



# Locator

## Руководство пользователя

---

#### **Авторские права**

Без предварительного письменного разрешения, полученного от ООО «НТЦ ПРОТЕЙ», документ и любые выдержки из него, с изменениями и переводом на другие языки, не должны быть воспроизведены или использованы.

---

## Содержание

<b>1</b>	<b>ТЕРМИНЫ И СОКРАЩЕНИЯ .....</b>	<b>5</b>
<b>2</b>	<b>ОБЩИЕ СВЕДЕНИЯ.....</b>	<b>10</b>
2.1	НАЗНАЧЕНИЕ ДОКУМЕНТА.....	10
2.2	СОСТАВ ДОКУМЕНТА .....	10
2.3	ТЕХНИЧЕСКАЯ ПОДДЕРЖКА.....	11
2.3.1	Производитель.....	11
2.3.2	Служба технической поддержки .....	11
2.4	ИСТОРИЯ ИЗМЕНЕНИЙ .....	12
<b>3</b>	<b>ОПИСАНИЕ МОДУЛЯ .....</b>	<b>13</b>
3.1	ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ .....	13
3.2	ВКЛЮЧЕНИЕ МОДУЛЯ LOCATOR В СЕТЬ ОПЕРАТОРА.....	14
3.3	ПОЛУЧЕНИЕ МЕСТОПОЛОЖЕНИЯ АБОНЕНТА.....	15
3.4	ПОЛУЧЕНИЕ IMEI.....	22
3.5	API ИНТЕРФЕЙС ДЛЯ РАБОТЫ С ВНЕШНИМИ ПОДСИСТЕМАМИ .....	22
3.5.1	Получение текущего местоположения.....	23
3.5.2	Получение текущего IMEI .....	26
3.5.3	Подписка на смену местоположения абонента .....	27
3.5.4	Подписка на смену IMEI абонента.....	34
3.5.5	Отмена подписки на смену местоположения абонента .....	37
3.5.6	Отмена подписки на IMEI абонента .....	39
3.6	ПОВЫШЕНИЕ НАДЕЖНОСТИ И ОРГАНИЗАЦИЯ РЕЗЕРВИРОВАНИЯ.....	39
<b>4</b>	<b>НАСТРОЙКА ПАРАМЕТРОВ КОНФИГУРАЦИИ .....</b>	<b>43</b>
4.1	УСЛОВНЫЕ ОБОЗНАЧЕНИЯ.....	43
4.2	НАСТРОЙКА ПАРАМЕТРОВ HTTP-СОЕДИНЕНИЯ (HTTP.CFG).....	44
4.2.1	Конфигурация SSL.....	47
4.2.2	Конфигурация авторизации Authorization .....	48
4.3	НАСТРОЙКА КОНФИГУРАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (LOCATOR.CFG) .....	48
4.3.1	Параметры подсчета статистики сигнальных сообщений TrafficStatistics .....	50
4.4	НАСТРОЙКА ПАРАМЕТРОВ OMI-ПОДКЛЮЧЕНИЙ (OM_INTERFACE.CFG).....	51
4.4.1	Конфигурация временных интервалов Timers .....	55
4.4.2	Конфигурация сетевой логики ConnectionLogics .....	56
4.4.3	Конфигурация сокет-интерфейса Sockets.....	56
4.4.4	Конфигурация направлений Directions .....	57
4.5	НАСТРОЙКА ПАРАМЕТРОВ СИСТЕМЫ ЖУРНАЛИРОВАНИЯ (TRACE.CFG) .....	58
4.5.1	Модификаторы task.....	64
4.5.2	Модификаторы period.....	65
4.5.3	Модификаторы buffering.....	65
4.5.4	Модификаторы type.....	66
4.6	НАСТРОЙКА ПАРАМЕТРОВ МАРШРУТИЗАЦИИ SCCP СООБЩЕНИЙ (SCCP_ROUTING.CFG).....	66
<b>5</b>	<b>ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ СИСТЕМЫ LOCATOR .....</b>	<b>68</b>
5.1	ДОСТУП С ИСПОЛЬЗОВАНИЕМ КОМАНДЫ TELNET .....	68
5.2	УПРАВЛЕНИЕ РАБОТОЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	69
5.3	ПЕРЕЗАПУСК СИСТЕМЫ .....	69
5.4	ПРОВЕРКА РАБОТОСПОСОБНОСТИ .....	69
5.4.1	Диагностика системы.....	69
5.4.2	Диагностика работы подсистемы M3UA.....	70
5.4.3	Диагностика взаимодействия с GTP-Probe .....	70
5.4.4	Диагностика трафика с M3UA.....	71
5.4.5	Проверка работы приложений .....	72
<b>6</b>	<b>СТАТИСТИКА.....</b>	<b>73</b>
6.1	СТАТИСТИЧЕСКАЯ ИНФОРМАЦИЯ ПО ТРАФИКУ .....	73

<b>7</b>	<b>ЖУРНАЛЫ .....</b>	<b>75</b>
7.1	ОБЩИЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ ЖУРНАЛОВ .....	75
7.1.1	Стандартный вывод и журнал скриптовой подсистемы .....	76
7.1.2	Базовые журналы приложения .....	78
7.1.3	Журналы подсистемы M3UA .....	84
7.1.4	Журналы подсистемы TCAP .....	86
7.1.5	Журналы подсистемы SCCP .....	89
7.1.6	Журналы подсистемы M2PA .....	91
7.1.7	Журналы подсистемы HTTP .....	93
7.1.8	Журналы подсистемы OM-interface .....	95
<b>8</b>	<b>АВАРИИ .....</b>	<b>99</b>
8.1	ОБЩИЕ АВАРИИ .....	99
8.2	АВАРИИ SIGTRAN .....	99
8.2.1	Аварии для Sg.SIGTRAN.M3UA.AS .....	99
8.2.2	Аварии для Sg.SIGTRAN.M3UA.ASP .....	100
8.3	АВАРИИ OMI-ИНТЕРФЕЙСА (OMI) .....	102

## 1 Термины и сокращения

В таблице ниже приведены используемые в настоящем документе термины и сокращения.

Таблица 1 — Используемые термины и сокращения

Термин	Описание
APC	Affected Point Code, код затронутой точки
API	Application Programming Interface, интерфейс программирования приложений
AS	Application Server, сервер приложений
ASP	Application Server Provider, провайдер сервисов приложений
ASPDW	Выключение ASP
ASPAC	ASP Active, активное состояние ASP
ASPIA	ASP InActive, неактивное состояние ASP
ASPUP	включение ASP
ATI	Any Time Interogation, MAP-запрос
CAMEL	Customized Applications for Mobile network Enhanced Logics, набор стандартов, реализующих интеллектуальные услуги в GSM и UMTS сетях
CellID (CI)	Cell Identifier, «идентификатор соты» (идентификатор сектора базовой станции)
CL	Connection Logics, сетевая логика
CPU	Central Processing Unit, центральное процессорное устройство
CS	Channel Switched, домен коммутации каналов, обрабатывающий трафик сигнализации и пользовательский трафик, обусловленные предоставлением услуг по коммутируемым каналам
DAVA	Destination AVailable, адресат доступен
DPC	Destination Point Code, код пункта назначения

Термин	Описание
DRST	Destination ReSTricted, доступ к пункту назначения ограничен
DUNA	Destination UNAvailable, адресат недоступен
DUPU	Destination User Part Unavailable, подсистема-пользователь в пункте назначения недоступна
GMSC	Gateway MSC, шлюзовой коммутатор мобильной сети для сетей GSM и UMTS
GPRS	General Packet Radio Service, мобильная пакетная сеть передачи данных
GSM	Global System for Mobile Communication, глобальный стандарт цифровой мобильной сотовой связи
GT	Global Title, адрес, использующийся в сигнальных сообщениях на уровне SCCP для определения маршрутов доставки в телекоммуникационных сетях
GTP	GPRS Tunneling Protocol, протокол туннелирования GPRS
HLR	Home Location Register, основная база данных по зарегистрированным абонентам
HTTP	HyperText Transfer Protocol, протокол передачи гипертекста
IAM	Identity and Access Management, управление учетными записями и доступом
ID	Identification, идентификатор
IDP	Initial Detection Point, начальная точка обнаружения
IMEI	International Mobile Equipment Identity, международный идентификатор мобильного оборудования
IMSI	International Mobile Subscriber Identity, международный идентификационный номер мобильного абонента
LAC	Local Area Code, код локальной зоны
LBS	Location-Based Service, сервис, основанный на определении текущего местоположения мобильного телефона пользователя

Термин	Описание
LTE	Long-Term Evolution, стандарт беспроводной высокоскоростной передачи данных для мобильных телефонов и других терминалов, работающих с данными
MAP	Mobile Application Part, прикладная часть (подсистема) контроля мобильной станции
MCC	Mobile Country Code, мобильный код страны
mCSI	Mobility Management CAMEL Subscription Information, профиль абонента, связанный с управлением мобильностью
MM	Moble Management, управление мобильностью
MNC	Mobile Network Code, код мобильной сети
MNO	Mobile Network Operator, оператор сотовой связи
MO-SMS	Mobile Originated SMS, подписка с оплатой за исходящее SMS-сообщение
MSC	Mobile Services Switching Center, центр коммутации мобильной связи
MSISDN	Mobile Station ISDN Number, номер мобильного абонента цифровой сети с интеграцией служб
MT-SMS	Mobile Terminated Short Message Service, подписка с оплатой за входящее SMS сообщение
MTP	Media Transfer Protocol, протокол передачи мультимедиа
MVNO	Mobile Virtual Network Operator, виртуальный оператор сотовой связи (оператор сотовой связи, использующий существующую инфраструктуру другого оператора, но продающий услуги под собственной маркой)
M2PA	MTP2 User Peer-to-Peer Adaptation Layer, протокол обеспечивает уровень адаптации SCTP для предоставления канала сигнализации SS7 MTP по IP-сети
M3UA	MTP3 User Adaptation Layer, протокол обеспечивает уровень адаптации SCTP для бесперебойной транзитной передачи или пиринга пользовательских сообщений MTP уровня 3 и интерфейса обслуживания в IP-сети
NI	Network Indicator, индикатор сети

Термин	Описание
NTFY	NoTiFy, уведомление
OMI	Operating and Maintenance Interface, интерфейс управления и технического обслуживания
OPC	Originating point code, код исходной точки
PDU	Protocol Data Unit, протокольная единица обмена
RAI	Routing Area Identifier, идентификатор зоны маршрутизации
RK	Routing Key, ключ маршрутизации
RT	Routing Table, таблица маршрутизации
SAI	Send-Authentication-Info, MAP-запрос
SCON	Signaling CONgestion, сообщение перегрузки сети
SGSN	Serving GPRS Support Node, узел обслуживания абонентов пакетной сети передачи данных сетей GSM и UMTS
SCCP	Signaling Connection Control Part, подсистема управления сигнальным соединением
SCP	Service Control Point, узел управления услугами интеллектуальной сети
SCTP	Stream Control Transmission Protocol, протокол транспортного уровня
SIC	Service Information Code, сервисный код
SIGTRAN	SIGnaling TRANsport, передача сигнальных сообщений телефонных сигнализаций по IP-сети
SIM	Subscriber Identity Module, модуль идентификации абонента, получаемый при заключении договора с оператором сети сотовой связи
SMS	Short Message Service, сервис коротких сообщений
SRF	Signaling Relay Function, функция ретрансляции сигнальных сообщений
SRIFSM	Send-Routing-Info-Forward-SM, MAP-запрос



Термин	Описание
SS7	Signaling System №7
SSL	Secure Sockets Layer, протоколы, обеспечивающие защищенную передачу данных в компьютерной сети
STP	Signalling Transfer Point, транзитный пункт сигнализации
TCAP	Transaction Capabilities Application Part, прикладная часть средств транзакций, часть общеканальной системы сигнализации № 7
UMTS	Universal Mobile Telecommunications System, технология сотовой связи для внедрения 3G
URL	Uniform Resource Locator, единообразный определитель местонахождения ресурса
VLR	Visitor Location Register, гостевой регистр местоположения
ОС	операционная система
ПО	программное обеспечение
ПУ	пульт управления
СОПМ	система оперативно-розыскных мероприятий

## **2 Общие сведения**

### **2.1 Назначение документа**

Настоящее руководство содержит руководство пользователя по работе с модулем Locator, разработанное ООО «НТЦ ПРОТЕЙ».

### **2.2 Состав документа**

Настоящее руководство состоит из следующих основных частей:

«Термины и сокращения» — раздел, содержащий расшифровку аббревиатур и сокращений, используемых в документе;

«Общие сведения» — раздел, описывающий назначение и состав документа, содержащий сведения о производителе и технической поддержке и историю изменений;

«Описание модуля» — раздел, описывающий назначение, функциональные возможности модуля, а также принципы работы;

«Настройка параметров конфигурации» — раздел, содержащий информацию о настройке основных параметров системы с помощью конфигурационных файлов;

«Техническое обслуживание системы Locator» — раздел, содержащий описание работы пользователя с программным обеспечением;

«Статистика» — раздел, содержащий описание файлов статистики, формируемых системой;

«Журналы» — раздел, содержащий описание CDR-файлов, формируемых системой;

«Аварии» — раздел, содержащий описание возможных аварий системы.

---

#### **Внимание!**

Перед установкой и началом эксплуатации изделия необходимо внимательно ознакомиться с паспортом изделия и эксплуатационной документацией.

Данный документ должен постоянно находиться при изделии.

---

## **2.3 Техническая поддержка**

Техническая поддержка, а также дополнительное консультирование по вопросам, возникающим в процессе установки и эксплуатации изделия, осуществляются производителем и службой технической поддержки.

### **2.3.1 Производитель**

ООО «НТЦ ПРОТЕЙ»

194044, Санкт-Петербург

Большой Сампсониевский пр., д. 60, лит. А

Бизнес-центр «Телеком»

Тел.: (812) 449-47-27

Факс: (812) 449-47-29

WEB: <http://www.protei.ru>

E-mail: [info@protei.ru](mailto:info@protei.ru)

### **2.3.2 Служба технической поддержки**

ООО «НТЦ ПРОТЕЙ»

194044, Санкт-Петербург

Большой Сампсониевский пр., д. 60, лит. А

Бизнес-центр «Телеком»

Тел.: (812) 449-47-27 доп. 5888 (круглосуточно)

Факс: (812) 449-47-29

WEB: <http://www.protei.ru>

E-mail: [mobile.support@protei.ru](mailto:mobile.support@protei.ru)

## 2.4 История изменений

В таблице ниже приведена история изменений.

Таблица 2 — История изменений

Дата	Версия документа	Версия продукта	Комментарий
03.07.2018	1.0.0.1	1.1.9.0	Первая версия документа
31.05.2023	1.0.0.2	1.1.18.0	Внесение новых параметров в запросы и ответы в разделе 3.5. Внесение новых параметров в конфигурационный файл locator.cfg в разделе 4.3.

## 3 Описание модуля

Locator — это программный модуль, который позволяет дополнительно извлекать информацию о местоположении пользователя из протоколов передачи данных в сетях связи оператора.

Данный программный модуль, производства ООО «НТЦ ПРОТЕЙ», предназначен для использования в сетях GSM и реализует все функции COPM по части сбора информации о местоположении абонентов. Все полученные данные модуль Locator обрабатывает и отправляет на COPM (GMSC).

Для всех произведенных транзакций осуществляется сбор статической информации и генерация CDR-файлов. Данная информация позволяет эффективно анализировать трафик и его структуру с помощью специализированных инструментов анализа, предусмотренных в системе.

### 3.1 Функциональные возможности

Модуль Locator обладает следующими функциональными возможностями:

- принимать информацию о местоположении, событиях Attach/Detach по всем абонентам;
- получать информацию об актуальном местоположении от SCP/SRF в момент вызова, MO-SMS-сообщения по проприетарному протоколу;
- обрабатывать подписки на изменение местоположения абонента по IMSI со стороны COPM (GMSC);
- записывать CDR-файлы по событиям MAP\_NOTE\_MM\_EVENT и информационным сообщениям от GTP-Probe;

GTP-Probe осуществляет отправку уведомлений на модуль Locator при создании или изменении связанных идентификаторов MSISDN-IMSI-IMEI, а также при обновлении информации о местоположении.

- направлять ATI-запрос на получение IMEI для обогащения таблицы местоположения абонентов;

Для получения IMEI по абонентам на модуль Locator отправляется ATI-запрос после получения регистрации в сети или первого события с местоположением, не содержащим связки IMSI-IMEI.

- сохранять в памяти связку IMEI-IMSI.

Сохранение данной связки позволяет поставить абонента на контроль по известному IMEI. COPM (GMSC) получит подписку на данный IMEI, который сохранен на модуле Locator, чтобы узнать связанный с ним IMSI.

При постановке абонента на контроль по IMSI, COPM (GMSC) имеет возможность получить подписку на данный IMSI, который сохранен на модуле Locator, чтобы узнать связанный с ним IMEI.

- осуществляет взаимодействие с внешними системами в качестве LBS-сервиса.

Модуль Locator выполняет функции определения местоположения абонентов мобильных сетей GSM, UMTS и LTE на основании параметров, получаемых от мобильных устройств посредством специального SIM-апплета.

Для реализации корректной работы ПО Locator для всех абонентов должен быть включен CAMEL-профиль и на модуле HLR активирована подписка mCSI на все доступные события. Модуль Locator устанавливается дополнительно в сети оператора для обработки событий, получаемых в сообщении MAP\_NOTE\_MM\_EVENT.

## 3.2 Включение модуля Locator в сеть оператора

На рисунке 1 изображена схема включения модуля Locator в сеть оператора мобильной связи.

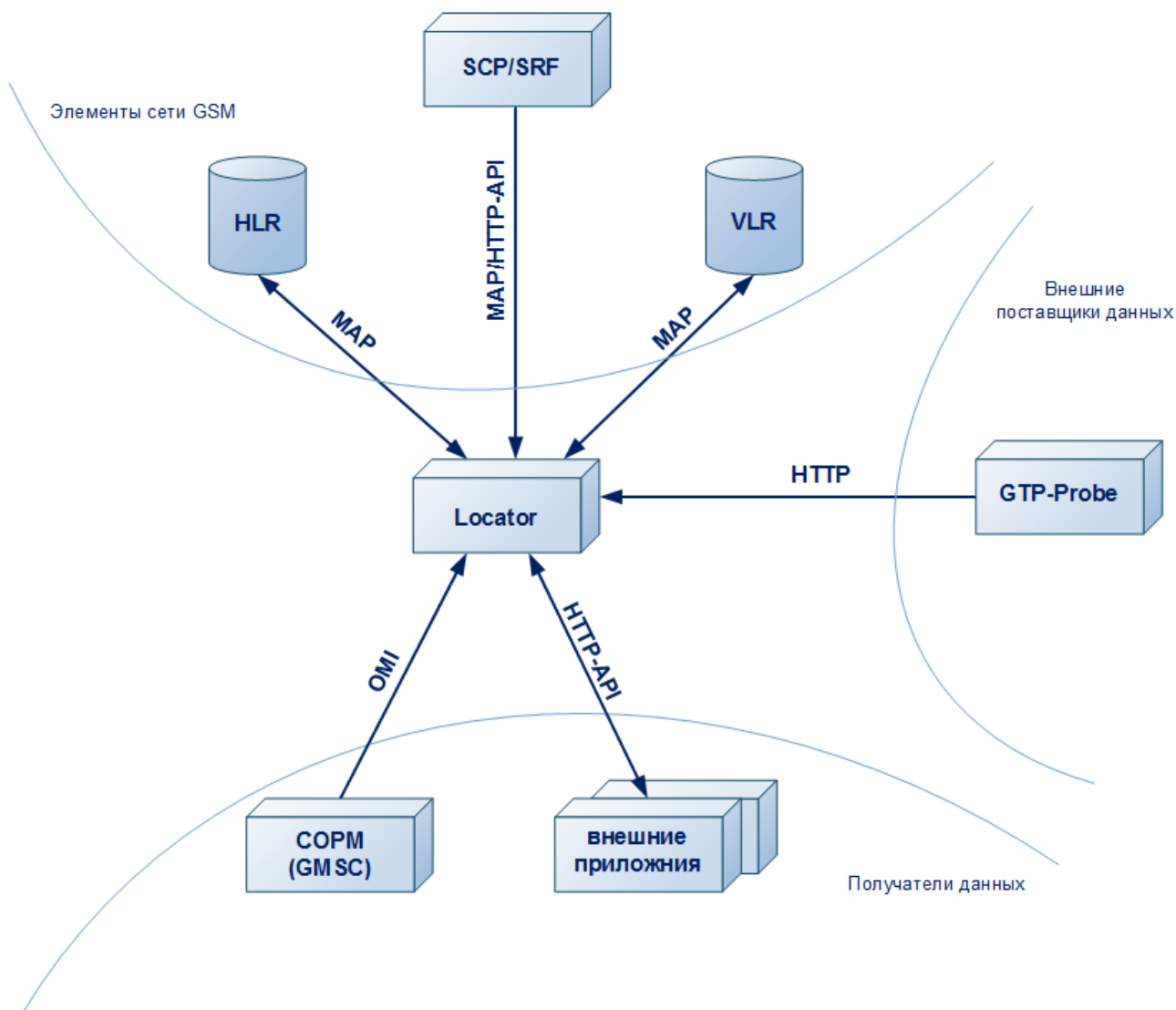


Рисунок 1 — Схема включения модуля Locator в сеть оператора

Основные элементы схемы:

- COPM (GMSC) — подсистема COPM, входящая в состав GMSC и обеспечивающая функции проведения оперативно-розыскных мероприятий;
- HLR — распределенная база данных, устанавливаемая в сети мобильных операторов, предоставляющих услуги связи в стандарте GSM. В этой базе данных хранится полная информация абонентских профилей Оператора. HLR осуществляет контроль процесса перемещения мобильных абонентов.
- SCP/SRF — элемент сети, отвечающий за логические операции и управление услугами интеллектуальной сети с функциями ретрансляции сигнальных сообщений.
- VLR — временная база данных абонентов, которые находятся в зоне действия определённого MSC. Все данные, которые хранятся на VLR поступают из HLR. VLR обеспечивает контроль абонента внутри своей зоны действия.

- GTP-Probe — приложение, которое дополнительно отправляет данные на модуль Locator из пакетного сегмента сети MVNO.

Взаимодействие между элементами схемы происходит по следующим протоколам:

- CAMEL — используется для реализации интеллектуальных услуг в сетях GSM/UMTS;
- HTTP — отправка данных между GTP-Probe и модулем Locator;
- HTTP API — взаимодействие модуля Locator с внешними приложениями. Формат HTTP-запросов, которые поддерживает модуль Locator при обработке различных сервисов, рассмотрен ниже;
- MAP — обеспечивает взаимодействие модуля Locator с внешним оборудованием сети оператора связи.
- SIGTRAN — протокол обмена сигнальных сообщений телефонных сигнализаций по IP-сети;
- OMI — используется в случае установки оборудования производства ООО «НТЦ ПРОТЕЙ».

В случае проблем с совместимостью с MAP обеспечивается корректное взаимодействие с сетями роуминговых партнеров. Непрерывное наблюдение позволяет постоянно контролировать параметры качества роуминга.

### 3.3 Получение местоположения абонента

Для получения информации о местоположении абонентов на модуле Locator формируется временное хранилище, которое содержит последние известные параметры об абоненте:

- гостевого регистра местоположения (VLR);
- кода локальной зоны (LAC);
- идентификатора соты (CellID).

Данные в нем обновляются в следующих случаях:

1. Изменение данных в VLR Host-оператора и получение актуальной информации в сообщении о местоположении абонента по триггеру mCSI;
2. При исходящем вызове из сообщений IDP при обработке его на SRF;
3. При исходящем SMS-сообщении (MO-SMS) из сообщения IDPforSM при обработке его на SRF;
4. В рамках периодического активного запроса информации о местоположении абонента (ATI-запрос с и без пейджинга);
5. Получение информации от модуля GTP-Probe при инициации/разрыве/модификации сессии передачи данных.
6. Получение информации о местоположении абонента от оборудования оператора сети сотовой связи, если модуль Locator выполняет функции LBS-сервиса.
7. При проверке данных с помощью подписки об изменении местоположения абонента и получении уведомления об этом изменении от оператора сети сотовой связи, если модуль Locator выполняет функции LBS-сервиса.

Далее представлены сценарии работы для различных ситуаций обновления местоположения абонента.

При изменении данных в VLR Host-оператора, данные об этом поступают на модуль Locator.

Подписка осуществляется в соответствии со сценарием, отображенным на рисунке 2.

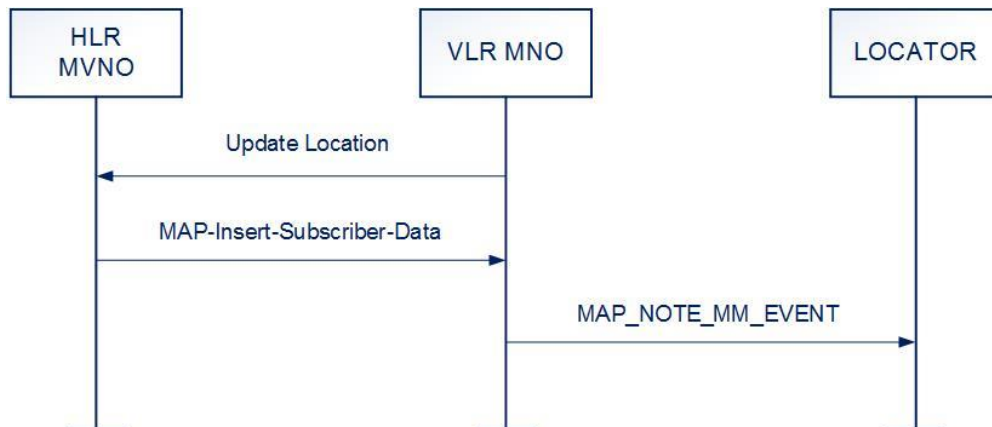


Рисунок 2 — Подписка на M-CSI и получение сообщений о событиях

Принцип работы:

1. После того как VLR MNO (Host-оператора) авторизует абонента, он отправит на HLR MVNO сообщение Update Location.

Это сообщение отправляется для того, чтобы HLR MVNO знал, в зоне действия какого VLR MNO находится абонент. Сообщение Update Location содержит в себе IMSI абонента и GT VLR MNO. HLR осуществит проверку полученных данных.

2. С помощью сообщения MAP-Insert-Subscriber-Data HLR MVNO пересылает абонентские данные на обслуживающий VLR MNO, запросивший эту информацию. Данное сообщение также содержит профиль CamelServiceInfo с кодами (MM-Codes) и адресом модуля SC\_Lite.
3. Затем VLR MNO в сообщении MAP\_NOTE\_MM\_EVENT передает данные (Event, IMSI, Basic MSISDN, Location Information) на модуль Locator о событиях, связанных с изменением местоположения абонента в рамках CS-домена.

Сообщение MAP\_NOTE\_MM\_EVENT отправляет VLR Host-оператора на модуль Locator в случае изменения следующих данных на VLR:

- Смена кода локальной зоны (LAC) и соты (CellID);
- Смена гостевого регистра местоположения (VLR);
- Изменение флага Attach/Detach.



По первым двум событиям формируется информационное сообщение №1.6 «Изменение местоположения наблюдаемого абонента». Схема получения информации представлена на рисунке 3.

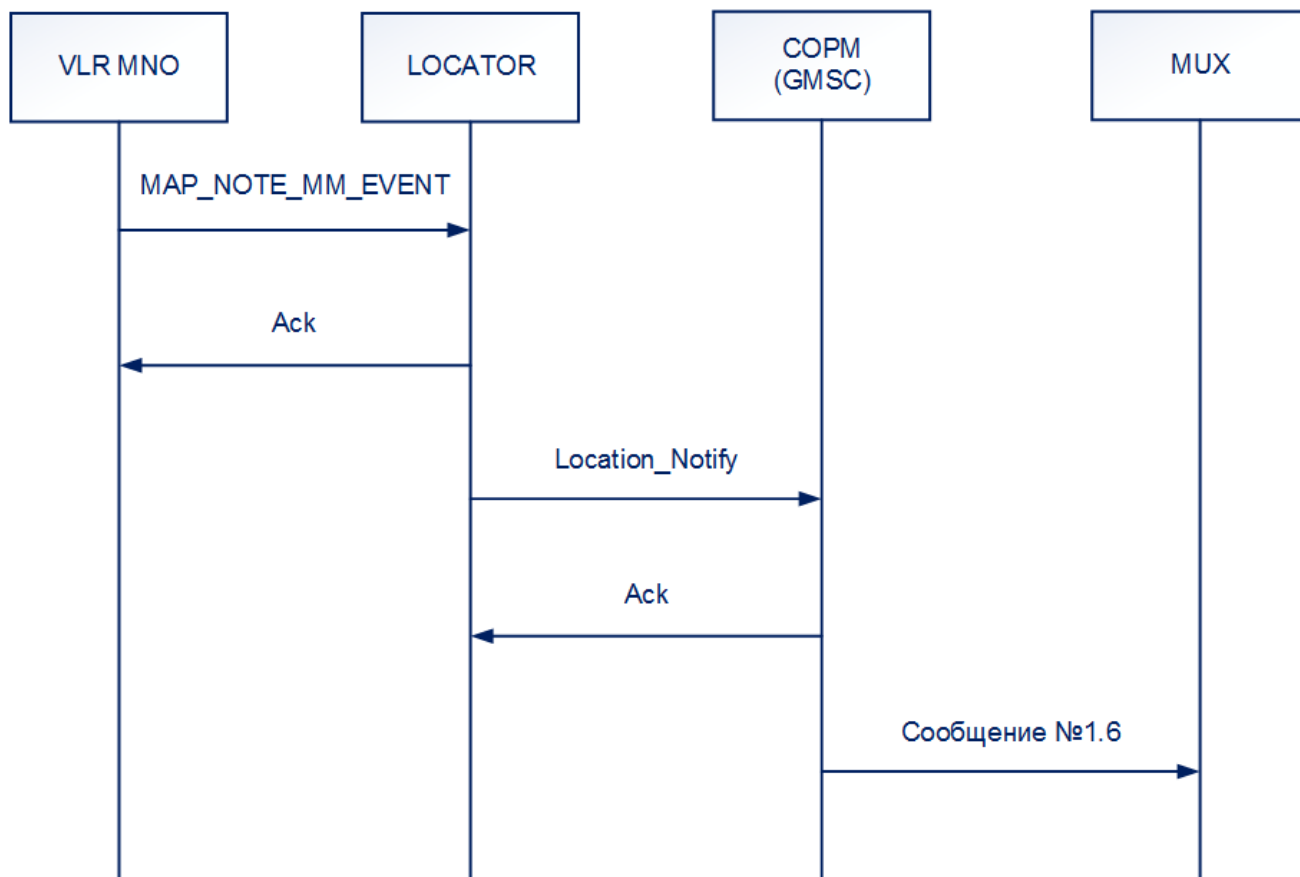


Рисунок 3 — Формирование сообщения №1.6

Принцип работы:

1. VLR MNO отправляет сообщение MAP\_NOTE\_MM\_EVENT на модуль Locator, в котором передает данные (Event, IMSI, Basic MSISDN, Location Information) о событиях, связанных с изменением местоположения абонента в рамках CS-домена.
2. Модуль Locator подтверждает получение информации от VLR MNO и отправляет на него сообщение «Ack».
3. Далее модуль Locator отправляет уведомление на COPM (GMSC) о смене местоположения абонента. Для этого отправляется сообщение Location\_Notify.
4. В ответ COPM (GMSC) отправляет сообщение Ack, подтверждающее получение сообщения от модуля Locator.
5. Затем COPM (GMSC) формирует информационное сообщение №1.6 «Изменение местоположения наблюдаемого абонента».

При получении сообщения MAP\_NOTE\_MM\_EVENT по событиям Attach/Detach в сторону ПУ COPM (GMSC) отправляется информационное сообщение №1.5 «Изменение статуса наблюдаемого абонента». Сценарий получения информации представлена на рисунке 4.

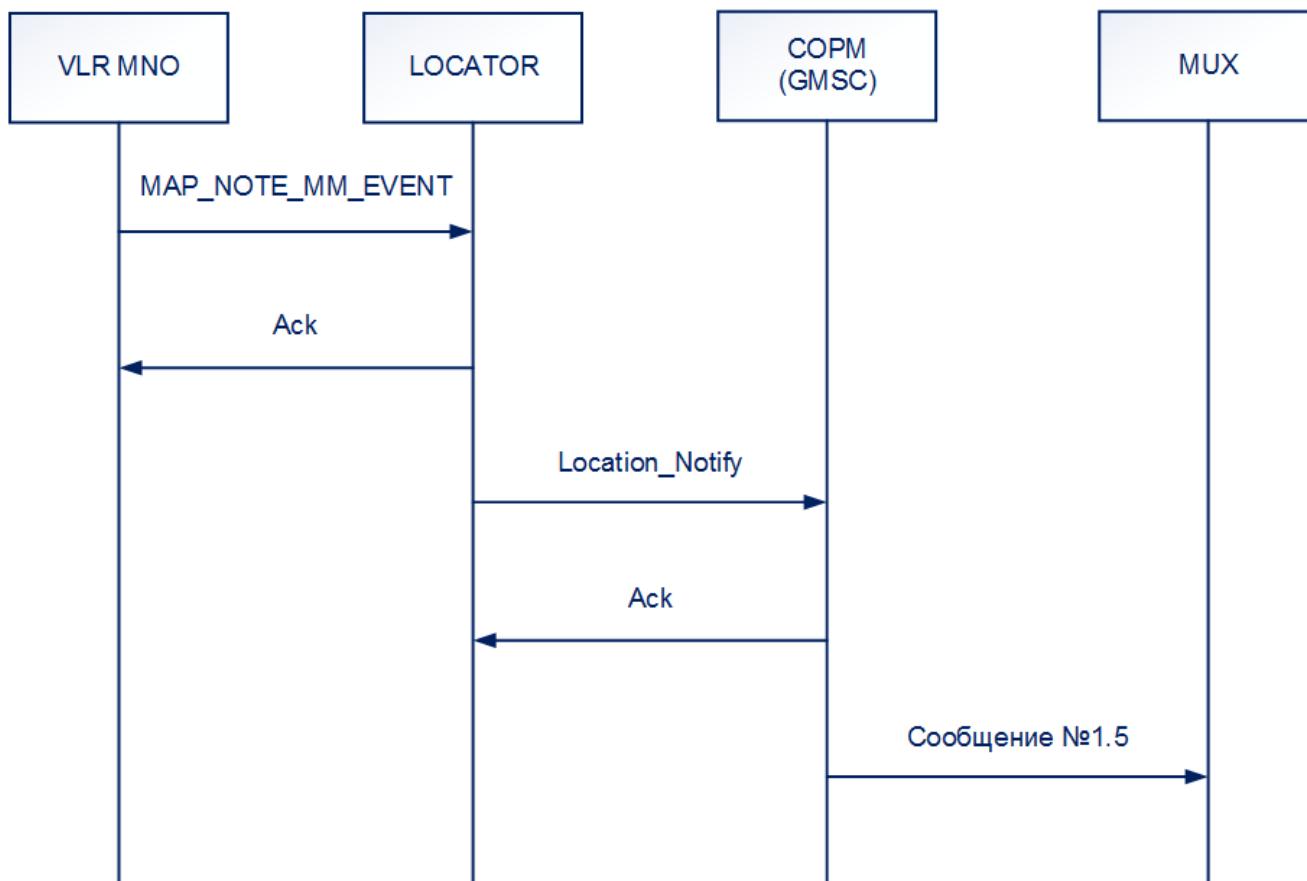


Рисунок 4 — Формирование сообщения №1.5

Принцип работы:

1. VLR MNO отправляет сообщение MAP\_NOTE\_MM\_EVENT на модуль Locator, в котором передает данные по событиям Attach/Detach, связанным с изменением местоположения абонента в рамках CS-домена.
2. Модуль Locator подтверждает получение информации от VLR MNO и отправляет на него сообщение Ack.
3. Далее модуль Locator отправляет уведомление на COPM (GMSC) о смене местоположения абонента. Для этого отправляется сообщение Location\_Notify.
4. В ответ COPM (GMSC) отправляет сообщение Ack, подтверждающее получение сообщения от Locator.
5. Затем COPM (GMSC) формирует информационное сообщение №1.5 «Изменение статуса наблюдаемого абонента», которое отправляется на MUX для передачи на ПУ COPM.

После того, как Locator получит уведомление от GTP-Probe с информацией об изменении местоположения абонента, COPM (GMSC) отправляется информационное сообщение №1.6 «Изменение местоположения наблюдаемого абонента» в сторону ПУ COPM. Сценарий получения информации представлена на рисунке 5.

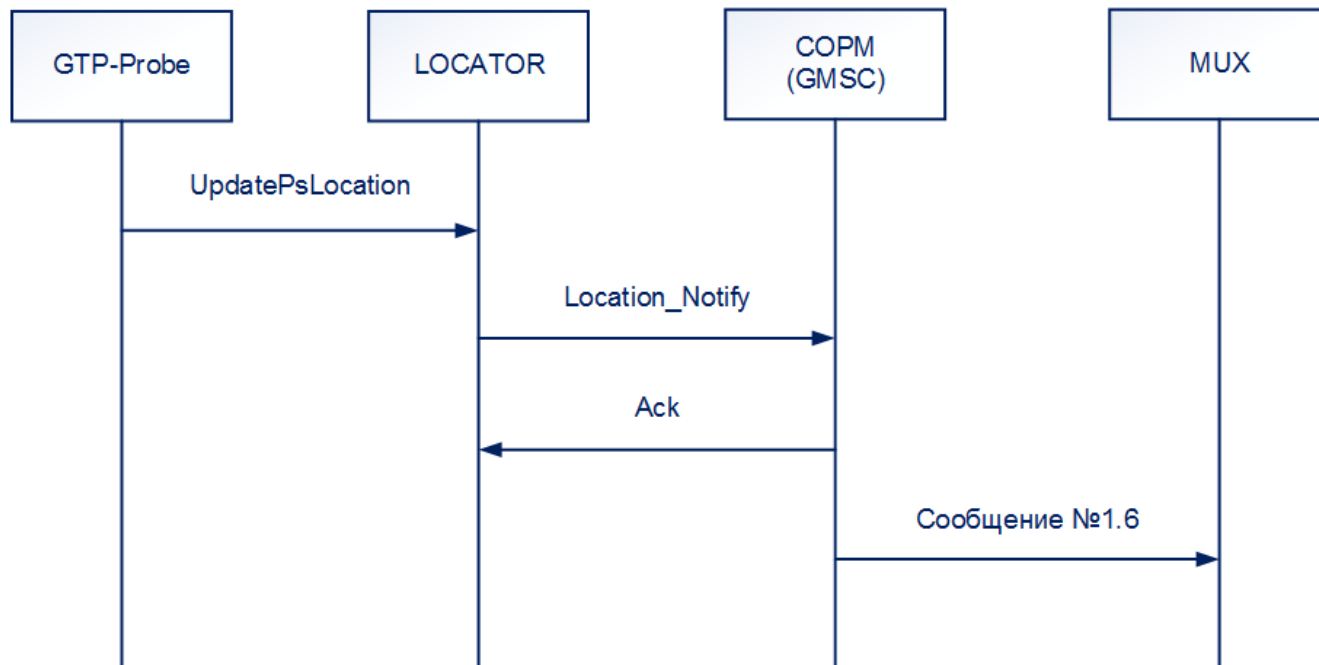


Рисунок 5 — Формирование сообщения №1.6 после уведомления от GTP-Probe

Принцип работы:

1. GTP-Probe по HTTP/JSON отправляет сообщение UpdatePsLocation на модуль Locator, в котором передает данные, связанные с изменением местоположения абонента.
2. Модуль Locator получает информацию и посылает уведомление на COPM (GMSC) о смене местоположения абонента. Для этого отправляется сообщение Location\_Notify.
3. В ответ COPM (GMSC) отправляет сообщение Ack, подтверждающее получение сообщения от модуля Locator.
4. Затем COPM (GMSC) формирует информационное сообщение №1.6 «Изменение местоположения наблюдаемого абонента» и посылает его на MUX.

В момент вызова модуль Locator получает информацию об актуальном местоположении абонента от SCP/SRF. Сценарий получения информации представлена на рисунке 6.

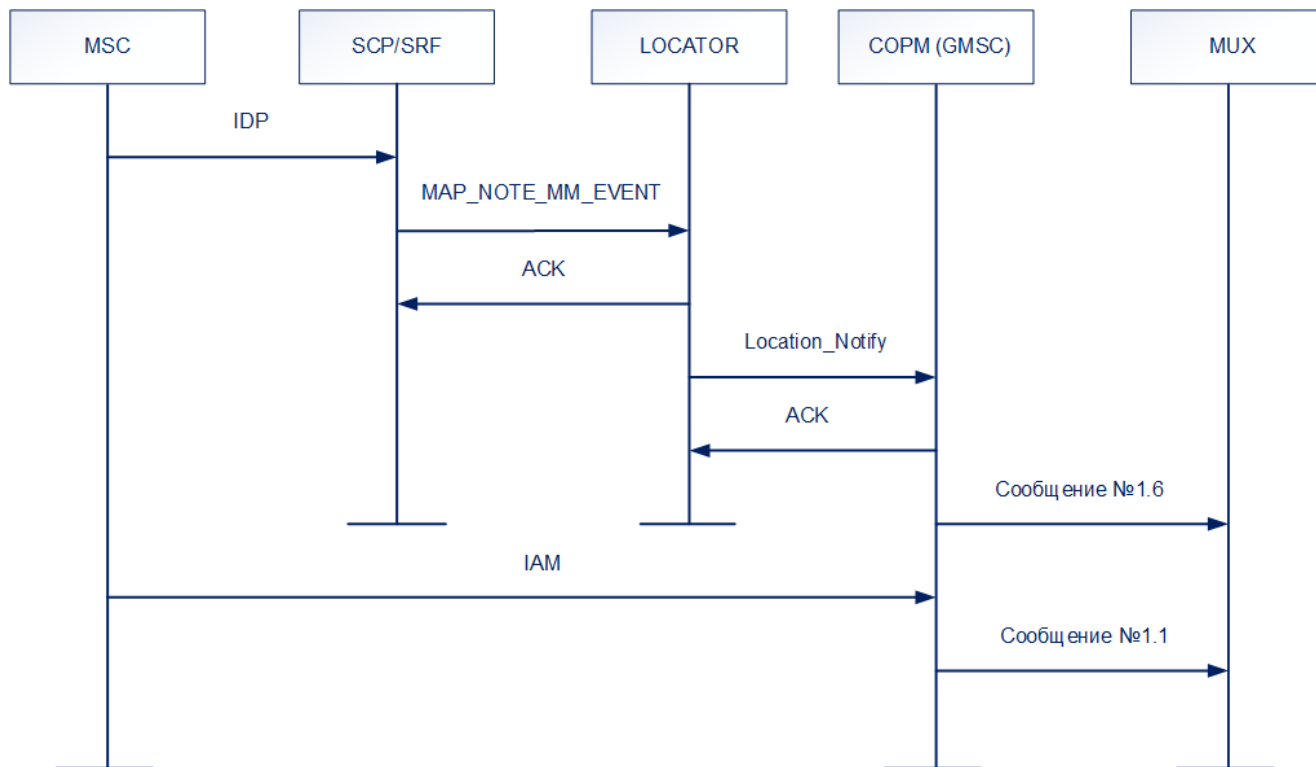


Рисунок 6 — Получение точного местоположения абонента при вызове

Принцип работы:

1. MSC инициирует сообщение IDP (InitialDP), в котором передаются параметры зарегистрированного абонента с указанием его текущего состояния.
2. SCP/SRF анализирует запрос и отправляет сообщение MAP\_NOTE\_MM\_EVENT на модуль Locator, в котором передает данные о событиях, связанных с изменением местоположения абонента.
3. Модуль Locator подтверждает получение информации от SCP/SRF и отправляет на него сообщение Ack.
4. Затем модуль Locator осуществляет передачу уведомления на COPM (GMSC) о смене местоположения абонента. Для этого отправляется сообщение Location\_Notify.
5. В ответ COPM (GMSC) отправляет сообщение Ack, подтверждающее получение сообщения от модуля Locator.
6. Затем COPM (GMSC) формирует информационное сообщение №1.6 «Изменение местоположения наблюдаемого абонента».

Сообщение №1.6 «Изменение местоположения наблюдаемого абонента» и Location\_Notify могут не приходить, в том случае, если текущее местоположение абонента на модуле Locator совпадает с полученным местоположением из SRF.

7. Обработка исходящего вызова происходит на MSC, в котором абонент зарегистрирован в данный момент. MSC осуществляет отправку сообщения IAM на COPM (GMSC) для установления соединения.

8. После получения сообщения IAM COPM (GMSC) формирует и отправляет в сторону MUX для передачи на ПУ COPM информационное сообщение №1.1 «Прием полного номера телефона вызываемого абонента».

Из диаграммы видно, что в данном сценарии сообщение №1.6 «Изменение местоположения наблюдаемого абонента» и сообщение №1.1 «Прием полного номера телефона вызываемого абонента» формируются асинхронно. Порядок формирования данных информационных сообщений на интерфейсе COPM может отличаться от приведенного на диаграмме, изображенной на рисунке 6.

При исходящем SMS сообщении (MO-SMS) сценарий для получения данных о местоположении абонента будет аналогичный получению точного местоположения абонента при исходящем вызове, приведенному на рисунке 5. Необходимые данные о местоположении будут обрабатываться на SRF и получены из сообщения IDPforSM протокола CAMEL.

Для получения информации о точном местоположении при входящих вызовах и SMS сообщений (MT-SMS) модуль Locator не используется. Информация о точном местоположении в таких случаях получает COPM (GMSC) из сигнального обмена.

Модуль Locator имеет возможность сохранять связки IMEI-IMSI. При постановке абонента на контроль по IMSI, COPM (GMSC) имеет возможность получить подписку на данный IMSI, который сохранен на модуле Locator, чтобы узнать связанный с ним IMEI.

Сценарий получения информации о местоположении абонента при обработке подписки по IMSI со стороны COPM (GMSC) представлен на рисунке 7.

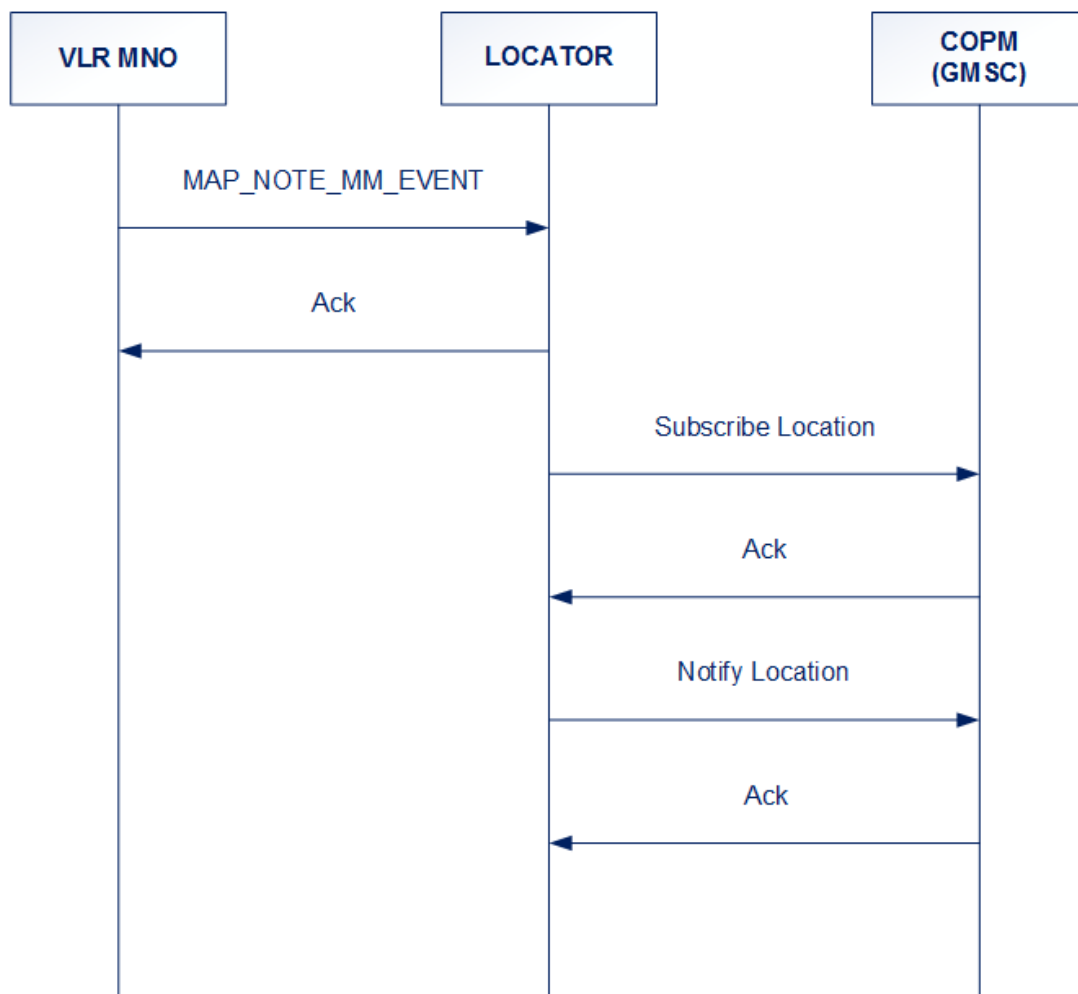


Рисунок 7 — Получение местоположения абонента при обработке подписки по IMSI

Принцип работы:

1. VLR MNO отправляет сообщение MAP\_NOTE\_MM\_EVENT на модуль Locator, в котором передает данные (Event, IMSI, Basic MSISDN, Location Information) об изменении местоположения абонента.
2. Модуль Locator подтверждает получение информации от VLR MNO с помощью сообщения Ack.
3. После того как модуль Locator получит новые данные местоположения абонента, он осуществит подписку на изменение статуса абонента по полученному от VLR MNO IMSI. Для этого отправляется сообщение Subscribe Location.
4. COPM (GMSC) отправляет сообщение Ack, подтверждающее получение сообщения с подпиской от модуля Locator.
5. Далее модуль Locator отправляет уведомление на COPM (GMSC) со статусом, которое оповещает о смене местоположения абонента. Для этого отправляется сообщение Location\_Notify.
6. В ответ COPM (GMSC) отправляет сообщение Ack, подтверждающее получение сообщения от модуля Locator.

### 3.4 Получение IMEI

Для получения международного идентификатора мобильного оборудования (IMEI) модуль Locator отправляет ATI-запрос на HLR при получении Attach по абоненту.

Если Locator получает событие Attach либо сообщение Update Location по IMSI, с которым не сопоставлен ни один IMEI, то модуль Locator осуществляет принудительное получение IMEI через ATI-запрос на HLR, после чего сохранит текущее значение IMEI абонента.

Сообщение Update Location может не содержать значение IMEI, если он получен от GTP-Probe. В этом случае модуль Locator не будет производить проверку IMEI.

В ответном сообщении от HLR на ATI-запрос и в сообщениях протокола GTP и GTPv2 IMEI передается IMEI-SV (16 знаков, в соответствии с ETSI TS 123 003).

### 3.5 API интерфейс для работы с внешними подсистемами

Для получения информации из базы данных о местоположении любого абонента сети MVNO оператором модуль Locator поддерживает HTTP-интерфейс к внешней LBS-системе.

LBS-запросы отправляются на модуль Locator по протоколу HTTP. Интерфейс поддерживает следующие запросы:

- получение текущего местоположения (см. раздел 3.5.1 «Получение текущего местоположения»);
- получение текущего IMEI (см. раздел 3.5.2 «Получение текущего IMEI»);
- подписка на смену местоположения абонента (см. раздел 3.5.3 «Подписка на смену местоположения абонента»);
- подписка на смену IMEI (см. раздел 3.5.4 «Подписка на смену IMEI абонента»);
- отмена подписки на смену местоположения абонента (см. раздел 3.5.5 «Отмена подписки на смену местоположения абонента»);
- отмена подписки на смену IMEI (см. раздел 3.5.6 «Отмена подписки на IMEI абонента»).

Типы данных и значимость параметра описаны в разделе 4.1 «Типы данных и условные обозначения».

### 3.5.1 Получение текущего местоположения

Модуль Locator имеет возможность реализовывать функции определения местоположения абонентов мобильных сетей GSM, UMTS и LTE на основании параметров, получаемых от мобильных устройств. Модуль Locator принимает на обработку текущие параметры от сети оператора из области нахождения терминала абонента, на основании этих данных вычисляет географические координаты.

Сценарий получения информации о местоположении абонента представлен на рисунке 8.

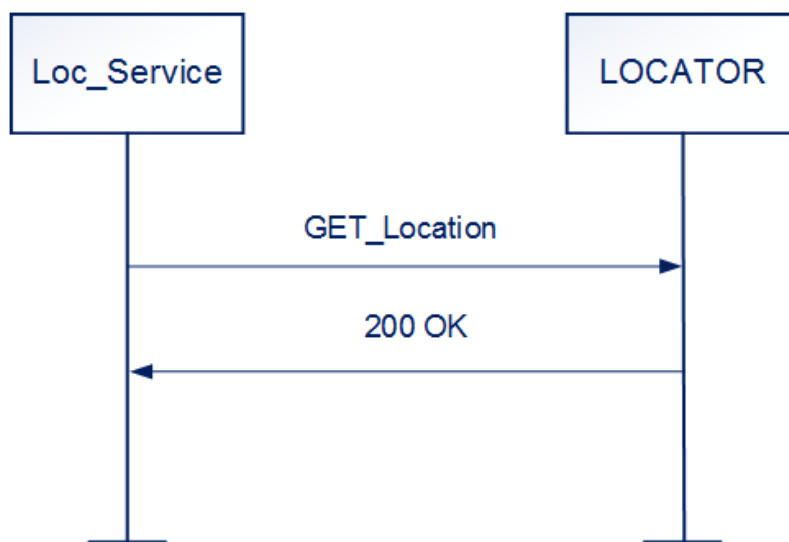


Рисунок 8 — Получение информации о текущем местоположении терминального оборудования абонента через HTTP-API

Принцип работы:

1. Оборудование сети оператора (Loc\_Service) отправляет запрос GET\_Location на модуль Locator, в котором просит предоставить информацию о нахождении терминального оборудования абонента.

Модуль Locator взаимодействует по протоколу HTTP с оборудованием оператора сети сотовой связи и поддерживает метод GET для передачи данных. Используемый метод указывается в первой строке HTTP-запроса (линия запроса).

Формат запроса: **http://<host>:<port>/Location?imsi=<IMSI>**,

где

- host — IP-адрес или домен Locator;
- port — порт для подключения через API;
- IMSI — IMSI абонента, по которому требуется узнать его текущее местоположение.

2. Модуль Locator подтверждает получение запроса от Loc\_Service с помощью сообщения «200 OK».

Если IMSI абонента в базе данных модуля Locator отсутствует или в параметрах запроса была допущена ошибка, то на Loc\_Service отправится сообщение «not\_found».

3. Далее модуль Locator должен осуществить обращение к базе данных и определить актуальные данные о местоположении терминального оборудования абонента.

Данный запрос используется для получения последнего известного местоположения абонента.

### Запрос

GET-запрос для запроса местоположения абонента: **/Location?imsi={IMSI}**.

GET-запрос для активного запроса местоположения абонента с пейджингом:  
**/Location?forced={0/1}&imsi={IMSI}**.

GET-запрос для активного запроса местоположения абонента без пейджинга:  
**/Location?forced={0/1}&imsi={IMSI}&no-paging={0/1}**.

Параметры запросов описаны в таблице ниже.

Таблица 3 — Параметры запросов

Параметр	Значимость параметра	Описание
IMSI	O	IMSI абонента, по которому требуется узнать его текущее местоположение. Тип — string.
forced	O	Параметр, определяющий необходимость активного запроса местоположения с помощью ATI-запроса в режиме пейджинга. Тип — bool.
no-paging	O	Параметр, который позволяет не использовать пейджинг в рамках активного запроса местоположения. Тип — bool.

### Ответ

Ниже приведён формат ответа на запрос:

```
{
  "data" : {
    "ageofLocation":<ageofLocation>,
    "vlr":<vlr>,
    "location" : {
      "ci" : <ci>,
      "lac" : <lac>,
      "location_type" : <location_type>,
      "mcc" : <mcc>,
      "mnc" : <mnc>
    },
  },
  "status" : "<status>"
}
```



Параметры ответа описаны в таблице ниже.

Таблица 4 — Параметры ответа

Параметр	Значимость параметра	Описание
ci	O	Идентификатор соты (Cell Identification), в которой находится абонент (оборудование пользователя). Тип — int.
lac	O	Код местоположения (Location Area Code) абонента, либо идентификатор узла B (EnodeBId), когда LOC_TYPE=2. Тип — int.
location_type	O	Тип сети, от которой получена индикация. Тип — int. Возможные значения: 0 — Местоположение абонента (пользовательского оборудования) в сети 2G/3G; 2 — Местоположение абонента (пользовательского оборудования) в сети LTE.
mcc	O	Мобильный код страны абонента. Тип — int.
mnc	O	Код сети сотовой связи, определяемый для каждого оператора в пределах страны абонента. Тип — int.
ageofLocation	O	Время с момента получения данных по местоположению. Тип — int (в секундах).
vlr	O	Номер VLR/MSC, на котором зарегистрирован абонент. Тип — int.
status	O	Статус выполнения запроса. Тип — string. Возможные значения: ok — запрос выполнен успешно; not_found — IMSI в базе данных модуля Locator отсутствует или в параметрах запроса допущена ошибка.

Ниже приведён пример запроса:

```
{
  "data" : {
    "ageOfLocation" : 648,
    "location" : {
      "ci" : 60472,
      "lac" : 27855,
      "location_type" : 0,
      "mcc" : 250,
      "mnc" : 20
    }
  },
  "status" : "ok"
}
```

### 3.5.2 Получение текущего IMEI

Данный запрос используется для получения последнего известного местоположения абонента.

#### **Запрос**

GET-запрос: **/Imei?imsi=<IMSI>**

Параметры запроса описаны в таблице ниже.

Таблица 5 — Параметры запроса

Параметр	Значимость параметра	Описание
IMSI	О	IMSI абонента, по которому требуется узнать его текущий IMEI. Тип — string.

#### **Ответ**

Формат ответа приведен в таблице ниже.

```
{
  "data" : {
    "imei" : "<imei>"
  },
  "status" : "ok"
}
```

Параметры ответа описаны в таблице ниже.

Таблица 6 — Параметры ответа

Параметр	Значимость параметра	Описание
imei	O	Международный идентификатор мобильного оборудования абонента. Тип — string.
status	O	Статус выполнения запроса. Тип — string. Возможные значения: ok — запрос выполнен успешно; not_found — IMSI в базе данных модуля Locator отсутствует или в параметрах запроса допущена ошибка.

### 3.5.3 Подписка на смену местоположения абонента

Еще одной функцией модуля Locator является проверка данных с помощью подписки об изменении местоположения абонента. При изменении информации о месте нахождения терминального оборудования абонента оператор сети сотовой связи подготовит и отправит уведомление на модуль Locator.

Сценарий получения информации о местоположении абонента представлен на рисунке 9.

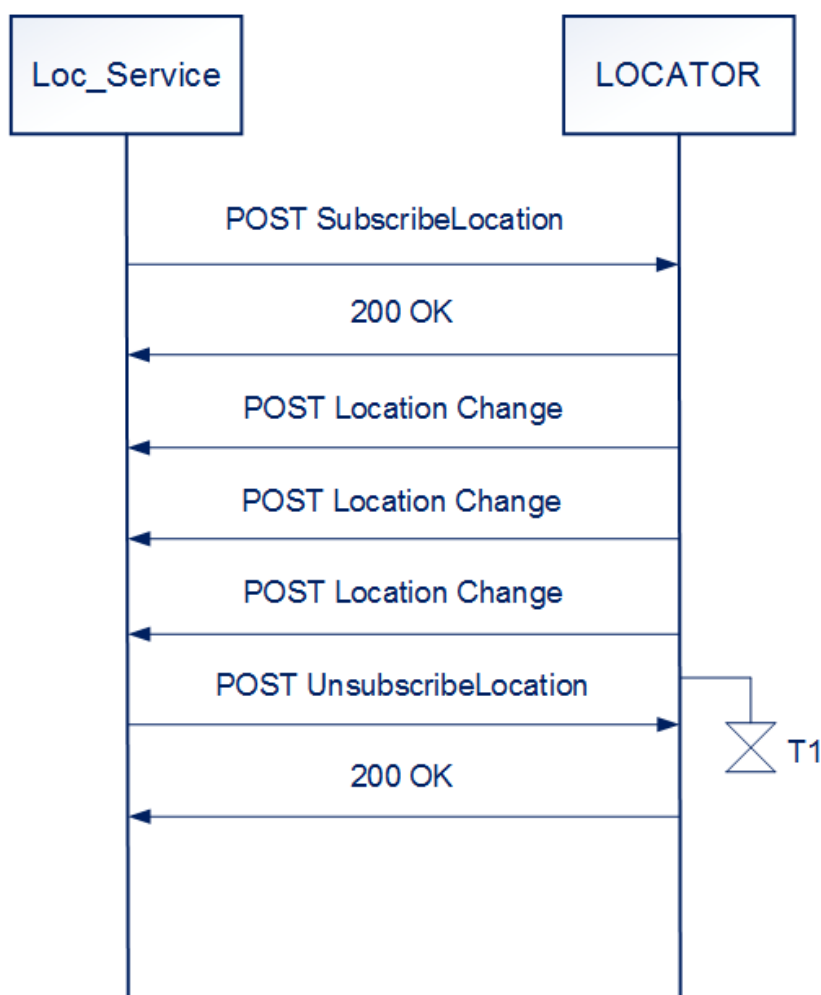


Рисунок 9 — Получение подписки на смену местоположения абонента в модуле Locator

Принцип работы:

1. Оборудование сети оператора (Loc\_Service) отправляет на модуль Locator POST-запрос SubscribeLocation, содержащий следующие параметры: IMSI, уникальный идентификатор и длительность подписки, а также URL-адрес, на который будут отправляться уведомления.

**Примечание** — Значение таймера для подписки задается в конфигурационном файле locator.cfg.

2. В случае успешно выполненной подписки модуль Locator отправит сообщение «200 OK».

Если модуль Locator не смог распознать запрос от оборудования сети оператора, то он может отправить следующие типы сообщений об итоге выполнения запроса:

- expires\_too\_brief — в запросе указан очень маленький интервал подписки. В ответ вернется минимально возможный интервал подписки;
- no\_subscription\_with\_that\_id — попытка продлить несуществующую подписку;
- no\_subscription\_on\_imsi\_with\_that\_id — попытка продлить существующую подписку, но для другого IMSI;
- invalid\_contact — URL-адрес не соответствует схеме URI (RFC 3986);

- `no_host_in_contact` — в URL-адресе отсутствует имя хоста;
  - `invalid` — запрос не соответствует JSON-формату или отсутствуют значения IMSI и URL-адреса, на который будут отправляться уведомления;
  - `unknown` — ошибка запроса.
3. Информация об изменении местоположения абонента отправляется сразу после осуществления подписки, а также после каждой смены местоположения. Вместе с идентификатором подписки модуль Locator будет отправлять индикацию подписки (активная или завершившаяся), IMSI, а также данные о местоположении абонента (CI, LAC, MCC, MNC)/
  4. Для отмены подписки на изменение местоположения `Loc_Service` отправляет на модуль Locator POST-запрос `UnsubscribeLocation`, в котором будет указан идентификатор отменяемой подписки.

Отмена подписки может быть произведена по истечению времени (таймер подписки). Значения таймера указываются в конфигурационном файле `locator.cfg`

5. Для подтверждения отмены подписки модуль Locator отправит сообщение «200 OK».

В случае отмены несуществующей подписки модуль Locator отправит сообщение «`no_subscription_with_that_id`». Если запрос, принятый от оборудования сети оператора не соответствует JSON-формату или отсутствует идентификатор отменяемой подписки, то модуль Locator отправит сообщение «`invalid`».

Данный запрос используется для подписки на смену местоположения абонента и её продления.

### **Запрос**

POST-запрос: **`/SubscribeLocation`**.

Запрос должен содержать:

- Заголовок — `Content-Type: application/JSON`.
- Тело POST-запроса в JSON-формате.

Ниже приведён формат запроса:

---

```
{
"contact" : "<contact>",
"force-update-period" : <force-update-period>,
"paging" : <paging>
"imsi" : "<imsi>",
"sub-expires" : <sub-expires>,
"sub-id" : "<sub-id>",
}
```

---

Параметры запросов описаны в таблице ниже.

Таблица 7 — Параметры запроса

Параметр	Значимость параметра	Описание
imsi	M	IMSI абонента, на которого осуществляется подписка. Тип — string.
sub-id		Идентификатор подписки. Тип — string. <b>Примечание</b> — Параметр должен отсутствовать в запросе на создание подписки. Параметр должен присутствовать в запросе на продление подписки.
sub-expires	M	Срок действия подписки. Тип — int (в секундах). <b>Примечание</b> — Если не задан данный параметр, то используется значение параметра DefExpires из файла конфигурации locator.cfg.
contact	M	Адрес, на который должны отправляться уведомления о смене местоположения. Тип — string.
force-update-period	O	Период активного запроса местоположения по абоненту. Тип — int (в секундах). Значения по умолчанию — 0. <b>Примечания.</b> 1. Если параметр равен 0 или не задан, то активный запрос не посылается. 2. Параметр актуален для запроса на создание подписки.
paging	O	Флаг активации активного запроса местоположения с процедурой пейджинга. Тип — bool. Значения по умолчанию — false. <b>Примечание</b> — Параметр актуален для запроса на создание подписки.

Ниже приведён пример запроса:

```
{
  "contact": "http://10.77.51.205:7779/event/imsi/location",
  "force-update-period": 120,
  "imsi": "250507800000053",
  "paging": true,
  "sub-expires": 3600,
  "sub-id": "c4acd96c-bd19-4071-aa9f-63c034a58557"
}
```

### Ответ

Ниже приведён формат ответа на запрос:

```
{
  "status" : "<status>",
  "sub-id" : "<sub-id>",
  "sub-expires" : <sub-expires>
}
```

Параметры ответа описаны в таблице ниже.

Таблица 8 — Параметры ответа

Параметр	Значимость параметра	Описание
status	О	<p>Статус выполнения запроса. Тип — string.</p> <p>Возможные значения</p> <p>ok — запрос выполнен успешно;</p> <p>expires_too_brief — в запросе слишком маленький интервал подписки. В ответе вернется минимально возможный интервал подписки;</p> <p>no_subscription_with_that_id — попытка продлить несуществующую подписку;</p> <p>no_subscription_on_imsi_with_that_id — попытка продлить существующую подписку, но не для другого IMSI;</p> <p>invalid_contact — поле contact не соответствует схеме URI (RFC 3986);</p> <p>no_host_in_contact — отсутствует хост в поле contact;</p> <p>Invalid — запрос не соответствует JSOB-формату или отсутствуют обязательные поля IMSI и contact;</p> <p>unknown — другая ошибка.</p>

Параметр	Значимость параметра	Описание
sub-id	O	Идентификатор подписки. Тип — string.
sub-expires	O	Срок действия подписки. Тип — int (в секундах).

Ниже приведён пример ответа на запрос:

```
{
  "status" : "ok",
  "sub-expires" : 3600,
  "sub-id" : "e4b5829c-47a6-4549-8c8f-bbc99d6d6a3d"
}
```

### **Уведомления**

При успешной подписке на смену местоположения абонента модуль Locator начинает отправлять уведомления о смене местоположения абонента на адрес, указанный в поле contact.

Уведомления начинают отправляться сразу после осуществления подписки, а также после каждой смены местоположения.

Уведомления отправляются в формате POST-запроса. Тело POST-запроса представлено в JSON-формате.

Ниже приведён формат запроса для уведомления:

```
{
  "sub-id" : "<sub-id>",
  "sub-status" : "<sub-status>"
  "data" : {
    "imsi" : "<imsi>",
    "location" : {
      "mcc" : <mcc>,
      "mnc" : <mnc> }
      "lac" : <lac>,
      "ci" : <ci>,
    },
    "ageofLocation" : <ageofLocation>,
    "VLR" : <VLR >,
  }
}
```



Параметры запроса описаны в таблице ниже.

Таблица 9 — Параметры запроса

Параметр	Значимость параметра	Описание
sub-id	O	Идентификатор подписки. Тип — string.
sub-status	O	Статус подписки. Тип — string. Возможные значения: active — индикация активной подписки; inactive — индикация завершившейся подписки. <b>Примечание</b> — В случае истечения таймера запрос на подписку посылается без данных data.
imsi	O	Международный идентификатор абонента мобильной сети (International Mobile Subscriber Identity). Тип — string.
mcc	O	Мобильный код страны абонента. Тип — int.
mnc	O	Код сети сотовой связи, определяемый для каждого оператора в пределах страны абонента. Тип — int.
lac	O	Код местоположения (Location Area Code) абонента либо идентификатор узла B (EnodeBId), когда LOC_TYPE=2. Тип — int.
ci	O	Идентификатор соты (Cell Identification), в которой находится абонент (оборудование пользователя). Тип — int.
ageofLocation	O	Время с момента получения данных по местоположению. Тип — int (в секундах).
VLR	O	Номер VLR/MSC, на котором зарегистрирован абонент. Тип — int.

Ниже приведен пример запроса для уведомления:

```
{
  "sub-id" : "f1158376c2e741d58dbb5f904a95e102",
  "sub-status" : "active"
  "data" : {
    "imsi" : "111111111111111",
    "location" : {
      "mcc" : 12,
      "mnc" : 34,
      "lac" : 56,
      "ci" : 78,
    },
    "ageofLocation" : 10,
    "VLR" : 123,
  }
}
```

### 3.5.4 Подписка на смену IMEI абонента

Если модуль Locator выполняет функции LBS-сервиса, то для получения IMEI оборудование сети оператора (Loc\_Service) должно отправить POST-запрос на модуль Locator, в котором просит предоставить информацию о IMEI.

Формат запроса: **http://<host>:<port>/Imei?imsi=<IMSI>**,

где

- host — IP-адрес или домен Locator;
- port — порт для подключения через API;
- IMSI — IMSI абонента, по которому требуется узнать его текущий IMEI.

В случае успешно принятого запроса от Loc\_Service, модуль Locator отправляет IMEI в десятичном виде и сообщает о результате выполнения данного запроса.

Если IMSI абонента в базе данных модуля Locator отсутствует или в параметрах запроса была допущена ошибка, то на Loc\_Service отправится сообщение «not\_found».

Также модуль Locator осуществляет проверку данных с помощью подписки об изменении IMEI. При изменении информации о IMEI оператор сети сотовой связи подготовит и отправит уведомление на модуль Locator.

Сценарий получения информации о IMEI представлен на рисунке 10.

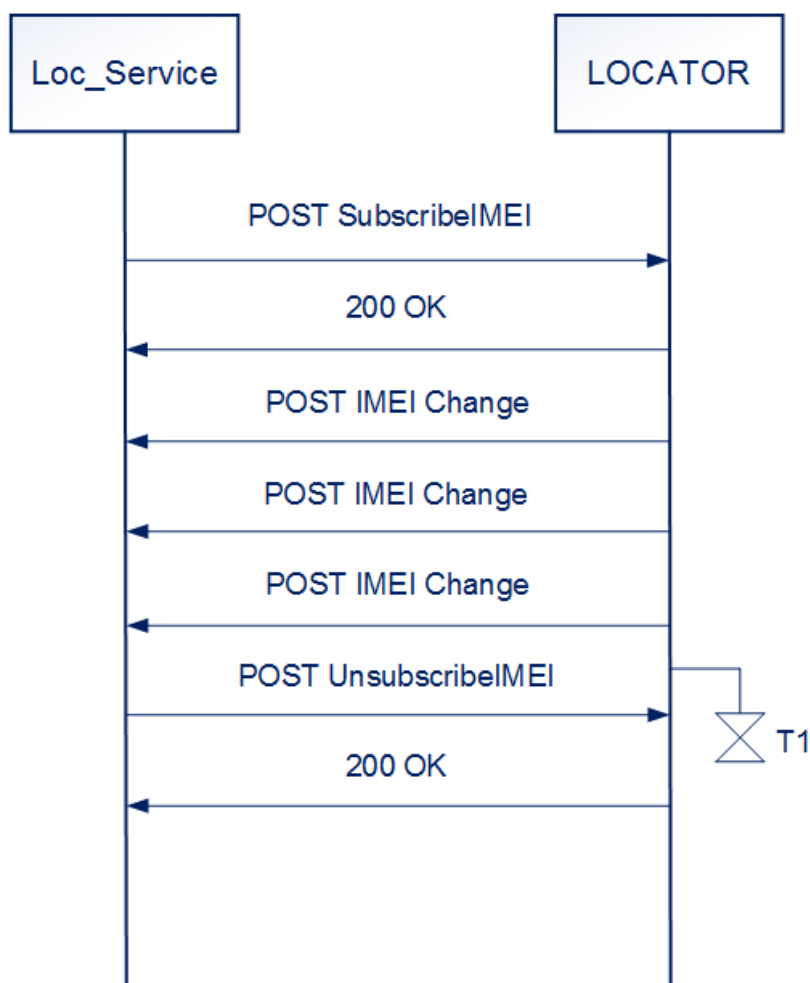


Рисунок 10 — Получение подписки на смену IMEI

Принцип работы:

1. Оборудование сети оператора (Loc\_Service) отправляет на модуль Locator POST-запрос SubscribeIMEI, содержащее следующие параметры: IMSI, уникальный идентификатор и длительность подписки, а также URL-адрес, на который будут отправляться уведомления.

**Примечание** — Значение таймера для подписки задается в конфигурационном файле locator.cfg.

2. В случае успешно выполненной подписки модуль Locator отправит сообщение «200 OK».

Если модуль Locator не смог распознать запрос на подписку, то он отправляет соответствующее сообщение об итоге выполнения запроса. Аналогичные типы сообщений, отправляемых на оборудование сети оператора, могут быть при получении подписки на смену IMEI в модуль Locator.

3. Информация об изменении IMEI отправляется сразу после осуществления подписки, а также после каждой смены IMEI. Вместе с идентификатором подписки модуль Locator будет отправлять индикацию подписки (активная или завершившаяся), IMSI, а также данные о старом и новом IMEI.

4. Для отмены подписки на изменение IMEI Loc\_Service отправляет на модуль Locator POST-запрос UnsubscribeIMEI, в котором будет указан идентификатор отменяемой подписки.

Отмена подписки может быть произведена по истечению времени (таймер подписки). Значения таймера указываются в конфигурационном файле locator.cfg

5. Для подтверждения отмены подписки модуль Locator отправит сообщение «200 OK».

В случае отмены несуществующей подписки модуль Locator отправит сообщение следующего вида «no\_subscription\_with\_that\_id». Если запрос, принятый от оборудования сети оператора, не соответствует JSON-формату или отсутствует идентификатор отменяемой подписки, то модуль Locator отправит сообщение «invalid».

### **Запрос**

POST-запрос: **/SubscribeImei**

Запрос описан в разделе 3.5.3 «Подписка на смену местоположения абонента».

**Примечание** — Если есть подписка на смену IMEI абонента, но в базе данных нет данного IMEI, POST-запрос не посылается.

### **Ответ**

Ответ описан в разделе 3.5.3 «Подписка на смену местоположения абонента».

### **Уведомления**

При успешной подписке на смену IMEI абонента модуль Locator начинает отправлять уведомления о смене IMEI абонента на адрес, указанный в поле contact.

Уведомления начинают отправляться сразу после осуществления подписки, а также после каждой смены IMEI.

Уведомления отправляются в формате POST-запроса. Тело POST-запроса представлено в JSON-формате.

Ниже приведён формат запроса для уведомления:

---

```
{
  "sub-id" : "<sub-id>",
  "sub-status" : "<sub-status>",
  "data" : {
    "imsi" : "<imsi>",
    "imei" : "<imei>"
  }
}
```

---

Параметры запроса описаны в таблице ниже.

Таблица 10 — Параметры запроса

Параметр	Значимость параметра	Описание
sub-id	O	Идентификатор подписки. Тип — string.
sub-status	O	Статус подписки. Тип — string. Возможные значения: active — индикация активной подписки; inactive — индикация завершившейся подписки. <b>Примечание</b> — В случае истечения таймера на подписку запрос посылается без данных data.
imsi	M	Международный идентификатор абонента мобильной сети. Тип — string.
imei	O	Международный идентификатор мобильного оборудования абонента. Тип — string.

Ниже приведён пример запроса для уведомления:

```
{
  "sub-id" : "f1158376c2e741d58dbb5f904a95e102",
  "sub-status" : "active",
  "data" : {
    "imsi" : "111111111111111",
    "imei" : "222222222222222"
  }
}
```

### 3.5.5 Отмена подписки на смену местоположения абонента

Данный запрос используется для отмены подписки на смену местоположения абонента.

#### **Запрос**

POST-запрос: **/UnsubscribeLocation**.

Запрос должен содержать:

1. Заголовок — Content-Type: application/JSON.
2. Тело POST-запроса в JSON-формате.

Ниже приведён формат запроса:

```
{  
  "sub-id" : "<sub-id>",  
}
```

Параметры запроса описаны в таблице ниже.

Таблица 11 — Параметры запроса

Параметр	Значимость параметра	Описание
sub-id	O	Идентификатор подписки. Тип — string.

Ниже приведён пример запроса:

```
{  
  "sub-id" : "f1158376c2e741d58dbb5f904a95e102",  
}
```

### Ответ

Ниже приведён формат ответа на запрос:

```
{  
  "status" : "<STATUS>"  
}
```

Параметры ответа описаны в таблице ниже.

Таблица 12 — Параметры ответа

Параметр	Значимость параметра	Описание
status	O	Статус выполнения запроса. Тип — string. Возможные значения: ok — запрос выполнен успешно; no_subscription_with_that_id — попытка продлить несуществующую подписку; invalid — запрос не соответствует JSON-формату или отсутствуют обязательные поля IMSI и contact; unknown — другая ошибка.

Ниже приведён пример ответа на запрос:

```
{  
  "status" : "ок"  
}
```

### 3.5.6 Отмена подписки на IMEI абонента

Данный запрос используется для отмены подписки на смену IMEI абонента.

#### **Запрос**

POST-запрос: **/UnsubscribeImei**.

Запрос описан в разделе 3.5.5 «Отмена подписки на смену местоположения абонента».

#### **Ответ**

Ответ описан в разделе 3.5.5 «Отмена подписки на смену местоположения абонента».

## 3.6 Повышение надежности и организация резервирования

Для повышения надежности работы модуля Locator осуществляется резервирование.

Резервирование осуществляется путем дублирования модулей. Каждому модулю Locator присваивается свой код пункта сигнализации (PC, Point Code). Глобальный адрес (GT, Global Title) выделяется один на всю систему.

Модули Locator не соединены между собой и могут работать в двух режимах: Active/Active и Active/Standby.

Модуль Locator осуществляет хранение данных обо всех абонентах в распределенном хранилище Data Grid с функциями In-Memory (таким образом хранение данных и их обработка происходит в оперативной памяти в реальном времени) базы данных на базе Hazelcast.

Схема резервирования приведена на рисунке 11.

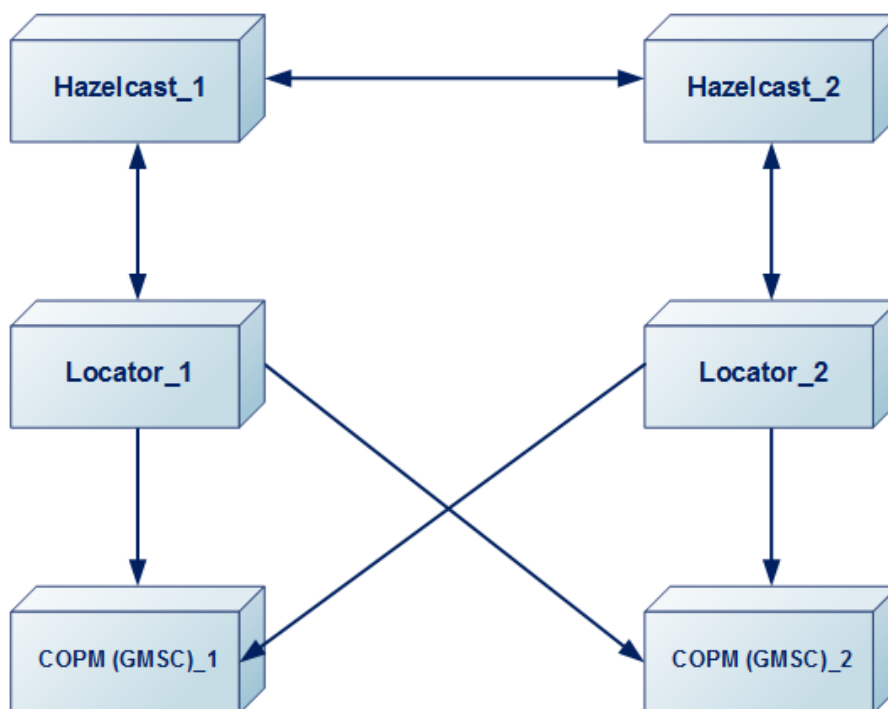


Рисунок 11 — Схема реализации резервирования

Требования к созданию и хранению резервных копий модулей Locator приведены в таблице 1.

Таблица 13 — Требования к созданию и хранению резервных копий Protei\_Locator

Объект	Вид резервной копии	Процедура создания резервной копии	Процедура восстановления из резервной копии	Объем
Настройки системы и приложений	Locator	Копировать в резервный файл согласно указанному пути:	Восстановить файл можно из резервной копии в место расположения на сервере	Восстановить файл можно из резервной копии в место расположения на сервере
		приложение /usr/protei/Protei_Locator/bin/Protei_Locator*		
		конфигурация /usr/protei/Protei_Locator/config/*.cfg		
		конфигурация /usr/protei/Protei_Locator/config/component/*.cfg		



Объект	Вид резервной копии	Процедура создания резервной копии	Процедура восстановления из резервной копии	Объем
		CDR /usr/protei/Protei_Locator/cdr/ *.cdr		
	Hazelcast	приложение /usr/protei/Hazelcast/bin/ Hazelcast*.jar		100M6
		приложение /usr/protei/Hazelcast/bin/lib/*		
		конфигурация /usr/protei/Hazelcast/config		
	eraser	приложение /usr/protei/eraser/bin/*		500к6
		приложение /usr/protei/eraser/Plugins/*		
		конфигурация /usr/protei/eraser/Config/ Setup.pm		

Объект	Вид резервной копии	Процедура создания резервной копии	Процедура восстановления из резервной копии	Объем
Операционная система	Операционная система	<p>Создание tarball ОС:</p> <p>Перед созданием tarball необходимо убедиться, что хватит свободного места.</p> <p>Для создания запускается команда от пользователя root</p> <pre>/usr/protei/utils/tarball/create.sh</pre> <p>После чего будет создан tarball в директории /usr/protei/log/backup с названием hel-<code>\$HOSTNAME</code>-YYYY-MMDD.tar.gz</p>	<p>Восстановление системы из tarball:</p> <ul style="list-style-type: none"> <li>– необходимо загрузиться с LiveCD Linux;</li> <li>– разбить и отформатировать разделы в соответствии с требованиями к системе;</li> <li>– выполнить mount разделов;</li> <li>– распаковать tarball командой <code>tar xvf rhel-<code>\$HOSTNAME</code>YYYY-MMDD.tar.gz</code>;</li> <li>– прописать загрузчик grub;</li> <li>– перезагрузить сервер.</li> </ul>	10Гб

Периодичность создания резервных копий объектов системы составляет:

- общие настройки системы и приложений — ежедневно;
- конфигурация операционной системы — раз в год.

Срок хранения резервных копий объектов составляет:

- общие настройки системы и приложений — три дня;
- конфигурация операционной системы — один год.

## 4 Настройка параметров конфигурации

Модуль Locator настраивается в следующих конфигурационных файлах:

- http.cfg — файл, содержащий настройки HTTP-соединения;
- om\_interface.cfg — файл, содержащий настройки OMI-подключений;
- locator.cfg — файл конфигурации ПО модуля Locator;
- sccp\_routing.cfg — файл, содержащий настройки маршрутизации сообщений SCCP по различным DPC;
- trace.cfg — файл, содержащий настройки подсистемы записи логов и CDR-файлов.

Настройки управления параметрами конфигурации сохраняются в файлах конфигурации в каталоге /<путь\_до\_Locator>/config.

### 4.1 Типы данных и условные обозначения

Типы данных описаны в таблице ниже.

Таблица 14 — Используемые обозначения для типов данных

Тип	Описание
bool	Логический тип. Принимает только значения 0 или 1, false или true соответственно. Используется для задания флага.
datetime	Тип для задания даты и времени. Используемые сокращения: YY/YYYY — год, записанный двумя/четырьмя цифрами, соответственно; MM — месяц, записанный двумя цифрами; DD — день, записанный двумя цифрами; hh — часы, записанные двумя цифрами; mm — минуты, записанные двумя цифрами; ss — секунды, записанные двумя цифрами; mss — миллисекунды, записанные тремя цифрами. Время задается в формате 24-часового дня.
int	Числовой тип. Задает целое 32-битное число, записанное цифрами 0–9 и знаком минуса "-". Диапазон: от $-2^{31}$ до $2^{31}-1$ .
hex	Числовой тип. Задает целое число в формате шестнадцатеричного числа, записанного цифрами 0–9 и буквами A–F. Числу может предшествовать обозначение 0x. При отсутствии обозначения определяется как строка.
list	Список, содержит несколько значений одной типа или структуры.
object	Кортеж, содержит фиксированное количество параметров различных типов.
string	Строковый тип. Может содержать буквы латинского алфавита, цифры 0–9, спецсимволы и знаки препинания.

Тип	Описание
ip	Строка типа string, имеет формат IPv4: xxx.xxx.xxx.xxx
regex	Строка типа string, регулярное выражение, задает маску, шаблон для формата данных.
float	Дробное число

При описании параметров также используются такие характеристики, как обязательность задания значения и возможность изменения значения без перезапуска. Указываются в колонке OMPR.

Таблица 15 — Буквенные коды

Тип	Описание
O	Optional. Опциональный параметр. Может отсутствовать в конфигурации, в таком случае используется значение по умолчанию.
M	Mandatory. Обязательный параметр. Его отсутствие не позволяет запустить систему, а после перезагрузки конфигурации отображается сообщение об ошибке.
P	Permanent. Параметр не переопределяется динамически, поскольку используется при запуске системы.
R	Reloadable. Параметр, значение которого можно переопределить без перезагрузки.

## 4.2 Настройка параметров HTTP-соединения (http.cfg)

Конфигурационный файл — *http.cfg*.

В файле настраиваются HTTP-соединения.

В таблице ниже описаны параметры конфигурационного файла.

Таблица 16 — Параметры http.cfg

Параметр	OMPR	Описание
Секция [Common]		
ParseAllHeaders	O/P	Параметр, определяющий разбор заголовков парсером. Тип — bool. Значение по умолчанию — 0. <b>Примечание</b> — Для разбора выборочных заголовков служит параметр Fields в секции [Server].
Секция [Server]		
ID	M/P	Идентификатор направления. Тип — int.

Параметр	OMPR	Описание
Address	O/R	Адрес для приема запросов. Тип — ip. Значение по умолчанию — «0.0.0.0» (слушаются все интерфейсы).
Port	M/R	Порт для приема запросов. Тип — int.
Fields	O/R	Список разбираемых заголовков. Тип — list, элементы — заголовки типа string. Формат записи: {<field_name1>; <field_name2>; ...}.
MaxBufferSize	O/R	Максимальный размер буфера для приема сообщений. Тип — int, измеряется в байтах. Значение по умолчанию — 65535.
RecvBufferSize	O/R	Минимальный размер буфера для приема сообщений для экономии ресурсов ввиду возможных специфик приложений. Тип — int, измеряется в байтах. Значение по умолчанию — 65536 б. <b>Примечание</b> — Если размер сообщения превышает размер буфера, то вместительность буфера будет увеличиваться вплоть до значения параметра MaxBufferSize.
ActivityTimer	O/R	Время ожидания активности во время соединения. По истечении заданного времени при отсутствии запросов разрывается соединение, а затем уничтожается объект HTTP_ServerCL. Тип — int, измеряется в секундах. Значение по умолчанию — 60 с.
SSL	M/R	Параметры, характеризующие безопасность передачи информации. См. раздел 4.2.1 «Конфигурация SSL». Тип — object.
Authorization	M/R	Параметры авторизации. См. раздел 4.2.2 «Конфигурация авторизации Authorization». Тип — object.
Секция [Client]		
ID	M/P	Идентификатор направления. Тип — int.

Параметр	OMPR	Описание
DestAddress	M/R	IP адрес и порт назначения, куда отправляются запросы. Тип — ip:port.
SrcAddress	O/R	IP-адрес источника, откуда направляются запросы. Тип — ip. Значение по умолчанию — 0.0.0.0.
Persistent	O/R	Активация режима работы с persistent-соединениями. Тип — bool. Значение по умолчанию — false.
ResponseTimer	O/R	Время ожидания ответа от сервера. Тип — int, измеряется в секундах. Значение по умолчанию — 60с.
MaxQueue	O/R	Максимальный размер очереди запросов для одного соединения (для persistent режима). Тип — int. Значение по умолчанию — 10.
MaxConnection	O/R	Максимальное количество одновременных соединений к серверу. Тип — int. Значение по умолчанию — 500.
ActivityTimer	O/R	Время ожидания запросов до разрыва постоянного соединения. Тип — int, измеряется в секундах. Значение по умолчанию — 60с.
MaxBufferSize	O/R	Максимальный размер буфера для приема сообщений. Тип — int, измеряется в байтах. Значение по умолчанию — "-1", т.е. 4 294 967 295 б (без ограничения).
RecvBufferSize	O/R	Минимальный размер буфера для приема сообщений для экономии ресурсов ввиду возможных специфик приложений. Тип — int, измеряется в байтах. Значение по умолчанию — 65536 б. <b>Примечание</b> — Если размер сообщения превышает размер буфера, то вместительность буфера будет увеличиваться вплоть до значения параметра MaxBufferSize.
SSL	M/R	Параметры, характеризующие безопасность передачи информации. См. раздел 4.2.1 «Конфигурация SSL». Тип — object.

Параметр	OMPR	Описание
Authorization	M/R	Параметры авторизации. См. раздел 4.2.2 «Конфигурация авторизации Authorization». Тип — object.

Пример файла конфигурации:

```
[Common]
```

```
[Server]
```

```
{
```

```
    ID = 1;
```

```
    Address = "127.0.0.1";
```

```
    Port = 7778;
```

```
}
```

## 4.2.1 Конфигурация SSL

Ниже описаны параметры SSL.

Таблица 17 — Параметры SSL

Параметр	OMPR	Описание
Секция [SSL]		
Version	O/P	Используемая версия SSL. Возможные значения: 1 — TLSv1; 2 — SSLv2; 3 — SSLv3. Тип — int. Значение по умолчанию — 2.
CertificatePath	M/R	Путь до файла сертификата сервера с расширением *.pem. Тип — string.
PrivateKeyPath	M/R	Путь до файла с приватным ключом сервера с расширением *.pem. Тип — string.
Ciphers	O/R	Поддерживаемые схемы шифрования. Тип — list, элементы — названия стандартов типа string. Полное описание возможных значений дано в <a href="#">OpenSSL Manpages</a> . Формат записи: c1:c2:...:cN.

Параметр	OMPR	Описание
PreferServerCiphers	O/R	Параметр, определяющий приоритет сервера при выборе шифра. Тип — bool. Значение по умолчанию — false.

#### 4.2.2 Конфигурация авторизации Authorization

Ниже описаны параметры авторизации Authorization на сервере.

Таблица 18 — Параметры Authorization

Параметр	OMPR	Описание
Type	O/R	Вид авторизации. Тип — string. Возможные значения: Basic/Digest. Значение по умолчанию — Basic. <b>Примечание</b> — Схема Digest на данный момент не реализована.
User	M/R	Логин для авторизации. Тип — string.
Password	O/R	Пароль для авторизации. Тип — string.

#### 4.3 Настройка конфигурации программного обеспечения (locator.cfg)

Конфигурационный файл — *locator.cfg*.

В файле настраиваются параметры конфигурации ПО модуля Locator.

В таблице ниже описаны параметры конфигурационного файла.

Таблица 19 — Параметры locator.cfg

Параметр	OMPR	Описание
Cluster	O/P	Список адресов узлов Hazelcast. Тип — list, элементы — object. Формат: IP_address:port, где IP_address — IP адрес узла Hazelcast (O/P, тип — string); port — порт узла Hazelcast (O/P, тип — int).



Параметр	OMPR	Описание
HlrGT	O/P	Глобальный адрес узла HLR, на который отправляются ATI-запросы. Тип — int. <b>Примечание</b> — Если значение данного параметра не задано, то ATI-запросы отправляются согласно MSISDN.
LocalGT	O/P	Параметр глобального адреса модуля Locator. Номер, который будет присвоен модулю Locator (GTNumberDigits Locator). Тип — int.
LocalNP	O/P	Параметр глобального адреса модуля Locator. План нумерации, который будет присвоен модулю Locator (GTNumberingPlan Locator). Тип — int.
NormalizeImei	O/P	Параметр, определяющий нормализацию получаемых IMEI. Значение IMEI должно содержать 15 знаков. Тип — bool. Значение по умолчанию — 0.
Profiling	O/P	Параметр, определяющий вывод информации о потреблении CPU определенными частями функционала (профилирование). Тип — bool.
AtiRpsLimit	O/P	Максимальное количество ATI-запросов в секунду со всех модулей Locator. Тип — int. Значение по умолчанию — 0 (без ограничения).
AllowEmptyCellID	O/P	Параметр, разрешающий нулевой идентификатор соты. Тип — bool.
TrafficStatistics	O/P	Параметры подсчета статистики сигнальных сообщений. См. раздел 4.3.1 «Параметры подсчета статистики сигнальных сообщений TrafficStatistics». Тип — object.

#### Пример файла конфигурации:

```
Cluster = {
  "10.77.161.68:5702";
}
#HlrGT=79585949977;
HlrGT=79809119977;
#HlrGT=79585949938;
LocalGT=79585949936;
```

```
LocalNP=1;
NormalizeImei = 1;
Profiling = 1;
AtiRpsLimit = 10;
AllowEmptyCellID = 0;
TrafficStatistics = {
  Intervals = {
    Check = 60000;
    Out = 3600;
  }
}
```

### 4.3.1 Параметры подсчета статистики сигнальных сообщений TrafficStatistics

В таблице ниже описаны параметры подсчета статистики сигнальных сообщений.

Таблица 20 — Параметры TrafficStatistics

Параметр	OMPR	Описание
LogName	O/P	Наименование журнала, в который будет выводиться статистика. Тип — string. Значение по умолчанию — traffic_stat.
Intervals	O/P	Параметры временных интервалов. См. раздел «Параметры временных интервалов Intervals». Тип — object.
Subscription	O/P	Параметры подписок. Тип — object.

#### **Параметры временных интервалов Intervals**

В таблице ниже описаны параметры временных интервалов Intervals.

Таблица 21 — Параметры Intervals

Параметр	OMPR	Описание
Check	O/P	Период подсчета количества транзакций в секунду. Тип — int, измеряется в миллисекундах. Значение по умолчанию — 60000.

Параметр	OMPR	Описание
Out	O/P	Период вывода максимального количества транзакций в секунду. Тип — int, измеряется в секундах. Значение по умолчанию — 3600.
Online	O/P	Период вывода статистики. Тип — int, измеряется в секундах. Значение по умолчанию — 15.

### **Параметры таймеров подписок Subscription**

В таблице ниже описаны параметры таймеров подписок Subscription.

Таблица 22 — Параметры Subsscription

Параметр	OMPR	Описание
MinExpires	O/P	Минимальный интервал подписки. Тип — int, измеряется в секундах. Значение по умолчанию — 600.
MaxExpires	O/P	Максимальный интервал подписки. Тип — int, измеряется в секундах. Значение по умолчанию — 86400.
DefExpires	O/P	Период подписки по умолчанию, если не задан в запросе от клиента. Тип — int, измеряется в секундах. Значение по умолчанию — 3600.

## **4.4 Настройка параметров OMI-подключений (om\_interface.cfg)**

Конфигурационный файл — *om\_interface.cfg*.

В файле настраивается связь между функциональными модулями по OMI-интерфейсу.

Файл перезагружается командой

---

```
./reload om_interface.cfg
```

---

В таблице ниже описаны параметры конфигурационного файла.

Таблица 23 — Параметры `om_interface.cfg`

Параметр	OMPR	Описание
Секция [General]		
ServerIP	O/R	Адрес динамического OM-сервера. Тип — ip.
ServerPort	O/R	Порт динамического OM-сервера. Тип — int. Значение по умолчанию — 0.
Timers	M/R	Параметры временных интервалов. См. раздел 4.4.1 «Конфигурация временных интервалов Timers». Тип — object.
Секция [Client]		
Timers	M/R	Параметры временных интервалов. См. раздел 4.4.1 «Конфигурация временных интервалов Timers». Тип — object.
ConnectionLogics	O/P	Параметры, описывающие сетевую логику ConnectionLogics. См. раздел 4.4.2 «Конфигурация сетевой логики ConnectionLogics». Тип — list, элементы — строки типа object.
Directions	M/P	Параметры направлений автоподключений. См. раздел 4.4.4 «Конфигурация направлений Directions». Тип — list, элементы — строки типа object.
Секция [Server]		
Timers	M/R	Параметры временных интервалов. См. раздел 4.4.1 «Конфигурация временных интервалов Timers». Тип — object.

Параметр	OMPR	Описание
Sockets	O/P	<p>Параметры сокет-интерфейса. См. раздел 4.4.3 «Конфигурация сокет-интерфейса Sockets».</p> <p>Тип — list, элементы — строки типа object. Формат:</p> <pre>Sockets= {   {     Address= #addr;     Port= #port;   } }</pre>
ConnectionLogics	O/P	<p>Параметры, описывающих сетевую логику ConnectionLogics. См. раздел 4.4.2 «Конфигурация сетевой логики ConnectionLogics».</p> <p>Тип — list, элементы — строки типа object.</p>
Directions	M/P	<p>Параметры направлений автоподключений. См. раздел 4.4.4 «Конфигурация направлений Directions».</p> <p>Тип — list, элементы — строки типа object.</p>

#### Пример файла конфигурации:

---

```
[General]
```

```
ServerIP= 0.0.0.0;
```

```
ServerPort= 31017;
```

```
Timers =
```

```
{
```

```
    SessionResponseTimeOut = 600000;
```

```
    TransactionResponseTimeOut = 30000;
```

```
    SegmentResponseTimeOut = 10000;
```

```
    MaxSegmentErrorCount = 3;
```

```
    LoginReqTimeOut = 10000;
```

```
    ReconnectTimeOut = 10000;
```

```
    KeepAliveTimeOut = 30000;
```

```
    KeepAliveResponseTimeOut = 10000;
```

```
}
```

```
[Server]
```

```
Timers=
```

```
{
```

```
TransactionResponseTimeOut=10000;  
MaxSegmentErrorCount= 1;  
LoginReqTimeOut= 10000;  
KeepAliveTimeOut= 30000;  
KeepAliveResponseTimeOut= 10000;  
}
```

Sockets=

```
{  
  {  
    Address= 0.0.0.0;  
    Port= 31018;  
  }  
}
```

ConnectionLogics=

```
{  
  CL.0={  
    Priority=1;  
    MaxTransactionCount=1000;  
    Login=support;  
    Password=elephant;  
  }  
}
```

Directions=

```
{  
  Dir.Locator=  
  {  
    CL_Monitoring=0;  
    ChangeOverTimeOut=5000;  
    Primary={  
      MaxTraffic= 10;  
      Connections= {0};  
    }  
  }  
}
```

---

#### 4.4.1 Конфигурация временных интервалов Timers

Ниже описаны параметры временных интервалов Timers.

Таблица 24 — Параметры Timers

Параметр	OMPR	Описание
SessionResponse TimeOut	O/R	Максимальное время существования сессии. Тип — int, измеряется в миллисекундах. Значение по умолчанию — 0.
Transaction ResponseTimeOut	O/R	Максимальное время существования транзакции. Тип — int, измеряется в миллисекундах. Значение по умолчанию — 0.
SegmentResponse TimeOut	O/R	Время ожидания ответа на запрос. Тип — int, измеряется в миллисекундах. Значение по умолчанию — 0.
MaxSegmentError Count	M/R	Максимальное количество ошибок SegmentError, при превышении которого сетевая логика закрывает текущую сессию. Тип — int.
LoginReqTimeOut	M/R	Временной интервал после установления соединения, по истечении которого в случае отсутствия сообщения LoginReq серверная сетевая логика разорвет соединение. Тип — int, измеряется в миллисекундах.
Reconnect TimeOut	M/R	Время ожидания клиентской сетевой логики до очередной попытки соединения. Тип — int, измеряется в миллисекундах.
KeepAliveTimeOut	M/R	Временной интервал, по истечении которого в случае отсутствия сетевой активности будет отправлено сообщение KeepAlive. Тип — int, измеряется в миллисекундах.
KeepAlive ResponseTimeOut	M/R	Время ожидания подтверждения KeepAlive_ACK на запрос KeepAlive. Тип — int, измеряется в миллисекундах.

#### 4.4.2 Конфигурация сетевой логики ConnectionLogics

Ниже описаны параметры сетевой логики ConnectionLogics.

Таблица 25 — Параметры ConnectionLogics

Параметр	OMPR	Описание
CL.#N={};#N	M/P	Идентификатор сетевой логики. Тип — int.
Priority	M/P	Приоритет сетевой логики при распределении новой транзакции. Тип — int.
MaxTransactionCount	M/P	Максимальное количество транзакций, одновременно обрабатываемых данной сетевой логикой. Тип — int.
IP	M/P	IP-адрес сервера автоподключения. Тип — ip.
Port	M/P	Номер порта сервера автоподключения. Тип — int.
Login	O/P	Логин для авторизации на сервере автоподключения. Тип — string.
Password	O/P	Пароль для авторизации на сервере автоподключения. Тип — string.
Info	O/R	Дополнительная информация. Тип — string.

#### 4.4.3 Конфигурация сокет-интерфейса Sockets

Ниже описаны параметры сокет-интерфейса Sockets.

Таблица 26 — Параметры Sockets

Параметр	OMPR	Описание
Address	O/P	IP-адрес сокет-интерфейса. Тип — ip.
Port	M/P	Прослушиваемый порт сокет-интерфейса. Тип — int.



#### 4.4.4 Конфигурация направлений Directions

Ниже описаны параметры направлений Directions.

Таблица 27 — Параметры Directions

Параметр	OMPR	Описание
Dir.#N={};#N	M/P	Идентификатор направления. Тип — int.
CL_Monitoring	O/P	Флаг наблюдения за сетевыми логиками. Возможные значения: 1 — сетевая логика будет посылать примитив OM_CONNECTION_RESUME_IND в случае установления соединения и OM_CONNECTION_PAUSE_IND в случае разрыва соединения; 0 — сетевая логика не посылает данные примитивы. Тип — bool. Значение по умолчанию — 0.
ChangeOverTimeOut	O/P	Время ожидания восстановления первичных подключений после их отключения, по истечении которого передача входящих запросов будет осуществляться на вторичные подключения. Тип — int. Значение по умолчанию — 0.
Primary	M/P	Параметры первичного подключения. См. раздел «Конфигурация подключений». Тип — object.
Secondary	O/P	Параметры вторичного подключения. См. раздел «Конфигурация подключений». Тип — object.

#### Конфигурация подключений

Ниже описаны параметры первичного и вторичного подключений.

Таблица 28 — Параметры подключений

Параметр	OMPR	Описание
Секция [Primary], [Secondary]		
MaxTraffic	M/P	Максимальное количество транзакций, одновременно обрабатываемых на данном направлении. Тип — int.

Параметр	OMPR	Описание
Connections	M/P	Список используемых логик CL. Тип — list, элементы — идентификаторы типа int. Формат: <code>Connections = { #idCL1, #idCLN };</code>

## 4.5 Настройка параметров системы журналирования (trace.cfg)

Конфигурационный файл — *trace.cfg*.

В файле настраивается система журналирования.

Файл перезагружается командой

```
./reload_trace.cfg
```

В таблице ниже описаны параметры конфигурационного файла.

Таблица 29 — Параметры trace.cfg

Параметр	OMPR	Описание
Секция [Trace]		
common — O/R — общие настройки системы журналирования, тип — object.		
tracing	O/R	Флаг активности системы журналирования. Тип — bool. Значение по умолчанию — 1.
dir	O/R	Путь к директории, в которой находятся журналы. Тип — string. ./ — путь берётся относительно текущего каталога; / — путь берется от корня. Иначе — от каталога по умолчанию. <b>Примечание</b> — Путь может содержать ".." и маску формата времени. Значение по умолчанию — <code>"/logs."</code>
no_signal	O/R	Набор сигналов, не перехватываемых системой журналирования. Все остальные сигналы отражаются в журналах. Тип — list, элементы — int, разделитель — ",", запятая. Значение all — не перехватывать никакие сигналы. Значение по умолчанию — перехватывать все сигналы (intercept all).

Параметр	OMPR	Описание
logs — O/R — конфигурация журналов, тип — object. Формат: name = { params }		
name	O/R	<p>Наименование журнала, служит идентификатором журнала в системе.</p> <p>Тип — string.</p> <p><b>Примечание</b> — В имени журнала допускаются символы, цифры, знак подчеркивания.</p> <p>Существует несколько зарезервированных имен:</p> <ul style="list-style-type: none"> <li>– trace — журнал «по умолчанию»;</li> <li>– info — отражает информационные события: создание и течение времени жизни компонентов;</li> <li>– warning — журнал предупреждений;</li> <li>– config — журнал, отображающий всю информацию, считанную с конфигурационных файлов;</li> <li>– OM_trace — журнал событий OMI-интерфейса;</li> <li>– OM_warning — журнал предупреждений OMI-интерфейса;</li> <li>– SCCP_trace — журнал SCCP «по умолчанию»;</li> <li>– SCCP_warning — журнал предупреждений SCCP;</li> <li>– TCAP_trace — журнал TCAP «по умолчанию»;</li> <li>– M3UA_trace — журнал M3UA «по умолчанию»;</li> <li>– M3UA_warning — журнал предупреждений M3UA;</li> <li>– M3UA_info — информационный журнал M3UA;</li> <li>– traffic_stat — журнал статистики по трафику;</li> <li>– loc_info — журнал, отображающий детальную информацию по обработке сообщений Note-MM-Event.</li> </ul>

Параметр	OMPR	Описание
period	O/R	<p>Период обновления лог-файла.</p> <p>Тип — object. Формат: interval + shift</p> <p>interval — промежуток времени между соседними обновлениями;</p> <p>shift — первоначальный сдвиг.</p> <p>interval и shift имеет подпараметры: count type.</p> <p>См. раздел 4.5.2 «Модификаторы period».</p> <p><b>Примечание</b> — Сдвиг не может быть больше длины периода и в случае некорректного значения игнорируется.</p> <p>Пример: day+3hour — файл обновляется каждый день в 03:00:00.</p>
mask	O/R	<p>Маска формата вывода автоматических полей в журнале.</p> <p>Тип — string.</p> <p>Подпараметры: date/time/tick/state/pid/tid/level/file.</p> <p>См. раздел 4.5.1 «Модификаторы mask».</p> <p>Пример маски: date &amp; time &amp; tick &amp; state &amp; pid &amp; tid &amp; level &amp; file.</p>
file	O/R	<p>Путь к файлу лог.</p> <p>Тип — string.</p> <p>./ — путь берётся относительно текущего каталога.</p> <p>/ — путь берется от корня</p> <p>Иначе — от каталога по умолчанию.</p> <p>Путь может содержать ".." и маску формата времени.</p> <p><b>Примечание</b> — При указании несуществующих директорий система создает все необходимые каталоги. Допускается задание пустого имени файла, если значение параметра level равно 0. В этом случае запись производится согласно параметру tee. В случае отсутствия этого параметра, запись на диск не производится.</p> <p>Пример: cdr/%Y/%m/%d/%H_%M_%S.log =&gt; cdr/2004/07/07/13_54_31.log.</p>
separator	O/R	<p>Разделитель автоматических полей.</p> <p>Тип — string. Значение по умолчанию — значение common.</p> <p><b>Примечание</b> — Вывод date, time, tick считается одним полем.</p>

Параметр	OMPR	Описание
buffering	O/R	<p>Настройки буферизированной записи.</p> <p>Тип — object.</p> <p>Подпараметры: cluster_size/clusters_in_buffer/overflow_action.</p> <p>См. раздел 4.5.3 «Модификаторы buffering».</p>
tee	O/R	<p>Дублирование потока вывода.</p> <p>Тип — string. Возможные значения: stdout/cout/info/имя любого лога.</p> <p><b>Примечание</b> — При знаке минуса "-" не пишется имя исходного лога при дублировании.</p> <p>Пример: tee = stdout &amp; cout &amp; info &amp; any_log_file.</p>
limit	O/R	<p>Максимальное количество строк в файле.</p> <p>Тип — int.</p> <p><b>Примечание</b> — Как только достигнут предел строк, лог автоматически открывается заново. При этом не исследуется реальное количество строк в файле на данный момент. Если имя файла зависит от времени, то открывается новый файл, иначе файл обнуляется.</p>
level	O/R	<p>Уровень журнала.</p> <p>Тип — int.</p> <p><b>Примечание</b> — Сообщения с уровнем большим, чем значение, игнорируются.</p>
type	O/R	<p>Тип журнала и дополнительные настройки.</p> <p>Тип — string.</p> <p>Подпараметры: name_now/name_period, truncate/append, log/cdr.</p> <p>См. раздел 4.5.4 «Модификаторы type».</p> <p>Примеры: type = cdr &amp; name_period — cdr с именем файла по началу периода; type = append — log без обнуления файлов.</p>

Пример файла конфигурации:

```
[Trace]
common={tracing=1; dir="./logs";}

logs=
{
    config =
    {
```

```
        file = "logs/config.log";
        mask = time & tick & file;
        level=10;
        period=hour;
};

trace =
{
    file="logs/trace/trace-%Y%m%d-%H%M.log";
    mask=file & date & time & tick;
    level=10;
    period=hour;
    separator=";";
};

info =
{
    file="logs/info.log";
    mask=date & time & tick;
    level=10;
};

warning =
{
    file=" logs/warning.log";
    mask=date & time & tick;
    level=10;
};

OM_trace =
{
    file = "logs/om_trace.log";
    period = hour;
    mask = date & time & tick & file;
    separator = ";";
    level = 10;
};

OM_warning =
{
    file = "logs/om_trace.log";
    period = hour;
    mask = date & time & tick & file;
    separator = ";";
```

```
        level = 6;

};

SCCP_trace =
{
    file=" logs/sccp_trace.log";
    mask=date & time & tick & pid & file;
    level=0;
};

SCCP_warning =
{
    file=" logs/sccp_warning.log";
    mask=date & time & tick & pid & file;
    period=1hour;
    level=0;
};

TCAP_trace =
{
    file=" logs/tcap/tcap_trace-%Y%m%d-%H%M.log";
    mask=date & time & tick & pid & file;
    level=10;
};

M3UA_trace =
{
    file=" logs/m3ua_trace.log";
    mask=date & time & tick & pid & file;
    level=10;
};

M3UA_info =
{
    file=" logs/m3ua_info.log";
    mask=date & time & tick & pid & file;
    level=0;
};

M3UA_warning =
{
    file=" logs/m3ua_warning.log";
    mask=date & time & tick & pid & file;
    level=0;
};
```

```

traffic_stat =
{
    file="logs/traffic_stat.log";
    mask=date & time;
    level=10;
};
loc_info =
{
    file="cdr/loc_info-%Y%m%d-%H%M.cdr";
    mask=date & time;
    separator=";";
    period=1hour;
    level=10;
}

```

#### 4.5.1 Модификаторы mask

Ниже описаны модификаторы параметра.

Таблица 30 — Модификаторы mask

Параметр	Описание
date	Дата создания. Тип — datetime. Формат — DD/MM/YY.
time	Время создания. Тип — datetime. Формат — hh:mm:ss.
tick	Миллисекунды. Тип — int. Формат: если задано time — .mss, три цифры; если не задано time — .mssmss, шесть цифр.
state	Состояние системы. Тип — int или string.
pid	Идентификатор процесса. Тип — int. Формат — шесть цифр.
tid	Идентификатор потока. Тип — int. Формат — шесть цифр.
level	Уровень журнала для записи. Тип — int.



Параметр	Описание
file	Файл и строка в файле с исходным кодом, откуда производится вывод. Тип — string.

#### 4.5.2 Модификаторы period

Ниже описаны модификаторы параметра.

Таблица 31 — Модификаторы period

Параметр	Описание
count	Текущее время для имени файла. Количество стандартных периодов. Тип — int. Значение по умолчанию — 1.
type	Вид временного интервала. Тип — string. Возможные значения: sec/min/hour/day/week/month/year.

#### 4.5.3 Модификаторы buffering

Ниже описаны модификаторы параметра.

Таблица 32 — Модификаторы buffering

Параметр	Описание
cluster_size	Размер кластера в килобайтах. Тип — int, измеряется в килобайтах. Значение по умолчанию — 128 Кб.
clusters_in_buffer	Длина буфера в кластерах. Тип — int. Значение по умолчанию — 0.
overflow_action	Действие при переполнении буфера. Тип — string. Возможные значения: erase — удаление; dump — запись. Значение по умолчанию — dump.

#### 4.5.4 Модификаторы type

Ниже описаны модификаторы параметра.

Таблица 33 — Модификаторы type

Параметр	Описание
name_now	Текущее время для имени файла.
name_period	Время для имени файла, начало периода.
truncate	Файл при открытии обнуляется.
append	Файл при открытии не обнуляется, а дописывается.
log	Состоит из truncate и name_now, при падении пишется информация о сигнале.
cdr	Состоит из append и name_now, при падении не пишется информация о сигнале.

#### 4.6 Настройка параметров маршрутизации сообщений SCCP (sccp\_routing.cfg)

Конфигурационный файл — *sccp\_routing.cfg*.

В файле настраиваются параметры маршрутизации сообщений SCCP.

Файл перезагружается командой

```
./reload sccp_routing.cfg
```

В таблице ниже описаны параметры конфигурационного файла.

Таблица 34 — Параметры sccp\_routing.cfg

Параметр	OMPR	Описание
Секция [DefaultRouting] Настройки по умолчанию		
DPC	M/R	Код основного пункта назначения. Тип — int. Диапазон значений: [0—65535]
RDPC	O/R	Код резервного пункта назначения. Тип — int.

Параметр	OMPR	Описание
NI	O/R	Идентификатор сети основного пункта назначения. Тип — int. Возможные значения: 0 — International network; 1 — Spare (for international use only); 2 — National network; 3 — Reserved for national use.
RNI	O/R	Индикатор сети резервного пункта назначения. Тип — int. Возможные значения аналогичны описанным для NI.
Секция [Routing]		
GT_AddrDigit	M/R	Маска номера GT. Тип — regex.
DPC	M/R	Код пункта назначения. Тип — int.
RDPC	O/R	Код резервного пункта назначения. Тип — int.
NI	O/R	Идентификатор сети основного пункта назначения. Тип — int.
RNI	O/R	Идентификатор сети резервного пункта назначения. Тип — int.

#### Пример файла конфигурации:

---

```
[DefaultRouting]
```

```
DPC={845;};
```

```
NI=3;
```

---

```
[Routing]
```

---

## 5 Техническое обслуживание системы Locator

### 5.1 Доступ с использованием команды telnet

Команда telnet служит для установления удаленного сеанса связи. Для этого необходима авторизация пользователя (ввод имени пользователя и пароля). Если номер порта не указан, telnet использует для связи с сервером с номером порта по умолчанию. Вместо имени сервера может использоваться его IP-адрес. Telnet работает на базе протокола TCP. Доступные команды приведены в таблице ниже.

Таблица 35 — Команды telnet

Команда	Значение
cd path	переход в каталог path
cd..	переход в родительский каталог
cd /	переход в корневой каталог
ps afx	просмотр запущенных процессов
Pwd	просмотр текущего каталога
ls -l	просмотр содержимого текущего каталога
df -h	информация об использовании места на диске (флеш-накопитель и жесткий диск)
date	просмотр текущей даты

Утилиты приведены в таблице ниже.

Таблица 36 — Утилиты

Утилита	Значение
./start	инициализирует загрузку данных и запускает ПО конвертера
./stop	остановка системы и всех утилит
./tracer	обеспечивает протоколирование обмена данными в сигнальном канале
./trace	обеспечивает вывод на экран обмена данными в сигнальном канале

Для запуска утилит необходимо находиться в каталоге:  
/usr/protei/<путь\_до\_Locator>/.

Перед именем утилиты обязательно нужно писать точку и знак слэш «/».

## 5.2 Управление работой программного обеспечения

Управление работой системы Locator осуществляется скриптами. Работа скриптов фиксируется в отдельном файле.

Основными скриптами управления работой программного обеспечения являются: `start`, `stop`, `restart`, `version`, `reload`, `trace`. Использование основных скриптов предоставляет следующие функциональные возможности:

- запуск/остановка системы;
- перезагрузка системы;
- просмотр версии программного обеспечения;
- динамическая перезагрузка системы;
- вывод в лог-файл информации о текущем состоянии системы.

Основные скрипты, необходимые для управления системой, находятся в корневом каталоге системы `/usr/protei/<путь_до_Locator>/`.

Ниже приведен перечень и описание основных скриптов:

- **start** — штатный запуск (перезапуск) системы. Возможен запуск скрипта при запущенной системе, так как скрипт при старте все останавливает. Сохраняет лог-файлы. Только затем запускается система.
- **stop** — останавливает систему и все утилиты.
- **reload** — интерфейс к динамической перегрузке конфигурации.
- **start\_utils** — вспомогательный скрипт. Запускается автоматически при старте системы (`start`). Запускает все вспомогательные утилиты. Перед запуском останавливает все запущенные утилиты. Запуск вручную возможен при отсутствии утилит в памяти.
- **stop\_utils** — вспомогательный скрипт. Останавливает все вспомогательные утилиты. Запуск вручную возможен.
- **trace** — вспомогательный скрипт. Выводит в лог-файл `./logs/out.log` информацию о потоках системы и вспомогательных скриптов, а также о занятой памяти.

## 5.3 Перезапуск системы

При изменении параметров должен производиться перезапуск системы.

**Примечание** — Путем автоматического перезапуска системы также может осуществляться автоматическая локализация аварий (сбой ПО системы).

Для перезапуска используется последовательность команд `./stop;./start`, которая инициализирует загрузку данных и запускает ПО.

Для остановки программного обеспечения используется команда `./stop`.

## 5.4 Проверка работоспособности

### 5.4.1 Диагностика системы

Диагностика системы осуществляется по лог-файлам, которые ведутся в системе Locator. Для доступа к лог-файлам необходимо:

1. запустить консоль.
2. ввести команду: `telnet hostname`, где `hostname` — IP-адрес сервера, на котором установлен программный модуль Locator.

3. после соединения ввести регистрационное имя (login) и далее пароль.
4. после успешной регистрации перейти в каталог `usr/protei/<путь_до_Locator>`.

В системе ведется лог-файл `err.log`, в котором фиксируются все ошибки, возникшие при эксплуатации системы.

Если после запуска системы в лог-файле `err.log` зафиксирована ошибка, то необходимо осуществить перезапуск системы. Подробное описание действий, выполняемых при перезагрузке системы, приведено в разделе 5.3 «Перезапуск системы».

В случае возникновения ошибки в лог-файле `err.log` повторно обратитесь в службу технической поддержки ООО «НТЦ ПРОТЕЙ».

Если перезапуск системы был выполнен успешно, то лог-файл `err.log` не будет содержать никаких записей. Для корректной работы необходимо, чтобы система Locator формировала журналы (CDR-записи различных типов). Описание работы с журналами приведено в разделе 7 «Журналы».

#### 5.4.2 Диагностика работы подсистемы M3UA

Диагностика работы подсистемы M3UA осуществляется по лог-файлу `m3ua_trace.log`, который ведется в системе Locator.

Для доступа к данному лог-файлу необходимо с помощью консоли выполнить подключение к системе. Подробное описание действий, выполняемых пользователем, приведено в предыдущем пункте. После успешной регистрации перейдите в каталог `usr/protei/<путь_до_Locator>`.

Далее необходимо открыть лог-файл `m3ua_trace.log` и найти запись нужного соединения и проверить. Если соединение работает в нормальном режиме, то в конце строки будет выведено значение `==[ UP ]==`.

Запись о состоянии каждого соединения выводится в отдельной строке.

Пример записи:

---

```
2018-09-11 16:28:19.991 032483 ASP.cpp(546) M3UA: Sg.SIGTRAN.M3UA.ASP.0: ==[ UP ]==
```

---

#### 5.4.3 Диагностика взаимодействия с GTP-Probe

Проверка корректного взаимодействия модуля Locator и GTP-Probe осуществляется с помощью лог-файла `http_trace.log`, который ведется в системе Locator.

Для доступа к данному лог-файлу необходимо с помощью консоли выполнить подключение к системе. Подробное описание действий, выполняемых пользователем, приведено в разделе 5.4.1 «Диагностика системы». После успешной регистрации перейдите в каталог `usr/protei/<путь_до_Locator>/trace`.

Далее необходимо открыть лог-файл `http_trace.log` и проверить наличие сообщений (received message), содержащих данные, связанные с получением информации о местонахождении абонентов и IMEI устройств.

Пример записи:

---

```
2018-07-20 00:00:01;FSM: Sg.HTTP.SDir.1.SCL.1859940 received message:
POST /UpdatePsLocation HTTP/1.1
Content-Type: application/json    //mime_part_header
```

---

[

```
{
  "IMEI" : "867xxxxxxxxxxxxx",
  "IMSI" : "250xxxxxxxxxxxxx",
  "MSISDN" : "8831xxxxxxxxxxxxx",
  "location" : {
    "CELL LAC" : 45030,
    "LAC" : 65534,
    "MCC" : 250,
    "MNC" : 1,
    "RAC" : 255,
    "SAC" : 31189
  },
  "location_type" : 0
}
```

---

#### 5.4.4 Диагностика трафика с M3UA

Проверка корректного получения трафика с M3UA осуществляется с помощью лог-файла `tcap_trace.log`, который ведется в системе Locator.

Для доступа к данному лог-файлу необходимо с помощью консоли выполнить подключение к системе. Подробное описание действий, выполняемых пользователем, приведено в разделе 5.4.1 «Диагностика системы». После успешной регистрации перейдите в каталог `usr/protei/<путь_до_Locator>/tcap`.

Далее необходимо открыть лог-файл `tcap_trace.log` и проверить наличие сообщений TCAP: Received PDU.

Пример записи:

---

```
2018-06-28 18:04:08.868 003121 TCAP_Handler.cpp(722) TCAP: Received PDU:
```

```
begin {
  otid = 4 octets {
    00 00 14 ff
  }
  dialoguePortion = {
    dialogueExternal = {
      objectID = 0.0.17.773.1.1.1
      asnl = {
        value = 17 octets {
          60 0f 80 02 07 80 a1 09 06 07 04 00 00 01 00 2a   *
          03
        }
      }
    }
  }
}
```

```
    }
  }
  components = 1 entries {
    [0]=invoke {
      invokeID = 0
      operationCode = 89
      parameter = 52 octets {
        30 32 02 01 17 80 01 00 81 08 52 40 72 37 47 63    02      R@r7Gc
        72 f2 82 07 91 97 85 67 99 19 f2 a3 17 02 01 00    r      g
        81 07 91 97 11 02 90 29 f6 a3 09 80 07 52 f0 10      )      R
        00 f0 2a 96                                          *
      }
    }
  }
}
```

---

#### 5.4.5 Проверка работы приложений

Для проверки наличия активного процесса работы приложения Locator следует воспользоваться консолью. Выполните открытие консоли и введите следующую команду: *ps afx| grep Locator*.

В результате выполнения данной команды будет выведена строка с процессом. Если процесс с именем Locator есть, то приложение работает корректно.

Если в результате выполнения данной команды не будет выведена никакая информация, то приложение работает некорректно.

В случае возникновения проблем обратитесь в службу технической поддержки ООО «НТЦ ПРОТЕЙ».



## 6 Статистика

### 6.1 Статистическая информация по трафику

Система Locator позволяет формировать статистику по трафику.

Статистика по трафику выводится в файл traffic\_stat.log. Каждая запись выводится в одной строке. Значения разделены точкой с запятой.

Формат:

```
DateTime; OpCode; MsgName; SuccessTrCount; FailTrCount; ErrorTrCount;  
TransitTrCount; MaxSpeed; %SuccessTrCount; %FailTrCount; %TransitTrCount;
```

В таблице ниже описаны поля статистики.

Таблица 37 — Поля traffic\_stat.log

№	Параметр	Описание
1	DateTime	Дата и время начала формирования статистики в формате YYYY-MM-DD hh-mm-ss. Тип — datetime.
2	OpCode	Код операции. Тип — int.
3	MsgName	Краткое наименование транзакции. Тип — string.
4	SuccessTrCount	Количество успешных транзакций. Тип — int.
5	FailTrCount	Количество неуспешных транзакций. Тип — int.
6	ErrorTrCount	Количество неуспешных транзакций с разбивкой по ошибкам. Формат: {ErrorCode1;Count2;};{ErrorCode2;Count2};...{ErrorCodeN;CountN} Тип — list, элементы — object.
7	TransitTrCount	Количество транзитных транзакций (SAI, SRIFSM). Тип — int.

№	Параметр	Описание
8	MaxSpeed	Максимальное количество транзакций в секунду, посчитанное за заданный промежуток времени. Тип — float. <b>Примечание</b> — Промежуток времени задается в конфигурационном файле locator.cfg.
9	%SuccessTrCount	Количество успешных транзакций в процентах. Тип — float.
10	%FailTrCount	Количество неуспешных транзакций в процентах. Тип — float.
11	%TransitTrCount	Количество транзитных транзакций в процентах (SAI, SRIFSM). Тип — float.

Пример файла статистики:

```

2018-01-09 08:44:55 ;0;All;0;0;{};-1;0.00;0.00;0.00;0.00
2018-01-09 08:45:55 ;0;All;0;0;{};-1;0.00;0.00;0.00;0.00
2018-01-09 08:46:55 ;0;All;0;0;{};-1;0.00;0.00;0.00;0.00
2018-01-09 08:47:55 ;0;All;0;0;{};-1;0.00;0.00;0.00;0.00
2018-01-09 08:48:55 ;0;All;0;0;{};-1;0.00;0.00;0.00;0.00
2018-01-09 08:49:55 ;0;All;0;0;{};-1;0.00;0.00;0.00;0.00
2018-01-09 08:50:55 ;0;All;1;0;{};-1;0.00;1.00;0.00;-1.00
2018-01-09 08:50:55 ;89;NoteMM;1;0;{};-1;0.00;1.00;0.00;-1.00
2018-01-09 08:51:55 ;0;All;0;0;{};-1;0.00;0.00;0.00;0.00
2018-01-09 08:52:55 ;0;All;0;0;{};-1;0.00;0.00;0.00;0.00
2018-01-09 08:53:55 ;0;All;0;0;{};-1;0.00;0.00;0.00;0.00
2018-01-09 08:54:55 ;0;All;0;0;{};-1;0.00;0.00;0.00;0.00

```

## 7 Журналы

### 7.1 Общие принципы организации журналов

Подсистема Tm\_Trace позволяет строить древовидную иерархическую структуру журналов системы. Каждая подсистема пишет свои журналы. Журналы представлены в виде отдельных папок. Каждое событие выводится в отдельной строке. Значения разделены знаком «;». Журналы всех подсистем организованы в виде дерева папок. Перемещаясь по его узлам, можно посмотреть файлы всех отдельных журналов событий. Корнем дерева журналов всегда является тип Trace.

Каждая подсистема ведет несколько типов журналов:

#### 1. Trace.

В данном журнале содержится последовательность выполненных команд и значения переменных на данном шаге выполнения программы (трассировка работы подсистемы). Глубина трассировки управляется параметром level. Сообщения, уровень которых выше, чем настроенный level данного журнала, будут игнорироваться системой. Допустимые уровни трассировки, доступные пользователю, представлены в таблице ниже.

Таблица 38 — Уровни трассировки

Уровень	Описание
1 – 3	Специфично для подсистемы.
4	Выводится информация о состояниях автоматов, обмене сигналами, событиях.
5	Выводится информация о состояниях автоматов, обмене сигналами, событиях, параметрах сигналов и событий, внутренние сообщения и переменные.
6 и выше	Специфично для подсистемы.

#### 2. Warning.

В данном журнале содержится информация о возникших ошибках в работе трассируемой подсистемы.

#### 3. Info.

В данном журнале содержится информация о событиях, которая представлена для пользователя в упрощенном виде. Система выводит следующую информацию:

- создание, удаление и модификация ресурсов;
- проход программой некоторых значимых точек.
- состояние внутренних подсистем или окружения приложения.

Помимо подсистемы Tm\_Trace часть журналов ведется скриптовой подсистемой, в том числе перенаправленные стандартные потоки вывода приложения.

### 7.1.1 Стандартный вывод и журнал скриптовой подсистемы

Стандартные потоки вывода в системе — потоки процесса, имеющие номер, зарезервированный для выполнения некоторых «стандартных» функций.

#### **Стандартный вывод (stdout)**

Поток stdout зарезервирован для вывода данных, фиксирующих версии запускаемых подсистем и процесс их запуска. Запись данных происходит только в процессе старта системы. С помощью глобального параметра OUT\_FILE, задаваемого в конфигурационном файле scripts.sh, осуществляется корректный вывод значений.

Каталог, в котором хранятся все данные журналов размещается в out.log. Каждая запись выводится в одной строке.

Пример конфигурации:

---

```
OUT_FILE=./logs/out.log
```

---

Пример записи:

---

```
*****
Protei Locator (Location DB via Note MM)
1.1.9.0
*****
```

Supported features:

SgGeneral	4.0.2.18
M3UA	4.0.1.43
SCCP	4.0.6.9
TCAP	3.3.14.1
MAP	4.1.1.0
HTTP	1.2.6.0

---

#### **Стандартный вывод ошибок (stderr)**

Поток stderr зарезервирован для вывода диагностических и отладочных сообщений в текстовом формате для пользователя. С помощью потока stderr фиксируются критичные ошибки в работе приложения. При корректной работе должен оставаться пустым.

Вывод значений осуществляется с помощью глобального параметра ERR\_FILE, задаваемого в конфигурационном файле scripts.sh.

Каталог, в котором хранятся все данные журналов размещается в err.log.

Пример конфигурации:

---

```
ERR_FILE=./logs/err.log
```

---

Пример записи:

---

```
terminate called after throwing an instance of 'std::length_error'
what():  basic_string::assign
```

---

### Стандартный вывод информации о системе (system)

Поток system зарезервирован для фиксирования информации от некоторых системных метрик (объектов системы) таких как утилита who, free, df -h, ps. С помощью данных, выводимых в данном потоке, пользователь может проследить за изменениями, происходящими с системой в определенный момент времени.

Вывод значений осуществляется с помощью глобального параметра SYSTEM\_LOG, задаваемого в конфигурационном файле scripts.sh.

Каталог, в котором хранятся все данные журналов размещается в out.log.

Пример конфигурации:

---

```
SYSTEM_LOG=./logs/out.log
```

---

Пример записи:

---

```
date: 28/08/18 17:42:40
who:
support pts/2          2018-08-28 11:39 (62.152.87.202)
free:
      total        used        free      shared    buffers     cached
Mem:      3924080    2844120    1079960         472     171960    1177724
-/+ buffers/cache:    1494436    2429644
Swap:      4063228          0     4063228
df:
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/vg_proteigmscorm-lv_root
                        50G       44G   3,1G   94% /
tmpfs                      1,9G         0   1,9G    0% /dev/shm
/dev/sda1                  477M       65M   388M   15% /boot
/dev/mapper/vg_proteigmscorm-lv_home
                        45G      553M   42G    2% /home
ps:
  PID   PPID  %CPU  %MEM    VSZ   STAT  CMD
15134     1   2.7   0.6 120028  Sl    ./bin/Locator-1
15138     1   0.1   0.0 106120  S     /bin/bash ./bin/utils/checker.sh
```

---

### Журнал скриптовой подсистемы (journal)

В данном журнале содержится информация о взаимодействии пользователя и скриптовой подсистемы, а также фиксируются действия, выполненные самими скриптами в автоматическом режиме.

Вывод значений осуществляется с помощью глобального параметра LOG\_FILE, задаваемого в конфигурационном файле scripts.sh.

Каталог, в котором хранятся все данные журналов, размещается в journal.txt.

Пример конфигурации:

---

```
LOG_FILE=./logs/journal.txt
```

---

**Пример записи:**

```
2018-08-28 17:37:08 ./restart by support:root from localhost
2018-08-28 17:37:08 move_log.sh : move 9 logs from ./logs to
./logs/history/2018.08.28_17.37.08
2018-08-28 17:38:33 arch_log.sh : archive 9 logs from ./logs/history before
2018-08-28 17:38:38 arch_log.sh : archive 9 logs from ./logs/history after
```

## 7.1.2 Базовые журналы приложения

### **Основной журнал приложения (trace)**

В данном журнале содержится информация о всех событиях подсистемы, а также полная информация о процессах обработки различных запросов.

Рекомендуемые значения глубины трассировки для различных режимов представлены в таблице ниже.

Таблица 39 — Значение глубины трассировки

Наименование режима	Рекомендуемое значение параметра level
Режим отладки (Debug)	10
Режим Production	0

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

**Пример конфигурации:**

```
trace = {
    file="logs/trace/trace-%Y%m%d-%H%M.log";
    mask=file & date & time & tick;
    level=10;
    period=hour;
    separator=" ";
};
```

**Пример записи:**

```
2018-08-28 17:00:00.005;PrimFSM.cpp(70);FSM: Locator id: 0; < St: 0 ACTIVE
2018-08-28 17:00:00.006;HTTP_Connection.cpp(173);HTTP_Connection id: 3776432 (27)
10.141.0.20:80->10.136.77.41:32816; disconnect by network cause: 0
2018-08-28 17:00:00.006;PrimFSM.cpp(35);FSM: Sg.HTTP.SDir.1.SCL.3776428 id: 0; >
St: 2 DISCONNECT Pr: 184:10 SI_DISCONNECT_CONF ((0, 3776431))
2018-08-28 17:00:00.006;PrimFSM.cpp(70);FSM: Sg.HTTP.SDir.1.SCL.3776428 id: 0; <
St: 0 NUL
2018-08-28 17:00:00.006;PrimFSM.cpp(35);FSM: Sg.HTTP.SDir.1.SCL.3776429 id: 0; >
St: 1 ACTIVE Pr: 184:13 SI_DISCONNECT_IND ((0, 3776432))
2018-08-28 17:00:00.006;PrimFSM.cpp(70);FSM: Sg.HTTP.SDir.1.SCL.3776429 id: 0; <
St: 2 DISCONNECT
```

```

2018-08-28 17:00:00.006;PrimFSM.cpp(35);FSM: Sg.HTTP.SDir.1 id: 0; > St: 0 NUL Pr:
34:3 CM_DESTROY_REQ ()
2018-08-28 17:00:00.006;HTTP_ServerDirection.cpp(54);destroying
Sg.HTTP.SDir.1.SCL.3776428
2018-08-28 17:00:00.006;PrimFSM.cpp(70);FSM: Sg.HTTP.SDir.1 id: 0; < St: 0 NUL
2018-08-28 17:00:00.007;PrimFSM.cpp(35);FSM: Sg.HTTP.SDir.1.SCL.3776429 id: 0; >
St: 2 DISCONNECT Pr: 184:10 SI_DISCONNECT_CONF ((0, 3776432))
2018-08-28 17:00:00.007;PrimFSM.cpp(70);FSM: Sg.HTTP.SDir.1.SCL.3776429 id: 0; <
St: 0 NUL
2018-08-28 17:00:00.007;PrimFSM.cpp(35);FSM: Sg.HTTP.SDir.1 id: 0; > St: 0 NUL Pr:
34:3 CM_DESTROY_REQ ()
2018-08-28 17:00:00.007;HTTP_ServerDirection.cpp(54);destroying
Sg.HTTP.SDir.1.SCL.3776429
2018-08-28 17:00:00.007;PrimFSM.cpp(70);FSM: Sg.HTTP.SDir.1 id: 0; < St: 0 NUL
2018-08-28 17:00:00.162;SCTP_Connection.cpp(174);(1, 1) RX(43616) 01 00 01 01 00
00 00 80 02 10 00 76 00 00 01 E3 00 00 00 0B 03 03 00 05 09 81 03 0E 19 0B 52 93
00 11 04 97 85 55 95 89 00 0B 12 06 00 11 04 97 85

```

### Журнал предупреждений (warning)

В данном журнале содержится информация о некритичных ошибках в работе приложения. Данные, выводимые в журнале, позволяют пользователю в короткие сроки определить причины некорректной работы приложения.

Рекомендуемые значения глубины трассировки для различных режимов представлены в таблице ниже.

Таблица 40 — Значение глубины трассировки

Наименование режима	Рекомендуемое значение параметра level
Режим отладки (Debug)	10
Режим Production	10

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

```

warning = {
    file="logs/warning.log";
    mask=date & time & tick;
    level=10;
};

```

Пример записи:

```

2018-08-20 15:41:23.248 AP: No LOG alarm_cdr1/alarm_cdr
2018-08-20 15:42:11.594 Absence imei in SubscriberInfo. IMSI=250xxxxxxxxxxxx
MSISDN=79xxxxxxxxxx VLR=

```





Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

---

```
config = {  
    file="logs/config.log";  
    mask=date & time & tick;  
    level=10;  
};
```

---

Пример записи:

---

```
17:37:13.860 OM_ConfigDynaData.cpp(281) config/om_interface.cfg [General] ServerIP  
= 0.0.0.0  
17:37:13.860 OM_ConfigDynaData.cpp(293) config/om_interface.cfg [General]  
ServerPort = 31017  
17:37:13.860 OM_ConfigDynaData.cpp(308) config/om_interface.cfg [General] Timers:  
SessionResponseTimeout isn't valid.  
17:37:13.860 OM_ConfigDynaData.cpp(330) config/om_interface.cfg [General] Timers:  
TransactionResponseTimeout = 10000  
17:37:13.860 OM_ConfigDynaData.cpp(338) config/om_interface.cfg [General] Timers:  
SegmentResponseTimeout isn't valid.  
17:37:13.860 OM_ConfigDynaData.cpp(361) config/om_interface.cfg [General] Timers:  
MaxSegmentErrorCount = 1  
17:37:13.860 OM_ConfigDynaData.cpp(376) config/om_interface.cfg [General] Timers:  
LoginReqTimeout = 10000  
17:37:13.860 OM_ConfigDynaData.cpp(391) config/om_interface.cfg [General] Timers:  
ReconnectTimeout = 2000  
17:37:13.860 OM_ConfigDynaData.cpp(406) config/om_interface.cfg [General] Timers:  
KeepAliveTimeout = 30000  
17:37:13.860 OM_ConfigDynaData.cpp(421) config/om_interface.cfg [General] Timers:  
KeepAliveResponseTimeout = 10000  
17:37:13.860 OM_ConfigDynaData.cpp(457) config/om_interface.cfg [Server] Timers:  
TransactionResponseTimeout = 10000
```

---

### ***Журнал статистики по трафику (traffic\_stat)***

В данном журнале содержится статистика по GSM MAP трафику.

Таблица 40 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

---

```
traffic_stat =  
{  
    file="cdr/stat/traffic_stat-%Y%m%d-%H%M.log";  
    mask=date & time;  
    period=1day;  
    level=10;  
};
```

---

Пример записи:

```
2018-08-29 15:41:13 ;0;All;11816;1041;{{11;24}};{34;8};{49;1009};};-1;0.00;0.92;0.
2018-08-29 15:41:13 ;71;ATI;936;1041;{{11;24}};{34;8};{49;1009};};-1;0.45;0.47;0.5
2018-08-29 15:41:13 ;89;NoteMM;10880;0;{};-1;0.00;1.00;0.00;-1.00
```

### **Журнал, отображающий детальную информацию по обработке сообщений Note-MM-Event (loc\_info)**

По умолчанию журнал находится по адресу /usr/protei/Protei\_Locator/cdr/loc\_info-%Y%m%d-%H%M.cdr

В данном журнале отображается информация, связанная с получением информации о местонахождении абонентов и IMEI устройств.

Формат имени файла: loc\_info-YYYYMMDD-hhmm.cdr, где YYYY — год, MM — месяц, DD — день, hh — часы, mm — минуты.

Формат:

```
DateTimeT; MSISDN; IMSI; IMEI; LAC; CellID; RAI; MSC; SGSN; EventMet; MessageID;
VLR; MCC; MNC; ServiceKey;
```

Для каждого принятого или отправленного сообщения Note-MM-Event пишется одна или несколько записей в журнал. Каждая запись выводится в одной строке. Значения разделены знаком «;».

В таблице ниже описаны поля журнала.

Таблица 41 — Поля журнала loc\_info

№	Параметр	Описание
1	DateTime	Дата и время формирования записи. Тип — datetime.
2	MSISDN	Номер мобильного абонента цифровой сети. Тип — string.
3	IMSI	Международный идентификатор мобильного абонента (индивидуальный номер абонента). Тип — string.
4	IMEI	Международный идентификатор мобильного оборудования. Тип — string. Значение по умолчанию — не задано.
5	LAC	Код зоны местоположения. Тип — int или hex.

№	Параметр	Описание
6	CellID	Идентификатор соты. Тип — int или hex.
7	RAI	Идентификатор зоны маршрутизации. Тип — hex.
8	MSC	Адрес MSC. Тип — string.
9	SGSN	Адрес SGSN. Тип — string.
10	EventMet	Код события MM, EventMet. Тип — int. Возможные значения: 0 — Location-update-in-same-VLR; 1 — Location-update-to-other-VLR; 2 — IMSI-Attach; 3 — MS-initiated-IMSI-Detach; 4 — Network-initiated-IMSI-Detach.
11	MessageID	Идентификатор сообщения. Тип — string. Значение по умолчанию — не задано.
12	VLR	Адрес VLR. Тип — string.
13	MCC	Мобильный код страны. Тип — int.
14	MNC	Код мобильной сети. Тип — int.
15	ServiceKey	Уникальный идентификатор каждой из имеющихся CAMEL-услуг. Тип — string.
16	Location_type	Код сети, откуда получена информация об абоненте. Тип — int. Возможные значения: 0 — 2G/3G; 2 — 4G/LTE.

Таблица 40 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

```
loc_info =  
{  
    file="cdr/loc_info-%Y%m%d-%H%M.cdr";  
    mask=date & time;  
    separator=" ";  
    period=5min;  
    level=10;  
};
```

Пример записи:

```
2018-02-07  
12:55:58;79xxxxxxxx;250xxxxxxxxxxxx;01366xxxxxxxx;592;53701;;;0;;;250;1;;2;  
2018-02-07  
12:56:01;79xxxxxxxx;250xxxxxxxxxxxx;35370xxxxxxxx;317;10101;;;0;;;250;1;;2;  
2018-02-07  
12:56:04;79xxxxxxxx;250xxxxxxxxxxxx;869205xxxxxxxx;860;64569;;;0;;;250;1;;2;
```

### 7.1.3 Журналы подсистемы M3UA

#### **Основной журнал подсистемы M3UA (M3UA\_trace)**

В данном журнале фиксируются все события, происходящие в подсистеме M3UA. С помощью данной информации пользователь может определить статус соединений, а также проследить этапы обработки сообщений на уровне M3UA.

Таблица 39 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

```
M3UA_trace = {  
    file="logs/m3ua/m3ua_trace -%Y%m%d-%H%M.log";  
    period = 1hour;  
    mask=date & time & tick;  
    level=10;  
};
```

Пример записи:

```
2018-08-28 17:37:17.016 015134 Decoder.cpp(57) M3UA: DECODER RAW_DATA: 01 00 02 03  
00 00 00 10 00 12 00 08 00 00 00 06  
2018-08-28 17:37:17.016 015134 ASP.cpp(424) M3UA: Sg.SIGTRAN.M3UA.ASP.0: -=[ DOWN  
]=-  
2018-08-28 17:37:17.016 015134 ASP.cpp(432) M3UA: Sg.SIGTRAN.M3UA.ASP.0: Got  
Pr_M3UA_ASPUP_ACK
```

```
2018-08-28 17:37:17.017 015134 AS.cpp(240) M3UA: Sg.SIGTRAN.M3UA.AS.0: --[
IN_SERVICE ]==
2018-08-28 17:37:17.017 015134 AS.cpp(250) M3UA: Sg.SIGTRAN.M3UA.AS.0:
Pr_CM_STATE_CHANGE_IND from Sg.SIGTRAN.M3UA.ASP.0 (ACTIVE)
2018-08-28 17:37:17.017 015134 AS.cpp(1301) M3UA:
Sg.SIGTRAN.M3UA.AS.0:[ProcessOStateChange] - Sg.SIGTRAN.M3UA.ASP.0 (ACTIVE)
2018-08-28 17:37:17.017 015134 AS.cpp(1800) M3UA: Sg.SIGTRAN.M3UA.AS.0:[SendASPAC]
- ASP:Sg.SIGTRAN.M3UA.ASP.0
2018-08-28 17:37:17.017 015134 ASP.cpp(546) M3UA: Sg.SIGTRAN.M3UA.ASP.0: --[ UP
]==
2018-08-28 17:37:17.017 015134 ASP.cpp(724) M3UA: Sg.SIGTRAN.M3UA.ASP.0: Got
Pr_M3UA_ASPAC from Sg.SIGTRAN.M3UA.ASP.0
2018-08-28 17:37:17.017 015134 ASP.cpp(546) M3UA: Sg.SIGTRAN.M3UA.ASP.0: --[ UP
]==
2018-08-28 17:37:17.017 015134 ASP.cpp(546) M3UA: Sg.SIGTRAN.M3UA.ASP.0: --[ UP
]==
2018-08-28 17:37:17.017 015134 ASP.cpp(546) M3UA: Sg.SIGTRAN.M3UA.ASP.0: --[ UP
]==
2018-08-28 17:37:17.017 015134 ASP.cpp(724) M3UA: Sg.SIGTRAN.M3UA.ASP.0: Got
Pr_M3UA_ASPAC from Sg.SIGTRAN.M3UA.AS.0
2018-08-28 17:37:17.017 015134 ASP.cpp(968) M3UA:
Sg.SIGTRAN.M3UA.ASP.0:[SendToSGP] - ASPAC
```

---

### **Журнал ошибок и предупреждений M3UA (M3UA\_warning)**

В данном журнале фиксируются все ошибки и предупреждения, происходящие в подсистеме M3UA. С помощью данной информации пользователь может увидеть причину обрыва соединения или отказа обработки сообщения.

Таблица 40 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

---

```
m3ua_warning = {
    file="logs/m3ua/m3ua_warning.log";
    mask=date & time & tick;
    level=10;
};
```

---

Пример записи:

---

```
2018-08-07 10:05:46.740 003650 ASP.cpp(1002) M3UA:
Sg.SIGTRAN.M3UA.ASP.0:[SendToAS] - cannot find AS(key=0)
2018-08-07 10:05:46.740 003650 ASP.cpp(1088) M3UA:
Sg.SIGTRAN.M3UA.ASP.0:[SendToDefaultAS] - Failed to send NTFY
```

---

### **Информационный журнал подсистемы M3UA (M3UA\_info)**

В данном журнале содержится информация о подсистеме M3UA. С помощью предоставляемых данных пользователь может определить версию подсистемы, с которой была собрана данная версия приложения.

Таблица 40 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

---

```
m3ua_info = {  
    file="logs/m3ua/m3ua_info.log";  
    mask=date & time & tick;  
    level=10;  
};
```

---

Пример записи:

---

```
2018-08-28 19:21:58.201 001087 ASP.h(54) M3UA:  
COMPANY:      Protei  
PRODUCT:      Signalling: SIGTRAN - M3UA  
PRODUCT_CODE: 4.0.1.32  
ADDITIONAL_INFO: RFC3332  
BUILD_NUMBER: 52  
VERSION_DATE:  Jul 27 2017 00:56:52  
HOST_NAME:    unknown host  
CVS_VERSION:  
MAKE_FILE:    /var/build/x86/2017-07-  
27/ATE/Signalling/SIGTRAN/M3UA/4_0/lib/makefile.SIGTRAN_M3UA
```

---

## **7.1.4 Журналы подсистемы TCAP**

### **Основной журнал подсистемы TCAP (TCAP\_trace)**

В данном журнале фиксируются все события, происходящие в подсистеме TCAP. С помощью данной информации пользователь может определить проследить все этапы обработки сигнальных сообщений на уровне TCAP.

Таблица 39 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

---

```
TCAP_trace = {  
    file="logs/tcap/tcap_trace -%Y%m%d-%H%M.log";  
    period = 1hour;  
    mask=date & time & tick;  
    level=10;  
};
```

---

**Пример записи:**

```
2018-08-20 14:12:16.154 018488 PrimFSM.cpp(35) FSM: Sg.SS7.TCAP(Sg.SS7.TCAP) id:
0; > St: 0 IN_SERVICE Pr
: 2112:0 TCAP_INVOKE_BEGIN_REQ ((3, -1) la: 72700076061 ra: 1111111111 i: 0 o: 3
s: 1, seg: 0)
2018-08-20 14:12:16.235 018488 PrimFSM.cpp(35) FSM: TCAP.Handler id: 3; > St: 0
NUL Pr: 2112:0 TCAP_INVOK
E_BEGIN_REQ ((3, -1) la: 72700076061 ra: 1111111111 i: 0 o: 3 s: 1, seg: 0)
2018-08-20 14:12:16.253 018488 TCAP_Handler.cpp(504) TCAP: Created message
begin {
    otid = 4 octets {
        00 00 00 04
    }
    dialoguePortion = {
        dialogueExternal = {
            objectID = 0.0.17.773.1.1.1
            asnl = {
                value = 17 octets {
                    60 0f 80 02 07 80 a1 09 06 07 04 00 00 01 00 02
                    03
                }
            }
        }
    }
    components = 1 entries {
        [0]=invoke {
            invokeID = 0
            operationCode = 3
            parameter = 20 octets {
                a3 12 30 10 04 08 74 00 04 10 83 32 10 f0 04 04
                00 00 00 00
            }
        }
    }
}
```

**Журнал ошибок и предупреждений TCAP (TCAP\_warning)**

В данном журнале фиксируются все ошибки и предупреждения, происходящие в подсистеме TCAP. С помощью данной информации пользователь может увидеть причину недоступности соединения с верхним/нижним уровнем или невозможность декодировать уровень TCAP.

Таблица 40 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

```
TCAP_warning = {
    file="logs/tcap/tcap_warning.log";
    mask=date & time & tick;
    level=10;
};
```

Пример записи:

```
2018-05-06 10:05:26.672 000365 TCAP.cpp(112) TCAP: Warning invalid binding: -1
2018-05-06 10:05:47.610 000365 TCAP.cpp(112) TCAP: Warning invalid binding: -1
2018-05-06 10:06:22.069 000365 TCAP.cpp(112) TCAP: Warning invalid binding: -1
```

### Журнал статистики TCAP (TCAP\_stat)

В данном журнале содержится статистика уровня TCAP.

Рекомендуемые значения глубины трассировки для различных режимов представлены в таблице ниже.

Таблица 42 — Значение глубины трассировки

Наименование режима	Рекомендуемое значение параметра level
Режим отладки (Debug)	-
Режим Production	10

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Формат:

```
<DateTime>; TCAP;Abort:< Abort.Sent.Total>;< Abort.Sent.Period>;<
Abort.Recv.Total>;< Abort.Recv.Period>;Begin:< Begin.Sent.Total>;<
Begin.Sent.Period>;< Begin.Recv.Total>;< Begin.Recv.Period>;Continue:<
Cont.Sent.Total>;< Cont.Sent.Period>;< Cont.Recv.Total>;< Cont.Recv.Period>;End:<
End.Sent.Total>;< End.Sent.Period>;< End.Recv.Total>;< End.Recv.Period>
```

Пример конфигурации:

```
tcap_stat = {
    file="logs/tcap/tcap_stat.log";
    mask=date & time & tick;
    level=10;
};
```



---

**Пример записи:**

---

```
2018-08-27 16:48:00
TCAP;Abort:0;0;0;0;Begin:29;2;50;2;Continue:90;0;171;0;End:27;0;54;0;

2018-08-27 16:48:01
TCAP;Abort:0;0;0;0;Begin:29;0;50;0;Continue:97;7;185;14;End:27;0;54;0;

2018-08-27 16:48:03
TCAP;Abort:0;0;0;0;Begin:29;0;50;0;Continue:98;1;187;2;End:29;2;58;4;

2018-08-27 16:48:04
TCAP;Abort:0;0;0;0;Begin:30;1;52;2;Continue:101;3;193;6;End:29;0;58;0;
```

---

## 7.1.5 Журналы подсистемы SCCP

### **Основной журнал подсистемы SCCP (SCCP\_trace)**

В данном журнале фиксируются все события, происходящие в подсистеме SCCP. С помощью данной информации пользователь может определить проследить все этапы обработки сигнальных сообщений на уровне SCCP.

Таблица 39 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

---

```
SCCP_trace = {
    file="logs/sccp/sccp_trace -%Y%m%d-%H%M.log";
    period = 1hour;
    mask=date & time & tick;
    level=10;
};
```

---

**Пример записи:**

---

```
2018-08-20 14:12:16.253 018488 PrimFSM.cpp(35) FSM: Sg.SS7.SCCP(Sg.SS7.SCCP) id:
0; > St: 0 IN_SERVICE Pr
: 2080:0 SCCP_UDT_REQ (Data size 560, SLS 3, CdPA
SSN=7(VLR);GT=(E.164);1111111111, CgPA SSN=6(HLR);GT=
(E.164);72700076061)

2018-08-20 14:12:16.253 018488 PrimFSM.cpp(35) FSM: SCLC id: 0; > St: 0 ACTIVE Pr:
2080:0 SCCP_UDT_REQ (D
ata size 560, SLS 3, CdPA SSN=7(VLR);GT=(E.164);1111111111, CgPA
SSN=6(HLR);GT=(E.164);72700076061)

2018-08-20 14:12:16.253 018488 PrimFSM.cpp(35) FSM: SCRC id: 0; > St: 0 ACTIVE Pr:
2080:0 SCCP_UDT_REQ (D
ata size 560, SLS 3, CdPA SSN=7(VLR);GT=(E.164);1111111111, CgPA
SSN=6(HLR);GT=(E.164);72700076061)

2018-08-20 14:12:16.294 018488 SCRC.cpp(364) SCCP: DefaultDest
2018-08-20 14:12:16.359 018488 SCRC.cpp(247) SCCP: Sending message:
UDT:
```

Protocol Class: 1

Message Handling: return on error

CdPA =

Routing Indicator = Route on GT  
SSN = 7 (VLR)  
Global title indicator = 4 (transln,n\_plan,enc\_sch,nai)  
TranslationType = 0 (not used)  
Numbering plan = 1 (E.164)  
Nature of address indicator = 4 (international number)  
Encoding Scheme = 2 (BCD)  
Address = 1111111111

CgPA =

Routing Indicator = Route on GT  
SSN = 6 (HLR)  
Global title indicator = 4 (transln,n\_plan,enc\_sch,nai)  
TranslationType = 0 (not used)  
Numbering plan = 1 (E.164)  
Nature of address indicator = 4 (international number)  
Encoding Scheme = 1 (BCD)  
Address = 72700076061

Data: 62 44 48 04 00 00 00 04 6B 1E 28 1C 06 07 00 11 86 05 01 01 01 A0 11 60 0F  
80 02 07 80 A1 09 06 07  
04 00 00 01 00 02 03 6C 1C A1 1A 02 01 00 02 01 03 A3 12 30 10 04 08 74 00 04 10  
83 32 10 F0 04 04 00 00  
00 00

2018-08-20 14:12:16.361 018488 PrimFSM.cpp(70) FSM: SCRC id: 0; < St: 0 ACTIVE

2018-08-20 14:12:16.361 018488 PrimFSM.cpp(70) FSM: SCLC id: 0; < St: 0 ACTIVE

---

### ***Журнал ошибок и предупреждений SCCP (SCCP\_warning)***

В данном журнале фиксируются все ошибки и предупреждения, происходящие в подсистеме SCCP.

Таблица 42 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

---

**Пример конфигурации:**

```
SCCP_warning = {  
    file="logs/sccp/sccp_warning.log";  
    period = 1hour;  
    mask=date & time & tick;  
    level=10;  
};
```

---

**Пример записи:**

```
2018-08-23 07:01:09.605 075172 SCRC.cpp(337) SCCP: PC present = 14191, but route  
on GT  
2018-08-23 07:06:25.364 075172 SCRC.cpp(337) SCCP: PC present = 6305, but route on  
GT
```

---

## 7.1.6 Журналы подсистемы M2PA

### **Основной журнал подсистемы M2PA (M2PA\_trace)**

В данном журнале фиксируются все события, происходящие в подсистеме M2PA. С помощью данной информации пользователь может определить состояние соединений, доступность персонального компьютера пользователя, а также проследить этапы обработки сообщений на уровне M2PA.

Таблица 39 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

**Пример конфигурации:**

---

```
M2PA_trace = {  
    file="logs/m2pa/m2pa_trace -%Y%m%d-%H%M.log";  
    period = 1hour;  
    mask=date & time & tick;  
    level=10;  
};
```

---

**Пример записи:**

---

```
2018-08-23 13:25:16.978 029585 5 HMRT.cpp(430) M2PA: route [ DPC=6981; APC=6981;  
Weight=1; Priority=0 ] in NI_2 takes 16 sls  
2018-08-23 13:25:16.978 029585 5 HMRT.cpp(436) M2PA: distributed sls count = 16  
2018-08-23 13:25:16.978 029585 5 HMRT.cpp(354) M2PA: update links distribution in  
linksets  
2018-08-23 13:25:16.978 029585 10 HMRT.cpp(270) M2PA: Sg.SIGTRAN.M2PA.L3.HMRT: new  
tables:  
Linksets distribution for routes:  
=====  
[ NI=2; DPC=6981 ]  
  { SLS(0;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15) -> Sg.SIGTRAN.M2PA.L2.LinkSet.1 };
```

---

Links distribution for linksets:

=====

```
[ Sg.SIGTRAN.M2PA.L2.LinkSet.1 ]
  { SLS(0;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15) ->
Sg.SIGTRAN.M2PA.L2.LinkSet.1.Link.0 };
```

2018-08-23 13:25:16.978 029585 5 L3.cpp(485) M2PA: Sg.SIGTRAN.M2PA.L3: Send NI=2  
DPC=6981 AVAILABLE to Sg.SS7.STP.0

2018-08-23 13:25:16.978 029585 10 L3.cpp(174) M2PA: Sg.SIGTRAN.M2PA.L3: known  
linksets:

```
[ Sg.SIGTRAN.M2PA.L2.LinkSet.1 -> {NI=2; OPC=1794; APC=6981; Weight=1;} ] -  
ACTIVE
```

```
[ Sg.SIGTRAN.M2PA.L2.LinkSet.0 -> {NI=2; OPC=1794; APC=6982; Weight=1;} ] -  
INACTIVE
```

---

### **Журнал ошибок и предупреждений M2PA (M2PA\_warning)**

В данном журнале фиксируются все ошибки и предупреждения, происходящие в подсистеме M2PA. С помощью данной информации пользователь может увидеть причину обрыва соединения или отказа обработки сообщения.

Таблица 40 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

---

```
M2PA_warning = {
    file="logs/sccp/sccp_warning.log";
    period = 1hour;
    mask=date & time & tick;
    level=10;
};
```

---

Пример записи:

---

```
2018-08-23 13:18:28.889 029585 1 IAC.cpp(87) M2PA:
Sg.SIGTRAN.M2PA.L2.LinkSet.1.Link.0.IAC: alignment is not possible(T2 expired)

2018-08-23 13:21:43.930 029585 1 IAC.cpp(87) M2PA:
Sg.SIGTRAN.M2PA.L2.LinkSet.1.Link.0.IAC: alignment is not possible(T2 expired)

2018-08-23 13:24:28.963 029585 1 IAC.cpp(87) M2PA:
Sg.SIGTRAN.M2PA.L2.LinkSet.1.Link.0.IAC: alignment is not possible(T2 expired)

2018-08-23 13:24:38.964 029585 1 IAC.cpp(87) M2PA:
Sg.SIGTRAN.M2PA.L2.LinkSet.1.Link.0.IAC: alignment is not possible(T2 expired)
```

---

### 7.1.7 Журналы подсистемы HTTP

#### **Основной журнал подсистемы HTTP (HTTP\_trace)**

В данном журнале фиксируются все события, происходящие в подсистеме HTTP. С помощью данной информации пользователь может просмотреть все входящие и исходящие HTTP-пакеты в декодируемом виде.

Рекомендуемые значения глубины трассировки для различных режимов представлены в таблице ниже.

Таблица 43 — Значение глубины трассировки

Наименование режима	Рекомендуемое значение параметра level
Режим отладки (Debug)	10
Режим Production	3

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

```
http_trace = {  
    file="logs/http/http_trace -%Y%m%d-%H%M.log";  
    period = 1hour;  
    mask=date & time & tick;  
    level=10;  
};
```

Пример записи:

```
HTTP_DATA_IND ()  
2018-07-20 07:00:00;FSM: Sg.HTTP.SDir.1.SCL.1907241 received message:  
POST /UpdatePsLocation HTTP/1.1  
Content-Type: application/json //mime_part_header  
  
[  
    {  
        "IMEI" : "864xxxxxxxxxxxxxxxx",  
        "IMSI" : "250xxxxxxxxxxxxxxxx",  
        "MSISDN" : "xxxxxxxxxxxxxxxx",  
        "location" : {  
            "CELL LAC" : xxxx,  
            "CID" : xxxx,  
            "LAC" : xxxxx,  
            "MCC" : 250,  
            "MNC" : x,  
            "RAC" : xxx
```

```

    },
    "location_type" : 0
  }
]

```

### **Журнал, содержащий данные, отправляемые по интерфейсу HTTP (http\_binary)**

В данном журнале фиксируются все данные полученные и отправленные по HTTP интерфейсу в не декодируемом виде.

Таблица 44 содержит рекомендуемые значения глубины трассировки для различных режимов.

Таблица 44 — Значение глубины трассировки

Наименование режима	Рекомендуемое значение параметра level
Режим отладки (Debug)	0
Режим Production	0

Параметры журнала настраиваются в файле конфигурации Trase.cfg.

Пример конфигурации:

```

http_binary = {
    file="logs/http/http_binary.log";
    period = 1hour;
    mask=date & time & tick;
    level=10;
};

```

Пример записи:

```

2018-08-28 20:24:38;HTTP_Connection id: 3(32) 10.136.77.41:7778-
>10.136.77.41:37906;

TCP connection established

2018-08-28 20:24:38;HTTP_Connection id: 3(32) 10.136.77.41:7778-
>10.136.77.41:37906;

received TCP packet:

504f5354 202f5375 62736372 6962654c 6f636174 696f6e20 48545450 2f312e31
0d0a5573 65722d41 67656e74 3a206375 726c2f37 2e31392e 37202878 38365f36
342d7265 64686174 2d6c696e 75782d67 6e752920 6c696263 75726c2f 372e3139
2e37204e 53532f33 2e32372e 31207a6c 69622f31 2e322e33 206c6962 69646e2f
312e3138 206c6962 73736832 2f312e34 2e320d0a 486f7374 3a203130 2e313336
2e37372e 34313a37 3737380d 0a416363 6570743a 202a2f2a 0d0a436f 6e74656e
742d5479 70653a20 6170706c 69636174 696f6e2f 6a736f6e 0d0a436f 6e74656e
742d4c65 6e677468 3a203939 0d0a0d0a 7b0a2020 2022696d 73692220 3a202222
2c0a2020 20227375 622d6578 70697265 7322203a 20333630 302c0a20 20202263

```

```
6f6e7461 63742220 3a202268 7474703a 2f2f3132 372e302e 302e313a 38303837
2f6c6f63 6174696f 6e5f6368 616e6765 220a7d
```

---

## 7.1.8 Журналы подсистемы OM-interface

### Основной журнал подсистемы OM-interface (OM\_trace)

В данном журнале фиксируются все события, происходящие в подсистеме OM-interface. С помощью данной информации пользователь может определить активность подключений к подсистемам COPM, этапы обработки сигнальных сообщений/

Таблица 39 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

---

```
OM_trace =
{
    file = "logs/om_trace.log";
    period = hour;
    mask = date & time & tick & file;
    separator = ";";
    level = 10;
};
```

---

Пример записи:

---

```
2018-08-29 16:03:36.753;PrimFSM.cpp(35);FSM: Ad.OMI.ASCL.0 id: 0; > St: 0 ACTIVE
Pr: 40705:6 OM_TRANSACTION_REQ (SessID = 0, TrID = 9741, SegID = 0, BegInd = 1,
EndInd = 1, ResponseTimeout = 0)

2018-08-29 16:03:36.753;OM_BaseLogic.cpp(919);Ad.OMI.ASCL.0 Create new
transaction. ( TrID = 9741)

2018-08-29 16:03:36.753;OM_CL.cpp(22);OM_TransactionReq_data: TransactionID = 9741
| SegmentID = 0 | BeginIndicator = 1 | EndIndicator = 1 | CommandList = 00 00 00
04 00 00 00 45 49 4d 53 49 3d 32 35 30 34 32 37 3

2018-08-29 16:03:36.753;PrimFSM.cpp(70);FSM: Ad.OMI.ASCL.0 id: 0; < St: 0 ACTIVE

2018-08-29 16:03:36.753;PrimFSM.cpp(70);FSM: Ad.OMI.Dir.Locator id: 0; < St: 1
ACTIVE

2018-08-29 16:03:36.755;PrimFSM.cpp(35);FSM: Ad.OMI.ASCL.0 id: 0; > St: 0 ACTIVE
Pr: 184:22 SI_DATA_IND ((536870912, 470915))

2018-08-29 16:03:36.755;OM_BaseLogic.cpp(272);Ad.OMI.ASCL.0
OM_TransactionConf_data: TransactionID = 9741 | SegmentID = 0 |
FinalResponseIndicator = 1 | ResultList = empty

2018-08-29 16:03:36.756;OM_AutoCL.cpp(314);Ad.OMI.ASCL.0 Remove transaction. (
TrID = 9741)

2018-08-29 16:03:36.756;OM_Direction.cpp(258);Ad.OMI.Dir.Locator Erase
transaction. ( ConID = 0)

2018-08-29 16:03:36.756;OM_Direction.cpp(261);Ad.OMI.Dir.Locator Current
transaction count = 0.
```

---

**Журнал, содержащий данные, отправляемые по ОМ-интерфейсу (OM\_binary)**

В данном журнале фиксируются все данные, полученные и отправленные по ОМ интерфейсу в не декодируемом виде.

Таблица 44 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример конфигурации:

---

```
http_binary = {  
    file="logs/http/http_binary.log";  
    period = 1hour;  
    mask=date & time & tick;  
    level=10;  
};
```

---

**Пример записи:**

---

```
2018-08-28 20:24:38;HTTP_Connection id: 3(32) 10.136.77.41:7778-  
>10.136.77.41:37906;  
  
TCP connection established  
  
2018-08-28 20:24:38;HTTP_Connection id: 3(32) 10.136.77.41:7778-  
>10.136.77.41:37906;  
  
received TCP packet:  
504f5354 202f5375 62736372 6962654c 6f636174 696f6e20 48545450 2f312e31  
0d0a5573 65722d41 67656e74 3a206375 726c2f37 2e31392e 37202878 38365f36  
342d7265 64686174 2d6c696e 75782d67 6e752920 6c696263 75726c2f 372e3139  
2e37204e 53532f33 2e32372e 31207a6c 69622f31 2e322e33 206c6962 69646e2f  
312e3138 206c6962 73736832 2f312e34 2e320d0a 486f7374 3a203130 2e313336  
2e37372e 34313a37 3737380d 0a416363 6570743a 202a2f2a 0d0a436f 6e74656e  
742d5479 70653a20 6170706c 69636174 696f6e2f 6a736f6e 0d0a436f 6e74656e  
742d4c65 6e677468 3a203939 0d0a0d0a 7b0a2020 2022696d 73692220 3a202222  
2c0a2020 20227375 622d6578 70697265 7322203a 20333630 302c0a20 20202263  
6f6e7461 63742220 3a202268 7474703a 2f2f3132 372e302e 302e313a 38303837  
2f6c6f63 6174696f 6e5f6368 616e6765 220a7d
```

---

**Информационный журнал ОМ-интерфейса (OM\_info)**

В данном журнале выводится информация о состоянии интерфейса для пользователя.

Таблица 40 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.



---

**Пример конфигурации:**

---

```
OM_info = {  
    file="logs/OM/OM_info.log";  
    mask=date & time & tick;  
    level=10;  
};
```

---

**Пример записи:**

---

```
2018-08-29 16:41:16.035;OM_Interface.cpp(124);Ad.OMI Sockets init...  
2018-08-29 16:41:16.036;OM_Interface.cpp(135);Ad.OMI ServerConnectionLogics  
init...  
2018-08-29 16:41:16.036;OM_Interface.cpp(387);Ad.OMI Create Ad.OMI.ASCL.0 ( Login  
= support , Password = elephant )  
2018-08-29 16:41:16.036;OM_Interface.cpp(161);Ad.OMI ServerDirections init...  
2018-08-29 16:41:16.036;OM_Interface.cpp(589);Ad.OMI: create server direction with  
address = Ad.OMI.Dir.Locator  
2018-08-29 16:41:16.036;OM_Interface.cpp(180);Ad.OMI Primary...  
2018-08-29 16:41:16.036;OM_Interface.cpp(184);Ad.OMI Link to 0 connection.  
2018-08-29 16:41:16.036;OM_Direction.cpp(147);Ad.OMI.Dir.Locator Link primary  
connection. ( ConID = 0)  
2018-08-29 16:41:16.036;OM_AutoCL.cpp(38);Ad.OMI.ASCL.0 Linked by direction  
Ad.OMI.Dir.Locator.  
2018-08-29 16:41:16.036;OM_Interface.cpp(192);Ad.OMI Secondary...  
2018-08-29 16:41:16.170;OM_AutoCL.cpp(38);Ad.OMI.ASCL.0 Linked by direction  
Ad.OMI.Dir.Locator.  
2018-08-29 16:41:16.170;OM_AutoCL.cpp(57);Ad.OMI.ASCL.0 Connection relinked by  
direction Ad.OMI.Dir.Locator.
```

---

***Журнал ошибок и предупреждений ОМ-интерфейса (OM\_warning)***

В данном журнале фиксируются все ошибки и предупреждения, происходящие на ОМ-интерфейсе.

Таблица 40 содержит рекомендуемые значения глубины трассировки для различных режимов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

**Пример конфигурации:**

---

```
OM_warning = {  
    file="logs/OM/OM_warning.log";  
    mask=date & time & tick;  
    level=10;  
};
```

---

---

**Пример записи:**

---

```
2018-08-14 11:34:36.396 017846 OM_MesHandler.cpp(98) Ad.OMI.CCL.9: Receive
unexpected message OM_LoginConf.

2018-08-14 11:34:36.401 017846 OM_MesHandler.cpp(232) Ad.OMI.CCL.9: Receive
unexpected message OM_LogoutConf.

2018-08-15 16:54:44.394 017846 OM_MesHandler.cpp(98) Ad.OMI.CCL.119: Receive
unexpected message OM_LoginConf.

2018-08-15 16:54:44.421 017846 OM_MesHandler.cpp(232) Ad.OMI.CCL.119: Receive
unexpected message OM_LogoutConf.
```

---

## 8 Аварии

### 8.1 Общие аварии

Возможна авария, связанная с отсутствием IMEI в ATI-запросе от модуля Locator на HLR.

Пример аварии:

```
2018-03-06 16:49:09.229 Absence imei in SubscriberInfo. IMSI=250427701358766
MSISDN=79584845765 VLR=
```

Для решения данной проблемы необходимо проанализировать сообщения сигнального обмена и осуществить настройку получения IMEI между HLR и VLR.

### 8.2 Аварии Sigtran

#### 8.2.1 Аварии для Sg.SIGTRAN.M3UA.AS

В таблице ниже описаны аварии узла AS в протоколе SIGTRAN.

Таблица 45 — Аварии AS

Описание	CA	Тип	Комментарий
Не найден UP для передачи данных (создается при каждой неуспешной попытке послать данные UP-у)	Alarm.UP.Invalid	string	DPC/SIC/NI — набор параметров, по которым осуществляется поиск UP (также можно передавать OPC и DPC)
	Alarm.UP.Num	int	Счетчик неуспешных поисков UP-ов
Не найден ASP для передачи данных (создается при каждой неуспешной попытке послать данные на ASP)	Alarm.DPC.Invalid	int	DPC, для которого не найден ASP (берется из входящего MTP сообщения)
	Alarm.DPC.Num	int	Счетчик неуспешных поисков ASP-ов
Неуспех изменения конфигурации (создается при каждой неуспешной попытке изменить конфигурацию AS)	Alarm.ChCfg.Invalid	string	Строковый параметр, передаваемый в обработчик On_ConfigChangeReq
Неуспех регистрации. (создается для каждого REG_RSP с полем STATUS не равным success)	Warn.Reg.Failed	string	ASP, через который не удалось зарегистрироваться
	Warn.Reg.Num	int	Число неуспешных попыток регистрации
Неуспех deregистрации (создается для каждого	Warn.Dereg.Failed	string	ASP, через который не удалось deregистрироваться

Описание	CA	Тип	Комментарий
DEREG_RSP с полем STATUS не равным success)	Warn.Dereg.Num	int	Число неуспешных попыток deregistracii
Неуспех активации (создается при каждой перепосылке ASPAC)	Warn.Act.Failed	string	ASP, который не удается активировать.
	Warn.Act.Num	int	Число попыток активации
Неуспех деактивации (создается при каждой перепосылке ASPIA)	Warn.Deact.Failed	string	ASP, который не удается деактивировать.
	Warn.Deact.Num	int	Число попыток деактивации
Получение NTFY со значением: ASP Failure (создается при каждом получении сообщения)	Warn.ASP.Failure	string	CA ASP (test not implemented)
Число залинкованных UP-ов на AS (создается при каждой успешной (раз)линковке UP(с)к AS)	Info.LinkUP.Num	int	
Инициализация AS	Info.ASInit	int (0,1,2,3)	0 — без rk и rt; 1 — с rk, но без rt; 2 — без rk, но с rt; 3 — с rk и rt.

**Примечание** — rt — routing table, rk — routing key

## 8.2.2 Аварии для Sg.SIGTRAN.M3UA.ASP

В таблице ниже описаны аварии узла ASP в протоколе SIGTRAN.

Таблица 46 — Аварии ASP

Описание	Адрес	Тип	Комментарий
Ошибка декодирования	Alarm.Decode	int	Код, возвращаемый функцией decode
Ошибка кодирования	Alarm.Encode	int	Код, возвращаемый функцией encode

Описание	Адрес	Тип	Комментарий
Падение SCTP-соединения	Alarm.CDI.Num	int	Общее число падений (ConnectionDownInd)
	Alarm.ASP.CDI	string	ASP, на котором упало соединение
Получение от сервера сообщения об ошибке M3UA_ERR (создается каждый раз при получении).	Warn.ErrCode	int	Код ошибки
Неуспех подъема ASP (создается каждый раз, когда происходит перепосылка ASPUP по таймеру)	Warn.ASPUP	string	ASP, который не удалось поднять (перевести в состояние UP)
	Warn.ASPUP.Num	int	Число попыток подъема ASP
Неуспех по ASPDN (создается каждый раз, когда происходит перепосылка ASPDN по таймеру)	Warn.ASPDN	string	ASP, который не удалось перевести в состояние DOWN
	Warn.ASPDN.Num	int	Число посылок ASPDN
Приход сообщения M3UA_DAVA	Info.DAVA	int	Affected PointCode (APC может быть несколько, одно сообщение M3UA_DAVA.Info создается на каждый APC) (DestinationAValiable)
Приход сообщения M3UA_DUNA	Info.DUNA	int	Affected PointCode (APC может быть несколько, одно сообщение M3UA_DUNA.Info создается на каждый APC) (DestinationUNAvailiable)
Приход сообщения M3UA_SCON	Info.SCON	int	Affected PointCode (APC может быть несколько, одно сообщение M3UA_SCON.Info создается на каждый APC) (SignallingCONgesion)

Описание	Адрес	Тип	Комментарий
Приход сообщения МЗUA_DUPU	Info.DUPU	int	Affected Point Code (DestinationUserPartUnavailable)
	Info.DUPU.UC	string	User/Cause — строка: User=#User; (SCCP, ISUP...) Cause=#Cause; (описание)
Приход сообщения МЗUA_DRST	Info.DRST	int	Affected Point Code (APC может быть несколько, одно сообщение МЗUA_DRST.Info создается на каждый APC) (DestinationReSTricted)

### 8.3 Аварии OMI-интерфейса (OMI)

В таблице ниже описаны аварии OMI-интерфейса.

Таблица 47 — Аварии OMI-интерфейса

Тип компоненты	Адрес компоненты	Имя и тип перемен ной	Область допустимых значений	Описание
Ad.OMI.ACCL	Ad.OMI.ACCL. <int>			Общее состояние системы
		OSTATE	{«ACTIVATE»,«FAIL»}	Оперативное состояние клиентской полуавтомати- ческой сетевой логики
Ad.OMI.ASCL	Ad.OMI.ASCL. <int>.<int>			Общее состояние системы
		OSTATE	{«ACTIVATE»,«FAIL»}	Оперативное состояние сессий для серверной полуавтомати- ческой сетевой логики

Тип компоненты	Адрес компоненты	Имя и тип переменной	Область допустимых значений	Описание
Ad.OMI.Traffic.Stat	Ad.OMI.Traffic.Stat			
		PARAM	{Stat=«<время_начала_подсчета_статистики>»; Statistics=«<количество_попыток_соединения>; <количество_успешных_соединений>; <количество_ошибок>;»}	Статистика