The GTT Mode hierarchy for this link set.
- The following parameter on the **GTT Set** page are used to perform the configurations:
  - `Gtt Set Type`: The parameter defines the type of GTT Set.
- The following parameter on the **SCCP GTT Selectors** page are used to perform the configurations:
  - `CdPA GTT Set Name`: The CdPA GTT set name (`/vstp/gttsets`) associated with this GTT Selector.
  - `CgPA GTT Set Name`: The CgPA GTT set name (`/vstp/gttsets`) associated with this GTT Selector.
  - `Linkset Name`: Linkset name (`/vstp/linksets`) associated with this GTT Selector.
- The following parameter on the **Global Title Addresses** page are used to perform the configurations:
  - `Application Context Name`: Application context name. This parameter specifies the ITU TCAP acn field in the incoming MSU.
  - `TCAP Opcode`: The TCAP opcode field in the incoming MSU.
  - `Package Type`: The ANSI and ITU TCAP package type.

- TCAP Opcode: Operation Code Tag.

For more information, see GUI Configurations section in *vSTP User Guide*.

### 2.3.1.1.3 TCAP Opcode Based Routing Alarms and Measurements

#### Alarms/Events

The following table lists the event specific to the TOBR functionality for vSTP:

**Table 2-8 Alarms/Events**

Alarm/Event ID	Name
70281	GTT FLOBR Duplicate Set Name Failed
70282	GTT FLOBR Duplicate Set Name Warning
70283	GTT FLOBR Max Search Depth Failed
70284	GTT FLOBR Max Search Depth Warning
70279	GTT MBR Duplicate Set Type Failed
70280	GTT MBR Duplicate Set Type Warning
70285	MBR Decoding Failed

For more details related to alarms and events, refer to *DSR Alarms and KPI Guide*.

#### Measuremet

The following table lists the measurements specific to the TOBR functionality for vSTP:

**Table 2-9 Measurements**

Measurements ID	Measurements Name
21465	VstpMSUCgpaFlexiGTT
21466	VstpMSUCdpaFlexiGTT
21467	VstpCgpaGTTNoSelectorMatch
21468	VstpCdpaGTTNoSelectorMatch
21469	VstpCgpaGTTFail
21470	VstpCdpaGTTFail

For more details related to measurements, refer to *DSR Measurement Reference Guide*.

### 2.3.1.2 Troubleshooting

- If selector match gets failed, then the VstpCdpaGTTNoSelectorMatch or VstpCgpaGTTNoSelectorMatch measurements are incremented as per GTT mode.
- If Global Title Translation fails, then VstpCdpaGTTFail or VstpCgpaGTTFail measurements are incremented as per GTT mode.
- If GTT is successful then VstpMSUCdpaFlexiGTT or VstpMSUCgpaFlexiGTT measurements are incremented as per GTT mode.
- If any of the above statement fails as per scenarios then verify the configuration.

In case of the error scenarios, the vSTP measurements are pegged. For information related to XUDT UDT Conversion measurements, see [TCAP Opcode Based Routing Alarms and Measurements](#).

### 2.3.1.3 Dependencies

The TOBR functionality for vSTP has no dependency on any other vSTP operation.

## 2.3.2 vSTP Multi Component Message Security

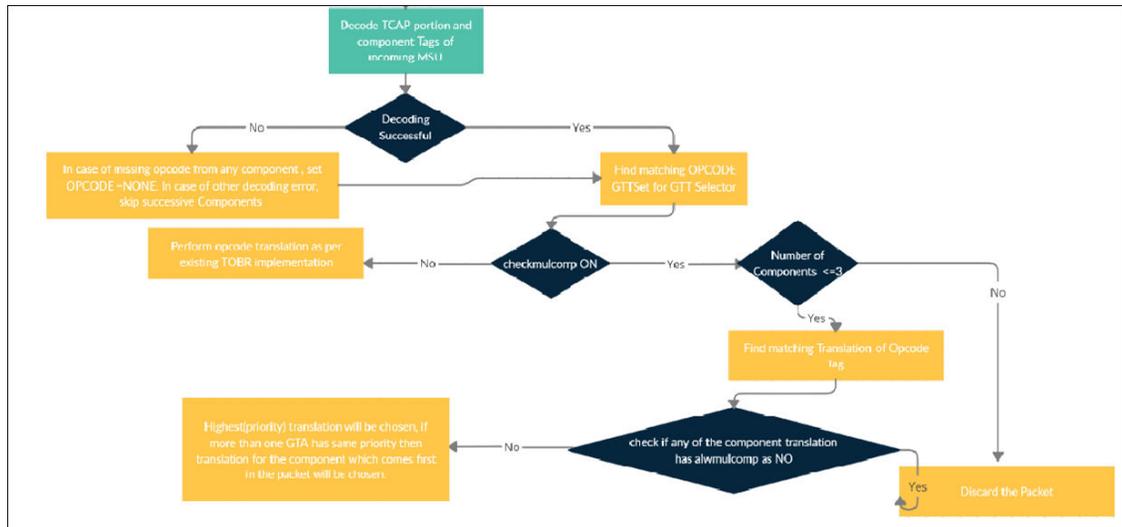
Along with TOBR (TCAP OpCode Based Routing), vSTP provides the capability to decode all the components of a TCAP message. The multi component message security functionality checks for the presence of multiple MAP operations in the message for which the MSU is processed under the GTTSet of type OPCODE. The highest priority translation is selected for the final translation of MSU.

### 2.3.2.1 Feature Description

The multi component message security feature enables vSTP to decode more than one components in a TCAP message for MAP operations and performs further translations. The basic workflow of the functionality is as follows:

- A `checkmulcomp` parameter in **VstpSccpGTTSet** of type OPCODE indicates that MSUs being processed under the respective set will check for the presence of multiple MAP operations in the same message and will do a translation lookup from more than one component in the MAP portion of the message to find matching translation.
- In addition, every translation of OPCODE GTTSet has a configurable parameter `alwmulcomp`. This parameter checks if the respective opcode in the MSU can be allowed to take part in TCAP Multicomponent Checking.
- The configurable parameter `prio` available in GTA of OPCODE GTTSet helps in selecting GTA for component in the MSU when more than one key has successful match in GTTSet. The lower value being the highest priority GTA. The MSUs belonging to Opcode **VstpSCCPGTTSet**, where TCAP MulComp feature is applicable, if the number of components are greater than three, then the packet is discarded.
- Every MAP operation in the message forms a separate key for lookup in the OPCODE GTTSet.
- All the keys are searched in the GTTSet. If there is only one successful lookup, then TOBR functionality is executed.
- If more than one key has a successful match in the GTTSet, then the translation with higher priority number is chosen for further TOBR processing.
- If more than one key matches the translation, whose priority number is equal, then the translation matching the component that occurs first in the message is chosen for further processing.
- The TCAP message component, which is selected (based on translation found and priority), is used for picking other parameters for MAP based routing or for SCPVAL & other GTT Action.
- TCAP Multiple Components supports both ANSI and ITU TCAP messages.
- In case MSU has multiple components, with any of the component having not allowed fields in its translation, then the packet is discarded.

The following figure describes the work flow:



### 2.3.2.2 Feature Configurations

This section provides procedures to perform the vSTP Multi Component Message Security functionality.

The SMulti Component Message Security is configured using the vSTP managed objects and vSTP GUI. The MMI API contains details about the URI, an example, and the parameters available for each managed object.

#### 2.3.2.2.1 MMI Managed Objects for Multi Component Message Security Support

MMI information associated with Multi Component Message Security support is accessed from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* gets opened, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for Multi Component Message Security support:

Managed Object Name	Supported Actions
GTT Sets	Insert, Update, Delete
Global Title Addresses	Insert, Update, Delete

#### gttset - Insert, Update, Delete

To configure MTP2 Interface channel “Set1”: Create a file with following content. File name could be anything, for example option name is used as filename:

```

{
    "checkmulcomp": "Yes",
    "domain": "Ansi",
    "gttSetType": "Opcode",
    "name": "Set1"
}
  
```

Execute the following command on Active SOAM to insert the data:

```
/vstp/gttsets -v POST -r <filename>.json
```

Sample Output:

```
{
  "checkmulcomp": "Yes",
  "configurationLevel": "1",
  "domain": "Ansi",
  "gttSetType": "Opcode",
  "name": "set1"
}
```

**Note**

URI for POST and GET operations:

```
/vstp/gttsets
```

URI for PUT, DELETE and GET operations:

```
/vstp/interfacemappings/{Name}
```

### globaltitleaddress - Insert, Update, Delete

Create a file with following content. File name could be anything, for example option name can be used as filename:

```
{
  "alwmulcomp": "No",
  "ccgt": false,
  "cgGtmod": false,
  "cgpcaction": "Dflt",
  "fallback": "Sysdflt",
  "family": "46",
  "gttSetName": "set1",
  "opcode": "99",
  "pkgtype": "Any",
  "prio": 1024,
  "translateIndicator": "None"
}
```

Execute the following command on Active SOAM to insert the data:

```
/vstp/globaltitleaddresses -v POST -r <filename>.json
```

Sample Output:

```
{
  "alwmulcomp": "No",
  "ccgt": false,
  "cgGtmod": false,
  "cgpcaction": "Dflt",
  "configurationLevel": "29",
  "fallback": "Sysdflt",
  "family": "46",
  "gttSetName": "set1",
  "opcode": "99",
  "pkgtype": "Any",
  "prio": 1024,
  "translateIndicator": "None",
  "uniqueIdentifier": "65b47059-0bb6-4fc1-adcd-f72145eff04f"
}
```

### 2.3.2.2.2 GUI Configurations for Multi Component Message Security Support

The Multi Component Message Security functionality can be configured from Active System OAM (SOAM). Select **VSTP**, and then **Configuration** page.

The following options are used to perform the configurations:

- GTT Sets
- Global Title Addresses

For more information, see GUI Configuration in *Oracle Communications vSTP User's Guide*.

### 2.3.2.2.3 Multi Component Message Security Alarms and Measurements

#### Alarms and Events

The following table lists the alarms or events specific to the Multi Component Message Security functionality for vSTP:

Event ID	Event Name
70441	vstpTobrDupOpcodeFoundDiscard
70443	vstpTobrMulCompTransNaDiscard
70444	vstpTobrMulCompMaxcompExceeded

For more details related to measurements, refer to *Diameter Signaling Router Alarms and KPIs Reference*.

#### Measurements

The following table lists the measurements specific to the Multi Component Message Security functionality for vSTP:

Measurement ID	Measurement Name
22172	VstpMSUTmulComp
22173	VstpMSUTmulCompGtaNa

Measurement ID	Measurement Name
22174	VstpMSUTmulCompMaxExc

For more details related to measurements, refer to *Diameter Signaling Router Measurement Reference*.

### 2.3.2.3 Troubleshooting

In case of the error scenarios, the measurements specific to Multi Component Message Security feature are pegged. For information related to Multi Component Message Security measurements, see [Multi Component Message Security Alarms and Measurements](#).

### 2.3.2.4 Dependencies

The Multi Component Message Security feature for vSTP has no dependency on any other vSTP operation.

#### Note

The Multi Component Message Security feature supports upto 3 opcode/components in MSU.

## 2.3.3 vSTP TCAP Decoding

The vSTP TCAP Decoding feature allows users to filter the messages, which have additional octets in TCAP layer.

The basic encoding rules often make it possible to encode same values in multiple ways. Therefore, an increase in the number of octets used for encoding an integer is possible. For example encoding in MAP Operation Code prepends 0x00 octets. Similarly, an increase is possible in the number of octets used for encoding an object identifier by pre-pending 0x80 octets to individual sub-identifier octets.

Both of the above scenarios are not allowed as per the security specifications. vSTP discards such messages using the the TCAP Decoding functionality.

### 2.3.3.1 Feature Description

The TCAP Decoding feature is applicable for ITU messages.

The `tcapErrorDiscard` parameter in SCCOPTIONS table is used to enable the feature.

In the scenario, when MSU is discarded, the **VstpTcapDecodeErr** event is generated. No UDTS is generated when MSUs are discarded.

### 2.3.3.2 TCAP Decoding Error Scenario

The TCAP Decoding functionality discards the MSUs with the following specifications:

- A valid opcode tag is present and opcode length is not equal to one
- ACN and ACN object identifier tags are present but ACN length is either zero or greater than 7

- Invalid length in Transaction portion
- Invalid length in dialog portion
- Invalid length in component portion
- TCAP portion is beyond the last byte of the message as specified by SCCP

### 2.3.3.3 Feature Configuration

This section provides procedures to configure the TCAP Decoding feature. The feature requires the vSTP managed objects. The MMI API contains details about the URI, an example, and the parameters available for each managed object.

#### 2.3.3.3.1 MMI Managed Objects for TCAP Decoding

MMI information associated with TCAP Decoding support is accessed from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* gets opened, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for TCAP Decoding support:

**Table 2-10 TCAP Decoding support Managed Objects and Supported Operations**

Managed Object Name	Supported Operations
sccoptions	Display, Update

#### sccoptions

For this feature, the `tcapErrorDiscard` parameter is added to the `sccoptions` MO.

The allowed values for this parameter with their interpretation are:

- **OFF:** TCAP Decoding feature is OFF. This is the default value.
- **ON:** TCAP Decoding feature is ON.

The example output for Display of `sccoptions` MO:

```
{
    "allowedFirstSegLen": 0,
    "alwMsgDuringRsmblErr": false,
    "classlseq": "Disabled",
    "dfltfallback": false,
    "dfltgttmode": "Cd",
    "isSegXUDTfeatureEnable": false,
    "mtprggt": "Off",
    "mtprggtfallback": "Mtproute",
    "reassemblyTimerDurationAnsi": 5000,
    "reassemblyTimerDurationItu": 10000,
    "segmentedMSULength": 200,
    "smsDelivery": "Off",
    "smsOrigination": "Off",
    "smsTermination": "Off",
    "tcapErrorDiscard": "On",
    "tgtt0": "None",
}
```

```

    "tgttl": "None",
    "tgttudtkey": "Mtp",
    "tgtxudtkey": "Mtp",
    "travelVelocity": 700
}

```

### 2.3.3.3.2 Alarms and Measurements

#### Alarms and Events

The following table lists the Alarms and Events specific to the TCAP Decoding support for vSTP:

**Table 2-11 TCAP Decoding Alarms**

Alarm/ Event ID	Name
70457	VstpTcapDecodeErr

For more details related to Alarms and Events, refer to *Alarms and KPIs Reference* document.

#### Measurements

There are no measurements specific to the TCAP Decoding functionality. However, the existing vSTP measurements are pegged during the TCAP decoding operations.

### 2.3.3.4 Troubleshooting

In case of the error scenarios, the vSTP measurements are pegged.

### 2.3.3.5 Dependencies

The TCAP Decoding feature for vSTP has no dependency on any other vSTP operation.

## 2.4 MAP Level Security

This section describes the MAP Level security features of vSTP:

### 2.4.1 Map Based Routing (MBR)

MBR provides vSTP with the ability to route messages based on its MAP components. This can be done by using either IMSI or MSISDN GTT set types, which are linked by OPCODE set type.

MBR works based on the following rules:

- TCAP package types BEGIN, CONTINUE, and END are supported for MAP based routing, so OPTSN with one of the MAP GTT set types are allowed to be provisioned for TOBR GTA entries that have "pkgtype" as BGN, CNT, or END.
- When an MSU is processed by the TOBR GTT translation with the OPTSN as one of these new set types, vSTP decodes the TCAP part and extracts the required TCAP parameter from the MSU. The digits in this parameter are used as the key to search for the translation in the GTT set.
- If Dialogue Portion is present in the message, pick the last byte of the ACN.

**Note**

MBR does not validate if the MAP operation is supported with the ACN in the message; it is only decoding the last byte of the ACN to determine the MAP version.

- If Dialogue Portion is not present, the MAP version provisioned with the Opcode translation is used as the MAP version.

## 2.4.1.1 Feature Configurations

This section provides procedures to perform the MBR functionality.

The MBR is configured using the vSTP managed objects and vSTP GUI. The MMI API contains details about the URI, an example, and the parameters available for each managed object.

### 2.4.1.1.1 MMI Managed Objects for MBR

MMI information associated with MBR can be configured from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* gets opened, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for MBR:

Managed Object Name	Supported Actions
Linkset	GET, PUT, POST, DELETE
GTT Sets	GET, PUT, POST, DELETE
GTT Selector	GET, PUT, POST, DELETE
GTA	GET, PUT, POST, DELETE

#### linksets

#### POST

Create a file with following content to set value for the `gttmode` parameter. File name could be anything, for example option name can be used as filename:

```
{
    "asNotification": false,
    "enableBroadcastException": false,
    "gttmode": "Fcd",
    "ituTransferRestricted": false,
    "linkTransactionsPerSecond": 10000,
    "localSignalingPointName": "VSTP1SP",
    "name": "V1V2LS",
    "remoteSignalingPointName": "VSTP2SP",
    "routingContext": 0,
    "type": "M2pa"
}
```

Execute the following command on Active SOAM to update the data:

```
/vstp/linksets -v POST -r /<Absolute path>/<File Name>
```

Execute the following command to display the content:

```
vstp/gttsets
{
    "configurationLevel": "1",
    "domain": "Itu",
    "gttSetType": "Skbcsm",
    "name": "set1"
}
```

Sample Output:

```
{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}
```

## GET

Execute the following command to display the content:

```
vstp/linksets
```

Sample Output:

```
{
    "data": [
        {
            "asNotification": false,
            "configurationLevel": "0",
            "enableBroadcastException": false,
            "gttmode": "Fcd",
            "ituTransferRestricted": false,
            "linkTransactionsPerSecond": 10000,
            "localSignalingPointName": "VSTP1SP",
            "name": "V1V2LS",
            "remoteSignalingPointName": "VSTP2SP",
            "routingContext": 0,
            "type": "M2pa"
        }
    ],
    "links": {},
    "messages": [],
    "status": true
}
```

## gttsets

### POST

Create a file with following content to set value for the `gttSetType` parameter. File name could be anything, for example option name can be used as filename:

```
{
    "configurationLevel": "5",
    "domain": "Itu",
    "gttSetType": "Cdgta",
    "name": "SET1"
}
```

Execute the following command on Active SOAM to create the data:

```
/vstp/gttsets -v POST -r <filename>.json
```

Sample Output:

```
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}
```

### GET

Execute the following command to display the content:

```
/vstp/gttsets
```

Sample Output:

```
{
  "data": [
    {
      "configurationLevel": "5",
      "domain": "Itu",
      "gttSetType": "Cdgta",
      "name": "SET1"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

## gttselectors

### POST

Create a file with following content. File name could be anything, for example option name can be used as filename:

```
{
    "domain": "Itui",
    "globalTitleIndicator": "TtOnly",
    "cdgttsn": "SET1",
    "name": "Sel2",
    "translationType": 155,
    "linksetName" : "V1V2LS"
}
```

Execute the following command on Active SOAM to update the data:

```
/vstp/gttselectors -v POST -r <filename>.json
```

Sample Output:

```
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}
```

## GET

Execute the following command to display the content:

```
/vstp/gttselectors
```

Sample Output:

```
{{
  "data": [
    {
      "cdgttsn": "SET1",
      "configurationLevel": "56",
      "domain": "Itui",
      "globalTitleIndicator": "TtOnly",
      "linksetName": "V1V2LS",
      "name": "Sel2",
      "translationType": 155
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}}
```

## globaltitleaddresses

### POST

Create a file with following content. File name could be anything, for example option name can be used as filename:

```
{
    "ccgt": false,
    "cgpcaction": "Dflt",
    "endAddress": "11109194412880",
    "fallback": "Sysdflt",
    "gttSetName": "SET1",
    "optsn": "SET401",
    "mapSetId": 0,
    "routingIndicator": "Ssn",
    "rspName": "MEAT3SP",
    "startAddress": "01209197411063",
    "subsystem": 18,
    "translateIndicator": "Dpcssn"
}
```

Execute the following command on Active SOAM to update the data:

```
/vstp/globaltitleaddresses -v POST -r <filename>.json
```

Sample Output:

```
{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}
```

## GET

Execute the following command to display the content:

```
vstp/globaltitleaddresses
```

Sample Output:

```
{
    "data": [
        {
            "ccgt": false,
            "cgpcaction": "Dflt",
            "configurationLevel": "54",
            "defmapvr": "V3",
            "endAddress": "11109194412880",
            "fallback": "Sysdflt",
            "gttSetName": "SET1",
            "optsn": "SET401",
            "mapSetId": 0,
            "mrnSetId": 0,
            "routingIndicator": "Ssn",
            "rspName": "MEAT3SP",

```

```

        "startAddress": "01209197411063",
        "subsystem": 18,
        "translateIndicator": "Dpcssn",
        "uniqueIdentifier": "3fe8634e-0aee-462a-9b46-cf842b739571"
    }
],
"links": {},
"messages": [],
"status": true
}

```

### 2.4.1.1.2 GUI Configurations for MBR

The MBR functionality can be configured from Active System OAM (SOAM). Select **VSTP**, and then **Configuration** page.

- The following parameter on the **Link Set** page used to perform the configurations:
  - `GTT Mode`: The GTT Mode hierarchy for this link set.
- The following parameter on the **GTT Set** page are used to perform the configurations:
  - `Gtt Set Type`: The parameter defines the type of GTT Set.
- The following parameter on the **SCCP GTT Selectors** page are used to perform the configurations:
  - `CdPA GTT Set Name`: The CdPA GTT set name (`/vstp/gttsets`) associated with this GTT Selector.
  - `CgPA GTT Set Name`: The CgPA GTT set name (`/vstp/gttsets`) associated with this GTT Selector.
  - `Linkset Name`: Linkset name (`/vstp/linksets`) associated with this GTT Selector.
- The following parameter on the **Global Title Addresses** page are used to perform the configurations:
  - `Application Context Name`: Application context name. This parameter specifies the ITU TCAP acn field in the incoming MSU.
  - `TCAP Opcode`: The TCAP opcode field in the incoming MSU.
  - `Package Type`: The ANSI and ITU TCAP package type.
  - `TCAP Opcode`: Operation Code Tag.

For more information, see GUI Configurations section in *vSTP User Guide*.

### 2.4.1.1.3 MBR Alarms and Measurements

#### Alarms/Events

The following table lists the event specific to the MBR functionality for vSTP:

**Table 2-12 Alarms/Events**

Alarm/Event ID	Name
70281	GTT FLOBR Duplicate Set Name Failed
70282	GTT FLOBR Duplicate Set Name Warning
70283	GTT FLOBR Max Search Depth Failed

**Table 2-12 (Cont.) Alarms/Events**

Alarm/Event ID	Name
70284	GTT FLOBR Max Search Depth Warning
70279	GTT MBR Duplicate Set Type Failed
70280	GTT MBR Duplicate Set Type Warning
70285	MBR Decoding Failed

For more details related to alarms and events, refer to *DSR Alarms and KPI Guide*.

### Measurement

The following table lists the measurements specific to the MBR functionality for vSTP:

**Table 2-13 Measurements**

Measurements ID	Measurements Name
21465	VstpMSUCgpaFlexiGTT
21466	VstpMSUCdpaFlexiGTT
21467	VstpCgpaGTTNoSelectorMatch
21468	VstpCdpaGTTNoSelectorMatch
21469	VstpCgpaGTTFail
21470	VstpCdpaGTTFail

For more details related to measurements, refer to *DSR Measurement Reference Guide*.

## 2.4.1.2 Troubleshooting

- If selector match gets failed, then the `VstpCdpaGTTNoSelectorMatch` or `VstpCgpaGTTNoSelectorMatch` measurements are incremented as per GTT mode.
- If Global Title Translation fails, then `VstpCdpaGTTFail` or `VstpCgpaGTTFail` measurements are incremented as per GTT mode.
- If GTT is successful then `VstpMSUCdpaFlexiGTT` or `VstpMSUCgpaFlexiGTT` measurements are incremented as per GTT mode.
- If any of the above statement fails as per scenarios then verify the configuration.

In case of the error scenarios, the vSTP measurements are pegged. For information related to XUDT UDT Conversion measurements, see [TCAP Opcode Based Routing Alarms and Measurements](#).

## 2.4.1.3 Dependencies

The MBR functionality for vSTP has no dependency on any other vSTP operation.

## 2.4.2 Stateful Application Feature

SS7 Firewall - Stateful Applications (SFAPP) allow vSTP to validate the messages coming in for a subscriber by validating them against the Visitor Location Register (VLR). The last seen details of the subscriber can be fetched from the Home Location Register (HLR). Once the HLR provides a validity of the new VLR, vSTP allows the message into the network. If the

message is not validated, it is handled as per configuration (either silent discard, fallback, or respond with error). Similarly for SGSN, the same functionality is provided.

For detailed information about this feature, refer to *vSTP SS7 Security User's Guide*.

## 2.4.2.1 Supported MAP Operations

The following MAP Operations are supported by the Stateful Applications feature.

**Table 2-14 Supported MAP Operations**

MAP Operation	OpCode	Application Context (AC)	AC Code
sendParameters	9	infoRetrieval /v1	14
Registers	10	networkFunctionalSs	18
Erases	11	networkFunctionalSs	18
Activates	12	networkFunctionalSs	18
deactivates	13	networkFunctionalSs	18
interrogates	14	networkFunctionalSs	18
authenticationFailureReport	15	authenticationFailureReport /v3	39
registerPassword	17	networkFunctionalSs	18
processUnstructuredSS-Data	19	networkFunctionalSs /v1	18
mo-forwardSM	46	shortMsgMO-Relay	21
noteSubscriberPresent	48	mwdMngt/v1	24
beginSubscriberActivity	54	networkFunctionalSs /V1	18
restoreData	57	networkLocUp/v2	1
processUnstructuredSS-Request	59	networkUnstructuredSs v2	19
readyForSM	66	mwdMngt /v2/v3	24
purgeMS	67	istAlerting /v2/v3	4
purgeMS	67	msPurging /v3	27
ss-Invocation-Notification	72	ss-InvocationNotification	36
statusReport	74	reporting	7
istAlert	87	istAlerting /v3	4
NoteMM-Event	89	mm-EventReporting	42
updateLocation	2	networkLocUp	1
updateGprsLocation	23	gprsLocationUpdate/v3	32
sendAuthenticationInfo	56	infoRetrieval /v2/v3	14

## 2.4.2.2 VLR Validation

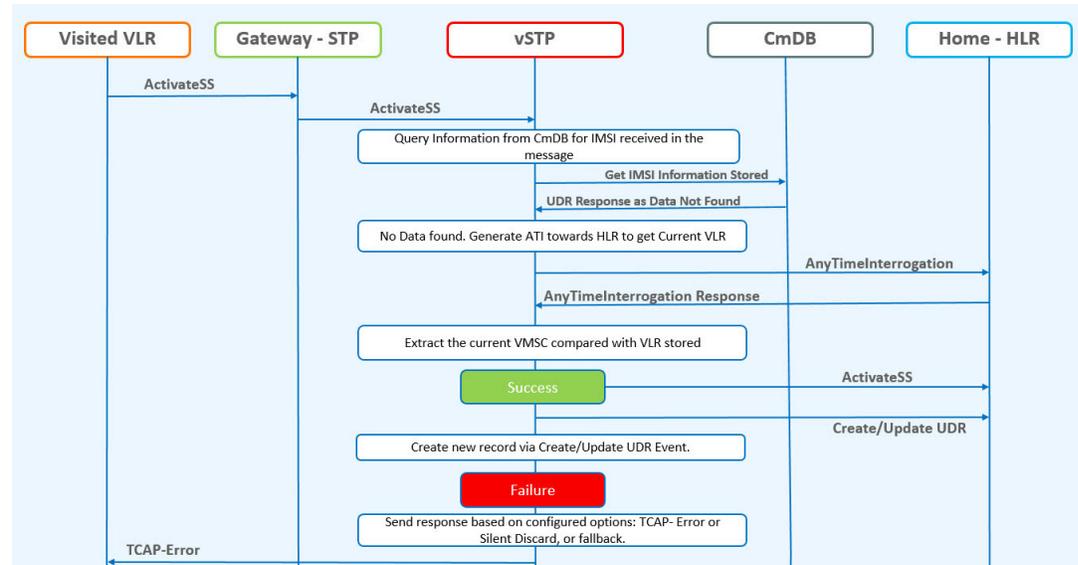
The VLR Validation uses the information stored in the HLR about the current VLR to validate the VLR from which the message is received.

The vSTP call flow for VLR Validation first lookup the Common DB that is UDR for IMSI. If the record is available, then the ATI is not sent to HLR and the UDR information is used further. But, in case the record is not available in UDR, then ATI is sent to HLR. Both the scenarios of vSTP call flow for VLR Validation are described below:

- **vSTP Call Flow - When no record is found in Common DB**

The following figure shows the vSTP call flow when IMSI record is not available in Common DB:

**Figure 2-7 VLR Validation - vSTP Call Flow when No IMSI Record Found in UDR**



1. The incoming message is decoded.
  - a. An Error is generated in case of decode failure.
2. The message information is stored in the local database.
3. The request to get the IMSI information is generated towards UDR.
4. If the IMSI record is not found in UDR, the Any Time Interrogation (ATI) request is generated towards the HLR.
  - a. The ATI Request is coded so that Acknowledgment is received on the same MP, as the DB is local.
5. For a successful response from the HLR:
  - a. The ATI Response is decoded to get the current VLR address.
  - b. The current VLR address is compared with the CgPA stored in the local database for the subscriber.
  - c. On a successful match, forward the message stored in the local DB to HLR. The UDR is updated with the new IMSI record by sending Update or Create Event to UDR corresponding to IMSI of query message.
  - d. In case of failure,
    - i. Send the configured response.
    - ii. Increment the measurement for failed messages.

The ATI sent to HLR must be formatted as follows:

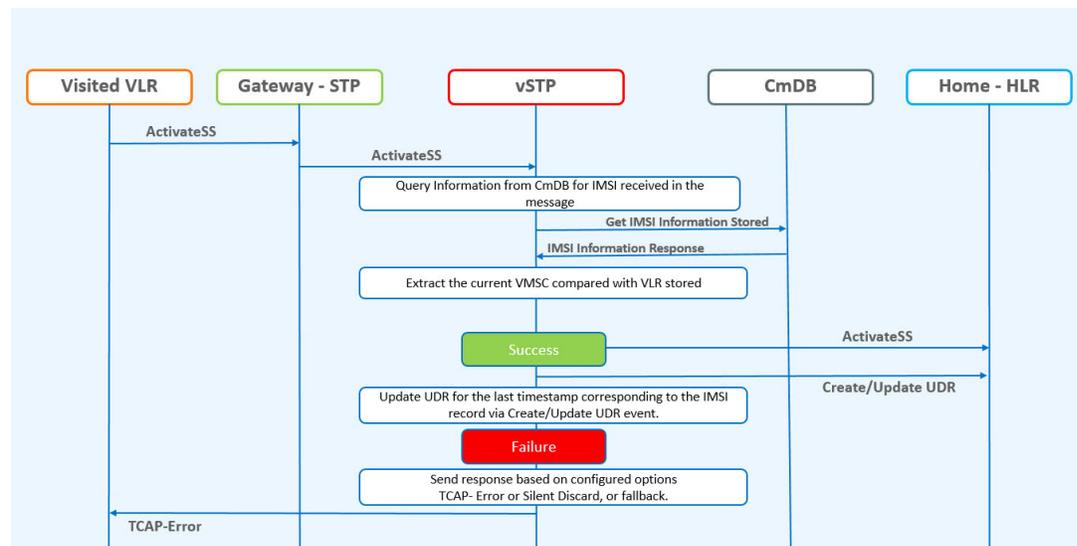
1. MTP OPC=vSTP SID, MTP DPC = HLR PC
2. SCCP CGPA (RI = SSN, PC = Local Signaling Point SID, SSN = <SSFAPP SSN>, SCCP CDPA (received message CDPA)

3. TCAP BEGIN with valid MAP dialogue portion (as per MAP specification)
4. TCAP DTID = unique OTID generated for each ATI (The DTID is not reused within 4.5 seconds)
5. ATI details: IMSI = IMSI/MSISDN received in received message, and other mandatory parameters

The Local Signaling Point validates the ATI\_ACK received from the HLR. A valid ATI\_ACK message is defined as:

1. It is a well formatted ANSI or ITU SCCP UDT, non-segmented XUDT message, with a valid TCAP END message, with valid dialogue portion, and single component in the component portion as return result with operation code = ATI\_ACK.
  2. Value of DTID received in TCAP END matches with one of the ongoing transactions.
  3. Component type is a return result and contains ATI\_ACK.
  4. VMSC digits are received in ATI\_ACK.
- **vSTP Call Flow - When IMSI record is found in Common DB**  
The following figure shows the vSTP call flow when the IMSI record is available in Common DB:

**Figure 2-8 VLR Validation - vSTP Call Flow when IMSI Record is Found in UDR**



1. The incoming message is decoded.
  - a. An Error is generated in case of decode failure.
2. The message information is stored in the local database.
3. The request to get the IMSI information is generated towards UDR.
4. The current VLR address from UDR response is compared with the CgPA stored in the local database for the subscriber.
5. On a successful Match,
  - a. Forward the Message stored in the local DB to HLR.
  - b. The UDR is updated with the latest timestamp for this IMSI record by sending Update event .

6. In case of failure,
  - a. Send the configured response.
  - b. Increment the measurement for failed messages.

### 2.4.2.3 Velocity Check

The Velocity Check use case uses the information stored in HLR about the current VLR and the age of location parameter to identify if the new VLR is reachable from the current VLR stored in HLR.

This use case is dependent on the validity of the information stored in the VLR and the T3212 timer (periodic update location timer). This timer governs the rate at which the mobile subscriber autonomously updates their location. In case the time distance between two networks is less than the value of T3212 timer configured for the network, this use case test would provide false positives since the location age information would not have been properly updated in the VLR.

The assumption for successful execution of this use case are:

- The First location update can be identified using the IMSI only in the address.
- The Age of Location provided by HLR is accurate.
- The quantum of information (Age of Location) will not be less than the time to get travel.

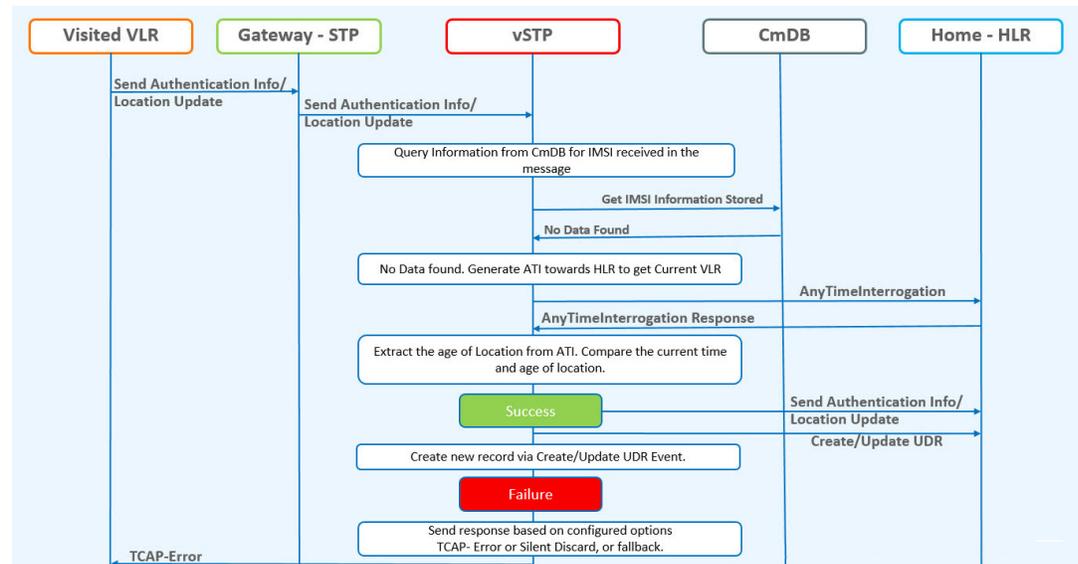
The vSTP call flow for Velocity Check, can be completed in a reasonable amount of time for Location Update to succeed.

The vSTP call flow for Velocity Check, first lookup the Common DB that is UDR, for IMSI. If the record is available in UDR, then the ATI is not sent to HLR and the UDR information is used further. But, in case the record is not available in UDR, then ATI is sent to HLR.

In both the scenarios, the UDR is updated in case of successful validation. If record is not found in UDR and validation is successful through ATI, then a new record is created in UDR with that IMSI. In case the IMSI record is available in UDR and validation is successful, then the last updated time of the record is updated in UDR.

Both the scenarios of vSTP call flow for VLR Validation are described below:

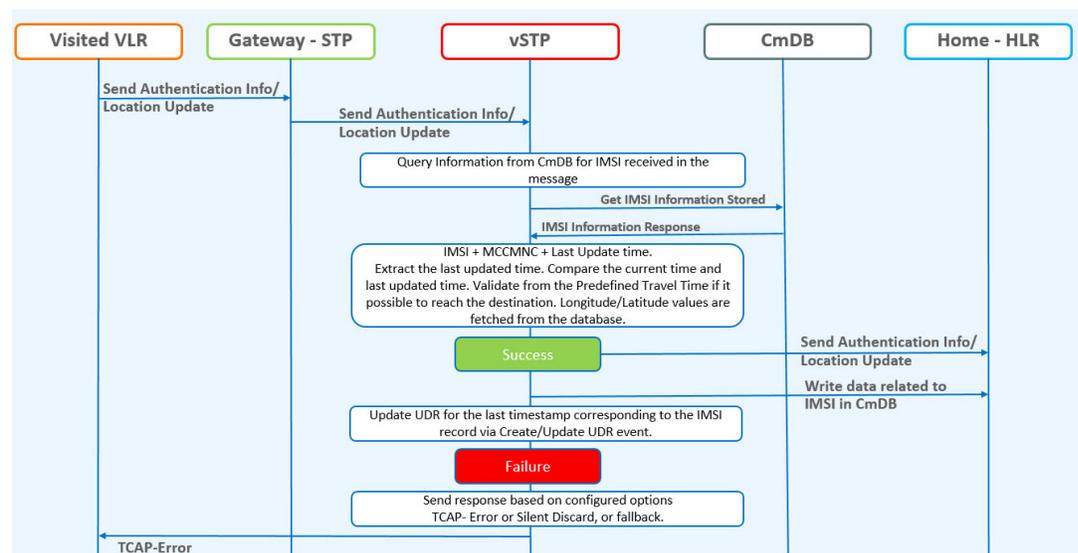
- **vSTP Call Flow for Velocity Check - When no IMSI record is found in Common DB**  
The following figure shows the vSTP call flow when there is no record available in Common DB:

**Figure 2-9 Velocity Check - vSTP Call Flow when IMSI Record is not Found in UDR**

1. A local database on vSTP will be configured to identify the network locations (using country codes for VLR addresses) and the shortest amount of time it may take to travel between them.
2. The incoming message will be decoded:
  - a. An error will be generated in case of decode failure.
  - b. A Measurement will be pegged for the decode failure with OpCode and CgPA.
3. The message information will be stored in the local database.
4. The request to get the IMSI information is generated towards UDR.
5. If the IMSI record is not found in UDR, the ATI request will be generated toward the HLR identified in the CdPA of the incoming message. The ATI request will be coded so that it is received on the same MP, as DB is local.
6. In case the HLR sends a failure in the ATI response:
  - a. A measurement will be pegged to identify HLR error corresponding message from CgPA (VLR).
7. For a success response if the old VLR is from ATI ACK, and the subscriber state in ATI ACK is *imsiDetached/ PsDetached/msPurged/notregistered/ notProvidedfromVLRnorMME/PSactive notreachable*, skip the velocity check. Forward message to HLR and update the UDR with relevant data with VLR number or SGSN, last updated Network, last updated time. Else continue with following steps.
8. Extract the Age of Location from the ATI Response message and the VMSC address in the HLR.
9. A record is created in UDR for the IMSI after successful validation.
10. The VLR GT or SGSN GT from which the SAI/LU was received must match the VLR GT or SGSN GT in the ATI response. In case the VLR GT or SGSN GT addresses do not match:
  - a. Calculate the time difference between the current time and the Age of Location.

- b. Verify the Age of Location is less than the travel time configured in the local database.
  - c. Calculate the distance between two country codes using Haversine Formula. Longitude/Latitude values are retrieved from database for corresponding entries.
  - d. In case the time value is not within limits:
    - i. The validation gets failed.
    - ii. A measurement is pegged.
    - iii. Response is generated based on the configured option.
11. If validation is successful, forward message to HLR and update the UDR with relevant data with VLR number or SGSN, last updated Network, last updated time.
- **vSTP Call Flow - When IMSI record is found in Common DB**  
The following figure shows the vSTP call flow when the IMSI record is available in Common DB:

**Figure 2-10 Velocity Check - vSTP Call Flow when IMSI Record Found in UDR**



1. A local database on vSTP will be configured to identify the network locations (using country codes for VLR addresses) and the shortest amount of time it may take to travel between them.
2. The incoming message will be decoded:
  - a. An Error will be generated in case of decode failure.
  - b. A Measurement will be pegged for the decode failure with OpCode and CgPA.
3. The message information will be stored in the local database.
4. In case VLR address do not match, then check for the neighboring countries:
  - a. Lookup into SfappCCMCCMap table and for corresponding country codes and retrieve the MCC.
  - b. The exception or neighboring list table is checked for with old MCC, if it is available in neighboring list then do nothing. Else, the velocity check will be performed.
5. The exception or neighboring list table is checked for with old MCC, if it is available in neighboring list then the process ends. Else, the following step will be performed.

6. The distance between two country codes is calculated using Haversine Formula. Use Longitude/Latitude values from database.
7. The Time ( = Distance/Velocity) shall be calculated and used to authenticate the subscriber location.
8. Validate the last time distance check based on last location with the VLR address received in the incoming message.
9. In case the VLR addresses do not match:
  - a. Calculate the distance between two country codes using Haversine Formula using longitude and latitude values (from SfappLongLat MO).
  - b. Calculate the time using distance calculated from above point and travel\_velocity value from vSTPScppOptions MO.
  - c. Verify that time calculated in step b is less than the (current time - last update time from UDR).
10. If validation is successful, forward message to HLR and update the UDR with relevant data with VLR or SGSN, last updated Network, last update time.
11. In case the validation fails:
  - a. A measurement will be pegged.
  - b. Response will be generated based on the configured option.

#### **Note**

- The T3212 timer is responsible for periodic location update for subscribers. If the timer is set to a value greater than the shortest travel time value between two network locations, then the location validation in the use case fails.
- Results of the use cases depends on the pre-configured values of SfappCCMCCMAP, SfappCountryLongLat, SfappCountryCodes, and SfappNeighboringCountries entries.
- SGSNLastTS and vlrLastTS are added to the UDR. These fields will be used when a subscriber transitions to a 2G or 3G network.
- The values of SGSNLastTS and vlrLastTS fields will be updated following the successful validation of messages received from the SGSN or VLR.
- The LastTS field will continue to be used for validations when a subscriber moves from 4G to 3G or 2G.

### 2.4.2.3.1 Velocity Check Flow Charts

The following flow charts provide an overview of the Velocity Check feature for Stateful Applications:

Figure 2-11 SFAPP Process Message

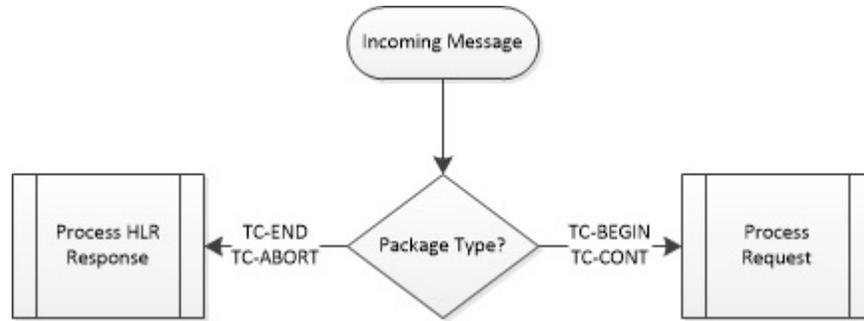


Figure 2-12 Perform VLR Validation

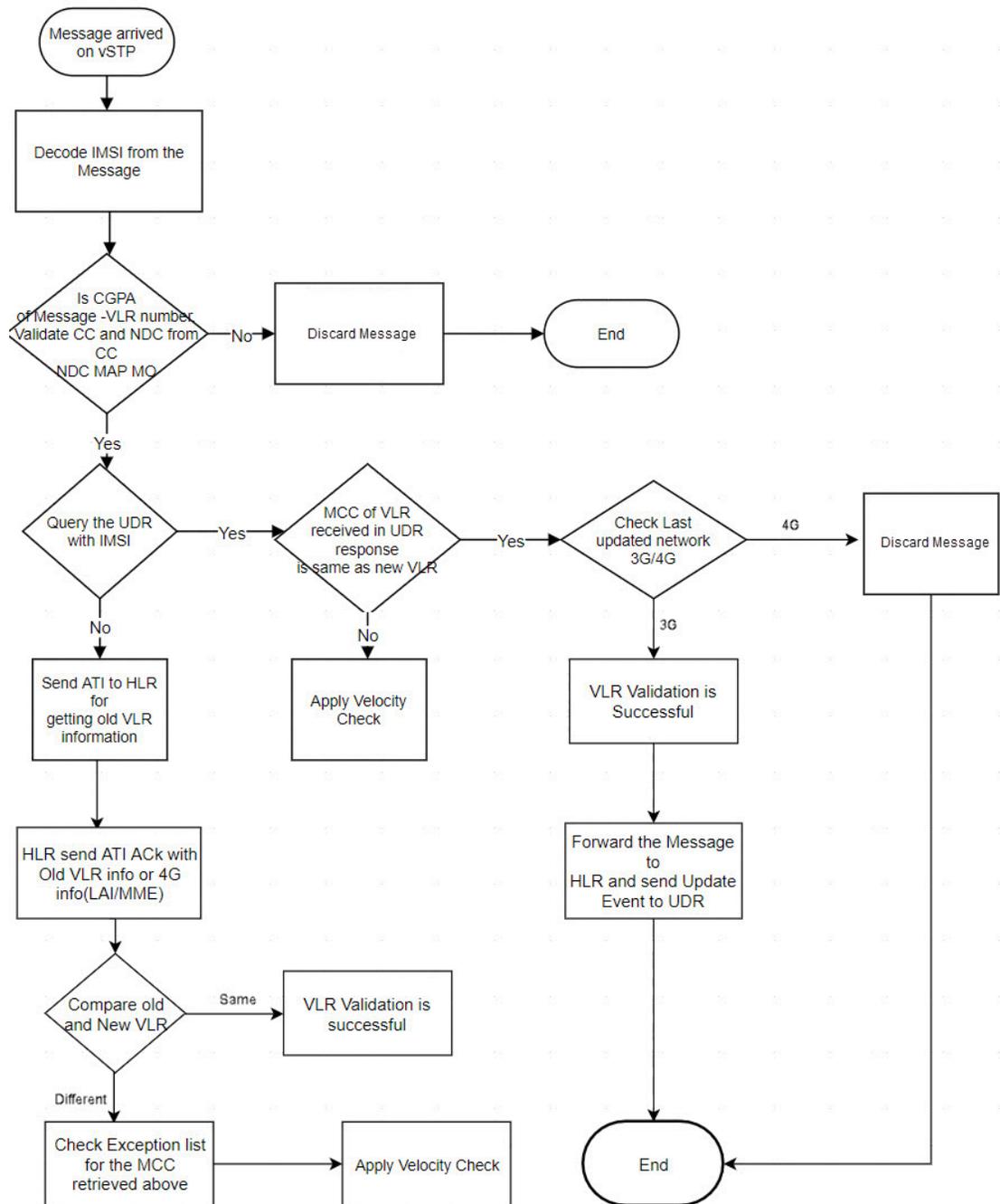
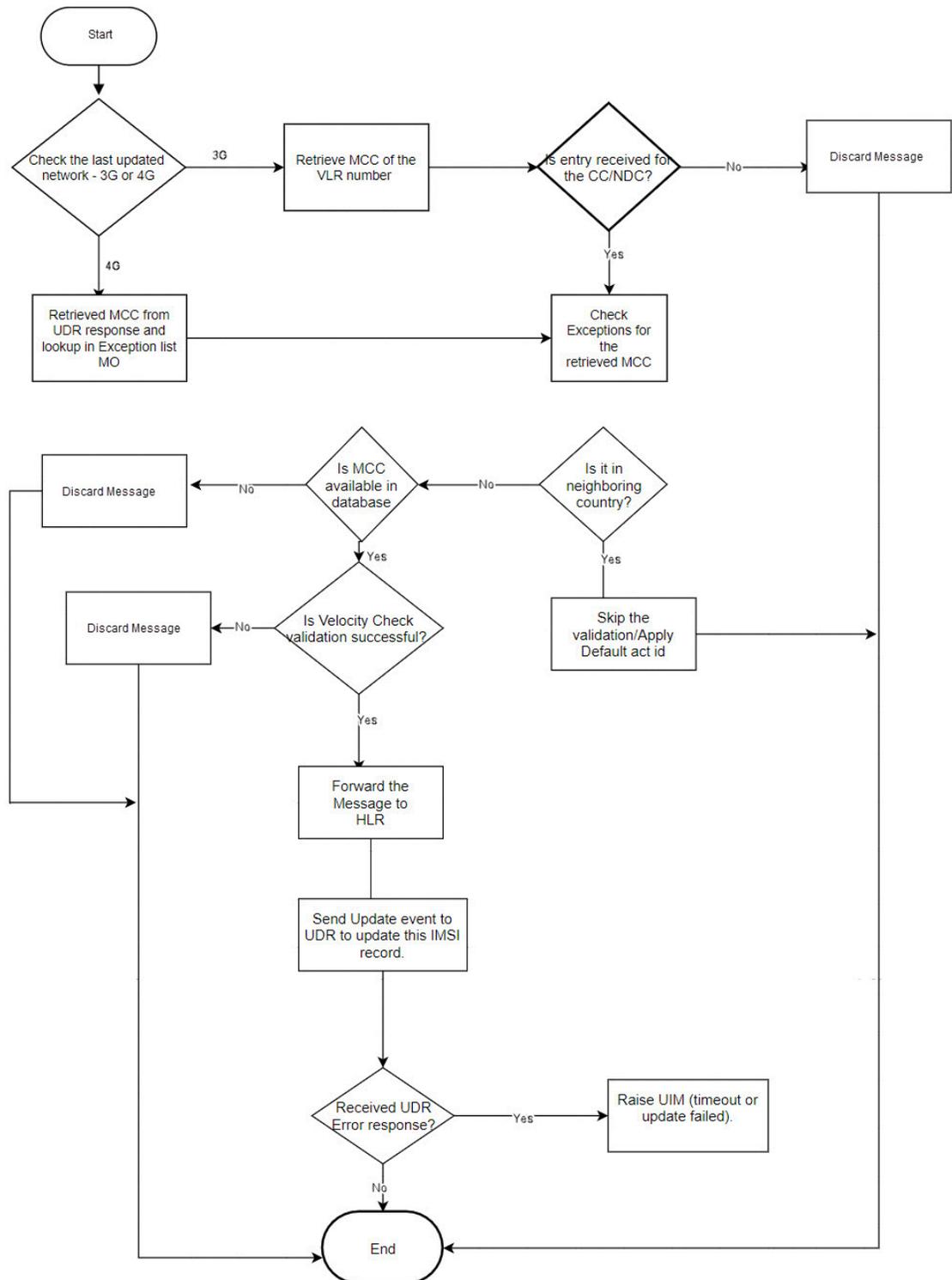


Figure 2-13 Perform Velocity Check



### 2.4.2.3.2 Verifying Aging of UDR Look Up Record

Before performing the functionality, the CommonDbAging parameter in Sfapp Option MO must be turned **On**.

User must configure home MCC parameter in sfapp option MO.

Perform the following steps if, home MCC matches with old VLR MCC in UDR:

1. Calculate minimum travel time (minTrav) between homeMCC and old VLR MCC using Long or Lat.
2. If current time VLR last timestamp on UDR is greater than minTrav using long or lat between home MCC and old VLR in UDR that means record is stale. Hence, get latest update from HLR through ATI.
3. Update UDR with latest old VLR and timestamp for stale record.
4. If Home MCC is not configured, then get updated info from HLR through ATI and update UDR.
5. vSTP has latest info either from UDR or HLRATI then continue performing validation check or velocity check.

When home MCC matches with old VLR MCC in UDR

- Calculate min travel time (minTrav) between newVLR and old VLR using long or lat.
- If current time VLR last timestamp on UDR is greater than minTrav using long or lat between home MCC and old VLR in UDR that means record is stale. Hence, get latest update from HLR through ATI.
- Update UDR with latest old VLR and timestamp for stale record.

## 2.4.2.4 Stateful Security Dynamic Learning

The Stateful Security Dynamic Learning feature enables vSTP to create and use a whitelist that is created as part of learning from the validation attempts defined in [VLR Validation](#). This feature is independent of the category of messages but it provides protection against all the messages coming from VLRs that fail the validation and are not part of the created whitelists. A grey list and black list is also created for the VLRs that fail the validation.

Learning is controlled by these modes using a mode parameter in the SFAPPOPTS table:

- **Learn Mode:** This mode allows to learn about new VLRs and no validations are performed. The newly learnt VLRs are considered as whitelisted.

### Note

The user can configure the amount of time for which the vSTP operates in Learn mode. The time is configured in SFAPPOPTS table. Hence, the switch from Learn to Test mode can happen either by configuring the timer, or manual switch.

- **Test Mode:** This mode validates all the learned VLRs. In case the VLR is not validated, the learnt VLRs remains greylisted and and measurements and alarms are generated.
- **Active Mode :** This mode allows validations based on the learned white lists in the system. New VLRs are also learned in this mode.

The status of dynamically learnt VLRs are changed to whitelist or blacklist as per their status.

- **OFF Mode:** When none of the above modes is active, it is considered as OFF mode. In this mode, if VLR entry is in whitelist, then no validation is performed for that VLR.

By default, the OFF mode remains enabled. That means the SFAPP dynamic learning functionality is disabled.

### Note

- In any mode, if VLR is in whitelist table, then it is considered as whitelisted, and the message is forwarded to HLR.
- If user has changed the mode from Learn/Test/Active mode to OFF mode, then the user has to wait for at least 10 mins before switching the mode again to Active/Learn/Test mode.

## 2.4.2.4.1 Call Flow in Learn Mode

This mode does not validate any VLRs and consider all the VLRs interacting with the network as valid. All the New VLRs that are used during this modes shall be added to the dynamic tables.

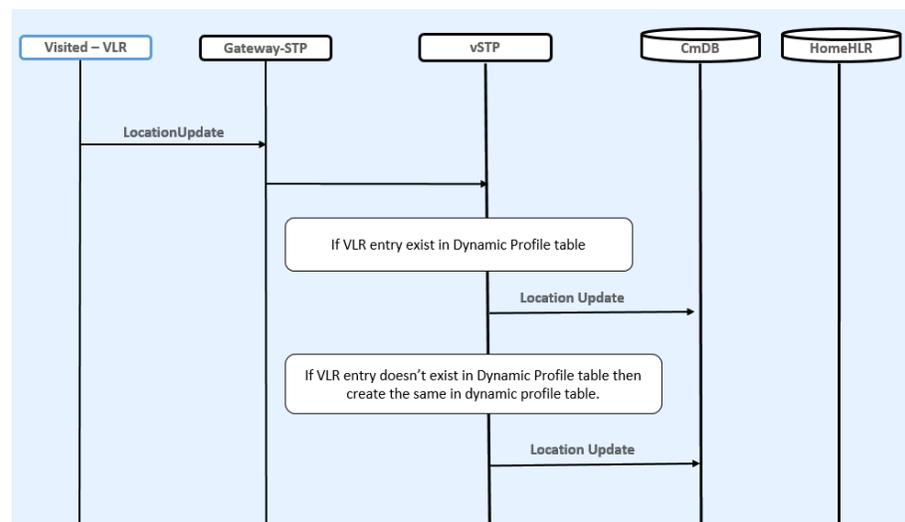
The Learn mode can be changed to Test mode in the following ways:

- Automatically, upon expiry of the configured learn mode time limit, if configured. You can configure the time in number of hours for which the vSTP operates in this mode in SFAPPOPTS table. The recommended time period for Learn mode is 6 to 12 hours.
- By manual switching of mode

### VLR Validation in Learning Mode

The following figure shows the vSTP call flow for VLR validation in learning mode:

**Figure 2-14 VLR Validation in Learning Mode**



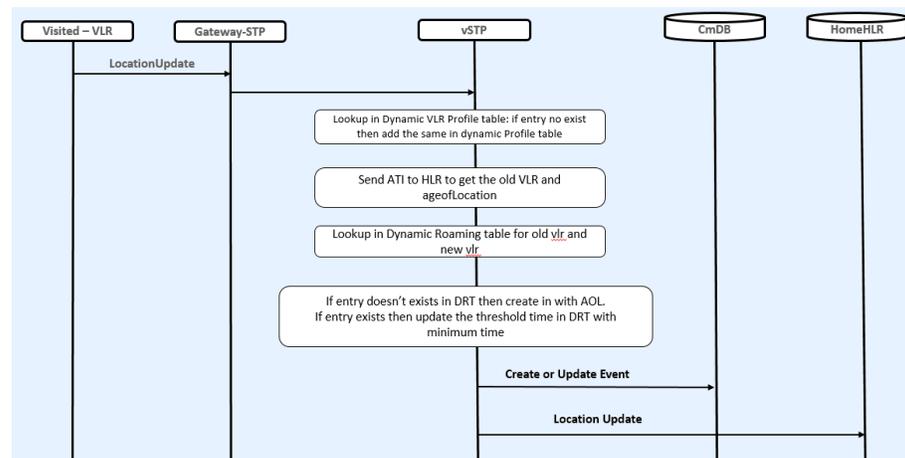
1. The incoming message will be decoded.  
An Error will be generated in case of decode failure.
2. The message information will be stored in the local database.

3. Lookup in VLR Whitelist table (static Profile).
  - If entry is found for new VLR, then the validation is skipped.
  - If entry is not found in static Whitelist VLR table, then the lookup is performed in Dynamic Profile Table (DPT).
4. If the entry is not available in DPT, then create it with filter as graylisted and forward the message to HLR. If entry is available in DPT and it is GL, then forward the message to HLR.
5. Also send the Create or Update event to UDR for IMSI record.

### Velocity Check in Learning Mode

The following figure shows the vSTP call flow for Velocity check in learning mode:

**Figure 2-15 Velocity Check in Learning Mode**



1. The incoming message will be decoded:
  - a. An Error will be generated in case of decode failure.
  - b. A Measurement will be pegged for the decode failure with OpCode and CgPA.
2. The message information will be stored in the local database.
3. If New VLR entry is not there is Dyn VLR Profile table then create it.
4. The ATI request will be generated toward the HLR identified in the CdPA of the incoming message.
5. In case the HLR sends a failure in the ATI response, a measurement will be pegged to identify HLR error corresponding message from CgPA (VLR).
6. For a success response, extract the Age of Location from the ATI Response message and the VMSC address in the HLR.
7. In case the New VLR from which the SAI/LU was received matches the old VLR in the ATI response, no action is required.
8. In case the VLR addresses does not match and old VLR is not available in Dynamic Profile table, then create the entry.
9. Lookup in Dynamic Roaming table (DRT):
  - If entry is not available, then create the DRT entry with threshold value as Age of Location value received in ATI ack.

- If DRT is already available for the same combination of old VLR and new VLR, and the value of age of location is different than that in the dynamic VLR roaming entry, age of location value in roaming entry is updated to a minimum of the age of location in dynamic VLR roaming entry and the age of location received in ATI ACK. Also Increment the entry\_usage\_threshold.
  - If subscriber state in ATI ack is `imsiDetached/PsDetached/msPurged/not registered/notProvidedfromVLRnorMME/PSactivenotreachable`, then create DRT entry and distance between two country codes is calculated using Haversine formula. Use Longitude or Latitude values from the database

10. Send the update location to HLR and also send CreateorUpdate event to UDR.

If user has configured the switch-mode timer then after expiry of that timer (in hrs), mode will switch to test mode.

### 2.4.2.4.2 Call Flow in Test Mode

This mode validates all the learned VLRs at all times. In case the VLR is not validated, it is added to greylist and measurements and alarms are generated. These measurements and alarms allows the operator to validate the whitelists and the overall solution, before they choose to go to the Active mode.

The mode can be changed to ACTIVE mode:

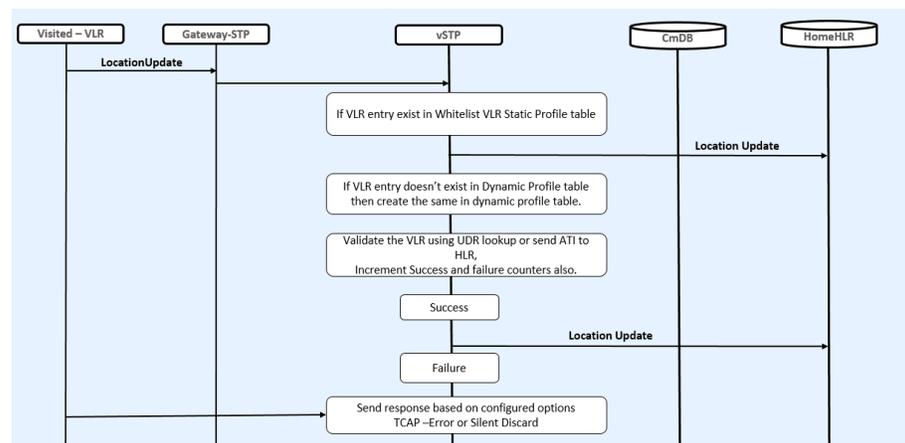
- Automatically, upon expiry of the test mode time limit, if configured
- By manual switching of mode

All the messages coming from the VLRs are allowed to the home network. This mode allows the operator to test the VLRs creation in learning mode.

#### VLR Validation in Test Mode

The following figure shows the vSTP call flow for VLR validation in test mode:

Figure 2-16 VLR Validation in Test Mode



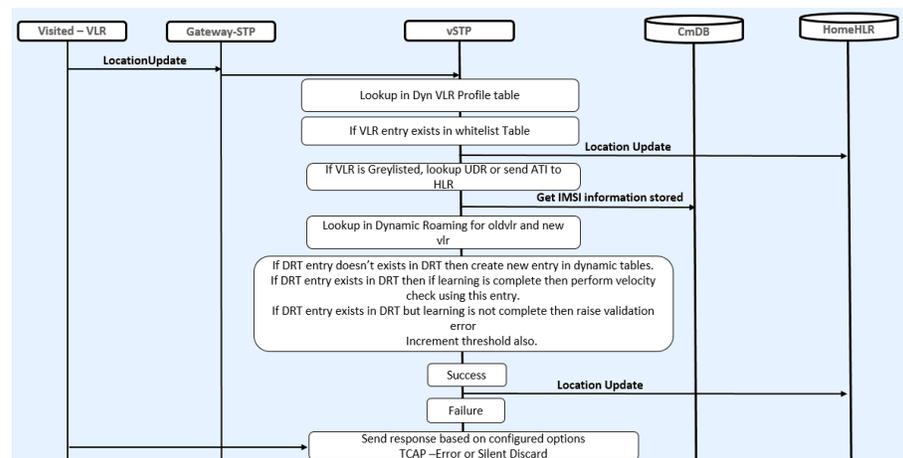
1. The incoming message is decoded. An error is generated in case of decode failure.
2. The message information gets stored in the local database.
3. Lookup in VLR Whitelist table (Profile).  
If entry is available for new VLR, then skip the validation. Otherwise, continue below steps.

4. Lookup in DPT:
  - If entry is not available for New VLR, then create the Entry in DPT.
  - If entry in DPT exists, then VLR validation is performed with lookup in UDR for that IMSI.
5. If record is not found in UDR then send ATI to HLR.
6. Update success & failure counts based on validation results.
7. If validation is success then send location update to HLR and send CreateorUpdate event to the UDR for latest timestamp.
8. If validation is failed then send response based on configured option:  
Fail Action Id is FALLBACK (do not discard messages even if the validation fails in test mode for dynamic VLRs)
9. The Greylisted dynamic VLRs remain unchanged. They are not moved to Whitelisted or Blacklisted VLRs. However, the event notification for status change (GL->BL, GL->WL, and so on) is raised, based on the threshold values.

### Velocity Check in Test Mode

The following figure shows the vSTP call flow for Velocity check in test mode:

**Figure 2-17 Velocity Check in Test Mode**



1. The incoming message is decoded.
  - a. An Error will be generated in case of decode failure.
  - b. A Measurement is pegged for the decode failure with OpCode and CgPA.
2. The message information is stored in the local database.
3. If no new VLR entry is available in Dyn VLR Profile table, then create the VLR entry.
4. Perform validation by sending ReadEvent to UDR for that IMSI record.
  - If the record is available in UDR, then extract the lastUpdatedTimestamp and VLR from UDR response.
  - If the record is not available in UDR, then ATI request is generated towards the HLR identified in the CdPA of the incoming message.  
For a success response from HLR, extract the Age of Location from the ATI Response message and the VMSC address in the HLR.

5. In case the new VLR from which the SAI/LU was received, matches the old VLR in the UDR/ATI response, no action is performed.
6. In case VLR address do not match, then check for the neighboring countries:
  - a. Lookup into SfappCCMCCMap table and for corresponding country codes and retrieve the MCC.
  - b. The exception or neighboring list table is checked with old MCC, if available in the neighboring list then do not perform any of the following steps, else proceed with the steps.
7. If the old VLR status is blacklisted and entry is not created in DPT, the velocity check results in Validation Failure.
8. If the old VLR is from ATI ACK, and the subscriber state in ATI ACK is `imsiDetached/PsDetached/msPurged/notregistered/notProvidedfromVLRnorMME/PSactive notreachable`, skip velocity check and continue with the following step, else proceed with step 9.
  - a. If the old VLR is not present in DPT, create it.
  - b. DRT entry is created or updated using long or lat data.
9. If the DRT entry does not exist in DRT, then create new entry in dynamic tables.
  - a. If DRT entry exists in DRT, and learning is complete, then perform velocity check using this entry.
  - b. If DRT entry exists in DRT and learning is not complete, then raise the validation error.
  - c. Include the increment threshold.

### 2.4.2.4.3 Call Flow in Active Mode

This mode enables the following actions:

- Learning of new VLRs. The status of existing VLRs get changed as per success or failure counts.
- Handling the dynamic VLRs based on their status. The following table describes the dynamic VLRs with status and respective results:

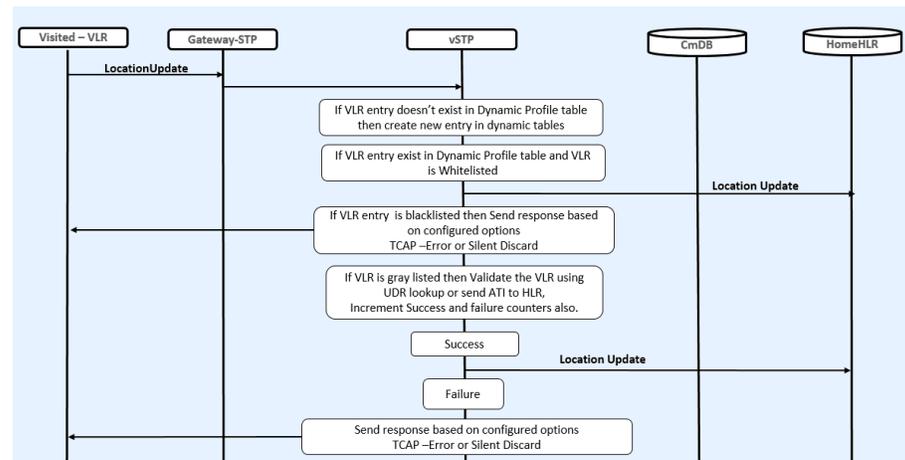
Status	Result
Whitelist	Validation Success. VLR validation is not performed
Blacklist	Validation Failure
Graylist	VLR validation is not performed <b>Note:</b> <ul style="list-style-type: none"> <li>– Success and Failure validation count is incremented based on validation result.</li> <li>– The GrayListed dynamic VLRs status can change to Whitelisted or Blacklisted</li> </ul>
Successful Validation Count	When the net <b>successful validation count</b> reaches threshold, then VLR status is changed to Whitelisted. <b>Note: success count - failure count &gt;= success threshold</b>

Status	Result
Failure Validation Count	When the net <b>failure validation count</b> reaches threshold, then VLR status is changed to Blacklisted. <b>Note: failure count - success count &gt; = failure threshold</b>

### VLR Validation in Active Mode

The following figure shows the vSTP call flow for VLR validation in active mode:

**Figure 2-18 VLR Validation in Active Mode**

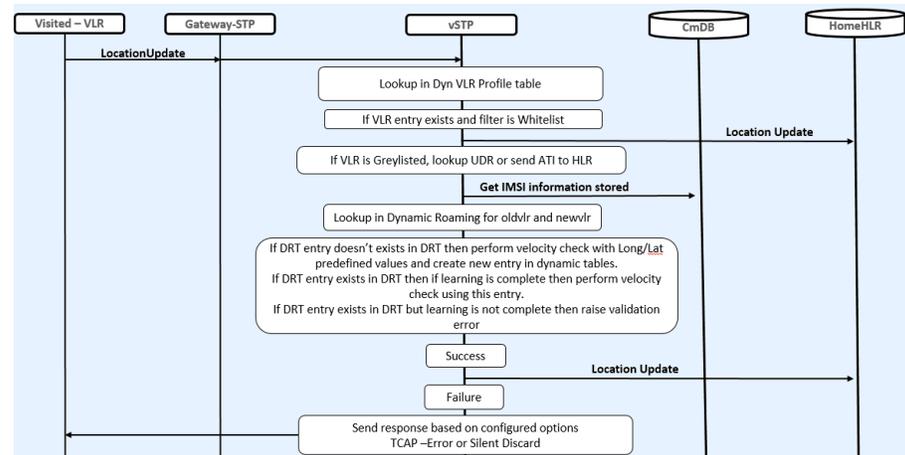


1. If the VLR is available in DPT or Whitelist Profile Table (WPT) as whitelist, then the validation is successful and LocUpdate is sent to HLR.
2. In case the VLR is available in DPT as blacklist, then the message is rejected.
3. If DPT entry is GL, then VLR validation is performed.
4. If entry does not exist, then create new entry in DPT (as learn mode is also enabled in active mode).
5. If entry is GL or entry doesn't exist in DPT, then perform validation such as lookup in UDR.
  - If IMSI record is found in UDR, then extract VLR from UDR response.
  - If record is not found in UDR then, send ATI to HLR and extract VLR from ATI ACK.
6. If old and new VLRs are same, then validation is success, otherwise the validation is failed.
7. Move VLR to BL/WL based on the threshold value, validation result, and raised event.
8. On successful validation, send CreateorUpdate event to UDR.

### Velocity Check in Active Mode

The following figure shows the vSTP call flow for Velocity check in active mode:

Figure 2-19 Velocity Check in Active Mode



1. In case the VLR address does not match, then check for the neighboring countries:
  - a. Check SfappCCMCCMap table for corresponding country codes and retrieve the MCC.
  - b. The exception or neighboring list table is checked with old MCC if it is available in neighboring list, then do nothing. Else, continue with the following steps.
2. If the old VLR is from ATI ACK, and the subscriber state in ATI ACK is `imsiDetached/PsDetached/msPurged/not registered/notProvidedfromVLRnorMME/PSactive not reachable`, continue with following steps else proceed with step 3.
  - a. Skip the velocity check.
  - b. If the old VLR is not present in DPT, create it.
  - c. If DRT entry, create or update it using long or lat data.
3. If the status of the old VLR Blacklisted, then entry is not created in DPT. The velocity check results in Validation Failure.
4. Perform lookup in DRT for old and new VLR combination. If entry exists and learning is complete (`velocity_threshold` exceeds), then perform velocity check using the entry.
5. If no DRT entry is available, then create entry using long/lat table time and perform velocity check using that DRT entry. Update UDR if validation is successful. Also update VLR success or failure count, based upon the validation result.
6. If entry is available but learning is not complete, then take time from long or lat table and reset the time also with Long or Lat table Time and Perform velocity check and VLRs count based upon the result.
7. Update UDR if validation is successful.

### 2.4.2.5 SFAPP Configurations

This section provides procedures to configure the connection required for SFAPP to access the database.

SFAPP is configured using the vSTP managed objects. The MMI API contains details about the URI, an example, and the parameters available for each managed object.

### 2.4.2.5.1 MMI Managed Objects for SFAPP

MMI information associated with SFAPP is accessed from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* displays, use the application navigation to locate specific vSTP managed object information.

[Table 2-15](#) lists the managed objects and operations supported for vSTP SFAPP feature.

**Table 2-15 vSTP SFAPP Managed Objects and Supported Operations**

Managed Object Name	Supported Operations
SfappNeighboringCountries	Insert, Delete
VstpMateStp	Insert, Update, Delete
SfappCountryCodes	No operations supported
SfappCountrylongLati	No operations supported
SfappCCMCCMap	Insert, Delete
VstpSccpApplications	Insert, Update, Delete
VstpSccpOptions	Update

#### SfappNeighboringCountries - Insert, Delete

Execute the MMI Client command from an active SOAM.

```
/vstp/SfappNeighboringCountries/
{
  "data": [
    {
      "mcc": 289,
      "name": "Abkhazia",
      "neighMcc": 250,
      "neighName": "Russia",
      "uniqueIdentifier": "289-250"
    },
    ...
    {
      "mcc": 648,
      "name": "Zimbabwe",
      "neighMcc": 655,
      "neighName": "South Africa",
      "uniqueIdentifier": "648-655"
    },
    {
      "mcc": 648,
      "name": "Zimbabwe",
      "neighMcc": 645,
      "neighName": "Zambia",
      "uniqueIdentifier": "648-645"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

```
}
```

Execute this command on an active SOAM for Delete operation:

```
/commonsecurity/neighboringcountries/<uniqueIdentifier> -v DELETE
```

Example output:

No output returned by URI: <https://localhost/mmi/dsr/v3.1/commonsecurity/neighboringcountries/648-645?> for 'DELETE' operation

### VstpMateStp - Insert, Update, Delete

#### Example:

Execute this command on an active SOAM to display entries.

```
/vstp/matestps/
{
  "data": [
    {
      "domain": "Itun",
      "pointCode": "13"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

Create a file as follows for insert:

```
$cat matestp.json
{
  "domain": "Itun",
  "pointCode": "13"
}
```

Execute this command on an active SOAM to insert:

```
/vstp/matestps/ -v POST -r <Absolute Path>/<filename>
```

Example output:

```
/vstp/matestps/ -v POST -r matestp.json
{
  "data": true,
```

```

    "links": {},
    "messages": [],
    "status": true
  }

```

Execute this command on an active SOAM to delete:

```
/vstp/matestps/<pointCode> -v DELETE
```

Example output:

```

/vstp/matestps/12 -v DELETE
No output returned by URI: https://localhost/mmi/dsr/v3.1/vstp/matestps/12?
for 'DELETE' operation

```

### SfappCCMCCMap - Insert, Delete

Execute the MMI Client command from an active SOAM.

```

/commonsecurity/mappings/
{
  "data": [
    {
      "cc": 1,
      "mcc": 310,
      "ndc": 1,
      "uniqueIdentifier": "1-1"
    },
    ...
    {
      "cc": 998,
      "mcc": 434,
      "uniqueIdentifier": "998-0"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}

```

Execute the following command to display:

```
/commonsecurity/mappings/<uniqueIdentifier>
```

Example output:

```

/commonsecurity/mappings/"998-0"{
  "data": {
    "cc": 998,
    "mcc": 434,
    "uniqueIdentifier": "998-0"
  },
  "links": {

```

```

        "delete": {
            "action": "DELETE",
            "description": "Delete this item.",
            "href": "/mmi/dsr/v3.1/commonsecurity/mappings/998-0",
            "type": "status"
        },
        "update": {
            "action": "PUT",
            "description": "Update this item.",
            "href": "/mmi/dsr/v3.1/commonsecurity/mappings/998-0",
            "type": "status"
        }
    },
    "messages": [],
    "status": true
}
[root@fixsetup-soal ~]# cat mapping.json
{
    "cc": 998,
    "mcc": 434
}

```

Create a file as follows for insert:

```

cat mapping.json
{
    "cc": 998,
    "mcc": 434
}

```

Execute the following command to insert:

```

/commonsecurity/mappings/ -v POST -r <Absolute Path>/<filename>

```

Example output:

```

/commonsecurity/mappings/ -v POST -r mapping.json
{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}

```

Execute this command on an active SOAM to delete:

```

/commonsecurity/mappings/<uniqueIdentifier> -v DELETE

```

Example output:

```

/commonsecurity/mappings/"998-0" -v DELETE
No output returned by URI: https://localhost/mmi/dsr/v3.1/vstp/gttactions/actid2006? for 'DELETE' operation

```

**VstpSccpApplications - Insert, Update, Delete**

Execute the MMI Client command from an active SOAM.

```
/vstp/sccpapplications/
{
  "data": [
    {
      "appType": "Sfapp",
      "ssn": 67
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

Execute the following command to display:

```
/vstp/sccpapplications/<appType>
```

```
/vstp/sccpapplications/"Sfapp"
{
  "data": {
    "appType": "Sfapp",
    "ssn": 67
  },
  "links": {
    "delete": {
      "action": "DELETE",
      "description": "Delete this item.",
      "href": "/mmi/dsr/v3.1/vstp/sccpapplications/Sfapp",
      "type": "status"
    },
    "update": {
      "action": "PUT",
      "description": "Update this item.",
      "href": "/mmi/dsr/v3.1/vstp/sccpapplications/Sfapp",
      "type": "status"
    }
  },
  "messages": [],
  "status": true
}
```

Example output:

```
"data": {
  "cc": 998,
  "mcc": 434,
  "uniqueIdentifier": "998-0"
},
```

```

    "links": {
      "delete": {
        "action": "DELETE",
        "description": "Delete this item.",
        "href": "/mmi/dsr/v3.1/commonsecurity/mappings/998-0",
        "type": "status"
      },
      "update": {
        "action": "PUT",
        "description": "Update this item.",
        "href": "/mmi/dsr/v3.1/commonsecurity/mappings/998-0",
        "type": "status"
      }
    },
    "messages": [],
    "status": true
  }
}
[root@fixsetup-soal ~]#
Insert
[root@fixsetup-soal ~]# cat mapping.json
{
  "cc": 998,
  "mcc": 434
}

```

Create a file as follows for insert:

```

$cat sccpapplication.json
{
  "appType": "Sfapp",
  "ssn": 68
}

```

Execute the following command to insert:

```

/vstp/sccpapplications/ -v POST -r <Absolute Path>/<filename>

```

Example:

```

/vstp/sccpapplications/ -v POST -r sccpapplication.json
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}

```

To update the file:

```

$cat sccpapplication.json
{

```

```

    "appType": "Sfapp",
    "ssn": 69
  }

```

Execute this command on an active SOAM to update:

```
/vstp/sccpapplications/ -v PUT -r <Absolute Path>/<filename>
```

Example output:

```

/vstp/sccpapplications/ -v PUT -r sccpapplication.json
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}

```

Execute this command on an active SOAM to delete:

```
/vstp/sccpapplications/<appType> -v DELETE
```

Example output:

```

/vstp/sccpapplications/"Sfapp" -v DELETE
No output returned by URI: https://localhost/mmi/dsr/v3.1/vstp/
sccpapplications/Sfapp? for 'DELETE' operation

```

### VstpSCCPOptions- Update

Execute the MMI Client command from an active SOAM.

```

/vstp/sccpoptions/
{
  "data": {
    "classlseq": "Disabled",
    "dfltfallback": false,
    "dfltgttmode": "Cd",
    "mtprgtt": "Off",
    "mtprgttfallback": "Mtproute",
    "tgtt0": "None",
    "tgtt1": "None",
    "tgttudtkey": "Mtp",
    "tgtxudtkey": "Mtp",
    "travelVelocity": 700
  },
  "links": {
    "update": {
      "action": "PUT",

```

```

        "description": "Update this item.",
        "href": "/mmi/dsr/v3.1/vstp/sccpoptions/",
        "type": "status"
    }
},
"messages": [],
"status": true
}

```

### Note

The **travelVelocity** is an existing MO and a new parameter "travel\_velocity" has been added as part of SFAPP feature.

Create a file as follows for update:

```

$cat sccpoption.json
{
    "class1seq": "Disabled",
    "dfltfallback": false,
    "dfltgttmode": "Fcd",
    "itun16ScmgEnabled": false,
    "tgtt0": "None",
    "tgtt1": "None",
    "tgttudtkey": "Mtp",
    "tgttxudtkey": "Mtp",
    "mtprggt": "Usemtppc",
    "mtprggtfallback": "Gttfail",
    "travelVelocity": 650
}

```

Execute the following command to update:

```
/vstp/sccpoptions/ -v PUT -r <Absolute Path>/<filename>
```

Example output:

```

/vstp/sccpoptions/ -v PUT -r sccpoption.json
{
    "data": true,
    "links": {},
    "messages": [],
    "status": true
}

```

Example output:

```

/vstp/sccpoptions/
{

```

```

"data": {
  "classlseq": "Disabled",
  "dfltfallback": false,
  "dfltgttmode": "Fcd",
  "mtprgtt": "Usemtppc",
  "mtprgttfallback": "Gttfail",
  "tgtt0": "None",
  "tgttl": "None",
  "tgttudtkey": "Mtp",
  "tgtxudtkey": "Mtp",
  "travelVelocity": 650
},
"links": {
  "update": {
    "action": "PUT",
    "description": "Update this item.",
    "href": "/mmi/dsr/v3.1/vstp/sccpoptions/",
    "type": "status"
  }
},
"messages": [],
"status": true
}

```

### SfappCountryCodes

There is no MMI support available for SfappCountryCodes, but a user can retrieve the data by executing get command on an active SOAM.

### SfappCountrylongLati

There is no MMI support available for SfappCountrylongLati, but a user can retrieve the data using get command on an active SOAM.

## 2.4.2.5.2 MMI Managed Objects for SFAPP Dynamic Learning

The following table lists the managed objects and operations supported for SFAPP Dynamic Learning feature.

**Table 2-16 SFAPP Dynamic Learning Managed Objects and Supported Operations**

Managed Object Name	Supported Operations
SfappOptions	Display, Update
whitelistvlrprofile	Insert, Update, Delete
vlrprofiles	Display
vlrromings	Display

### SfappOptions - Display, Update

Execute the MMI Client command from an active SOAM to display:

```
/vstp/sfappoptions
```

Example Output:

```
{
  "data": [
    {
      "agingTimer": "None",
      "failureThreshold": "4",
      "learnTimer": "5",
      "sfappMode": "Test",
      "successThreshold": "5",
      "velocityThreshold": "40"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

Create a file as follows for insert:

```
cat <filename.json>
{
  "failureThreshold": "5"
}
```

Execute this command on an active SOAM for Update operation:

```
/vstp/sfappoptions -v PUT -r /tmp/<filename.json>
```

Example output:

```
{
  "data": [
    {
      "agingTimer": "None",
      "failureThreshold": "5",
      "learnTimer": "5",
      "sfappMode": "Test",
      "successThreshold": "5",
      "velocityThreshold": "40"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

**Whitelist Vlr Profiles - Insert, Update, Delete**

**Example:**

Execute this command on an active SOAM to display entries.

```
/vstp/whitelistvlrprofiles/  
  
"data": [  
  {  
    "filter": "WhiteList",  
    "vlr": 1  
  }  
]
```

Create a file as follows for insert:

```
Cat <filename>  
  {  
    "filter": "WhiteList",  
    "vlr": 1  
  }
```

Execute this command on an active SOAM to insert:

```
/vstp/whitelistvlrprofiles -v POST -r /tmp/<filename>
```

Example output:

```
{  
  "data": true,  
  "links": {},  
  "messages": [],  
  "status": true  
}
```

Execute this command on an active SOAM to delete:

```
/vstp/whitelistvlrprofiles/16 -v DELETE
```

Example output:

```
/vstp/whitelistvlrprofiles/12 -v DELETE
```

### VLR Profiles - Display

Execute the MMI Client command from an active SOAM.

```
/vstp/vlrprofiles  
{  
  "data": [  
    {  
      "failureCount": 0,  
      "filter": "GrayList",
```

```

        "lastUsedTime": "1969-12-31T19:00:00-05:00",
        "successCount": 0,
        "vlr": "4114001133"
    }
],
"links": {},
"messages": [],
"status": true
}

```

### VLR Roaming - Display

Execute the MMI Client command from an active SOAM.

```

/vstp/vlrroamings
{
  "data": [
    {
      "entryUsageCount": 2,
      "lastUsedTime": "1969-12-31T19:00:00-05:00",
      "newVlr": 65746892,
      "oldVlr": 65746892,
      "time": 4085,
      "uniqueIdentifier": "65746892-65746892"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}

```

## 2.4.2.5.3 SFAPP Alarms and Measurements

### Alarms and Events

The following table lists the Alarms and Events specific to the SFAPP support for vSTP:

Alarm/ Event ID	Name
70293	SFAPP Validation Error
70294	SFAPP Validation Matching State not found
70295	SFAPP Validation Encoding Error
70296	SFAPP Validation Response Timeout Error.
70297	SFAPP Validation Velocity Chk Failed.
70298	SFAPP Validation Failed <b>Note:</b> The parameter <code>ageOfLoc</code> and <code>Threshold</code> with zero can be ignored if not relevant for scenarios where this UIM is observed.
70299	SFAPP Invalid CC/NDC received
70300	Updation failed in UDR
70301	VSTP SFAPP Stack Event Queue Utilization
<b>SFAPP Dynamic Learning</b>	
70429	VstpDynVlrStatusChanged

Alarm/ Event ID	Name
70430	VstpDynVeloThreshCrossed
70431	VstpDynVLRProfAging
70432	VstpDynVLRRoamAging
70433	VstpVlrDynLearningOFF
70434	VstpVlrDynLearningLearntimer

For more details related to Alarms and Events, refer to Alarms and KPIs Reference document.

### Measurements

The following table lists the measurements specific to the SFAPP support for vSTP:

Measurement ID	Measurement Name
21702	VstpSfappMsgSuccess
21703	VstpSfappMsgFailed
21704	VstpSfappMsgError1
21705	VstpSfappMsgError2
21706	VstpRxSfappMsg
21707	VstpRxSfappMsgDiscard
21708	VstpSfappInternalError
21709	VstpSfappCADecodeFail
21710	VstpSfappCATimeOut
21711	VstpSfappCAAvgProcessTime
21712	VstpSfappCAMaxProcessTime
21713	VstpSfappSubsNotFound
21714	VstpSfappCATx
21715	VstpSfappCATxFail
21716	VstpSfappPduFull
21717	VstpSfappCAProcesTime
21718	VstpSFAPPStackQueuePeak
21719	VstpSFAPPStackQueueAvg
21720	VstpSFAPPStackQueueFull
21782	VstpTxSfappMsg
21783	VstpTxSfappMsgPeak
21784	VstpTxSfappMsgAvg
<b>SFAPP Dynamic Learning</b>	
21937	VstpDynNewVLR
21938	VstpDynNewRoamEntry
21939	VstpDynVLRBL
21940	VstpDynVLRWL
21941	VstpDynVLRGL
21942	VstpDynVelCrossed
21943	VstpDynVLRProfAging
21944	VstpDynVLRRoamAging

For more details related to measurements, refer to Measurement Reference document.

#### 2.4.2.5.4 UDR Configuration for SFAPP

Configuring UDR for SFAPP involves adding vSTP MP(s) to UDR and then configuring UDR on the ComAgent server.

As a prerequisite for UDR configuration, it is assumed that the user is aware of UDR and ComAgent functionality. Also, UDR must be installed and the UDR topology must be configured.

Perform the following steps:

1. Add details about the vSTP MP on the ComAgent Remote Servers screen as a client by navigating to **Communication Agent**, and then **Configuration**, and then **Remote Servers** and clicking **Insert** on an active OCUDR NOAMP.
2. Select the OCUDR server group from the *Available Local Server Groups* that needs to communicate with vSTP MP.
3. From the active OCUDR GUI, navigate to **Communication Agent**, and then **Maintenance**, and then **Connection Status** and verify connection are *InService*.
4. From the active OCUDR GUI, navigate to **Communication Agent**, and then **Maintenance**, and then **Routed Services Status** and verify the *STPDbSvc* status is *Normal*.
5. From an active DSR NOAM, navigate to **Communication Agent**, and then **Configuration**, and then **Remote Servers** and click **Insert**.
6. Add the UDR NO IP in the ComAgent Remote Server screen as a Server.
7. Select the STP MP server group from the *Local SG* that needs to communicate with UDR.
8. Also add the Standby and DR NOs to the Local SG.
9. Navigate to **Communication Agent**, and then **Configuration**, and then **Connection Groups**, then select *STPSvcGroup* and click **Edit**.
10. Add all available UDR NO servers.
11. Navigate to **Communication Agent**, and then **Maintenance**, and then **Connection Status**, select the server name, and check the connection status.

UDR Configuration: SOAP Provisioning Request for IMSI

##### Steps to Enable SFAPP Feature on UDR:

Enable SFAPP feature on UDR by running the **enableSecurityApp** loader on the Active NOAM Server console. Follow the below steps:

1. Go to the path: `/usr/TKLC/udr/prod/maint/loaders/upgrade`
2. On the path, execute the **enableSecurityApp** script.

##### Note

See [Manually Adding SGSN Field in UDR](#) section, for manual steps to add SGSN field in UDR.

Here's an example of provisioning SFAPP data with the Type as RN and GRN in an individual IMSI.

```
<?xml version="1.0" encoding="UTF-8"?>
<subscriber>
<field name="IMSI">6912347700</field>
<field name="VPLMN">816308</field>
<field name="MCC">611</field>
<field name="MMER">epc.mnc905.mcc679.org</field>
<field name="MMEH">s6amme-epc.mnc905.org</field>
<field name="HSSR">hss@3gppnetwork.org</field>
<field name="HSSH">hss-epc.mnc905.mcc679.3gppnetwork.org</field>
<field name="lastUN">3G</field>
<field name="VLR">12340000</field>
<field name="SGSN">56780000</field>
</subscriber>
```

### Note

An UPDATE request from vSTP is assigned to an Active UDR only. However, a READ request from vSTP can be assigned to both Active or Standby UDR. To check the status of the UDR, navigate to **Communication AgentMaintenanceHA Services Status**. Check value for the **Active SRs** field for the UDR. If the value is **1**, the UDR is in active status. Therefore, an UPDATE request will be sent to this UDR.

## 2.4.2.6 Dependencies

The SFAPP support for vSTP has no dependency on any other vSTP operation.

## 2.4.2.7 Troubleshooting

The vSTP SFAPP sends default response in case of the following scenarios:

- **SFAPP thread CPU utilization exceeds congestion level**  
Check if the SFAPP thread CPU utilization exceeded Congestion Level 2. This check is performed at the beginning of the message processing cycle and if set, vSTP immediately responds with default response.  
  
The equipment status value set against eirDefResplnErr option of EirOptions table is sent right away.
- **SFAPP operational state**  
Check if the SFAPP operational state is **Unavailable**. vSTP performs this check before sending the message to UDR and if the state **Unavailable**, the default response is sent and the query is not sent to UDR. The VstpSfappCATimeOut meal is pegged in this scenario.

The following points must be considered while configuring SFAPP over vSTP:

- The J1 and ATM interfaces are not supported.
- Single vSTP MP VM can support only one 4-Port ADAX HDC3 Card.
- An ADAX HDC3 card cannot be accessed from Multiple VSTP MP VMs .

- The ADAX HDC3 driver components and RPMs needs to be installed separately.
- The DSR patch is required to be applied on vSTP MP VM that is connected to ADAX HDC3 card.
- In SFAPP dynamic learning, when no new VLRs get reflected in the replicated tables (VstpSfappVlrProfile/ VstpSfappVlrRoaming), then ensure that the vSTP OAM process is up and running on SOAM and its not under reboot.

## 2.4.3 Support for CAT2 SS7 Security

The CAT2 SS7 Security functionality allows vSTP to detect anomalies on inbound Category 2 packets through bulk upload of customer IR.21 documents.

### ① Note

The IR.21 document contains operator wise network information such as, MCC-MNC, Node GT (HLR/VLR/MSC), and CC-NDC.

For detailed information about this feature, refer to *vSTP SS7 Security User's Guide*.

### 2.4.3.1 Feature Overview

vSTP provides the IR.21 Utility to read and record the information present in GSMA IR.21 document.

The SCPVAL GTT Action addresses the SS7 CAT2 security checks. This GTT action ensures that the MSU details such as, CGPA and IMSI belongs to same operator after validating it with the newly generated table.

The CAT2 SS7 functionality is described as follows:

The IR.21 xml file is parsed through IR.21 utility. The information required for message validation is extracted from the file. The data is stored in vSTP tables.

### ① Note

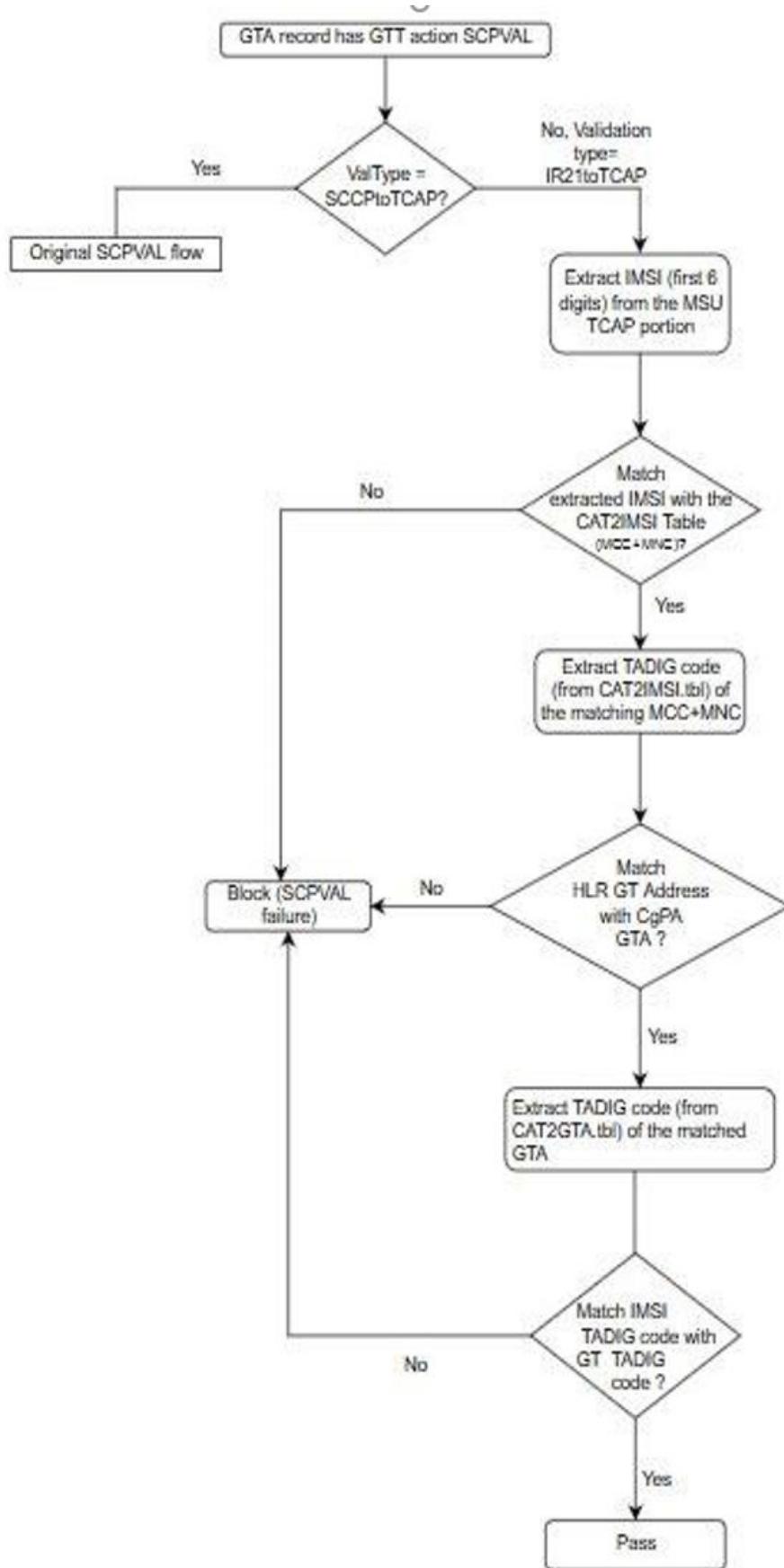
- The information can also be populated using MMIs. However, it is not the preferred method.
- When test mode is enabled for SCPVAL, GTTAction SCPVAL validations are executed, and all relevant events and alarms are generated. However, failed validations do not discard messages, they are allowed to pass through. Validation results are not enforced in test mode.

The GTT is configured to enforce CAT 2 validation on the received MSUs. The validation is performed based on the data available in IR21RoutingInfo and IR21NetworkElement tables.

### CAT2 SS7 Security Workflow

The following flow chart provides an overview of the CAT2 SS7 Security functionality:

Figure 2-20 CAT2 SS7 Security Workflow



The CAT2 SS7 Security functionality is described as follows:

- **Conversion of IR21 xml file**

- vSTP provides the IR21 Utility on SOAM. The IR21 Utility accepts operator IR21 input file in XML format and generate error message in case of no or other than IR21 XML files.
- The output is generated in the form of two CSV files named `IR21NetworkElement.csv` and `IR21RoutingInfo.csv`.
- The entries in the CSV files have length based validation for all fields. For example, sender TADIG code and TADIG code must be of 5 digits, IMSI must be of 6 digits, Node Type must be of 1 digit, GT Address range must be of 15 digits.
- The `IR21NetworkElement` table stores value 0 for HLR and 1 for MGT. Therefore, no validation is performed on this value.

**Note**

The IR21 utility supports parsing of 1000 IR.21.xml input files in alphabetical order in an instance. For more details on IR21 Utility, see [GUI Configurations for CAT2 SS7 Security Support](#).

- **Bulk upload after conversion**

The generated CSV files are imported using the **Import** option under **Diameter Common** on SOAM.

The following data is extracted from IR21 file and stored on vSTP:

- Sender TADIG code (RAEX IR.21 Information) : It is retrieved from the RAEX IR21 FileHeader tag and used to identify the operator. It consist of two fields, with a total length of five characters consisting of three-character country code and a two character operator or company identifier. Sender TADIG code is stored against each entry.
- Routing Information Data (Section ID 4) : It is a mandatory section in IR21 document of the operator. The vSTP `IR21RoutingInfo` table stores the MCC-MNC (E.212) along with TADIG code from this section. The vSTP `IR21NetworkElement` stores the CC-NC (from E.214) along with TADIG code from this section.
- Network Element Information Data (Section ID 13) – It is an optional section in IR21 document of the operator. The vSTP `IR21NetworkElement` table stores the HLR Node type GT address or Address range along with the TADIG code from this section.

- **Handling Multiple Countries with Shared MCCMNC Using TADIGIT Codes in CAT2IMSI Validation**

The following details describe how to handle multiple countries with shared MCCMNC using TADIGIT codes in CAT2IMSI validation:

- Multiple countries can share the same MCCMNC.
- To differentiate between these countries, associate multiple TADIGIT codes with a single MCCMNC entry in the CAT2IMSI table.
- Store these TADIGIT codes as comma separated values in the CAT2IMSI table.
- During validation, compare the TADIGIT value from CAT2GTA against all TADIGIT values listed (comma separated) in CAT2IMSI.
- Validation succeeds if a match is found, otherwise, it fails.

- **Validation**

The SCPVAL GTT action validates that the MSU details: CgPA and IMSI belongs to same operator. The validation is performed using the data available in `IR21RoutingInfo` and `IR21NetworkElement` tables.

The following OPCODES are applicable for CgPA and IMSI validation:

- provideRoamingNumber (4)
- provideSubscriberInfo (70)
- provideSubscriberLocation (83)
- cancelLocation (3)
- insertSubscriberData (7)
- deleteSubscriberData (8)
- getPassword (18)
- reset (37)
- activateTraceMode (50)
- unstructuredSS-Request (60)
- unstructuredSS-Notify (61)
- informServiceCentre (63)
- alertServiceCentre (64)
- setReportingState (73)
- remoteUserFree (75)
- istCommand (88)

The IMSI has upto 15 digits value. The value is composed of three parts:

- **Mobile Country Code (MCC):** Consists of 3 digits
- **Mobile Network Code (MNC):** Consists of 2 or 3 digits
- **Mobile Subscriber Identification Number (MSIN):** 9 or 10 digits

The MCC and MNC parameters (first 5-6 digits) determine the Operator ID. Hence, these values are used during CAT2 validation.

At first, the match is performed with 6 digit, and if the match is not found, then it is performed with 5 digits. In case, the match is not found, the validation gets failed.

## 2.4.3.2 Feature Configurations

This section provides procedures to perform the CAT2 SS7 Security functionality.

CAT2 SS7 Security is configured using the vSTP managed objects and vSTP GUI. The MMI API contains details about the URI, an example, and the parameters available for each managed object.

### 2.4.3.2.1 MMI Managed Objects for CAT2 SS7 Security Support

MMI information associated with CAT2 SS7 Security support is accessed from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* gets opened, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for CAT2 SS7 Security support:

**Table 2-17 CAT2 SS7 Security support Managed Objects and Supported Operations**

Managed Object Name	Supported Operations
cat2imsi	Insert, Delete
cat2gta	Insert, Delete
gttactions	Insert, Delete, Update

#### cat2imsi - Insert, Delete

Create a file with following content. File name could be anything, for example option name can be used as filename:

```
$ cat cat2imsi.json
{
  "tadigitCode": "TEST",
  "stadigitCode": "TEST",
  "mccmnc": "12345"
}
```

Execute the following command on Active SOAM to update the data:

```
/vstp/cat2imsi -v POST -r cat2imsi.json
```

#### Sample Output:

```
{
  "data": [
    {
      "mccmnc": "12345",
      "stadigitCode": "TEST",
      "tadigitCode": "TEST"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

#### Cat2Gta - Insert, Delete

Create a file with following content. File name could be anything, for example option name can be used as filename:

```
$cat cat2gta.json
{
  "gttStartAddress": "22345678",
  "uniqueIdentifier": "23405678-23405678-HLR",
  "stadigitCode": "TEST",
  "gttEndAddress": "22345678",
```

```
"nodeType": "HLR",
"tadigitCode": "TEST"
}
```

Execute the following command on Active SOAM to insert the data:

```
/vstp/cat2gta -v POST -r cat2gta.json
```

Sample Output:

```
{
  "data": [
    {
      "gttEndAddress": "22345678",
      "gttStartAddress": "22345678",
      "nodeType": "HLR",
      "stadigitCode": "TEST",
      "tadigitCode": "TEST",
      "uniqueIdentifier": "22345678-22345678-HLR"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

### gttactions - Insert

Execute the following command on Active SOAM to display table data:

Create a file with following content. File name could be anything, for example option name can be used as filename:

```
$ cat gtt_act.json
{
  "valType": "IR21ToTcap",
  "ndgt": "All",
  "actid": "actvall",
  "act": "Scpval"
}
```

Execute the following command on Active SOAM to insert the data:

```
/vstp/gttactions -v POST -r gtt_act.json
```

Sample Output:

```
{
  "data": [
    {
      "act": "Scpval",

```

```

    "actid": "actvall",
    "defactid": "fallback",
    "ndgt": "All",
    "uimreqd": false,
    "useicmsg": false,
    "valType": "IR21ToTcap"
  }
],
"links": {},
"messages": [],
"status": true
}

```

### 2.4.3.2.2 GUI Configurations for CAT2 SS7 Security Support

The CAT2 SS7 Security functionality can be configured from Active System OAM (SOAM).

#### To convert IR21 File

On the Active System OAM (SOAM), select **VSTP > IR21 Utility > Conversion**.

**Figure 2-21 IR21 Utility**

Main Menu: VSTP -> IR21 Utility -> Conversion

Tasks

IR21 Utility: IR21 Utility to convert IR21 XML files to IR21 csv files. Converted IR21(IR21NetworkElement.csv and IR21RoutingInfo.csv) files can be imported using Main Menu: Diameter Common -> Import screen. This screen lists all the IR21 xml files under File Management 'IR21XMLGUI' directory.

File Name	Line Count	Time Stamp
IR21_AFGEA_Emirates.xml	Unknown*	2020-May-07 03:26:43
IR21_INDT0_Tata.xml	Unknown*	2020-May-07 03:26:43
IR21_USACG_AT_T.xml	Unknown*	2020-May-07 03:26:43

IR21 Utility

\*Files which shows "unknown" for Line Count may not be converted, provide read file permission to that file for conversion. Converted IR21 (IR21NetworkElement.csv and IR21RoutingInfo.csv) files can be seen by clicking File Management button.

Convert All Files | Convert Selected Files | File Management

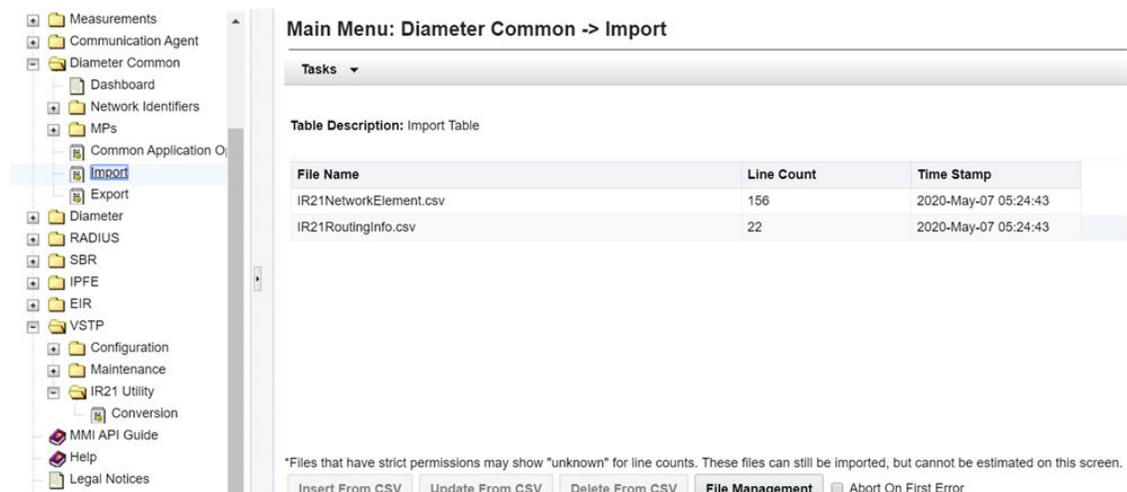
Copyright © 2010, 2020, Oracle and/or its affiliates. All rights reserved.

The IR21 Utility converts the IR21 XML files to CSV files.

#### Importing CSV Files

The converted `IR21NetworkElement.csv` and `IR21RoutingInfo.csv` files can be imported from Active System OAM (SOAM).

The **Group Code** parameter must be configured in the **Local Signalling Points** and **Remote Signalling Points** options. Select **VSTP > Diameter Common > Import**. The page lists all the IR21 files under **File Management > IR21XMLGUI** directory.

**Figure 2-22 Importing IR21 CSV Files**

For more details on IR21 Utility GUI configurations, see IR21 Utility section in *vSTP User Guide*.

### 2.4.3.2.3 CAT2 SS7 Security Alarms and Measurements

#### Alarms and Events

There are no alarms or events specific to the CAT2 SS7 Security functionality.

#### Measurements

The following table lists the measurements specific to the CAT2 SS7 Security support for vSTP:

Measurement ID	Measurement Name
21971	VstpGttActScpvalCat2Total
21972	VstpGttActScpvalCat2Discard
21973	VstpGttActScpvalCat2NotApplied
21974	VstpCgpaGttActScpvalCat2Total
21975	VstpCgpaGttActScpvalCat2Discard
21976	VstpCgpaGttActScpvalCat2NotApplied

For more details related to measurements, refer to Measurement Reference document.

### 2.4.3.3 Troubleshooting CAT2 SS7 Security

In case of the error scenarios, the measurements specific to CAT2 SS7 Security feature are pegged. For information related to CAT2 SS7 Security measurements, see [CAT2 SS7 Security Alarms and Measurements](#).

### 2.4.3.4 Dependencies

The CAT2 SS7 Security support for vSTP has no dependency on any other vSTP operation.

## 2.4.4 vSTP SMS Home Router

The vSTP SMS Home Router feature provides network monitoring for abnormal SMS activities, by obtaining statistics or reports from the SS7 FW. This feature enables vSTP to filter and accept the SMS traffic as per GSMA guidelines.

In order to address spoofing and spamming issues, the SMS Home Router functionality enables all the Managed Objects (MOs) and MT SMS to be routed via an SMS-SC-like logical entity (SMS ProxyService) located in the HPLMN of the receiving MS. An SMS signaling FW is provided to analyze MO and MT packets, before submitting or delivering.

### 2.4.4.1 Feature Description

With the Home SMS Router feature, a new **vSTPService** MP or node is introduced in the vSTP architecture. This node analyzes and validates the MO and MT packets before submitting or delivering.

All the configurations related to Home SMS Routing feature are managed using **vSTPService**. If SMS Service is enabled for a Linkset on vSTP MP, then vSTP MP forwards the SMS message to **vSTPService** for SMS routing to handle the possible attacks, such as spoofing, spamming, and DOS attacks.

If more than one **vSTPService** is present in the architecture, then message are forwarded to different **vSTPService** MPs in round-robin fashion. vSTP MP forwards the message to **vSTPService** MP through ComAgent Connection. **vSTPService** MP does not send the message to any external node directly. It is done through vSTP MP.

The following sections describes the possible SMS attacks and the call flows used by vSTP to prevent them.

#### 2.4.4.1.1 MO Spoofing

This section explains MO Spoofing. MO Spoofing occurs when a sender manipulates address information.

It is an SMS MO with a manipulated A-MSISDN (real or wrong) coming into the HPLMN network from a foreign VLR (real or wrong SCCP Address). As per HPLMN, it is the roaming scenario where one subscriber is in roaming and sending SMS. However, it is not a real subscriber. The message is not sent by a real mobile but is generated from a specific system with a C7 application. The A-MSISDN being used can be real or not depending on the screening in place in the HPLMN SMS-C (Screening on CC+NDC or No A-MSISDN screening in place).

The following are some other fields that can be manipulated in SMS MO:

- SM-RP-DA (Short-Message Relay-Protocol Destination-Address) field
- RP-Destination Address (Relay Protocol-Destination Address) field
- TP-Destination Address (Transfer Protocol-Destination) field

#### Note

SPAM messages can be sent from a valid originator. MO Spoofing is not used to block the spam messages. MO Spoofing is handled by processing MO\_FSM packets and perform extra validation on vSTPService MP.

### 2.4.4.1.1.1 Handling MO Spoofing

The SMS Home Router functionality provides a new service MP that enables vSTP to perform extra validations address the MO-Spoofing issue.

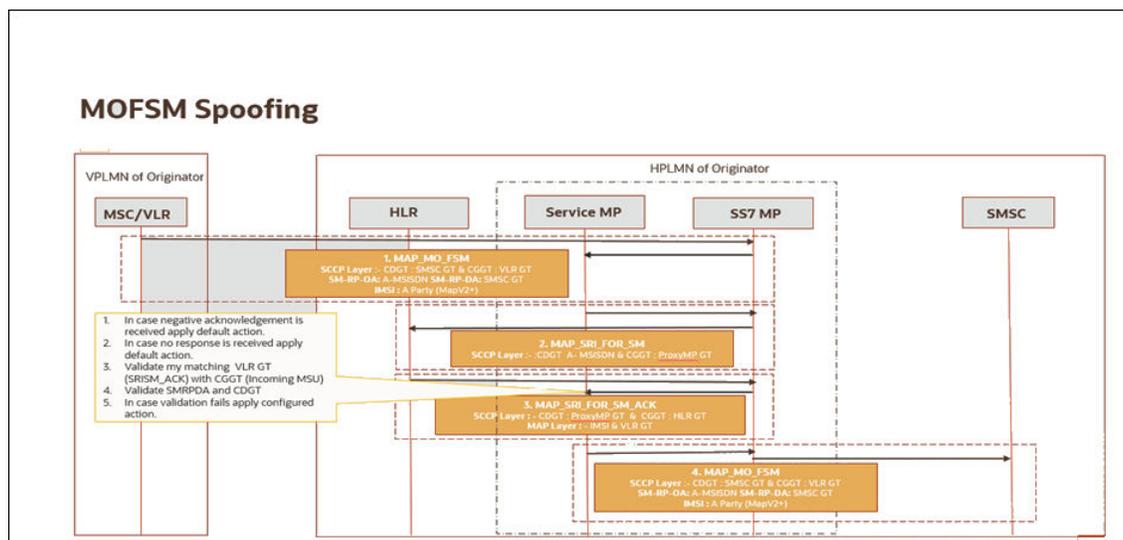
The call flow when MO Spoofing is enabled on Linkset :

1. Forward MO\_FSM (Opcode 46) to ServiceMP
2. Extract A party details (MSISDN) and perform SRISM
  - CDPA Digits: A MSISDN
  - CGPA Digits : ProxyVM
  - GTMAP Layer : A MSISDN
3. Extract VLR & IMSI from SRISM\_ACKMatch VLR ID (from SRISM\_ACK) and CGPA of the original MSU
4. Match IMSI (SRISM\_ACK) and Original MSU
5. Match SM-RP-DA and CDPA in original MOFSM MSU (SCCP VAL MOFSM).  
After successful validation forward the MSU back to vSTP MP for further routing.

**Note**

This action is not performed, if CGPA of the incoming MSUs matches the trusted MSC/VLR list.  
The vstpSmsProxySMSCStatus MO is used to configure the MSC/VLR list, whether it is allowed or blocked. If it is allowed, then no validation on service message is performed and the message is forwarded back to SS7 for further routing. If the MSC is blocked, then it is considered as validation failure. The default action is applied on the service MP.

The following figure describes the MO-FSM call flow:



### 2.4.4.1.2 MT Spoofing

MT Spoofing involves the manipulation of SCCP or MAP addresses. A fake SMS is originated from the international C7 network and terminated to a mobile network. The originating MS's HPLMN delivers the Short Message directly to the receiving MS's VPLMN after querying the HLR for the current location of the receiving MS. Thus, the HPLMN is not present in the MT routing of the actual data, such as SM.

There is currently no specific correlation between the MAP\_SRI\_For\_SM MAP operation and the subsequent MAP\_MT\_Forward\_Short\_Message MAP operation(s). This missing correlation is exploited by hackers to make spam, flooding and faking.

Below points describes the MT-spoofing:

- Fraudulently manipulate MTFSM to send SMS on another operator's account.
  - SMSC of an MTFSM doesn't reflect the actual originating network.
  - MAP and SCCP layer both can be spoofed.
  - Terminating charges are billed to spoofed network.
  - MTFSM spoofing is used for premium rate service.
  - Fraudster send SRISM for subscriber and get the serving VLR details.
  - Terminates MTFSM directly to subscriber without association of any HPLMN nodes.
- MT Spoofing is handled by processing 2 packets (SRISM and MTFSM) on vSTPService MP. In HomeSMSC routing feature, SMSPROXY performs the validation to handle the MT spoofing issues.

#### 2.4.4.1.2.1 Handling MT Spoofing - SRI\_SM

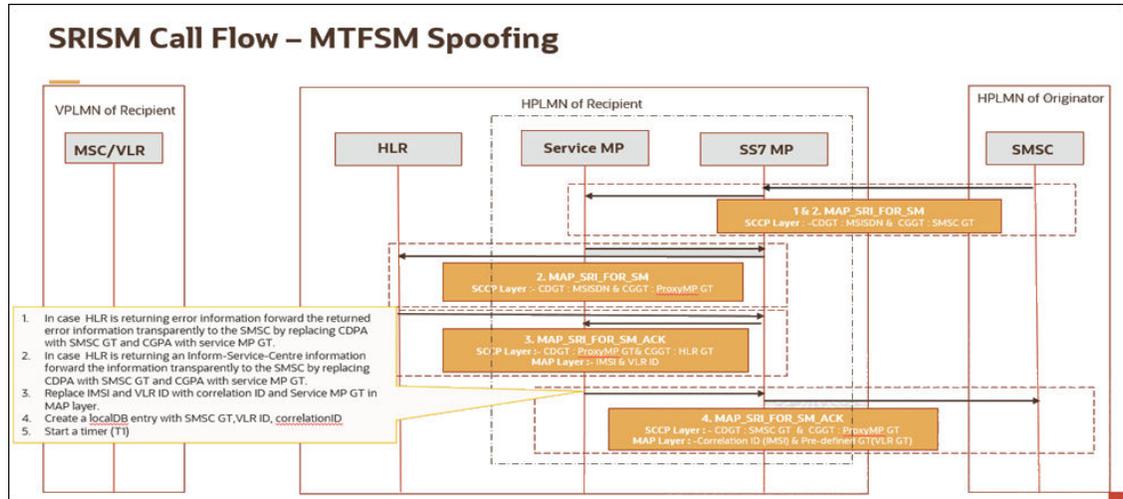
The SMS Home Router functionality provides a new service MP that enables vSTP to perform extra validations address the MT-Spoofing issue.

The call flow when MT Spoofing is enabled on Linkset :

1. Forward SRISM (Opcode 45) to vSTPServiceMP
  - Store CGPA (SMSC GT) in a local DB.
  - Modify SRISM CGPA with SMSPROXY GT.
  -
2. Route the modified SRISM to HLR.
3. SRISM\_ACK is received at vSTPServiceMP(GT routed).
  - Generate scrambled IMSI using random key.
  - Store actual IMSI, scrambled IMSI and MSC/VLR GT with the SMSC GT.
4. Modify SRISM\_ACK and send it to originating node.
  - IMSI replaced with scrambled IMSI
  - MSC/VLR GT replaced with one of the multiple SMSPROXY GT
  - Remove LMSI if present
  - CDPA is replaced with SMSC GT
  - CGPA is replaced with SMSPROXY GT

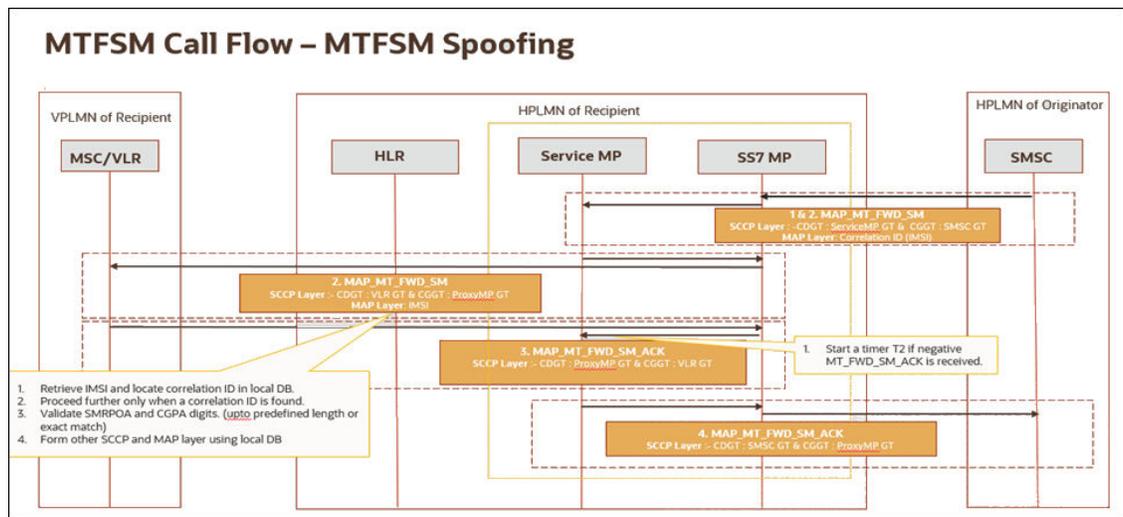
The following figure describes the MT FSM call flow:

Figure 2-23 MT Spoofing\_SRI SM



### 2.4.4.1.2.2 Handling MT Spoofing - MT\_FSM

A brief description of the call flow for MT\_FSM packet is available below:-



- SM-RP-DA with scrambled IMSI
- SM-RP-OA with fake/real GT of SMSC
- RP-OA with fake/real GT of SMSC
- TP-OA with fake/real A MSISDN
- Forward MT FSM to vSTPService MP(GT routed).
- Use scrambled IMSI to correlate with the stored information.
- Match SMSC GT from DB with the CGPA & MAP field.

Scrambled IMSI replaced with actual IMSI from DB

SMSPROXY GT replaced with actual VLR/MSC GT from DB.

The following figure describes the MT FSM call flow:

### 2.4.4.1.3 DOS Attack

A brief description of DOS Attack is as follows:

- MWI flag is enabled in the HLR and MSC after a failed delivery event or failed paging attempt to a specific subscriber. In the HLR, the MWI is set only by the MAP-REPORT-SM-DELIVERY-STATUS message from the SMSC.
- While MWI is set in the HLR, the HLR will NOT respond to any non-priority MAP-SEND-ROUTING-INFO-FOR-SM messages with MAP-INFORM-SERVICE-CENTRE.
- This prevents the SMSC to get routing information for the subscriber and to deliver the SMS using MT\_FSM •Abnormal use of the MAP-REPORT-SM-DELIVERY-STATUS Message is used to achieve a MT-SMS Denial of Service (DOS) attack on a specific or a set of customer.

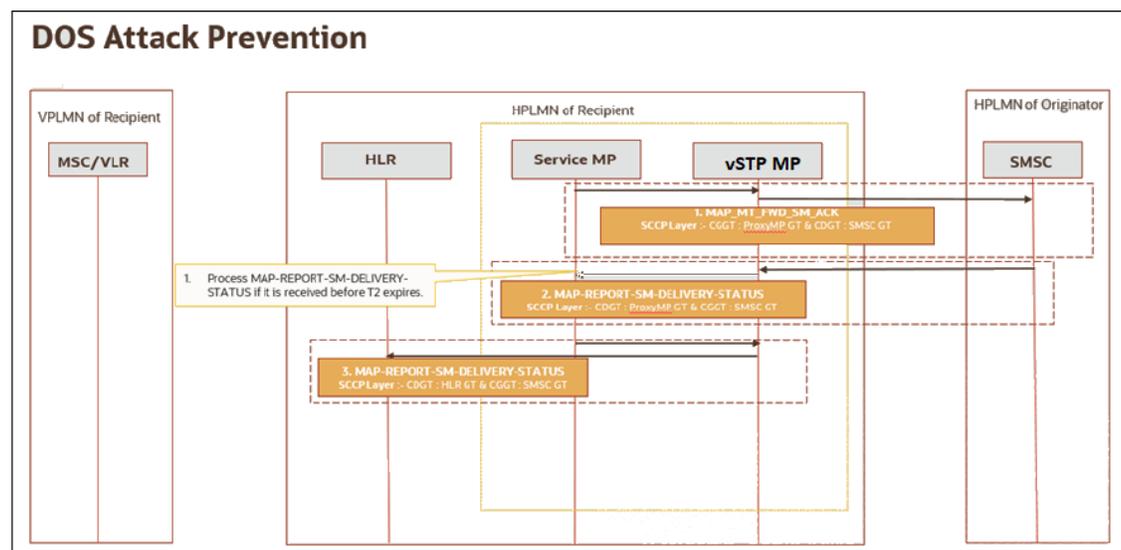
DOS Attack is handled by processing MAP\_DELIVERY\_STATUS\_REPORT on vSTPService MP. Extra validations are performed in order to handle the DOS attack.

### 2.4.4.1.3.1 Handling DOS Attack

A new service MP provides the ability to perform extra validation in order to address the DOS attack issue.

1. SMS SFW receives MT\_FSM and IMSI is searched in local DB
2. If the record is found in the DB, then the MT\_FSM record is created in local DB and forwarded to VLR ID stored in the DB. A timer starts after MT\_FSM is received at DB.
  - If the delivery status report is received before the timer expires, it is processed and sent to HLR. The MT\_FSM entry is deleted from local DB, once the timer expires.
  - If the delivery status report is received after timer is expired (No MT\_FSM entry is found in DB), then it is discarded.
3. If the record is not found in the DB, then the delivery status report is discarded.

The following figure describes the call flow to prevent DOS attacks:

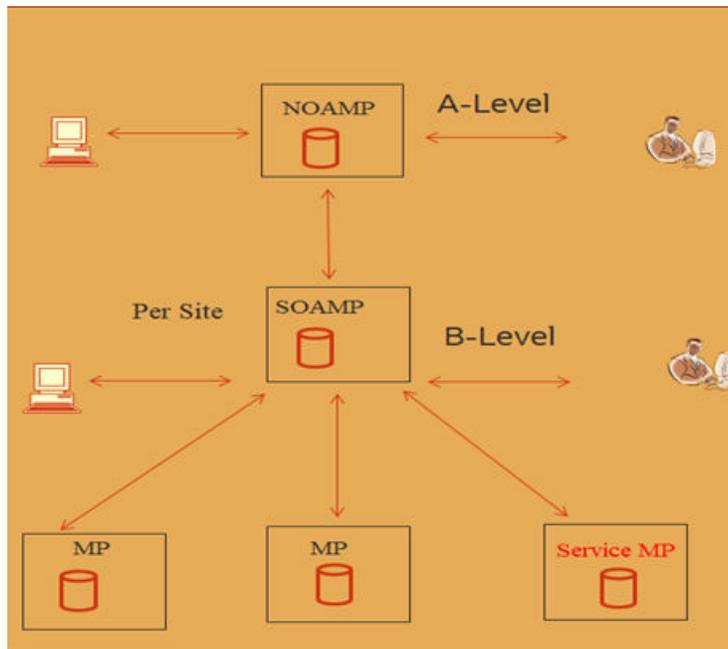


### 2.4.4.2 vSTP Architecture

To address the possible attacks, a service MP is introduced for SMS SFW functionality in vSTP. The vSTP MPs and ServiceMPs remain in same site (SO), but in different server groups. Hence, total MPs per SO is an addition of vSTP MPs and Service MPs.

The following figure shows the vSTP architecture with new service MP:

**Figure 2-24 vSTP Architecture**



In the architecture,

NOAM is the A-Level Server Group.

SOAM is the B-Level Server Group.

vSTP MP & Service MP are the C- level Server Group

There is an automatic ComAgt connection between vSTP MPs and Service MPs.

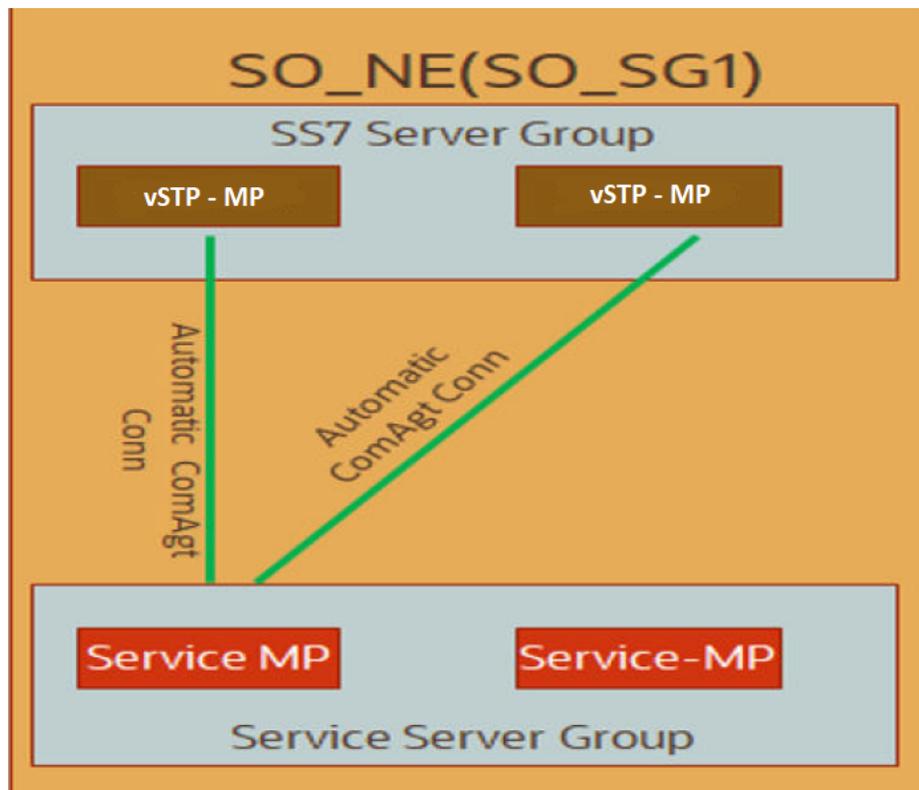
vSTPServiceVMs remain in HA Active/Standby Mode.

vSTPService process runs on all the ServiceVMs, so that traffic gets processed by all the Service VMs

### Processing on vSTP MP

The MSUs received at SS7 after the validation at vSTPService MP or after applying default action (fallback) on vSTPService MP, follows the GTT framework to route the packets, such as GTT selector and other GT and MNP related features. Applicable features are SMSNP/ FLOBR/TOBR/MBR.

The following figure describes the connection of vSTP with vSTP MPs



By default, the **vSTPService** process remains OFF. Turn On the **vstpService** process on the service MP to enable the Home SMS Router functionality. To enable the feature, run the following commands:

1. Run the `p1` command on the service MP.
2. Verify that **vstpService** process is not available.
3. Start the **vstpService** application on service MP.  
Run `pm.set` on `vstpService`.
4. Execute `p1` command to verify that `vstpService` process is running.

### 2.4.4.3 Feature Configurations

This section provides procedures to perform the vSTP SMS Home Router Security functionality.

The SMS Home Router is configured using the vSTP managed objects and vSTP GUI. The MMI API contains details about the URI, an example, and the parameters available for each managed object.

#### 2.4.4.3.1 MMI Managed Objects for SMS Home Router Support

MMI information associated with SMS Home Router support is accessed from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* gets opened, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for SMS Home Router support:

Managed Object Name	Supported Actions
Link Set	Insert, Update, Delete
Sccp Application	Insert, Update, Delete
Sccp Options	Update
Vstp SMS Proxy Options	Update
Vstp SMS Proxy SMSC Status	Insert, Update, Delete

### linkset - Insert, Update, Delete

Create a file with following content. File name could be anything, for example option name can be used as filename:

```
$cat linkset.json
{
    "asNotification": true,
    "asls8": false,
    "cgGtmod": false,
    "cgpnb1Set": "None",
    "configurationLevel": "18",
    "enableBroadcastException": true,
    "gnameSet": "Both",
    "gttmode": "Fcd",
    "islsrsb": 1,
    "ituTransferRestricted": false,
    "l2TimerSetName": "Default",
    "l3TimerSetName": "Default",
    "linksetAccMeasOption": "No",
    "localSignalingPointName": "lsp2",
    "maximumLinkTransactionsPerSecond": 1000,
    "name": "ls2",
    "numberSignalingLinkAllowedThreshold": 1,
    "numberSignalingLinkProhibitedThreshold": 1,
    "randsls": "Off",
    "remoteSignalingPointName": "rsp2",
    "reservedLinkTransactionsPerSecond": 1000,
    "routingContext": 0,
    "rsls8": false,
    "securityLogging": "Off",
    "slsci": false,
    "slsrsb": 1,
    "smsProxy": "Off",
    "type": "M3ua"
}
```

Execute the following command on Active SOAM to insert the data:

```
/vstp/linksets/ -v POST -r linkset.json
{
  "data": true,
  "links": {},
  "messages": [],
}
```

```
"status": true
}
```

#### Sample Output:

```
{
  "data": [
    {
      "asNotification": true,
      "asls8": false,
      "cgGtmod": false,
      "cgpnb1Set": "None",
      "configurationLevel": "18",
      "enableBroadcastException": true,
      "gnameSet": "Both",
      "gttmode": "Fcd",
      "islsrsb": 1,
      "ituTransferRestricted": false,
      "l2TimerSetName": "Default",
      "l3TimerSetName": "Default",
      "linksetAccMeasOption": "No",
      "localSignalingPointName": "lsp2",
      "maximumLinkTransactionsPerSecond": 1000,
      "name": "ls2",
      "numberSignalingLinkAllowedThreshold": 1,
      "numberSignalingLinkProhibitedThreshold": 1,
      "randsls": "Off",
      "remoteSignalingPointName": "rsp2",
      "reservedLinkTransactionsPerSecond": 1000,
      "routingContext": 0,
      "rsls8": false,
      "securityLogging": "Off",
      "slsci": false,
      "slsrsb": 1,
      "smsProxy": "On",
      "type": "M3ua"
    }
  ],
  "links": {},
  "messages": [],
  "status": true
}
```

### sccpapplication - Insert, Update, Delete

#### Attributes

Parameter	Value	Default	Description
appType	Eir, Atinp, Inpq, Sfapp, SmsProxy	-	Type of Application.

Create a file with following content. File name could be anything, for example option name can be used as filename:

```
$cat sccpapplication.json
{
    "appType": "SmsProxy",
    "ssn": 28
}
```

Execute the following command on Active SOAM to insert the data:

```
/vstp/sccpapplications/ -v POST -r sccpapplication.json
```

Sample Output:

```
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}
```

## sccpoptions – Update, Display

### Attributes

Parameter	Value	Default	Description
smsDelivery	On, Off	Off	SMS Proxy Delivery Functionality Status.
smsOrigination	On, Off	Off	SMS Proxy Origin Functionality Status.
smsTermination	On, Off	Off	SMS Proxy Terminate Functionality Status.

Update a file with following content. File name could be anything, for example option name can be used as filename:

```
$cat sccpoption.json
{
    "allowedFirstSegLen": 0,
    "alwMsgDuringRsemblyErr": false,
    "classlseq": "Disabled",
    "dfltfallback": false,
    "dfltgttmode": "Cd",
    "isSegXUDTfeatureEnable": false,
    "mtprgtt": "Off",
    "mtprgttfallback": "Mtproute",
    "reassemblyTimerDurationAnsi": 5000,
    "reassemblyTimerDurationItu": 10000,
    "segmentedMSULength": 200,
    "smsDelivery": "Off",
    "smsOrigination": "On",
}
```

```

    "smsTermination": "On",
    "tgtt0": "None",
    "tgtt1": "None",
    "tgttudtkey": "Mtp",
    "tgttxudtkey": "Mtp",
    "travelVelocity": 700
  }

```

Execute the following command on Active SOAM to insert the data:

```
/vstp/sccpoptions/ -v PUT -r sccpoption.json
```

Sample Output:

```

{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}

```

## VstpSmsProxyOptions – Update, Display

### Attributes

Parameter	Value	Default	Description
MofsmDfltActn	Fallback, Discard, Udts, Tcaperr	Discard	Default Action for MOFSM message validation failure.
MofsmErrCode	0-255	0	If Default action is Udts or TcapError, this error code is sent in response.
MtfsmDfltActn	Fallback, Discard, Udts, Tcaperr	Discard	Default Action for MT-FSM message validation failure.
MtfsmErrCode	0-255	0	If Default action is Udts or TcapError, this error code is sent in response.
MofsmScpval	On, Off	On	Whether to perform ScpVal for MO-FSM message.
MtfsmScpval	On, Off	On	Whether to perform ScpVal for MT-FSM message.
MtfsmInvkTimer	30-120	60	MT-FSM Timer. The MT-FSM should be received within this timer once the SRI-SM-Ack is sent to the originator.

Parameter	Value	Default	Description
SmdsDosTimer	30-120	60	Initiated after MTFSM is forwarded to the VLR. The SMS Delivery Status (if required) should be received before this timer expires.
SmsProxyGta	a-f,A-F,0-9 Min Len : 5 Max Len : 15	""	Global Title Address digits to identify the SMS Proxy Service.
SmsProxyTT	0-255	0	Translation type of CGPA to be used by the SMS Proxy service when generating Messages towards HLR.
ScmblImsiPrefix	a-f,A-F,0-9 Min Len : 5 Max Len : 10	""	Prefix Digits for the Scrambled IMSI. Also defines the range of Scrambled IMSIs to be used.
ScmblImsiLen	14-15	15	Total length of the IMSI to be sent as Scrambled IMSI in SRI-SM Ack.
Defcc	a-f,A-F,0-9, None Max Len : 4	-	Default country code.

Update a file with following content. File name could be anything, for example option name can be used as filename:

```
$cat smsproxyoptions.json
{
    "Defcc": "12",
    "MofsmDfltActn": "Udts",
    "MofsmErrCode": 2,
    "MofsmScpval": "On",
    "MtfsmDfltActn": "Tcaperr",
    "MtfsmErrCode": 0,
    "MtfsmInvkTimer": 120,
    "MtfsmScpval": "On",
    "ScmblImsiLen": 15,
    "ScmblImsiPrefix": "1234567",
    "SmdsDosTimer": 60,
    "SmsProxyGta": "12345678",
    "SmsProxyTT": 0
}
```

Execute the following command on Active SOAM to insert the data:

```
/vstp/smsproxyoptions/ -v PUT -r smsproxyoption.json
```

Sample Output:

```
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}
```

### VstpSMSProxySMSCStatus – Insert, Update, Display, Delete

#### Attributes

Parameter	Value	Default	Description
smscGttAddr	a-f, A-F, 0-9 Max Len : 21	-	Global Title Address of SMSCs to be allowlisted or blocklisted.
smscStatus	AllowList, BlockList	-	Indicates allowlist or blocklist status of SMSC.

Update a file with following content. File name could be anything, for example option name can be used as filename:

```
$cat smsproxystatus.json
{
  "smscGttAddr": "2234567891",
  "smscStatus": "ALLOWLIST"
}
```

Execute the following command on Active SOAM to insert the data:

```
/vstp/smscstatus -v POST -r smsproxystatus.json
```

Sample Output:

```
{
  "data": true,
  "links": {},
  "messages": [],
  "status": true
}
```

### 2.4.4.3.2 GUI Configurations for Home SMS Router Support

The SMS Home Router functionality can be configured from Active System OAM (SOAM). Select **VSTP**, and then **Configuration** page.

The following options are used to perform the configurations:

- SCCP Options
- Linkset
- SCCP Application

- SMS Proxy Options
- SMS Proxy SMSC Status

For more information, see GUI Configuration in *Oracle Communications vSTP User's Guide*.

## 2.4.4.4 Home SMS Router Alarms and Measurements

### Alarms and Events

The following table lists the alarms or events specific to the Home SMS Router functionality for vSTP:

Event ID	Event Name
70447	smsProxyValidationFailed
70448	smsProxyValRspTimeout
70449	smsProxyEcdError
70450	smsProxyDcdError
70452	smsProxyAllowlist
70453	smsProxyBlocklist
70454	smsProxySccpValidFail
70455	smsProxyMtfsmInvkTimeout
70456	smsProxyDosInvkTimeout
70446	VstpServiceStackEventQueueUtil
70451	serviceMpUnavailable

### Measurements

The following table lists the measurements specific to the Home SMS Router support for vSTP:

Measurement ID	Measurement Name
22180	VstpSMSProxyMOSMTx
22181	VstpSMSProxyMOSMTxFail
22182	VstpSMSProxyMOSMRx
22183	VstpSMSProxyMOSMValSuc
22184	VstpSMSProxyMOSMValFail
22185	VstpSMSProxyMOSMMsgDiscard
22186	VstpSMSProxyDecodingFail
22187	VstpSMSProxyEncodingFail
22188	VstpSMSProxyInternalError
22189	VstpSMSProxyPduFull
22190	VstpSMSProxyMOSMTxPeak
22191	VstpSMSProxyMOSMTxAvg
22192	VstpSMSProxyMOSMRxPeak
22193	VstpSMSProxyMOSMRxAvg
22194	VstpServiceStackQueuePeak
22195	VstpServiceStackQueueAvg
22196	VstpServiceStackQueueFull
22197	VstpSMSProxyMOSMAllowlist
22198	VstpSMSProxyMOSMBlocklist
22199	VstpSMSProxyMOSscpValFail

Measurement ID	Measurement Name
22200	VstpSMSPProxyMOSccpValSucc
22201	VstpSMSPProxyMOTx
22202	VstpSMSPProxyMORx
22203	VstpSMSPProxyMOValSuc
22204	VstpSMSPProxyMOValFail
22205	VstpSMSPProxyMOAllowlist
22206	VstpSMSPProxyMOBlocklist
22207	VstpSMSPProxyMOSccpValFail
22208	VstpSMSPProxyMOSccpValSucc
22210	VstpMOSMTxtoServiceMp
22211	VstpMTSMTxtoServiceMp
22212	VstpSRISMTxtoServiceMp
22213	VstpSMSPProxyMTSMTx
22214	VstpSMSPProxyMTSMTxFail
22215	VstpSMSPProxyMTSMRx
22216	VstpSMSPProxySRISMTx
22217	VstpSMSPProxySRISMRx
22218	VstpSMSPProxyMTSMMsgDiscard
22219	VstpSMSPProxyMTSMTxPeak
22220	VstpSMSPProxyMTSMTxAvg
22221	VstpSMSPProxyMTSMRxPeak
22222	VstpSMSPProxyMTSMRxAvg
22223	VstpSMSPProxyMTSMInvkTimeout
22224	VstpSMSPProxyMTSMBlocklist
22225	VstpSMSPProxyMTSMValFail
22226	VstpSMSPProxySRISMBlocklist
22227	VstpSMSPProxyMTSMSccpValFail
22228	VstpSMSPProxyMTSMSccpValSucc
22229	VstpSMSPProxyMTTx
22230	VstpSMSPProxyMTRx
22231	VstpSMSPProxySRITx
22232	VstpSMSPProxySRIRx
22233	VstpSMSPProxyMTAllowlist
22234	VstpSMSPProxyMTBlocklist
22235	VstpSMSPProxyMTValFail
22236	VstpSMSPProxySRIBlocklist
22237	VstpSMSPProxyMTSccpValFail
22238	VstpSMSPProxyMTSccpValSucc
22239	VstpSMSPProxyMsgDiscard
22240	VstpSMSPProxySRISMSucc
22241	VstpSMSPProxyMTSMSSucc
22242	VstpACKTxtoServiceMp
22243	VstpSMSPProxyMOSMMMsgFallback
22244	VstpSMSPProxyMTSMMMsgFallback
22245	VstpSMSPProxyDRTx
22246	VstpSMSPProxyDRRx

Measurement ID	Measurement Name
22247	VstpSMSProxyDELRPSMTx
22248	VstpSMSProxyDELRPSMRx
22249	VstpDELREPTxttoServiceMp
22250	VstpSMSProxyMTNACKTimeout
22251	VstpSMSProxyDelRepMsgDiscard
22252	VstpSMSProxyDELRPValFail

For more details related to measurements, refer to *Diameter Signaling Router Measurement Reference* document.

### 2.4.4.5 Troubleshooting

In case of the error scenarios, the measurements specific to Home SMS Router feature are pegged. For information related to Home SMS Router measurements, see [Home SMS Router Alarms and Measurements](#).

### 2.4.4.6 Dependencies

The Home SMS Router support for vSTP has no dependency on any other vSTP operation.

**Note**

For TC-Continue, no message processing is supported on the Service VM.

# 3

## GSMA Categorization

For protection against attacks, a comprehensive approach to information security is taken. The security of a signaling network is analyzed, which allows detection of current vulnerabilities in the network and helps in assessing information security risks. The possible attacks are described in the Appendix sections of this document.

GSMA recommendations specify the use of a monitoring system, which can perform analysis in real time. This enables detecting phishing or anomalies in a network at an early stage.

This section describes the different recommended details of SS7 security aspects for vSTP, in order to counter the potential SS7 attacks.

### 3.1 Supported Message Categories

#### 3.1.1 Category 1

This category includes messages that should only be received from within the same network and/or are unauthorized at interconnect level, and should not be sent between operators unless there is an explicit bilateral agreement between the operators to do so.

To handle the category 1 vulnerabilities, you must have the opcodes listed in [Table 3-1](#). These opcodes ensures blocking of MAP messages that are for intra-PLMN use only.

**Table 3-1 Message Category 1**

opCode	Originating SSN	Destination SSN	Description
provideRoamingNumber		MSC	This opcode is used in FLOBR/TOBR feature. when received from a VLR, the MAP packet can be used to pass information associated with the handover of active calls. Where, a network does not support handover across network boundaries, these messages gets blocked.
sendParameters	VLR	HLR	This opcode is used in FLOBR/TOBR feature. When addressed to a VLR, the MAP packet can be used to pass information associated with the handover of active calls. These MAP messages gets blocked, where a network does not support the handover across network boundaries.

Table 3-1 (Cont.) Message Category 1

opCode	Originating SSN	Destination SSN	Description
insertSubscriberData	HLR	VLR	This opcode is used in FLOBR/TOBR feature. When addressed to a VLR, the MAP packet can be used as part of the CUG and GroupCall services. Where CUG and GroupCall services are not supported across network boundaries, these MAP messages gets blocked.
registerSS	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
eraseSS	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
activateSS	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
deactivateSS	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
interrogateSS	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
registerPassword	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
getPassword	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
processUnstructuredSS-Data	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
processUnstructuredSS-Request	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
unstructuredSS-Request	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.

**Table 3-1 (Cont.) Message Category 1**

opCode	Originating SSN	Destination SSN	Description
unstructuredS S-Notify	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
readyForSM	MSC	VLR	This opcode is used in FLOBR/TOBR feature. Block MAP readyForSM packets.

### 3.1.2 Category 2

This category includes messages that should only be received from visiting subscribers home network. These should normally only be received from an inbound roamer's home network.

To handle the category 2 vulnerabilities, you must have the opcodes listed in [Table 3-2](#). These opcodes ensures the following:

- Blocking all messages from home-PLMN (messages where target IMSI is using the MCC+MNC of the own network.
- Blocking all messages from home-PLMN for inbound roamers, where `OperatorID` referenced by all parameters in MAP and `OperatorID` in CgPA do not match.
- Blocking the messages for which the `HLRid` is not consistent with the CgPA.

Following is the list of vulnerable category 2 opcodes:

**Table 3-2 Message Category 2**

opCode	Originating SSN	Destination SSN	Description
provideRoamingNumber	HLR	MSC	This opcode is used in FLOBR/TOBR feature. when received from an HLR, the MAP message is screened for any Category 2 vulnerabilities. Blocks the received HLR-to VLR provideRoamingNumber packets.
insertSubscriberData	HLR	MSC SGSN	This opcode is used in FLOBR/TOBR feature. Blocks the insertSubscriberData packets with application context values 16, 1, or 32.
mt-forwardSM	MSC	MSC	This opcode is used in FLOBR/TOBR feature.
getPassword	HLR	MSC	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
reset			

**Table 3-2 (Cont.) Message Category 2**

opCode	Originating SSN	Destination SSN	Description
unstructuredS S-Request	HLR	MSC SGSN	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
unstructuredS S-Notify	HLR	MSC SGSN	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.

### 3.1.3 Category 3

This category includes messages that should only be received from the subscriber's visited network. Specifically, MAP packets that are authorized to be sent on interconnects between mobile operators.

To handle the category 3 vulnerabilities, you must have the opcodes listed in [Message Category 3](#). These opcodes ensures the following:

- Blocking of messages in relation to outbound roaming subscribers, where MCC+MNC of IMSI and CdPA or prefix ID of the HLR do not match
- Blocking of messages in relation to outbound roaming subscribers, where VLR Id and CgPA do not match.

Following is the list of vulnerable category 3 opcodes:

**Table 3-3 Message Category 3**

opCode	Originating SSN	Destination SSN	Description
sendParameters	SGSN	HLR	This opcode is used in FLOBR/TOBR feature. When addressed to an HLR, the MAP message can be used to request authentication vectors and subscriber data.
registerSS	HLR	VLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
eraseSS	HLR	VLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
activateSS	HLR	VLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.

Table 3-3 (Cont.) Message Category 3

opCode	Originating SSN	Destination SSN	Description
deactivateSS	HLR	VLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
interrogateSS	HLR	VLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
registerPassw ord	HLR	VLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
processUnstru cturedSS-Data	HLR	VLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.
readyForSM	MSC	HLR	This opcode is used in FLOBR/TOBR feature. Block MAP readyForSM packets when send to a HLR.
mt-forwardSM	MSC	SGSN	This opcode is used in FLOBR/TOBR feature.
updateLocatio n	HLR	VLR	This opcode is used in Stateful Security Support (Velocity Check) features. Block inbound map packets if the received VLR or SGSN address in the CgPA is not reasonable compared with the last known location.
updateGprsLoc ation	HLR	VLR	This opcode is used in Stateful Security Support (Velocity Check) features. Block inbound map packets if the received VLR or SGSN address in the CgPA is not reasonable compared with the last known location.
sendAuthentic ation	HLR	VLR	This opcode is used in Stateful Security Support (Velocity Check) features. Block inbound map packets if the received VLR or SGSN address in the CgPA is not reasonable compared with the last known location.
processUnstru cturedSS- Request	VLR	HLR	This opcode is used in FLOBR/ToBR and Stateful Security Support (VLR Validation) features. It blocks the SS related MAP packets.

# 4

## Security Logging and Visualization

vSTP provides the SS7 Firewall Logging support. The logging support offers a holistic view of all the interconnect transactions and helps identify possible threats.

The logging data can be exported to an external server. It can also be ingested and analyzed in any 3rd party analytics or visualization platform.

### 4.1 Feature Description

The vSTP Security Logging feature generates and sends log messages from the vSTP MPs to an external visualization server. vSTP can export the feature logs for defined variables and generated logs to an external platform for analysis and visualization.

In addition, it is important to note the following points with respect to the Security Logging functionality:

- Support for 10k basic GTT traffic logging per MP
- Support for 2.5k SFAPP traffic logging per MP
- Support for 50K message logging per site.

#### 4.1.1 Overview

vSTP Security Logging generates and sends log messages from the SCCP and SFAPP applications to an external visualization server. The exported log messages are in `.csv` format, which can be ingested, stored and visualized by an external analytics tool. The logging is divided into two tasks:

- **SCCP/SFAPP Task:** This task includes:
  - Copying all the required fields in logging event in the format as present on vSTP
  - Sending the logging event to the logging task
- **Logging Task:** This task includes:
  - Fetching data from logging event
  - Performing data transformation, filling location information and category type
  - Writing the data in a `.csv` file and transferring the files to an external server

#### **Logging Rate and TPS supported per vSTP MP:**

- Support for 10k basic GTT traffic logging per MP
- Support for 2.5k SFAPP traffic logging per MP
- Support for 50K message logging per site

#### 4.1.2 Supported Operation Codes

The following lists define the supported Operation Codes (OpCodes) with vSTP Security Logging.

The category includes messages that should only be received from within the same network and/or are unauthorized at interconnect level and should not be sent between operators unless there is an explicit bilateral agreement between the operators to do so.

### Category 1

This category includes messages that should only be received from within the same network and/or are unauthorized at interconnect level and should not be sent between operators unless there is an explicit bilateral agreement between the operators to do so.

Following is the list of vulnerable category 1 opcodes:

- provideRoamingNumber
- sendParameters
- registerSS
- eraseSS
- activateSS
- deactivateSS
- interrogateSS
- registerPassword
- getPassword
- processUnstructuredSS-Data
- sendRoutingInfo
- sendRoutingInfoForGprs
- sendIdentification
- sendIMSI
- processUnstructuredSS-Request
- unstructuredSS-Request
- unstructuredSS-Notify
- anyTimeModification
- anyTimeInterrogation
- sendRoutingInfoForLCS
- subscriberLocationReport

### Category 2

This category includes messages received from visiting subscribers' home networks. These messages are typically received from an inbound roamer's home network.

Following is the list of vulnerable category 2 opcodes:

- provideRoamingNumber
- provideSubscriberInfo
- provideSubscriberLocation
- insertSubscriberData
- deleteSubscriberData

- cancelLocation
- getPassword
- reset
- unstructuredSS-Request
- unstructuredSS-Notify
- informServiceCentre

### Category 3

This category of messages is received from the subscriber's visited network. Specifically, MAP packets authorized to be sent on interconnect between mobile operators.

Following is the list of vulnerable category 3 opcodes:

- updateLocation
- updateGprsLocation
- sendParameters
- registerSS
- eraseSS
- activateSS
- deactivateSS
- interrogateSS
- registerPassword
- processUnstructuredSS-Data
- mo-forwardSM
- mt-forwardSM
- beginSubscriberActivity
- restoreData
- processUnstructuredSS-Request
- purgeMS
- sendRoutingInfoForSM
- sendAuthenticationInfo
- reportSmDeliveryStatus
- NoteMM-Event

## 4.2 Feature Configuration

### 4.2.1 MMI Managed Objects for Security Logging

MMI information associated with Security Logging support is accessed from a DSR NOAM or SOAM from **Main Menu**, and then **MMI API Guide**.

Once the *MMI API Guide* gets opened, use the application navigation to locate specific vSTP managed object information.

The following table lists the managed objects and operations supported for security logging.

**Table 4-1 Security Logging support Managed Objects and Supported Operations**

Managed Object Name	Supported Operations
linksets	Insert, Update, Delete
securitylogconfig	Update

### linksets

For this feature, the `securityLogging` parameter is added to the linkset MO.

The allowed values for this parameter with their interpretation are:

- **OFF:** No Logging will be done when traffic is run through the linkset.
- **ALL:** Logging of all messages on the particular linkset will be done.
- **RISKY:** Logging of only risky opcode messages coming on that linkset will be done.

The example output for Display of linkset MO:

```
{
    "asNotification": true,
    "asls8": false,
    "cgGtmod": false,
    "configurationLevel": "32",
    "enableBroadcastException": true,
    "gttmode": "Fcd",
    "islsrsb": 1,
    "ituTransferRestricted": false,
    "l2TimerSetName": "Default",
    "l3TimerSetName": "Default",
    "linkTransactionsPerSecond": 10000,
    "linksetAccMeasOption": "No",
    "localSignalingPointName": "LSP1",
    "name": "Linkset777",
    "numberSignalingLinkAllowedThreshold": 1,
    "numberSignalingLinkProhibitedThreshold": 1,
    "randsls": "Off",
    "remoteSignalingPointName": "RSP777",
    "routingContext": 8,
    "rsls8": false,
    "securityLogging": "All",
    "slsci": false,
    "slsrsb": 1,
    "type": "M3ua"
}
```

### securitylogconfig

The **securitylogconfig** MO manages all the attributes essential for Security Logging support. The following table describes these parameters:

Table 4-2 securitylogconfig MO Parameters

Parameter Name	Description
securityLoggingFeature	This is the global parameter for this feature. Users have to enable this parameter before configuring the <code>securityLogging</code> parameter for linkset. When disabled, there is no logging on that linkset. Also, the other parameters for this MO can only be modified after disabling this parameter. Allowed values: On, Off
siteIdentifier	This parameter identifies the logging site. The value entered here will be logged in the .CSV logs formed and can be used to identify the logging site. Allowed values: Alphanumeric characters of maximum length 20
logMpDirPath	The path at MP, where the user wants to form temporarily form .CSV logs before they are transferred to SOAM. Example: <code>/var/TKLC/db/filemgmt/securityLog</code>
logFileTimeout	The maximum time interval in seconds until which the MP waits before starting to open new .CSV log files. Allowed Values: Integer values from 60-120
maxLogsPerFile	Maximum messages to be logged in a single .CSV log file before closing it and beginning a new one for logging. Allowed Values: Integer values from 600000 up to 3000000
minDiskSpaceForLogging	Minimum disk space required for logging as % of available disk space in filemanagement area. An alarm is raised if available disk space is below the configured % value. Allowed Values: Integer values from 10 up to 100

The example output for Display of securitylogconfig MO:

```
{
  "logFileTimeout": 90,
  "logMpDirPath": "/var/TKLC/db/filemgmt/
securityLog",
  "maxLogsPerFile": 1500000,
  "minDiskSpaceForLogging": 30,
  "securityLoggingFeature": "On",
  "siteIdentifier": "ABC"
}
```

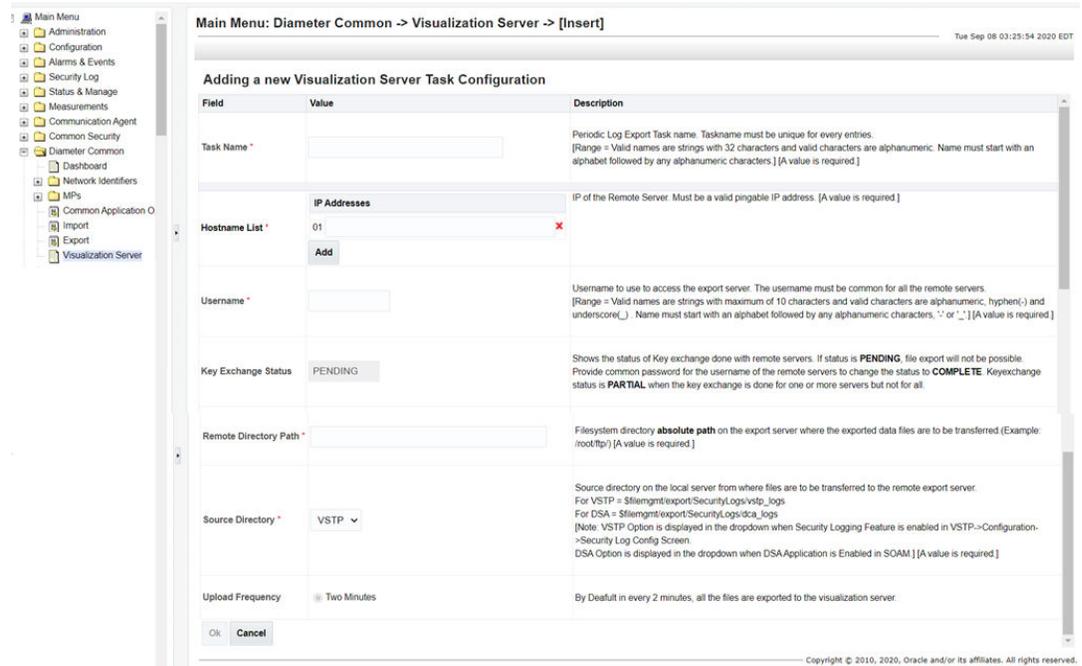
## 4.2.2 GUI Configuration

The Security Logging functionality can be configured from Active System OAM (SOAM) using the following steps:

1. On the Active System OAM (SOAM), select **VSTP**, navigate to **Configuration** and click **Security Log Config**.

2. On the **Security Log Config** page perform the configurations that governs the functionality of security logging in the file directory of SOAM. For more details, refer to Security Log Config section in *vSTP User's Guide*.
3. On the Active System OAM (SOAM), select **Diameter Common** click **Visualization Server**.

Figure 4-1 Visualization Server Page



The following table describes the key parameters on this page:

Table 4-3 Visualization Server Parameter Description

Parameter	Description	Allowed Values
Task Name	Name of the task.	Alphanumeric Characters of maximum length 32
Hostname List	IPv4 addresses of Remote Server for Log transfer.	Maximum of 8 remote servers can be configured.
Username	Username to access remote server.	Alphanumeric Character words of maximum length 10
Key Exchange Status	Shows the keyexchange status for the remote servers with SO. This field cannot be edited.	

**Table 4-3 (Cont.) Visualization Server Parameter Description**

Parameter	Description	Allowed Values
Source Directory	Name of the source directory.	VSTP or DSA <b>Note:</b> The VSTP Option is displayed in the dropdown when Security Logging Feature is enabled in VSTP using the option <b>VSTP</b> by selecting <b>Configuration</b> and clicking on <b>Security Log Config</b> GUI page.
Upload Frequency	Time interval between which logs are exported from SOAM to Remote Server. This field cannot be edited.	

Use this page to configure IP Addresses (IPv4) of remote servers and perform SSH Keyexchange of the SO with the Remote servers so that the export of logs (.CSV) happens without hassle. The remote server must have a common username and password combination, as the GUI screen allows a single username for all the remote servers.

After filling all the required details in the GUI Screen and performing SSH Keyexchange, the log files present at the source directory of SOAM are moved to the destination directory of remote server every 2 minutes time interval.

The page supports Insert, Edit, Delete, and SSH Key exchange operations.

- This completes the Security Logging feature configurations for vSTP.

## 4.3 Alarms and Measurement

### Alarms

The following table lists the measurements specific to the Security Logging and Visualization support for vSTP:

Alarm ID	Alarm Name
70437	VstpSecuLogEventQueue
70438	VstpSecuLogErro
70439	VstpSecuLogFetchError
70440	VstpSecuLogRemoteServerError

For more details related to Alarms, refer to *Alarms and KPIs Guidelines* document.

### Measurements

The following table lists the measurements specific to the Security Logging and Visualization support for vSTP:

Measurement ID	Measurement Name
21977	VstpSecuLogDiscQueueFull
21978	VstpSecuLogQueuePeak
21979	VstpSecuLogQueueAvg

Measurement ID	Measurement Name
21980	VstpSecuLogRate
21981	VstpSecuLogRatePeak
21982	VstpSecuLogRateAvg

For more details related to measurements, refer to *Measurement Reference Guide*.

## 4.4 Troubleshooting

In case of the error scenarios, the measurements specific to Security Logging and Visualization feature are pegged. For information related to CAT2 SS7 Security measurements, see [Alarms and Measurement](#).

## 4.5 Dependencies

The Security Logging and Visualization feature for vSTP has no dependency on any other vSTP operation.

Consider the following points while configuring this feature:

- If MP crashes and does not come up, then the log files present on that MP get lost.
- The VM profile does not have space to store logs at 30 minutes on SOAM at 50K site TPS. Hence if the transfer of logs to the remote server fails, logging may stop due to low disk space.

# A

## Subscriber Information Disclosure

Following is the type of information that could be disclosed in a subscriber information disclosure attack:

- IMSI disclosure
- Subscriber location discovery
- Disclosure of subscriber profile information
- Cryptographic material retrieval
- Call details gathering

To obtain routing information about a subscriber during an incoming voice call, the `SendRoutingInfo` message is used. It must be transmitted only within the operator's home network.

To determine a subscriber's location, the `ProvideSubscriberInfo` message is used.

# B

## Network Information Disclosure

Network information disclosure is fraught with the leakage of SS7 network configuration data.

To obtain the relevant information, the following two messages are used:

- `AnyTimeInterrogation`
- `SendRoutingInfo`

Both of the messages allow network information disclosure.

# C

## Subscriber Traffic Interception

Following are the types of attacks in a subscriber traffic interception:

- Call redirection with interception
- SM interception/monitoring

The message `UpdateLocation` is used to inform the HLR about a change in a mobile switch. Terminating SMSs or calls are intercepted by sending a fake request to register a subscriber in an intruder's network. When a terminating call is received, the operator's network sends a request to a fake network to obtain the subscriber's roaming number. An attacker can send the number of their telephone exchange in response, and the incoming traffic will be transmitted to the attacker's equipment. After sending another request to register the subscriber in the real network, the attacker can redirect the call to the subscriber's number. As a result, the conversation will pass through the equipment controlled by the attacker.

The same principle is used for the interception of terminating calls via `RegisterSS`. However, in such a case, terminating calls are unconditionally redirected to the intruder's telephone exchange.

Originating calls are tapped by using a similar pattern. The `InsertSubscriberData` message replaces the address of the billing platform in the subscriber's profile stored in the VLR database. When a request is sent to the changed address, the attacker first redirects the originating call to their equipment and then redirects it to the called subscriber. Therefore, the attacker can tap any conversation of the subscriber.

# D

## Fraud

Following are the categories into which a fraud can be classified:

- Illegitimate redirection of terminating or originating calls
- USSD request manipulation
- SMS message manipulation or spoofing
- Subscriber profile modification or spoofing
- Online charging evasion

### D.1 Illegitimate Redirection of Terminating or Originating Calls

An attacker can redirect voice calls of subscribers to premium-rate numbers or to a third-party number. The call will be paid by the subscriber when establishing unconditional redirection, or by the operator when the subscriber is registered in a fake network and the subscriber's roaming number is spoofed.

Calls are redirected by using `UpdateLocation`, `RegisterSS`, `InsertSubscriberData` as well as by using `AnyTimeModification` that allows making changes to a subscriber.

### D.2 USSD Request Manipulation

An attacker can transfer money from the account of a subscriber or an operator's partners by sending fake USSD requests using the `ProcessUnstructuredSSRequest` message. Also, `UnstructuredSSNotify` is used to send notifications to subscribers from various services and the operator.

An attacker can send a fake notification on behalf of a trusted service containing instructions for the subscriber. That may include sending an SMS message to a paid number to subscribe to a service, calling a fake bank number due to suspicious transactions, or following a link to update an application.

### D.3 SMS Message Manipulation or Spoofing

Phishing or ad messages can be sent on behalf of arbitrary subscribers or services using the `MT-ForwardSM` and the `MO-ForwardSM` methods.

`MT-ForwardSM` is designed for delivering incoming messages and can be used by attackers to generate forged incoming SMS messages. Unauthorized usage of `MO-ForwardSM` allows sending messages from subscribers at their expense.

### D.4 Subscriber Profile Modification or Spoofing

A subscriber's profile stores data about the billing platform and service subscriptions. To bypass a billing system in real time, it is necessary to delete the subscriber's `O-CSI` subscription, which is used to make originating calls or to substitute the billing system address.

In order to prevent non-fare calls, O-CSI parameters imply that the call must be terminated if the billing platform is unavailable. However, this parameter can be changed, so that the call continues without addressing the platform. As a result, the legitimate platform does not receive information about the calls, and they are not billed.

## D.5 Denial of Service

Following are the types of attacks in a denial of service attack:

- Service unavailability for subscriber
- Recourses depletion

If the VLR address where the subscriber is currently registered is removed from the HLR via `PurgeMS` initiated by a certain third-party host, terminating calls cannot be routed to the subscriber's VLR/MSC. The reason is that there is no registration address in the HLR. In such a case, originating calls are available for the subscriber because the registration record in the VLR is not changed.

Rebooting the device does not help to restore the record in the HLR, because the VLR does not initiate the `UpdateLocation` procedure, assuming that there are no changes in the subscriber's registration data.

It is possible to restore the registration record and the subscriber's availability only by registering in the coverage area of another serving MSC. For example, first manually selecting the network of another operator and then selecting the home network again. Another method is to move to another MSC of the home network.

# E

## Manually Adding SGSN Field in UDR

To manually add SGSN field in UDR, perform the following steps:

1. Go to **Subscriber Entity configuration**, and then **Transparent Entity**, and then **Base Field**. Then, select **SprProfileBFS** base field set name and click **Edit**.
2. Add SGSN field.

16	MMER_OLD	RegEx		0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>			0	1	^[x20-xxi
17	MMEH_OLD	RegEx		0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>			0	1	^[x20-xxi
18	HSSR_OLD	RegEx		0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>			0	1	^[x20-xxi
19	HSSH_OLD	RegEx		0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>			0	1	^[x20-xxi
20	SGSN	RegEx		0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>			0	1	^[x20-xxi