



PROTEI RD_NAS

Руководство администратора

Авторские права

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

Содержание

1	Термины и сокращения	4
2	Общие сведения.....	6
2.1	Назначение документа.....	6
2.2	Состав документа	6
2.3	Техническая поддержка	7
2.3.1	<i>Производитель</i>	<i>7</i>
2.3.2	<i>Служба технической поддержки</i>	<i>7</i>
2.4	История изменений.....	7
3	Описание модуля PROTEI RD_NAS	8
3.1	Введение	8
3.2	Функциональные возможности	8
3.3	Настройка работы модуля RD_NAS	9
3.3.1	<i>Общая информация</i>	<i>9</i>
3.3.2	<i>Настройка алгоритма отправки сообщений.....</i>	<i>9</i>
3.3.3	<i>Настройка операции «Авторизация/Аутентификация».....</i>	<i>10</i>
3.3.4	<i>Настройка операции «Начисление платы».....</i>	<i>10</i>
3.3.5	<i>Настройка операции «Авторизация на AAA при регистрации по SIP».....</i>	<i>11</i>
3.4	Алгоритм работы	11
3.5	Сценарии обмена сообщениями	12
3.6	Описание perl-скрипта translate.pl	22
3.6.1	<i>Параметры взаимодействия mCore.MKD с RD_NAS</i>	<i>24</i>
3.6.2	<i>Журнал perl.log</i>	<i>26</i>
4	Настройка системы	27
4.1	Условные обозначения в таблицах параметров конфигурации	27
4.2	Настройка параметров взаимодействия с AAA-сервером (rd_nas.cfg).....	27
4.3	Настройка параметров сервера RD_NAS (rd_nas_server.cfg)	28
4.4	Настройка параметров OMI- соединения (om_interface.cfg)	29
4.5	Настройка параметров системы журналирования (trace.cfg)	30
4.6	Настройка подсистемы сбора аварий (ar.cfg)	33

1 Термины и сокращения

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

Таблица 1 — Термины и сокращения

Термин	Описание
AAA	Authentication/ Authorization/ Accounting, Аутентификация/ Авторизация/ Отчетность
NAS	Network Access Server, сервер удаленного доступа к сети
OMI	Operation and Maintenance Interface, интерфейс эксплуатации и технического обслуживания
Perl	Practical Extraction and Report Language (алгоритмический язык программирования)
RADIUS	Remote Authentication in Dial-In User Service, служба удаленной аутентификации телефонной линии — протокол для реализации аутентификации, авторизации и сбора сведений об использованных ресурсах
Клиент RADIUS	сервер удаленного доступа (NAS) - отвечает за передачу сведений о пользователе заданным серверам RADIUS. Клиент RADIUS использует специальный формат для передачи сведений - RADIUS-Message (сообщение RADIUS).
Протокол RADIUS	отраслевой стандартный протокол, используемый для осуществления проверки подлинности, авторизации и учета.
Сервер RADIUS (AAA-сервер)	отвечает за прием запросов от клиентов RADIUS, идентификацию пользователей и возврат клиенту RADIUS всех конфигурационных параметров, требуемых для предоставления пользователю соответствующих услуг.
Сообщение RADIUS	всегда состоит из заголовка и атрибутов, каждый из которых содержит ту или иную информацию о попытке доступа: например, имя и пароль пользователя, запрашиваемые услуги и IP-адрес сервера доступа. Главной задачей атрибутов RADIUS является передача информации между клиентами и серверами.

Термин	Описание
Оборудование mCore.MKD	Программный коммутатор, выполняющий функции управляющего узла в IP-сети, маршрутизации вызова, управления шлюзовым оборудованием (например, mGate.ITG), оборудованием доступа (например, mAccess.МАК, mAccess.MTU), SIP-терминалами и применяющийся как в городских, так и в сельских телефонных сетях.

2 Общие сведения

2.1 Назначение документа

Настоящее руководство содержит сведения о функциональных возможностях системы, настройке и работе модуля RD_NAS в составе mCore.MKD, а также описание конфигурации системы.

2.2 Состав документа

Настоящее руководство состоит из следующих основных частей:

«Термины и сокращения» — описание используемых терминов и сокращений;

«Общие сведения» — назначение и состав документа, а также контактная информация производителя;

«Описание модуля PROTEI RD_NAS» — сведения о назначении и функциональных возможностях модуля PROTEI RD_NAS;

«Настройка системы» — сведения о настройке системы и параметрах конфигурационных файлов.

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 модуля, 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>
Email: sales@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. 5999 (круглосуточно)
(812) 449–47–31 (круглосуточно)
Факс: (812) 449–47–29
Web: <http://www.protei.ru>
Email: mak.support@protei.ru

2.4 История изменений

История изменений настоящего документа фиксируется в таблице 2.

Таблица 2 — История изменений

Дата	Версия документа	Изменения
16.11.2011	1.0.0	Выпущена первая версия документа.
04.04.2023	1.1.0	Документ переименован в «Руководство администратора». Добавлен раздел 1 «Термины и сокращения». Обновлен раздел 3.3 «Настройка работы модуля RD_NAS».

3 Описание модуля PROTEI_RD_NAS

3.1 Введение

PROTEI_RD_NAS (в дальнейшем - RD_NAS) представляет собой программное приложение, выполняющее функции клиента RADIUS. Модуль RD_NAS используется в mCore.MKD для взаимодействия с AAA-сервером (сервером RADIUS).

Модуль RD_NAS предназначен для преобразования запросов от mCore.MKD в сообщения протокола RADIUS.

Модуль RD_NAS устанавливается на том же сервере, что и mCore.MKD. Взаимодействие модулей mCore.MKD и RD_NAS осуществляется по внутреннему OMI-протоколу. Общая схема взаимодействия модулей приведена ниже (Рисунок 1).

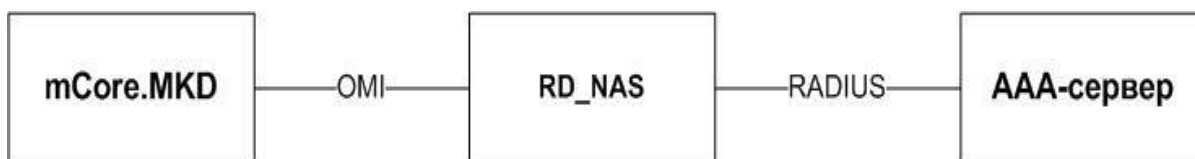


Рисунок 1 — Общая схема взаимодействия модулей

3.2 Функциональные возможности

Модуль RD_NAS выполняет следующие функции:

- преобразует параметры сообщений RADIUS в параметры пользовательского запроса и наоборот;
- по командам от mCore.MKD отправляет на AAA-сервер следующие сообщения RADIUS:
 - Access-Request — "запрос доступа", с которого начинается аутентификация и авторизация пользователя при попытке получить доступ к услугам сети;
 - Accounting-Request (Start/Stop) — "запрос учета", в котором отправляется учетная информация для расчета стоимости телефонного разговора или предоставляемой услуги;
 - Interim-Update-Request — сообщения, отправляемые абоненту на AAA-сервер для промежуточного учета состояния телефонного соединения.
- принимает от AAA-сервера следующие сообщения:
 - Access-Accept — "доступ разрешен", положительный ответ на запрос доступа;
 - Access-Reject — "доступ запрещен", отрицательный ответ на запрос доступа. Возможен в том случае, если пользовательских данных недостаточно для успешной аутентификации или доступ для пользователя не авторизован;
 - Accounting-Response — подтверждение приема запроса с учетной информацией. Посылается в процессе учета стоимости предоставляемых услуг.

3.3 Настройка работы модуля RD_NAS

3.3.1 Общая информация

Настройка алгоритма параметров RADIUS осуществляется с помощью Web-интерфейса для управления коммутатором PROTEI SSW5, блок «Параметры RADIUS» (Рисунок 2).

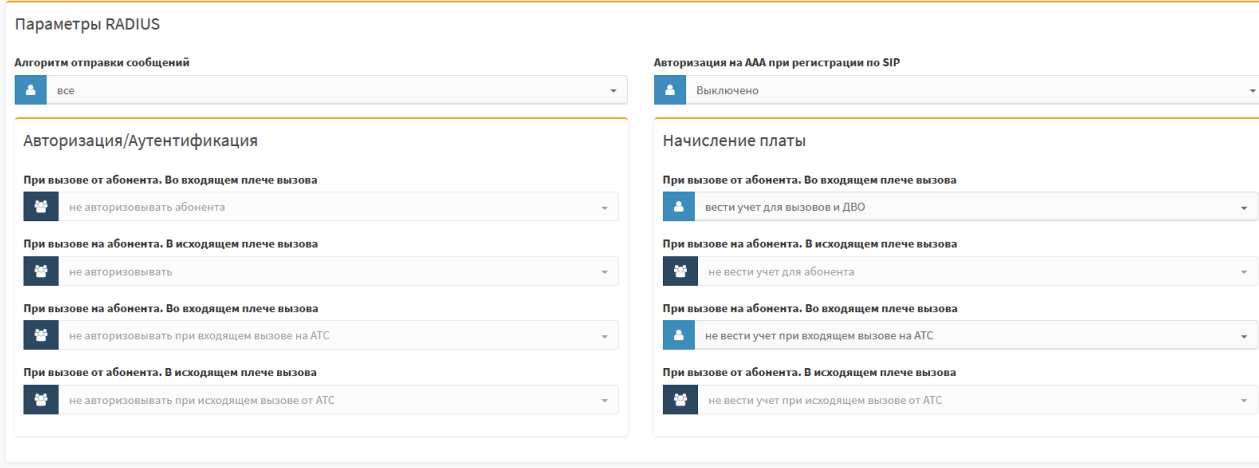


Рисунок 2 — Параметры RADIUS


Блок «Параметры RADIUS» отображается в настройках абонентов, в следующих пунктах интерфейса Web TO:

- индивидуальные настройки абонента (Главное меню | Абоненты VPBX | <номер абонента>);
- остальные абоненты (Главное меню | Остальные абоненты);
- групповые настройки абонентов (VPBX | Групповые настройки абонентов).

Примечание. Подробную информацию о настройках абонентов и их использовании см. «PROTEI SSW5. Руководство пользователя Web-интерфейса».

По умолчанию параметры RADIUS не используются при создании профиля абонента.

При необходимости использования какого-либо параметра следует выполнить следующие шаги:

1. В строке с параметром, который требуется изменить (Рисунок 2), нажать на иконку , а затем нажать в выпадающем меню на «Настройки администратора».
2. Установить необходимое значение параметра по инструкциям в разделах 3.3.2 - 3.3.5.

3.3.2 Настройка алгоритма отправки сообщений

Настройка алгоритма отправки сообщений протокола RADIUS осуществляется с помощью пункта «Алгоритм отправки сообщений». Необходимо выбрать из списка используемые алгоритмы отправки сообщений (можно выбрать несколько алгоритмов одновременно):

- Accounting-Start – отправлять при начале учета оплаты за соединения или услуги;
- Interim-Update – отправлять промежуточные сообщения;

- Accounting-Stop – отправлять по окончании учета оплаты.

3.3.3 Настройка операции «Авторизация/Аутентификация»

В пункте «Авторизация/Аутентификация» задается необходимость отправки сообщений Access-Request (сообщение с запросом доступа) при установке соединения.

Для настройки операции «Авторизация/Аутентификация» необходимо заполнить следующие поля:

- «При вызове от абонента. Во входящем плече вызова» — выпадающий список для указания необходимости авторизации/аутентификации вызовов, входящих от абонентов на MKD, по номеру вызывающего абонента. Возможные значения:
 - авторизовывать вызовы и ДВО;
 - авторизовывать только вызовы;
 - не авторизовывать абонента.
- «При вызове на абонента. В исходящем плече вызова» — выпадающий список для указания необходимости авторизации/аутентификации вызовов, входящих на MKD, по номеру вызывающего абонента. Возможные значения:
 - авторизовывать при всех вызовах;
 - авторизовывать при вызове от внешних;
 - авторизовывать при вызове от внутренних;
 - не авторизовывать.
- При вызове на абонента. Во входящем плече вызова» — выпадающий список для указания необходимости авторизации/аутентификации вызовов, исходящих от MKD, по номеру вызываемого абонента. Возможные значения:
 - авторизовывать при исходящем вызове от АТС;
 - не авторизовывать при исходящем вызове от АТС.
- «При вызове от абонента. В исходящем плече вызова» — выпадающий список для указания необходимости авторизации/аутентификации вызовов, исходящих от MKD, по номеру вызываемого абонента. Возможные значения:
 - авторизовывать при исходящем вызове от АТС;
 - не авторизовывать при исходящем вызове от АТС.

3.3.4 Настройка операции «Начисление платы»

В пункте «Начисление платы» задается необходимость отправки сообщений Accounting-Request (start/stop) при установлении и разрушении соединения.

Для настройки операции «Начисление платы» необходимо заполнить следующие поля:

- «При вызове от абонента. Во входящем плече вызова» — в поле определяется необходимость проводить начисление платы вызываемому абоненту за входящие на MKD вызовы. Требуемое значение выбирается из выпадающего списка:
 - вести учет для вызовов и ДВО;
 - вести учет только для вызовов;
 - не вести учет для абонента.

- «При вызове на абонента. В исходящем плече вызова» — в поле определяется необходимость проводить начисление платы вызываемому абоненту за вызовы, исходящие из MKD. Требуемое значение выбирается из выпадающего списка:
 - вести учет при всех вызовах;
 - вести учет при вызове от внешних;
 - вести учет при вызове от внутренних;
 - не вести учет для абонента.
- «При вызове на абонента. Во входящем плече вызова» — в поле определяется необходимость проводить начисление платы вызываемому абоненту за входящие на MKD вызовы. Требуемое значение выбирается из выпадающего списка:
 - вести учет при входящем вызове на АТС;
 - не вести учет при входящем вызове на АТС.
- «При вызове от абонента. В исходящем плече вызова» — в поле определяется необходимость проводить начисление платы вызывающему абоненту за вызовы, исходящие из MKD. Требуемое значение выбирается из выпадающего списка:
 - вести учет при исходящем вызове от АТС;
 - вести учет при исходящем вызове от АТС.

3.3.5 Настройка операции «Авторизация на AAA при регистрации по SIP»

В пункте меню «Авторизация на AAA при регистрации по SIP» задается необходимость отправки RADIUS-сообщения Access-Request при получении сообщения SIP REGISTER от абонента.

Для настройки операции «Авторизация на AAA при регистрации по SIP» используется выпадающий список, из которого следует выбрать одно из значений:

- включено — проводить авторизацию;
- выключено — не проводить авторизацию.

Отправлять информацию на AAA-сервер можно для всех типов абонентов, как внешних, так и внутренних.

3.4 Алгоритм работы

В этом разделе описываются алгоритмы работы для процедур, описанных в 3.3.2 «Настройка алгоритма отправки сообщений».

Алгоритм обработки процедуры «Авторизация/аутентификация» по протоколу RADIUS:

1. Модулю RD_NAS от mCore.MKD передается информация пользователя, инициировавшего вызов или запрос услуги.
2. После того, как RD_NAS получает данные, осуществляет идентификацию по протоколу RADIUS — создает запрос Access-Request с параметрами:
 - регистрационное имя пользователя;
 - пароль;
 - идентификатор клиента, используемый для регистрации;
 - идентификатор порта, используемый для регистрации.
3. Пакет Access-Request передается AAA-серверу.

Примечание. Если в течение продолжительного времени на запрос не будет получено отклика, передача запроса повторяется несколько раз.

4. После получения AAA-сервер проверяет запрос. Если проверка завершилась успешно (то есть пароль пользователя, идентификатор клиента и порт корректные), AAA-сервер обращается к базе данных пользователей для поиска.
5. При выполнении всех условий в отклик Access-Асcept включается список всех конфигурационных параметров для данного пользователя.

Алгоритм обработки процедуры «Начисление платы» по протоколу RADIUS:

1. Модулю RD_NAS от mCore.MKD передается информация, связанная с конкретным пользователем, инициировавшим вызов или запрос услуги.
2. Если RD_NAS настроен на учет услуг, то он генерирует старт-пакет Accounting-Request, описывающий тип предоставляемого пользователю сервиса.
3. Пакет Accounting-Request передается AAA-серверу.

Примечание. Если в течение продолжительного времени на запрос не будет получено отклика, передача запроса повторяется несколько раз.

4. После получения клиентского запроса AAA-сервер проверяет передавшего этот запрос - RD_NAS. Если проверка завершилась успешно, AAA-сервер обращается к базе данных о пользователях для поиска указанного в запросе имени.
5. Пользовательская запись в базе данных содержит список требований, которым пользователь должен удовлетворять для получения доступа к услуге.
6. При выполнении всех условий AAA-сервер отправляет ответ Accounting-Response.
7. После того, как услуга предоставлена, RD_NAS генерирует запрос Accounting-Request Stop, описывающий тип предоставленного пользователю сервиса. Может включать статистические сведения:
 - продолжительность сеанса;
 - количество принятых/переданных пакетов;
 - количество принятых/переданных байтов.
8. AAA-сервер отвечает подтверждением Accounting-Response.

3.5 Сценарии обмена сообщениями

3.5.1 Авторизация/Аутентификация при вызове от абонента во входящем плече вызова

Процедура авторизации/аутентификации осуществляется по номеру вызывающего абонента. Сценарий обработки вызова представлен ниже (Рисунок 3).

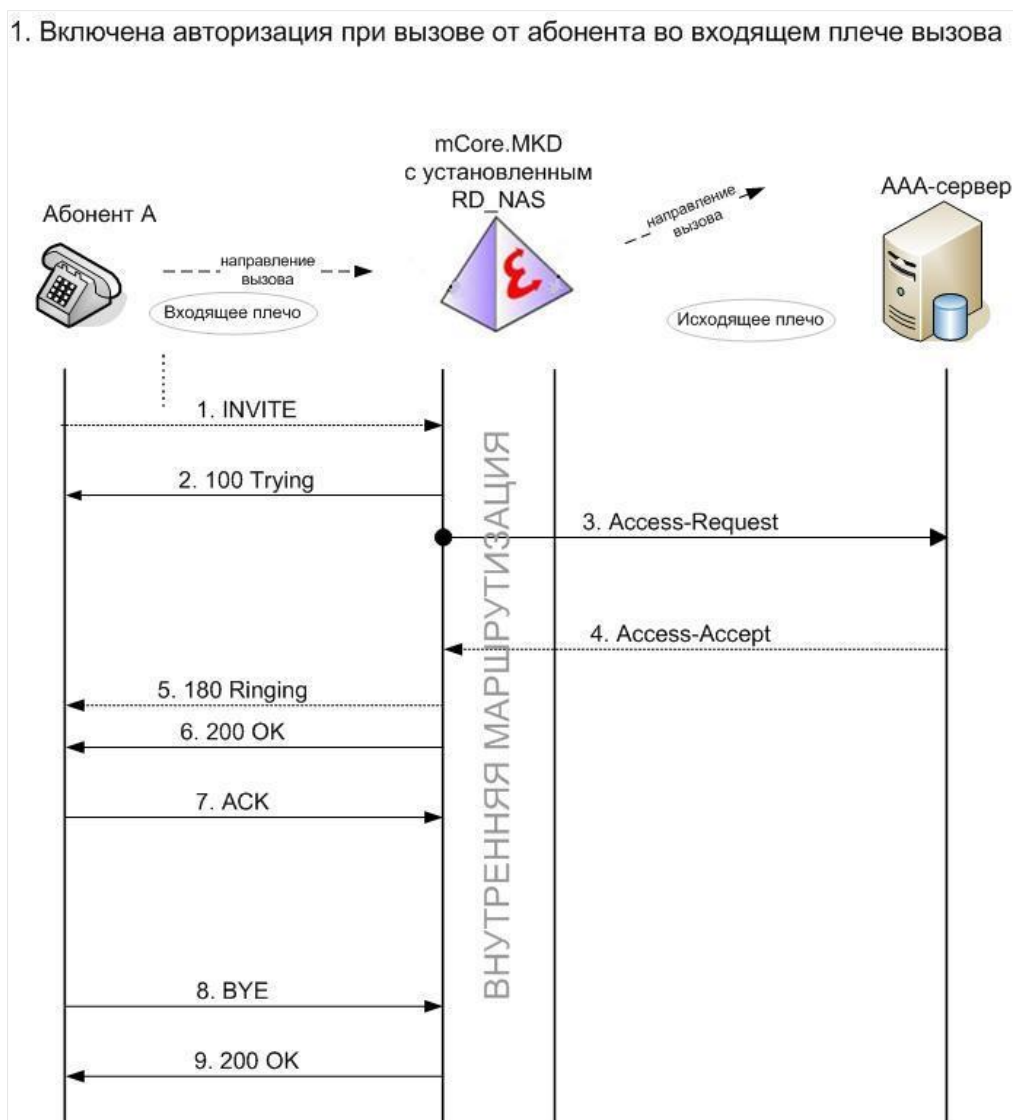


Рисунок 3 — Сценарий обмена сообщениями

1. абонент А снимает трубку, набирает телефонный номер - формируется запрос INVITE, инициирующий установление сеанса связи;
2. mCore.MKD оповещает абонента А, что запрос принят и обрабатывается (100 Trying);
3. модуль RD_NAS запрашивает у AAA-сервера разрешение на доступ абонента А к телефонной связи - запрос Access-Request;
4. AAA-сервер посылает ответ - Access-Асерт, на разрешение доступа вызывающего абонента к телефонной связи;
5. mCore.MKD посылает абоненту А сообщение 180 Ringing;
6. вызываемый абонент снимает трубку, формируется сообщение 200 OK;
7. станция абонента А отправляет подтверждение ACK;
8. после того как абонент А положит трубку, в mCore.MKD передается запрос BYE на разрыв соединения;
9. mCore.MKD подтверждает успешное завершение соединения отправкой сообщения 200 OK.

3.5.2 Авторизация/Аутентификация при вызове на абонента в исходящем плече вызова

Процедура авторизации/аутентификации осуществляется по номеру вызываемого абонента. Сценарий обработки вызова представлен ниже (Рисунок 4).



Рисунок 4 — Сценарий обмена сообщениями

В mCore.MKD поступает запрос на осуществление соединения с абонентом Б. mCore.MKD обрабатывает запрос по следующему сценарию:

1. модуль RD_NAS запрашивает у AAA-сервера разрешение на доступ абонента Б к телефонной связи - запрос Access-Request;
2. AAA-сервер посылает ответ - Access-Асерт, на разрешение доступа абонента Б к телефонной связи;
3. mCore.MKD выполняет маршрутизацию в направлении абонента Б — формируется запрос INVITE;
4. станция абонента Б оповещает mCore.MKD что запрос принят и обрабатывается (100 Trying);
5. станция абонента Б оповещает mCore.MKD о получении входящего вызова посредством сообщения 180 Ringing;
6. абонент Б снимает трубку, формируется сообщение 200 ОК;
7. mCore.MKD отправляет стороне абонента Б сообщение ACK, подтверждающее прием сообщения 200 ОК;

8. после окончания соединения вызывающей стороной, mCore.MKD передает запрос BYE станции абонента Б на разрушение соединения;
9. станция абонента Б подтверждает успешное разрушение соединения отправкой сообщения 200 ОК.

3.5.3 Авторизация/Аутентификация при вызове на абонента во входящем плече вызова

Процедура авторизации/аутентификации осуществляется по номеру вызывающего абонента. Сценарий обработки вызова представлен ниже (Рисунок 5).

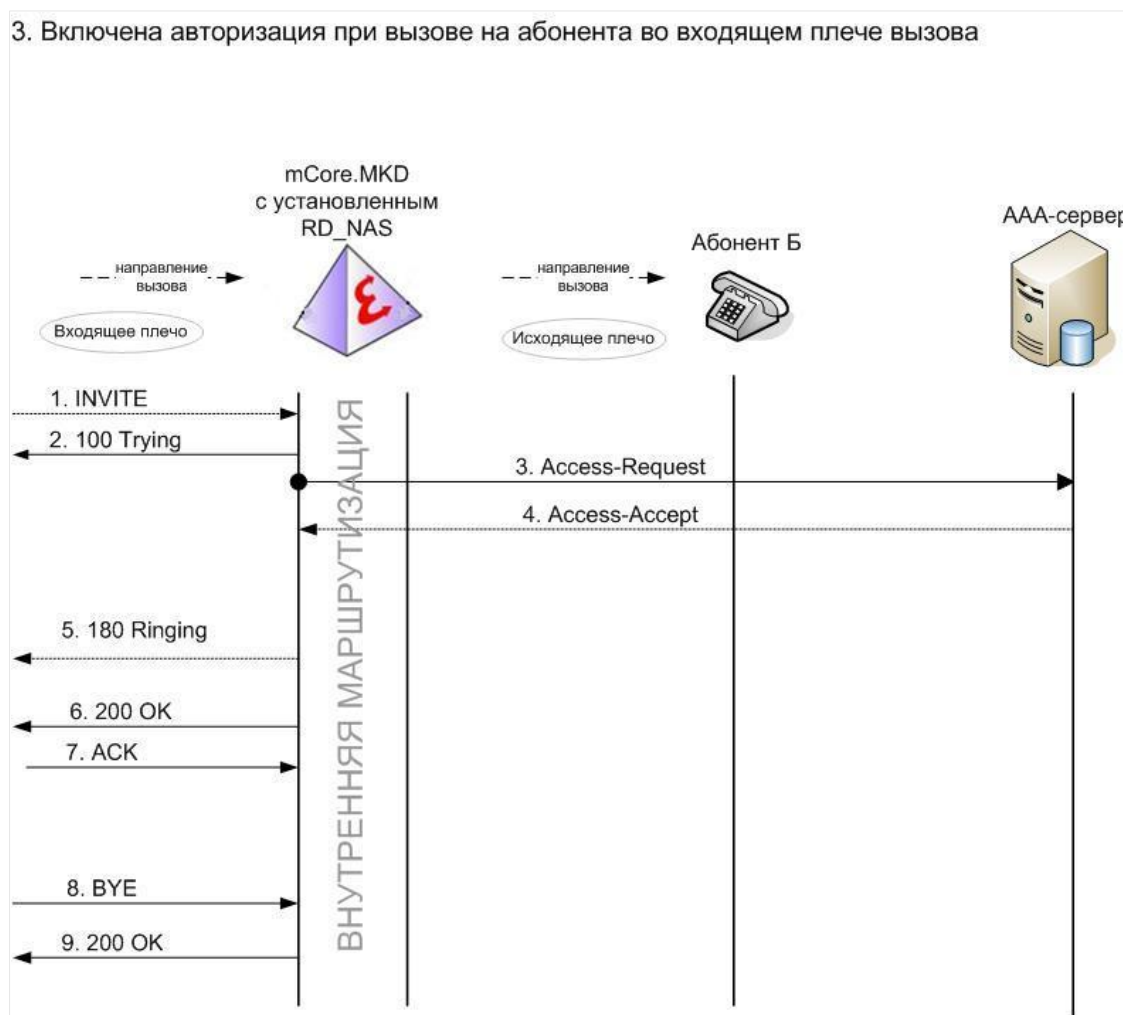


Рисунок 5 — Сценарий обмена сообщениями

1. Абонент А снимает трубку, набирает телефонный номер - формируется запрос INVITE, инициирующий установление сеанса связи;
2. mCore.MKD оповещает абонента А, что запрос принят и обрабатывается (100 Trying);
3. модуль RD_NAS запрашивает у AAA-сервера разрешение на доступ абонента А к телефонной связи - запрос Access-Request;
4. AAA-сервер посылает ответ - Access-Асcept, на разрешение доступа абонента А к телефонной связи;
5. mCore.MKD посылает абоненту А сообщение 180 Ringing;
6. вызываемый абонент снимает трубку, формируется сообщение 200 ОК;

7. станция абонента А отправляет подтверждение ACK;
8. после того как абонент А положит трубку, в mCore.MKD передается запрос BYE на разрыв соединения;
9. mCore.MKD подтверждает успешное завершение соединения отправкой сообщения 200 OK.

3.5.4 Авторизация/Аутентификация при вызове от абонента в исходящем плече вызова

Процедура авторизации/аутентификации осуществляется по номеру вызываемого абонента.

Сценарий обработки вызова представлен ниже (Рисунок 6).

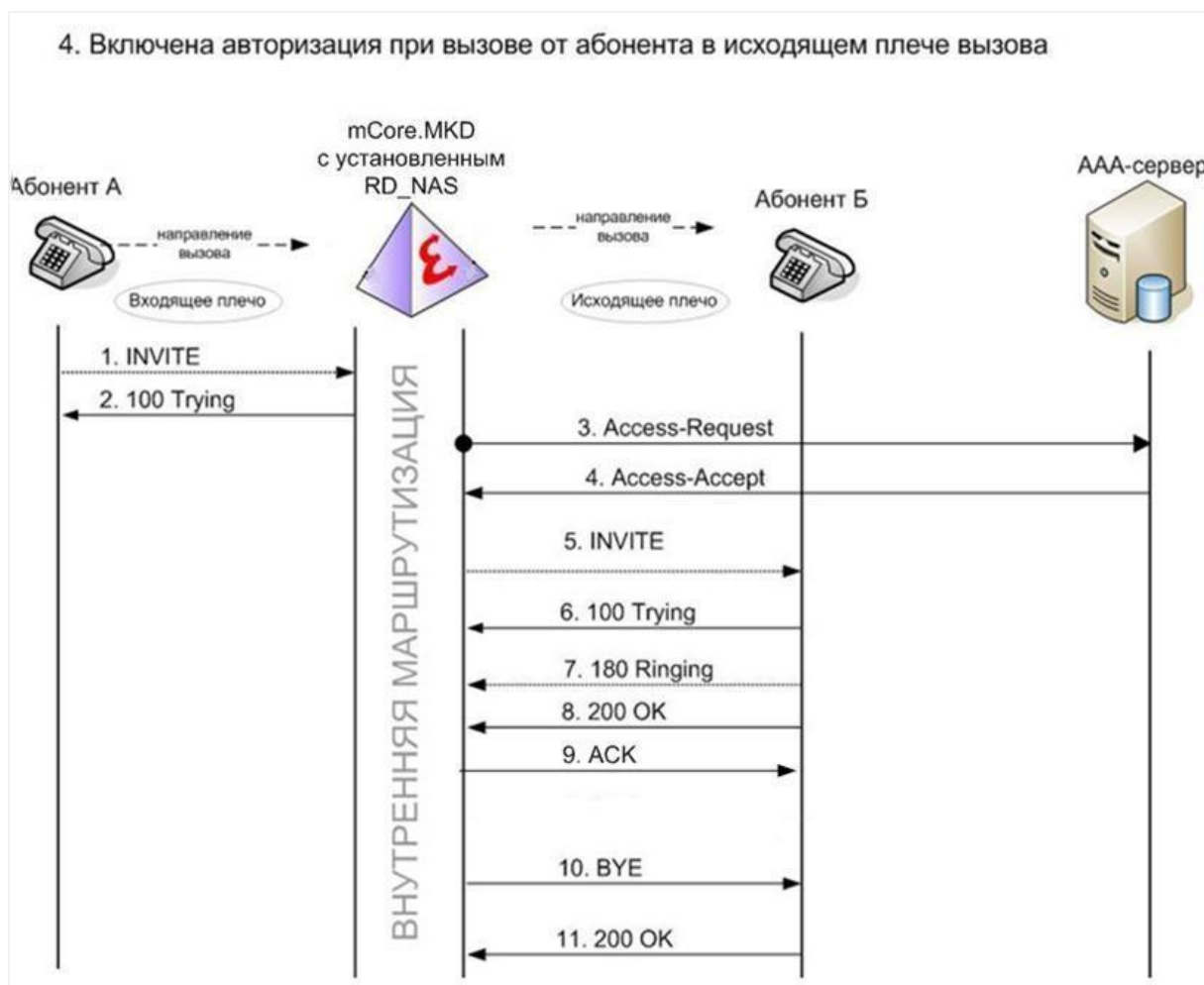


Рисунок 6 — Сценарий обмена сообщениями

1. Абонент А снимает трубку, набирает телефонный номер абонента Б - формируется запрос INVITE, инициирующий установление сеанса связи;
2. mCore.MKD оповещает абонента А, что запрос принят и обрабатывается (100 Trying);
3. модуль RD_NAS запрашивает у AAA-сервера разрешение на доступ абонента Б к телефонной связи - запрос Access-Request;
4. AAA-сервер посылает ответ - Access-Асcept, на разрешение доступа абонента Б к телефонной связи;
5. mCore.MKD выполняет маршрутизацию в направление абонента Б — формируется запрос INVITE;

6. станция абонента Б оповещает mCore.MKD что запрос принят и обрабатывается (100 Trying);
7. станция абонента Б оповещает mCore.MKD о получении входящего вызова посредством сообщения 180 Ringing;
8. абонент Б снимает трубку, формируется сообщение 200 ОК;
9. mCore.MKD отправляет стороне абонента Б сообщение ACK, подтверждающее прием сообщения 200 ОК;
10. после окончания соединения вызывающей стороной, mCore.MKD передает запрос BYE станции абонента Б на разрушение соединения;
11. станция абонента Б подтверждает успешное разрушение соединения отправкой сообщения 200 ОК.

3.5.5 Начисление платы при вызове от абонента во входящем плече вызова

Операция «Начисление платы» за входящие на MKD вызовы производится по номеру вызывающего абонента. Сценарий обработки вызова представлен ниже (Рисунок 7).

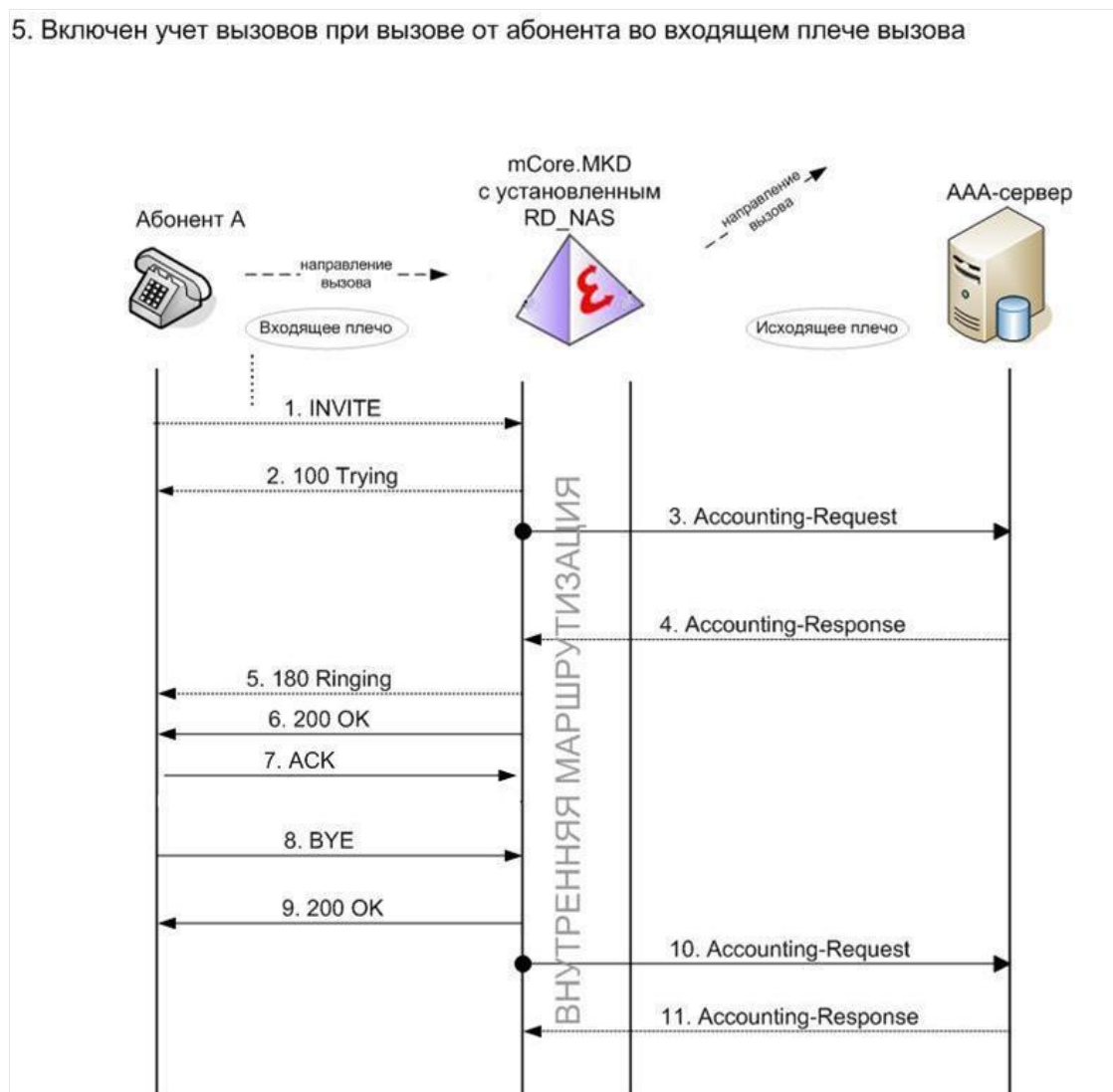


Рисунок 7 — Сценарий обмена сообщениями

1. Абонент А снимает трубку, набирает телефонный номер - формируется запрос INVITE, инициирующий установление сеанса связи;
2. mCore.MKD оповещает абонента А, что запрос принят и обрабатывается (100 Trying);
3. модуль RD_NAS посылает на AAA-сервер старт-пакет Accounting-Request, содержащий информацию, используемую для учета предоставляемых абоненту А услуг;
4. AAA-сервер подтверждает получение старт-пакета, возвращая пакет Accounting-Response;
5. mCore.MKD посылает абоненту А сообщение 180 Ringing;
6. вызываемый абонент снимает трубку, формируется сообщение 200 OK;
7. станция абонента А отправляет подтверждение ACK;
8. после того как абонент А положит трубку, в mCore.MKD передается запрос BYE на разрыв соединения;
9. mCore.MKD подтверждает успешное завершение соединения отправкой сообщения 200 OK.
10. модуль RD_NAS посылает на AAA-сервер стоп-пакеты Accounting-Request;
11. AAA-сервер подтверждает получение стоп-пакетов, возвращая пакеты Accounting-Response.

3.5.6 Начисление платы при вызове на абонента в исходящем плече вызова

Операция «Начисление платы» за вызовы, исходящие из MKD, производится по номеру вызываемого абонента. Сценарий обработки вызова представлен ниже (Рисунок 8).

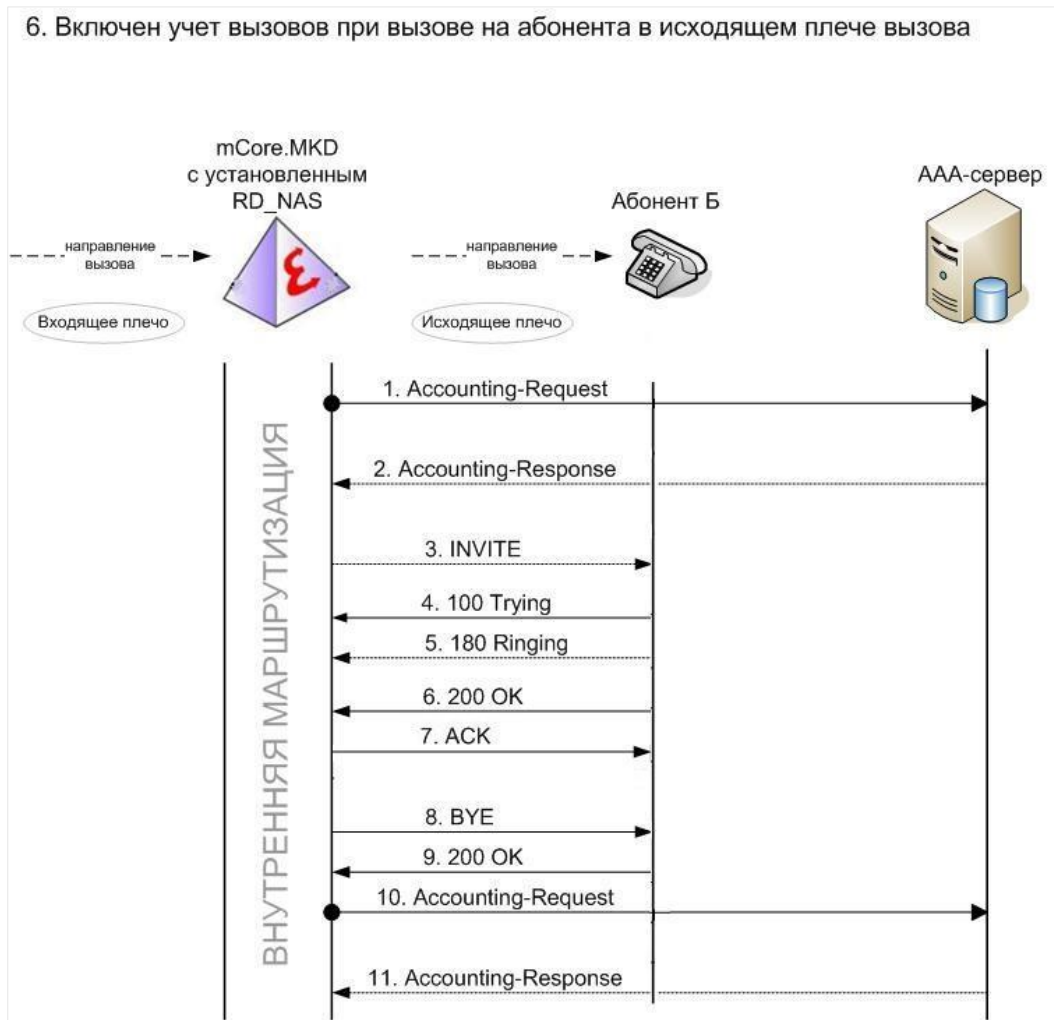


Рисунок 8 — Сценарий обмена сообщениями

В mCore.MKD поступает запрос на осуществление соединения с абонентом Б. mCore.MKD обрабатывает запрос по следующему сценарию:

1. модуль RD_NAS посылает на AAA-сервер старт-пакет Accounting-Request, содержащий информацию, используемую для учета предоставляемых абоненту Б услуг;
2. AAA-сервер подтверждает получение старт-пакета, возвращая пакет Accounting-Response;
3. mCore.MKD выполняет маршрутизацию в направлении абонента Б — формируется запрос INVITE;
4. станция абонента Б оповещает mCore.MKD что запрос принят и обрабатывается (100 Trying);
5. станция абонента Б оповещает mCore.MKD о получении входящего вызова посредством сообщения 180 Ringing;
6. абонент Б снимает трубку, формируется сообщение 200 ОК;

7. mCore.MKD отправляет стороне абонента Б сообщение ACK, подтверждающее прием сообщения 200 OK;
8. после окончания соединения вызывающей стороной, mCore.MKD передает запрос BYE станции абонента Б на разрушение соединения;
9. станция абонента Б подтверждает успешное разрушение соединения отправкой сообщения 200 OK.
10. модуль RD_NAS посылает на AAA-сервер стоп-пакеты Accounting-Request;
11. AAA-сервер подтверждает получение стоп-пакетов, возвращая пакеты Accounting-Response.

3.5.7 Начисление платы при вызове на абонента во входящем плече вызова

Операция «Начисление платы» за входящие на MKD вызовы производится по номеру вызываемого абонента. Сценарий обработки вызова представлен ниже (Рисунок 9).

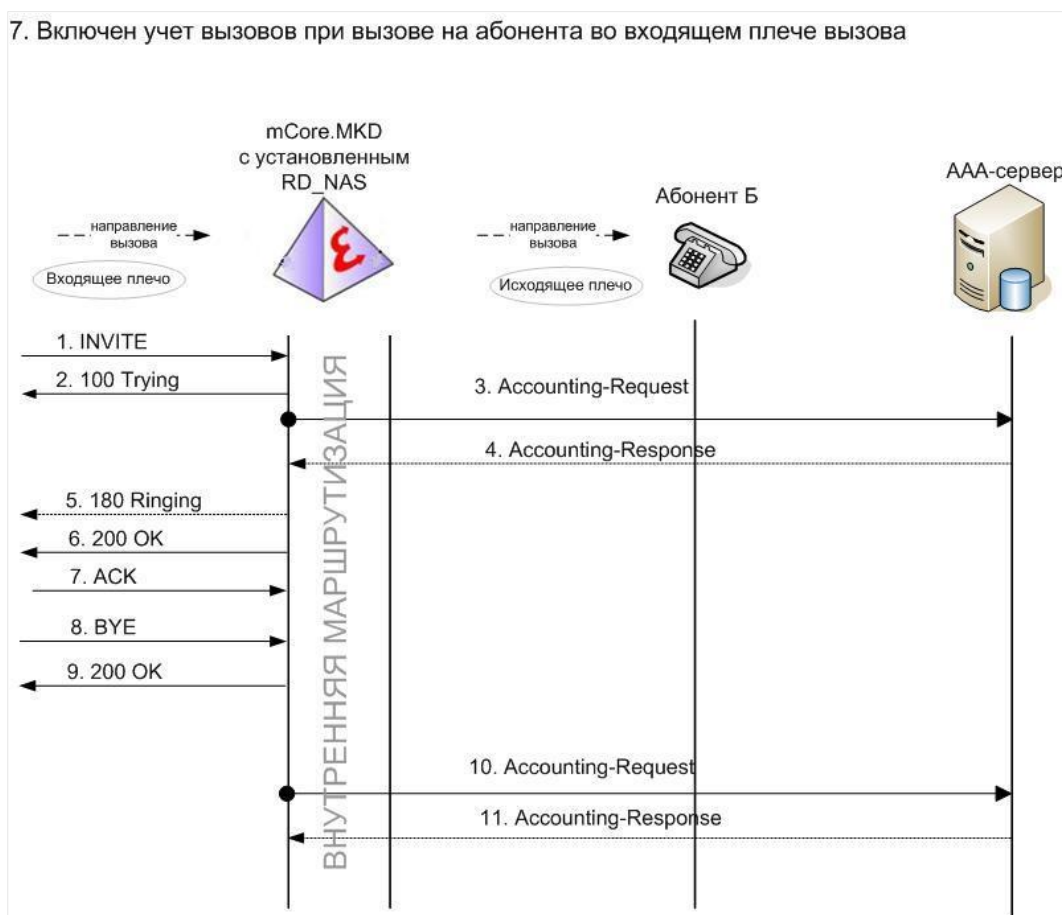


Рисунок 9 — Сценарий обмена сообщениями

1. абонент А снимает трубку, набирает телефонный номер - формируется запрос INVITE, инициирующий установление сеанса связи;
2. mCore.MKD оповещает абонента А, что запрос принят и обрабатывается (100 Trying);
3. модуль RD_NAS посылает на AAA-сервер старт-пакет Accounting-Request, содержащий информацию, используемую для учета предоставляемых абоненту Б услуг;

4. AAA-сервер подтверждает получение старт-пакета, возвращая пакет Accounting-Response;
5. mCore.MKD посылает абоненту А сообщение 180 Ringing;
6. вызываемый абонент снимает трубку, формируется сообщение 200 OK;
7. станция абонента А отправляет подтверждение ACK;
8. после того как абонент А положит трубку, в mCore.MKD передается запрос BYE на разрыв соединения;
9. mCore.MKD подтверждает успешное завершение соединения отправкой сообщения 200 OK.
10. модуль RD_NAS посылает на AAA-сервер стоп-пакеты Accounting-Request;
11. AAA-сервер подтверждает получение стоп-пакетов, возвращая пакеты Accounting-Response.

3.5.8 Начисление платы при вызове от абонента в исходящем плече вызова

Операция «Начисление платы» за вызовы, исходящие из MKD, производится по номеру вызывающего абонента. Сценарий обработки вызова представлен ниже (Рисунок 10).

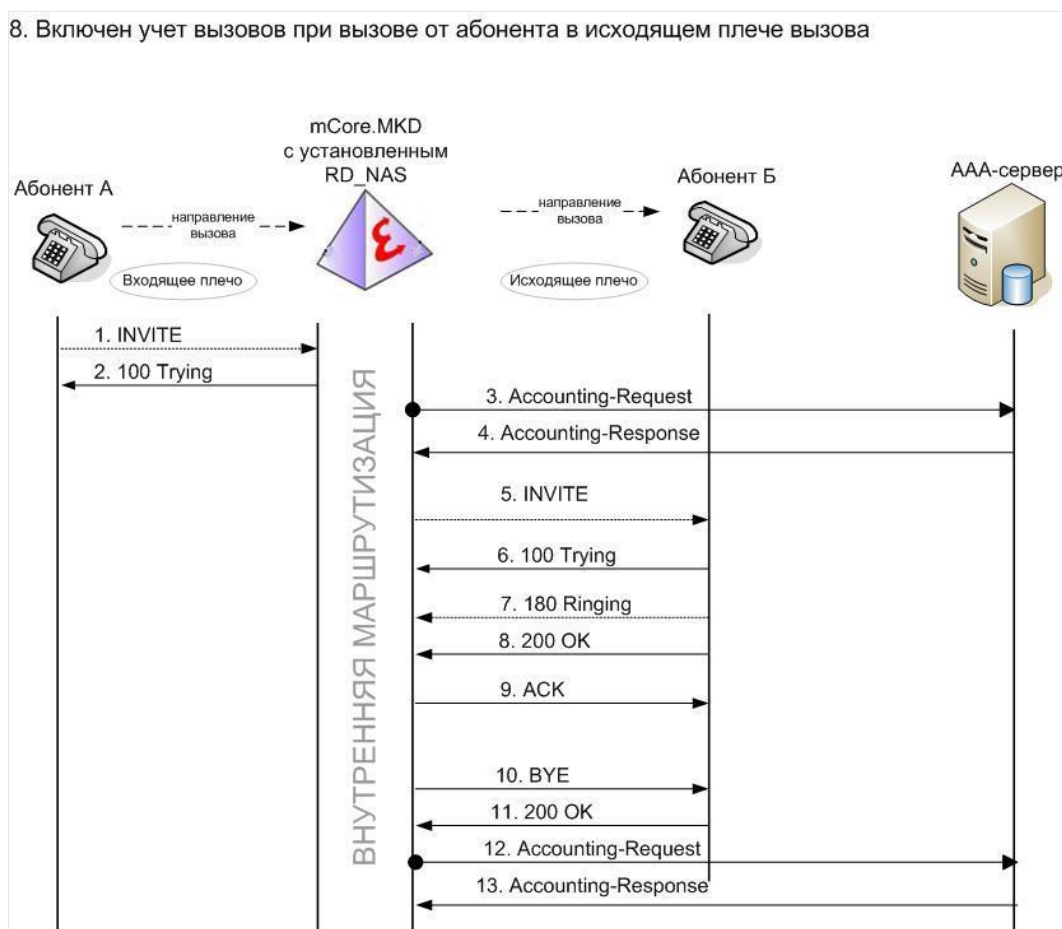


Рисунок 10 — Сценарий обмена сообщениями

1. абонент А снимает трубку, набирает телефонный номер абонента Б - формируется запрос INVITE, инициирующий установление сеанса связи;
2. mCore.MKD оповещает абонента А, что запрос принят и обрабатывается (100 Trying);

3. модуль RD_NAS посылает на AAA-сервер старт-пакет Accounting-Request, содержащий информацию, используемую для учета предоставляемых абоненту А услуг;
4. AAA-сервер подтверждает получение старт-пакета, возвращая пакет Accounting-Response;
5. mCore.MKD выполняет маршрутизацию в направлении абонента Б — формируется запрос INVITE;
6. станция абонента Б оповещает mCore.MKD что запрос принят и обрабатывается (100 Trying);
7. станция абонента Б оповещает mCore.MKD о получении входящего вызова посредством сообщения 180 Ringing;
8. абонент Б снимает трубку, формируется сообщение 200 ОК;
9. mCore.MKD отправляет стороне абонента Б сообщение ACK, подтверждающее прием сообщения 200 ОК;
10. после окончания соединения вызывающей стороной, mCore.MKD передает запрос BYE станции абонента Б на разрушение соединения;
11. станция абонента Б подтверждает успешное разрушение соединения отправкой сообщения 200 ОК.
12. модуль RD_NAS посылает на AAA-сервер стоп-пакеты Accounting-Request;
13. AAA-сервер подтверждает получение стоп-пакетов, возвращая пакеты Accounting-Response.

3.6 Описание perl-скрипта translate.pl

Конвертация запросов mCore.MKD в сообщения RADIUS построена на базе perl-скрипта `translate.pl`. По умолчанию данный скрипт находится в каталоге `/usr/protei/RD_NAS/config/perl`.

В файле `translate.pl` задаются правила формирования сообщений RADIUS из запросов, принятых от mCore.MKD.

Файл содержит секции для каждого сообщения RADIUS, которое можно отправлять на сервер RADIUS. Каждая из секций определяет содержимое RADIUS-запроса, устанавливает соответствие между параметрами сообщений RADIUS.

Таблица 3 — Описание секций скрипта `translate.pl`

Секция в файле	Сообщения, отправляемые на сервер RADIUS
<code>sub AccessReq</code>	Access-Request
<code>sub AccessAccept</code>	Access-Accept
<code>sub AcctStartReq</code>	Accounting-Request (Start)
<code>sub AcctStopReq</code>	Accounting-Request (Stop)
<code>sub InterimUpdateReq</code>	Interim-Update-Request

Формат параметров:

```
$perlContext->{"<radius_param>"} = NumTrans($callContext->{<mkd_param>});
```

где:

- radius_param — контекстные данные вызова с параметрами пользовательского запроса, инициировавшего услугу (передаваемые от mCore.MKD на RD_NAS по OMI-протоколу), см. Таблица 5;
- mkd_param — значения параметров сообщений RADIUS (параметры, формируемые в perl-процедуре).

Параметру сообщения RADIUS присваивается разрешенное текстовое значение или комбинация - в этом случае, текст берется в кавычки и отделяется "точкой".

Перечень разрешенных значений параметров сообщений RADIUS находится в папке ./config/rds_dictionary. Названия параметров попадают под маску dictionary*. Список параметров представлен в виде таблицы (Таблица 4):

Таблица 4 — Пример строк из таблицы параметров

Атрибут	Имя параметра сообщения RADIUS	Идентификатор параметра сообщения RADIUS	Тип параметра	Имя вендора
ATTRIBUTE	Protei-CallLeg	1	integer	Protei
ATTRIBUTE	Protei-Code	2	string	Protei

Пример perl-процедуры из файла translate.pl:

```
local $delimiter = " ";
sub AccessReq {
    my ($callContext,$serviceContext,$perlContext) = @_;
    $perlContext->{"User-Name"} = $callContext->{SerialNumber};
    $perlContext->{"Calling-Station-Id"} = $callContext->{CgPN};
    $perlContext->{"Called-Station-Id"} = $callContext->{CdPN};
    $perlContext->{"h323-conf-id"} = $callContext->{CallID};
    $perlContext->{"Cisco-AVPair"} = "param-id:45".$delimiter."param-name:test";
    $perlContext->{"Direction"} = "cisco_billing_master";
    return "OK";
}
sub AcctStartReq {
    my ($callContext,$serviceContext,$perlContext) = @_;
    $perlContext->{"Calling-Station-Id"} = $callContext->{CgPN};
    $perlContext->{"Called-Station-Id"} = $callContext->{CdPN};
    $perlContext->{"h323-conf-id"} = $callContext->{CallID};
    return "OK";
}
sub AcctStopReq {
    my ($callContext,$serviceContext,$perlContext) = @_;
    $perlContext->{"Calling-Station-Id"} = $callContext->{CgPN};
    $perlContext->{"Called-Station-Id"} = $callContext->{CdPN};
    $perlContext->{"h323-conf-id"} = $callContext->{CallID};
    $perlContext->{"Acct-Session-Time"} = $callContext->{CallDuration};
    return "OK";
}
```

```

}
sub InterimUpdateReq {
    my ($callContext,$serviceContext,$perlContext) = @_;
    $perlContext->{"Calling-Station-Id"} = $callContext->{CgPN};
    $perlContext->{"Called-Station-Id"} = $callContext->{CdPN};
    $perlContext->{"h323-conf-id"} = $callContext->{CallID};
    $perlContext->{"Acct-Session-Time"} = $callContext->{CallDuration};
    return "OK";
}
sub AccessAccept {
    my ($callContext,$serviceContext,$perlContext) = @_;
    my $CAVP = $callContext->{"Cisco-AVPair"};
    if ($CAVP =~ /\bh323-ivr-in=reauth:(\d+).*/ ) {
        $perlContext->{"ReauthIn"} = $1;
    }
    if ($CAVP =~ /.*\h323-ivr-out=reauth:(\d+).*/ ) {
        $perlContext->{"ReauthOut"} = $1;
    }
    $perlContext->{ServiceType} = $callContext->{"Service-Type"};
    return "OK";
}
}

```

3.6.1 Параметры взаимодействия mCore.MKD с RD_NAS

Список параметров, содержащихся в CallContext, передаваемых от mCore.MKD к RD_NAS и от RD_NAS к mCore.MKD по внутреннему OMI-протоколу, приведен ниже (Таблица 5 и Таблица 6).

Конкретные значения параметров, принятых и переданных RD_NAS по протоколу OMI, можно увидеть в отдельном журнале perl.log. Данный журнал расположен в каталоге /usr/protei/RD_NAS/logs.

Таблица 5 — Список параметров, передаваемых от mCore.MKD к RD_NAS

Параметр	Описание
PBX_ID	Идентификатор v(h)PBX.
PN	Номер абонента, для которого выполняется запрос
CgPN	Номер вызывающего абонента
CdPN	Номер вызываемого абонента
OgPN	Номер абонента, которому изначально предназначался вызов Примечание. Содержит значения "CallCompletion" или "AutoDial" при использовании этих услуг.
RdPN	Номер переадресации
RdPN_<1...n>	Цепочка номеров переадресаций 1 номер - OgPN, все последующие - RdPN
CgPC	Категория вызывающего абонента

Параметр	Описание
Call_ID	Идентификатор вызова
CallLeg_ID	Идентификатор плеча вызова
Domain	Домен абонента /vPBX Примечание. Может использоваться для заполнения некоторых параметров.
ACCESS_T	Время отправки ACCESS_REQ
ACCOUNT_T	Время отправки ACCOUNT_START
ANSWER_T	Время ответа абонента
STOP_T	Время отправки ACCOUNT_STOP
DURATION_T	Продолжительность разговора
SendPacketCount	Количество отправленных на MCU пакетов
SendOctetCount	Количество отправленных на MCU октетов
FracLost	Доля потерянных пакетов за последний отчетный период (0-1)
Lost	Количество потерянных пакетов от MCU за вызов
CName	Каноническое имя устройства, с которым работало MCU.
RTCP_Addr	Адрес устройства-отправителя RTCP
Cause	Код причины отбоя
Side	Сторона соединения, h323-call-origin (Cisco) = 26 Originate — входящая на MKD; Answer» — исходящая с MKD.
Local_HP	IP-адрес и порт сигнализации для взаимодействия с абонентом данного соединения
Remote_HP	IP-адрес и порт сигнализации для взаимодействия со встречным абонентом соединения
Insider	Является ли данный абонент внутренним
URI	URI внутреннего абонента
REGISTER	если это ACCESS на SIP::REGISTER
Password	Пароль профиля абонента Примечание. Если не задан, то пишется пробел « ».
Contact	Значение Contact абонента при регистрации

Таблица 6 — Список параметров, передаваемых от RD_NAS к mCore.MKD

Параметр	Описание
Limit_time	Максимальная продолжительность вызова для абонента

3.6.2 Журнал perl.log

Для мониторинга взаимодействия RD_NAS с mCore.MKD и AAA-сервером, существует отладочный журнал perl.log. Журнал находится в каталоге /usr/protei/RD_NAS/logs.

В данном журнале отображаются значения CallContext, принимаемые RD_NAS от mCore.MKD, то есть инициирование таких сообщений RADIUS запрашивает mCore.MKD.

Каждая запись выводится в одной строке, значения разделены пробелом или знаком «;» и располагаются в порядке, показанном ниже (Таблица 7).

Таблица 7 — Пример записи в журнале

Параметр	Описание
2009-11-27 16:51:16.683	Дата и время запроса
1	Идентификатор запроса
CallProcedure AccessReq callContext	Название вызываемой процедуры или
param->s=value	Параметры передаваемые от mCore.MKD к RD_NAS.

Пример записи в журнале perl.log:

```
2009-11-27 16:51:16.683 1 PerlInterpreter.cpp(208) Perl: translate.pl; 0 CallProcedure AccessReq
callContext: ACCESS T->s=1259315476669,ACCOUNT T->s=0,ANSWER T->s=0,CallLeg_ID-
>s=57784879173664832,Call_ID->s=57784879172616216,CdPN->s=921450,CgPC->s=-1,CgPN-
>s=921553,Contact->s=,Domain->s=62.231.191.8,Insider->s=1,Local_HP->s=62.231.161.221:5060,OgPN-
>s=,PBX_ID->s=2,PN->s=921553>Password->s=7558ssy3,REGISTER->s=0,RdPN->s=,Remote_HP->s=,STOP_T-
>s=0,Side->s=originate,URI->s=sip:921553@62.231.191.8 serviceContext: perlContext:
```

4 Настройка системы

Настройка системы осуществляется с помощью следующих конфигурационных файлов:

- rd_nas.cfg — настройка параметров взаимодействия с AAA-сервером;
- rd_nas_server.cfg — настройка параметров сервера RD_NAS;
- om_interface.cfg — настройка параметров OMI-соединения;
- trace.cfg — настройка параметров системы журналирования;
- ap.cfg — настройка параметров подсистемы сбора аварий.

4.1 Условные обозначения в таблицах параметров конфигурации

Графа «Значимость параметра/перезапуск» в таблицах конфигурации содержит буквенные коды, которые описаны в таблице ниже (Таблица 8).

Таблица 8 — Буквенные коды

Код	Описание
O	Optional. Опциональный параметр. Может отсутствовать в конфигурации, тогда параметру задается значение по умолчанию.
M	Mandatory. Обязательный параметр. При его отсутствии в конфигурации система не стартует, а при перегрузке конфигурации выдается ошибка.
X	Параметр зарезервирован. Использование запрещено.
N	Non reloadable. Параметр не перегружается динамически. Считывается при старте системы.

4.2 Настройка параметров взаимодействия с AAA-сервером (rd_nas.cfg)

Для связи модуля RD_NAS с AAA-сервером используется протокол RADIUS. Необходимые параметры данного соединения описаны в конфигурационном файле rd_nas.cfg.

Расположение: /usr/protei/RD_NAS/config/rd_nas.cfg

Параметры файла описаны ниже (Таблица 9).

Таблица 9 — Параметры конфигурационного файла rd_nas.cfg (Секция [Directions])

Параметр	OMPR	Описание
Секция [Directions]		
<name>	M/P	Название направления. Тип — string.
aaa_server	M/P	IP-адрес RADIUS-сервера. Тип — ip.

Параметр	OMPR	Описание
auth_port	M/P	Номер порта для авторизации. Тип — int.
acct_port	M/P	Номер порта для тарификации. Тип — int.
aaa_nas_ip	M/P	IP-адрес RD_NAS. Тип — ip.
aaa_secret	M/P	Секрет для декодирования пароля. Тип — string.

Примечание. Обязательно в конфигурации должно присутствовать направление с именем default.

Пример файла конфигурации:

```
[Directions]
default = {
    aaa_server = 192.168.100.187;
    auth_port = 1645;
    acct_port = 1646;
    aaa_nas_ip = 192.168.1.222;
    aaa_secret = testMKD;
};
```

4.3 Настройка параметров сервера RD_NAS (rd_nas_server.cfg)

Расположение: /usr/protei/RD_NAS/config/rd_nas_server.cfg

Параметры файла конфигурации ниже (Таблица 10).

Таблица 10 — Параметры конфигурационного файла rd_nas_server.cfg (секция **[General]**)

Параметр	OMPR	Описание
Секция [General]		
HandlersCount	O/P	Количество обработчиков. Тип — int.
OM_ServiceName	O/P	Имя сервиса в OMI v 4.1 (автоматические соединения). Тип — string.
RD_NAS_DI_Name	O/P	Название используемого интерфейса. Тип — string.

Пример файла конфигурации:

```
[General]
HandlersCount = 50;
OM_ServiceName = "RD_NAS";
RD_NAS_DI_Name = perl;
```

4.4 Настройка параметров OMI- соединения (om_interface.cfg)

Для связи mCore.MKD с модулем RD_NAS используется OMI-протокол. Необходимые параметры данного соединения описаны в конфигурационном файле om_interface.cfg.

Расположение: /usr/protei/RD_NAS/config/om_interface.cfg

Параметры файла конфигурации ниже (Таблица 11).

Таблица 11 — Параметры конфигурационного файла om_interface.cfg (секция **[Automatic]**)

Параметр	OMPR	Описание
ConnectionLogics		Обязательная подсекция. Задаются параметры сетевой логики.
SCL.RD_NAS	M/R	имя серверной сетевой логики;
port	M/R	Порт серверной сетевой логики. Тип — int.
Services		подсекция
Service.RD_NAS	M/R	Имя сервиса. Тип — string.
Connections	M/R	Перечень наименований сетевых подключений. Тип — list, элементы — string.

Пример файла конфигурации:

```
[Automatic]
ConnectionLogics = {
    SCL.RD_NAS = {
        Port = 35555;
    }
}
Services = {
    Service.RD_NAS = {
        Connections = { SCL.RD_NAS; };
    }
}
```

4.5 Настройка параметров системы журналирования (trace.cfg)

В данном конфигурационном файле настраиваются журналы, которые создает система.

Расположение: /usr/protei/RD_NAS/config/trace.cfg

Параметры файла описаны для разных секций в таблицах ниже.

Таблица 12 — Параметры конфигурационного файла trace.cfg (секция **[Trace]**, подсекция **Common**)

В данной подсекции задаются общие настройки системы журналирования.

Параметр	Значимость параметра	Описание
tracing	M	Флаг. Активность системы журналирования. Возможные значения: 1 - система активна; 0 - система отключена.
dir	O	Строка. Путь к каталогу, где будут храниться журналы. При необходимости система создаст недостающие каталоги. Значение по умолчанию = «./logs».
no_signal	O	Набор чисел через запятую или строка «all». Сигналы ОС Linux, не перехватываемые системой журналирования. Остальные сигналы система перехватывает и пишет об этом в основные журналы. Значение по умолчанию - перехватывать все. Значение all - вообще не перехватывать сигналы.

Таблица 13 — Параметры конфигурационного файла trace.cfg (секция **[Trace]**, подсекция **logs**)

В данной подсекции задаются конфигурации журналов.

Параметр	Значимость параметра	Описание
<p>Описание каждого журнала является опциональным. Параметры следующего формата: имя_журнала = { список_параметров };</p>		

Параметр	Значимость параметра	Описание
имя_журнала		<p>Зарезервированы следующие журналы:</p> <ul style="list-style-type: none"> - stdout - стандартный вывод; - trace - журнал "по умолчанию" - info — журнал для OMI-соединение вывода дополнительной информации; - warning - журнал предупреждений; - config — лог, отображающий все информацию, считанную с конфигурационных файлов; - perl — журнал параметров передаваемых от mCoreMKD к RD_NAS; - alarm_trace - детальная информация об авариях системы; - alarm_cdr — журнал типа cdr, содержит информацию об авариях системы.
file	O	<p>Строка. Путь к файлу лога. При необходимости недостающие каталоги создаются. Допускается задание пустого имени файла, если level=0, в этом случае запись производится согласно параметру tee. В случае отсутствия этого параметра, запись на диск не производится.</p> <p>Если путь начинается с "./" то путь берётся относительно текущего каталога, если с "/", то от корня, иначе - от каталога по умолчанию. Путь может содержать ".." и маску формата времени.</p> <p>Пример: cdr/%Y/%m/%d/%H_%M_%S.log преобразуется в cdr/2004/07/07/13_54_31.log.</p>

Параметр	Значимость параметра	Описание
mask	0	<p>Маска формата вывода автоматических полей в журнале. Пример маски: date & time & tick & state & pid & tid & level & file</p> <p>Модификаторы:</p> <p>date - дата в формате DD/MM/YY.</p> <p>time - время в формате HH:MM:SS.</p> <p>tick - миллисекунды. Если указан time, то подписываются как .MMM, иначе в формате MMMMMM.</p> <p>state - внутреннее состояние системы, может быть числом или строкой в зависимости от нужд разработчика.</p> <p>pid: идентификатор процесса. Число, 6 знаков.</p> <p>tid: идентификатор потока. Число, 6 знаков.</p> <p>level: уровень журнала, заданный для записи. Число.</p> <p>file: файл и строка в файле с исходным кодом, откуда производится вывод.</p>
level	0	Число. Уровень журнала. Специфично для журнала. Сообщения с уровнем, большим, чем level, игнорируются.
tee	0	<p>Строка. Дублирование потока вывода. Возможные значения: stdout, stderr, trace, info, warning или имя любого другого лога. Если перед именем написать «минус», например «-trace», то при дублировании не пишется имя исходного лога.</p> <p>Пример: tee=stdout & stderr & trace & info & warning or any your log.</p>

Пример файла конфигурации:

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

logs = {
    stdout = {
        mask = date & time & tick & level;
        level = 7;
        tee = trace
```



```
};
alarm_cdr = {
    file = alarm/alarm_cdr.log;
    mask = date & time & tick & level & file;
    level = 10;
};
alarm_trace = {
    file = alarm_trace.log;
    mask = date & time & tick & level & file;
    level = 10;
};
config = {
    file = config.log;
    mask = date & time & tick & level & file;
    level = 10;
};
trace = {
    file = trace.log;
    mask = date & time & tick & level & file;
    level = 10;
};
perl = {
    file = perl.log;
    mask = date & time & tick & level & file;
    level = 10;
};
config = {
    file = config.log;
    mask = date & time & tick & level & file;
    level = 10;
};
info = {
    file = info.log;
    mask = date & time;
    level = 4;
    tee = stdout:4;
};
warning = {
    file = warning.log;
    mask = date & time & tick & pid & tid & level & file;
    level = 4;
    tee = stdout:1 & my;
};
}
```

4.6 Настройка подсистемы сбора аварий (ap.cfg)

Данный конфигурационный файл не используется модулем RD_NAS и может быть пустым. Сам файл должен присутствовать в каталоге конфигурационных файлов /usr/protei/RD_NAS/config/.