

# **RD\_NAS**

Rrotei\_RD\_NAS

Основная версия

## **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

### **Авторские права**

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

---

## Оглавление

1 Общие сведения.....	4
1.1 Назначение документа.....	4
1.2 Состав документа.....	4
1.3 Техническая поддержка.....	5
1.3.1 Производитель.....	5
1.3.2 Служба технической поддержки.....	5
2 Глоссарий .....	6
3 Protei_RD_NAS.....	7
3.1 Функциональные возможности.....	7
3.2 Настройка работы модуля RD_NAS .....	8
3.2.1 Алгоритм работы .....	11
3.3 Описание perl-скрипта translate.pl.....	21
3.3.1 Параметры взаимодействия mCore.MKD с RD_NAS .....	23
3.3.2 Журнал perl.log.....	25
4 Конфигурация системы.....	27
4.1 Настройка параметров взаимодействия с AAA-сервером (rd_nas.cfg) .....	27
4.2 Настройка параметров сервера RD_NAS (rd_nas_server.cfg).....	28
4.3 Настройка параметров OMI- соединения (om_interface.cfg).....	29
4.4 Настройка параметров системы журналирования (trace.cfg).....	30
4.5 Настройка подсистемы сбора аварий (ap.cfg) .....	33

---

# 1 Общие сведения

## 1.1 Назначение документа

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

## 1.2 Состав документа

Настоящее руководство состоит из следующих основных частей:

«Общие сведения» – раздел, описывающий назначение изделия, режимы работы и их применение, основные характеристики изделия.

«Глоссарий» - раздел, содержащий расшифровки основных сокращений и аббревиатур, использованных в документе.

«Protei\_RD\_NAS» - раздел, раскрывающий назначение системы Protei\_RD\_NAS, особенности ее функциональности, настройка и работа модуля RD\_NAS в составе mCore.MKD.

«Конфигурация системы» - раздел, содержащий описание конфигурационных файлов, служащих для настройки системы.

### ***Внимание!***

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

Данный документ должен постоянно находиться при изделии.

## 1.3 Техническая поддержка

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

### 1.3.1 Производитель

НТЦ «ПРОТЕЙ»

194044, Санкт-Петербург

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

Бизнес-центр «Телеком СПб»

Тел.: (812) 449-47-27

Факс: (812) 449-47-29

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

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

### 1.3.2 Служба технической поддержки

НТЦ «ПРОТЕЙ»

194044, Санкт-Петербург

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

Бизнес-центр «Телеком СПб»

Тел.: (812) 449-47-31

(812) 449-47-27 доб. 5999 (круглосуточно)

Факс: (812) 449-47-29

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

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

---

## 2 Глоссарий

**AAA** - Authentication, Authorization, Accounting

**NAS** - Network Access Server

**Perl** — Practical Extraction and Report Language

**Протокол RADIUS** (Remote Authentication Dial-In User Service)— это отраслевой стандартный протокол, используемый для осуществления проверки подлинности, авторизации и учета.

**Клиент RADIUS** —сервер удаленного доступа (NAS) - отвечает за передачу сведений о пользователе заданным серверам RADIUS и дальнейшие действия в зависимости от возвращенной сервером информации. Клиент RADIUS использует специальный формат для передачи сведений - RADIUS-Message (сообщение RADIUS).

**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 транспортировка информации между клиентами и серверами.

**Оборудование mCore.MKD** – это программный коммутатор, выполняющий функции управляющего узла в IP-сети, маршрутизации вызова, управления шлюзовым оборудованием (например, mGate.ITG),оборудованием доступа (например, mAccess.MAK, mAccess.MTU), SIP-терминалами как отдельных, так и групп пользователей и применяющийся как в городских, так и в сельских телефонных сетях.

### 3 Protei\_RD\_NAS

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 (сервер 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 на рисунке 1.

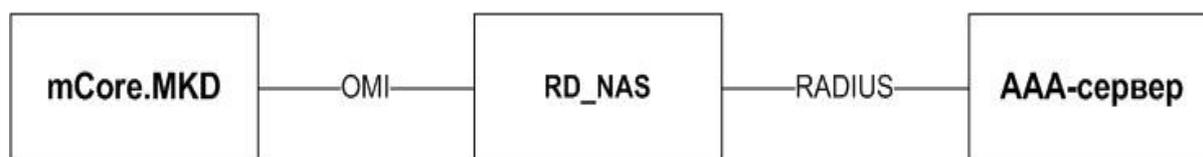


Рисунок 1 Общая схема взаимодействия модулей.

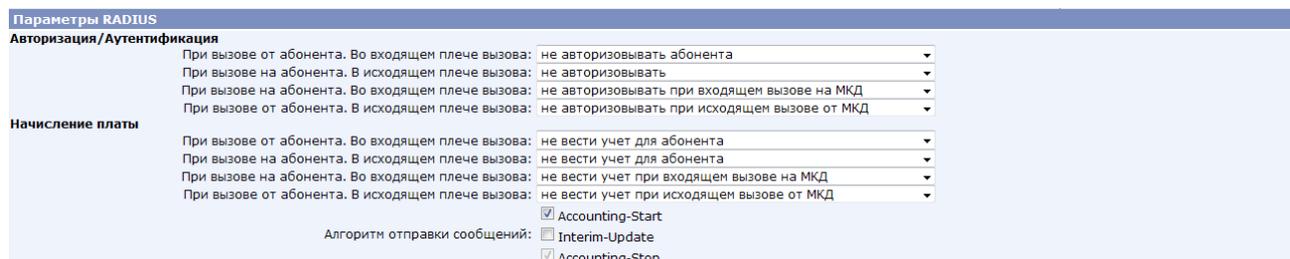
#### 3.1 Функциональные возможности

Модуль RD\_NAS выполняет следующие функции:

- RD\_NAS заполняет параметры сообщений RADIUS из параметров пользовательского запроса и наоборот;
- RD\_NAS по поступившим командам от mCore.MKD посылает на AAA-сервер сообщения RADIUS:
  - Access-Request - "запрос доступа". Запрос RD\_NAS, с которого начинается аутентификация и авторизация пользователя при попытке получить доступ к услугам сети;
  - Accounting-Request (Start/Stop) - "запрос учета". Данный запрос отправляется для учета стоимости телефонного разговора или предоставляемой услуги;
  - Interim-Update-Request – промежуточные сообщения отправляемые абоненту на AAA-сервер для промежуточного учета телефонного соединения.
- RD\_NAS получает от AAA-сервера следующие сообщения:
  - Access-Acception - "доступ разрешен". Положительный ответ на запрос доступа;
  - Access-Reject - "доступ не разрешен". Отрицательный ответ на запрос доступа. Возможен в том случае, если пользовательских данных недостаточно для успешной аутентификации или доступ для пользователя не авторизован;
  - Accounting-Response - ответ на запрос Accounting-Request (ведение/прекращение ведения учета стоимости). Посылается AAA-сервером на RD\_NAS в процессе учета стоимости предоставляемых услуг.

## 3.2 Настройка работы модуля RD\_NAS

Настройка алгоритма отправки сообщений RADIUS протокола RADIUS осуществляется с помощью Web TO mCoreMKD, блок «Параметры RADIUS» (Рисунок 2).



Параметры RADIUS	
<b>Авторизация/Аутентификация</b>	
При вызове от абонента. Во входящем плече вызова:	не авторизовывать абонента
При вызове на абонента. В исходящем плече вызова:	не авторизовывать
При вызове на абонента. Во входящем плече вызова:	не авторизовывать при входящем вызове на МКД
При вызове от абонента. В исходящем плече вызова:	не авторизовывать при исходящем вызове от МКД
<b>Начисление платы</b>	
При вызове от абонента. Во входящем плече вызова:	не вести учет для абонента
При вызове на абонента. В исходящем плече вызова:	не вести учет для абонента
При вызове на абонента. Во входящем плече вызова:	не вести учет при входящем вызове на МКД
При вызове от абонента. В исходящем плече вызова:	не вести учет при исходящем вызове от МКД
Алгоритм отправки сообщений:	
<input checked="" type="checkbox"/>	Accounting-Start
<input type="checkbox"/>	Interim-Update
<input checked="" type="checkbox"/>	Accounting-Stop

Рисунок 2

**Примечание.** По умолчанию параметры RADIUS не используются при создании профиля абонента. При необходимости использования какого-либо параметра следует убрать флажок в поле «По умолчанию» напротив требуемого параметра.

Алгоритм отправки запросов по протоколу RADIUS, при совершении вызова, определяется в абонентских настройках, так же в групповых настройках и в пункте «Остальные абоненты».

### Настройка операции «Авторизация/Аутентификация»

В пункте «**Авторизация/Аутентификация**» задается необходимость отправки сообщений Access-Request при установлении соединения.

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

- При вызове от абонента. Во входящем плече вызова: — в поле определяется необходимость проводить авторизацию/аутентификацию вызовов, входящих от абонентов на МКД, по номеру вызывающего абонента. Требуемое значение выбирается из выпадающего списка:
  - авторизовывать вызовы и ДВО;
  - авторизовывать только вызовы;
  - не авторизовывать абонента.
- При вызове на абонента. В исходящем плече вызова: - в поле определяется необходимость проводить авторизацию/аутентификацию вызовов, исходящих от МКД к абонентам, по номеру вызываемого абонента. Требуемое значение выбирается из выпадающего списка:
  - авторизовывать при всех вызовах;
  - авторизовывать при вызове от внешних абонентов;
  - авторизовывать при вызове от внутренних абонентов;
  - не авторизовывать.
- При вызове на абонента. Во входящем плече вызова: - в поле определяется необходимость проводить авторизацию/аутентификацию вызовов, входящих на МКД, по номеру вызывающего абонента. Требуемое значение выбирается из выпадающего списка:
  - авторизовывать при входящем вызове на МКД;
  - не авторизовывать при входящем вызове на МКД.
- При вызове от абонента. В исходящем плече вызова: - в поле определяется

необходимость проводить авторизацию/аутентификацию вызовов, исходящих от МКД, по номеру вызываемого абонента. Требуемое значение выбирается из выпадающего списка:

- авторизовывать при исходящем вызове от МКД;
- не авторизовывать при исходящем вызове от МКД.

## 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 МКД 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 МКД. 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 МКД вызовы. Требуемое значение выбирается из выпадающего списка:
  - вести учет при входящем вызове на МКД;
  - не вести учет при входящем вызове на МКД.
- При вызове от абонента. В исходящем плече вызова: - в поле определяется необходимость проводить начисление платы вызываемому абоненту за вызовы, исходящие из МКД. Требуемое значение выбирается из выпадающего списка:
  - вести учет при исходящем вызове от МКД;
  - вести учет при исходящем вызове от МКД.

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

В пункте меню «Авторизация на AAA при регистрации по SIP» задается необходимость отправки Access-Request при получении сообщения 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 в AAA-сервер можно для всех типов абонентов, как внешних, так и внутренних.

---

## **Настройка операции «Отправка InterimUpdate»**

В пункте меню «Отправка InterimUpdate» задается, нужно ли отправлять на AAA-сервер промежуточные сообщения (InterimUpdate).

### 3.2.1 Алгоритм работы

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

1. Модулю **RD\_NAS** от mCore.MKD передается информация, связанная с конкретным пользователем, инициировавшим вызов или запрос услуги.
2. После того, как **RD\_NAS** получает идентификационные данные пользователя, он производит процесс идентификации с использованием протокола RADIUS. Для этого RD\_NAS создает пакет Access-Request, содержащий такие атрибуты, как регистрационное имя пользователя, пароль, идентификатор клиента и порта (Port ID), через который регистрируется в системе данный пользователь.
3. Пакет Access-Request передается **AAA-серверу** через сеть. 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 в сеть. К таким требованиям относится проверка пароля, но в базе данных может также указываться клиент (клиенты) и порт (порты), через которые разрешен доступ пользователя.
6. При выполнении всех условий в отклик Access-Accept 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** через сеть. 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 отклик Accounting-Response в качестве подтверждения приема и успешной записи пакетов Accounting-Request.
7. После того как услуга предоставлена, **RD\_NAS** генерирует стоп-пакет Accounting-Request, описывающий тип предоставленного пользователю сервиса. Этот пакет может также включать статистические сведения о работе пользователя – продолжительность сеанса, число пакетов и байтов, принятых и переданных пользователем.
8. Стоп-пакет Accounting-Request передается **AAA-серверу**, который возвращает подтверждение приема в отклике Accounting-Response.

## Сценарии обмена сообщениями

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

Сценарий обработки вызова представлен на рисунке 3. Процедура авторизации/аутентификации осуществляется по номеру вызывающего абонента.

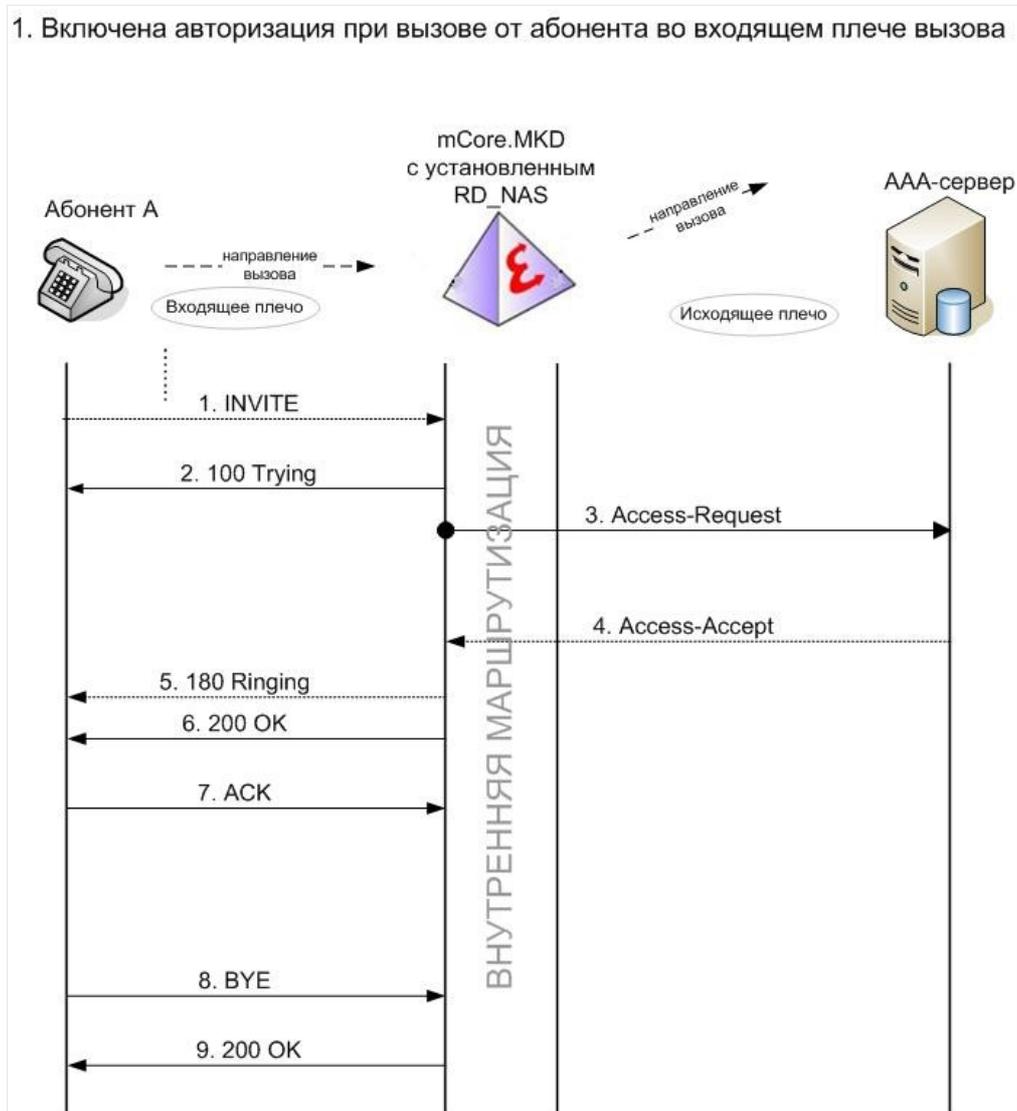


Рисунок 3. Сценарий обмена сообщениями

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-Ассерпт, 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 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 ОК.

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

Сценарий обработки вызова представлен на рисунке 4. Процедура авторизации/аутентификации осуществляется по номеру вызываемого абонента.



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-Ассерпт, на разрешение доступа абонента Б к телефонной связи;
3. mCore.MKD выполняет маршрутизацию в направление 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.

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

Сценарий обработки вызова представлен на рисунке 5. Процедура авторизации/аутентификации осуществляется по номеру вызывающего абонента.

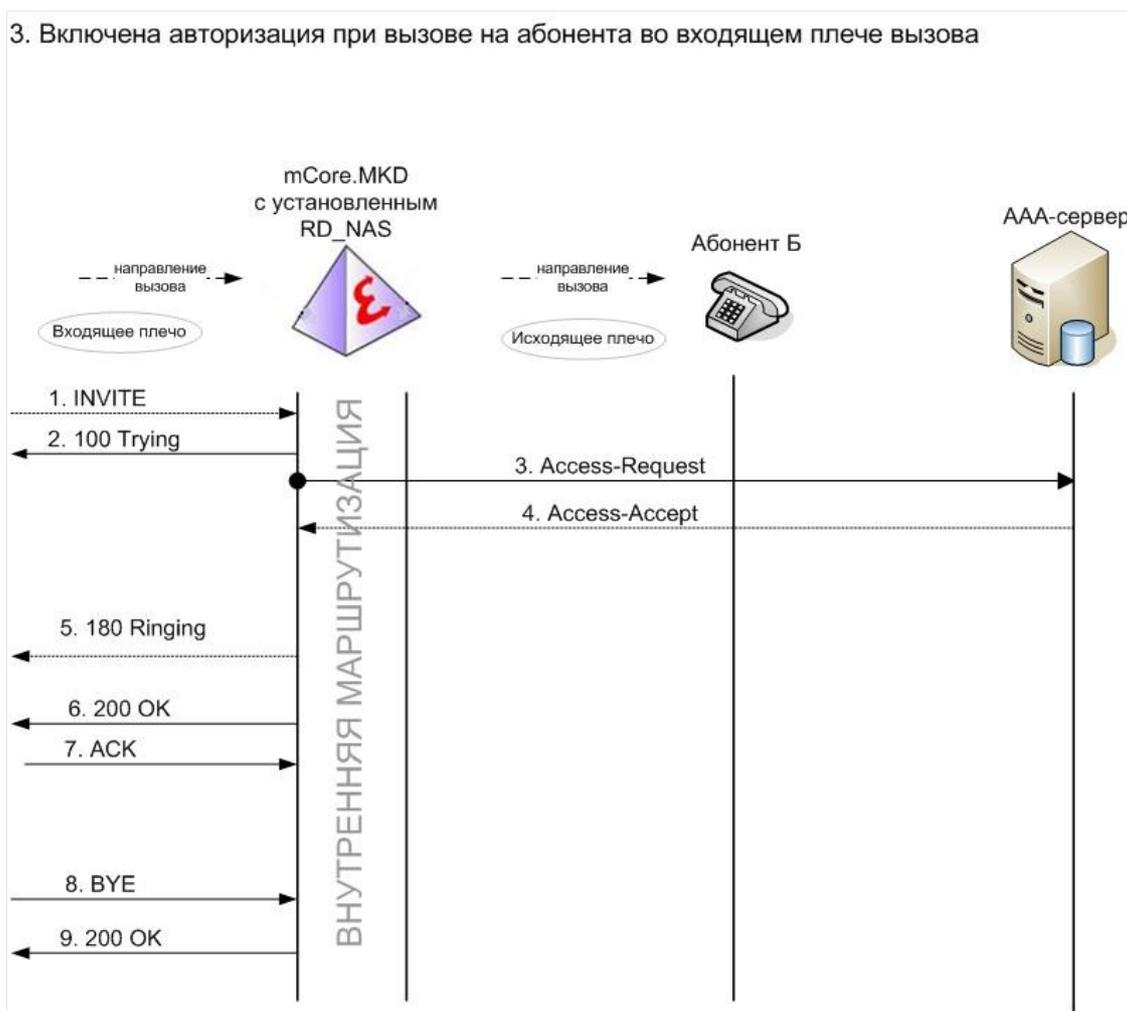


Рисунок 5. Сценарий обмена сообщениями

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-Асерт, 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 ОК.

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

Сценарий обработки вызова представлен на рисунке 6. Процедура авторизации/аутентификации осуществляется по номеру вызываемого абонента.

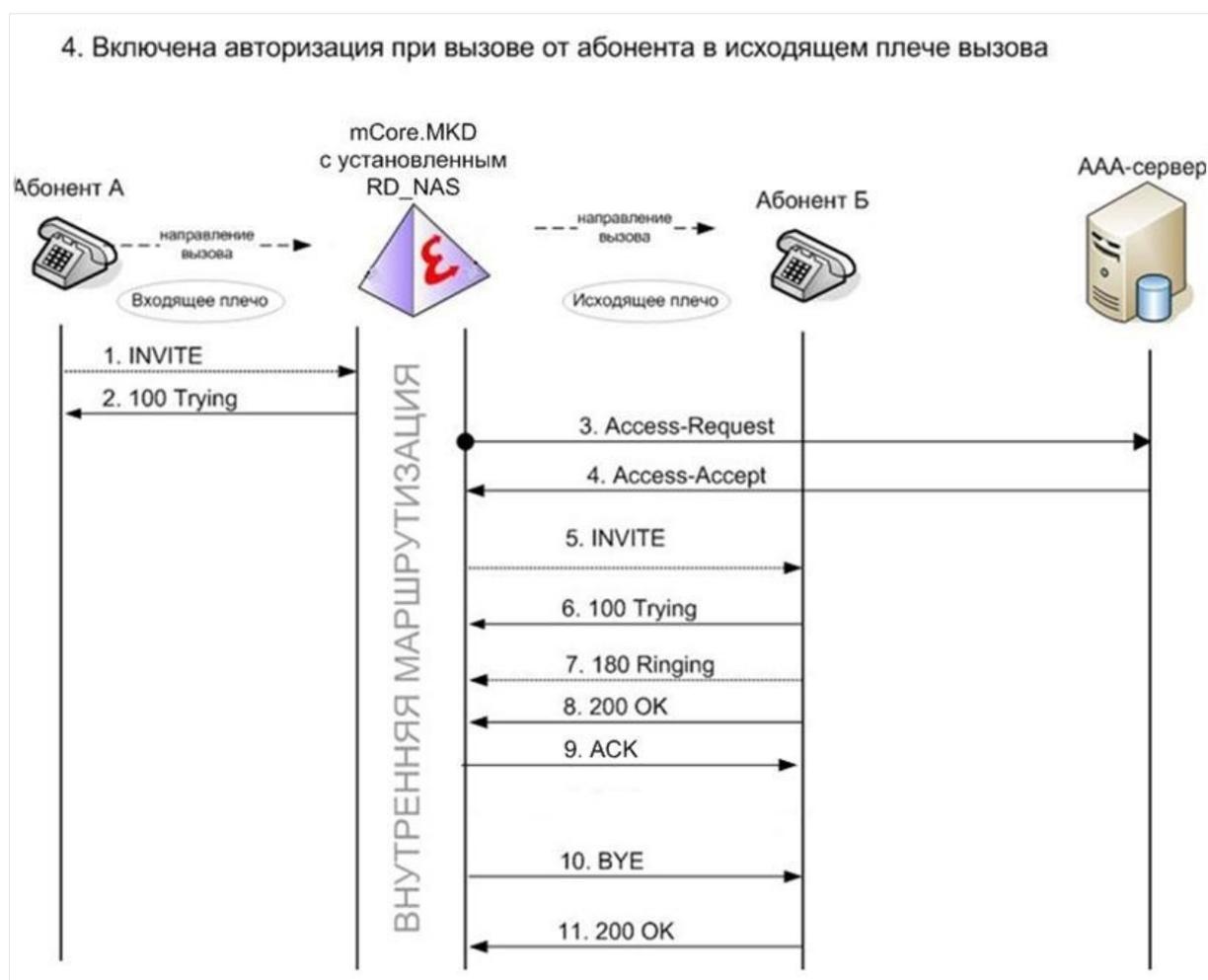


Рисунок 6. Сценарий обмена сообщениями

- 
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 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.

### Начисление платы при вызове от абонента во входящем плече вызова

Сценарий обработки вызова представлен на рисунке 7. Операция «Начисление платы» за входящие на МКД вызовы производится по номеру вызывающего абонента.

5. Включен учет вызовов при вызове от абонента во входящем плече вызова

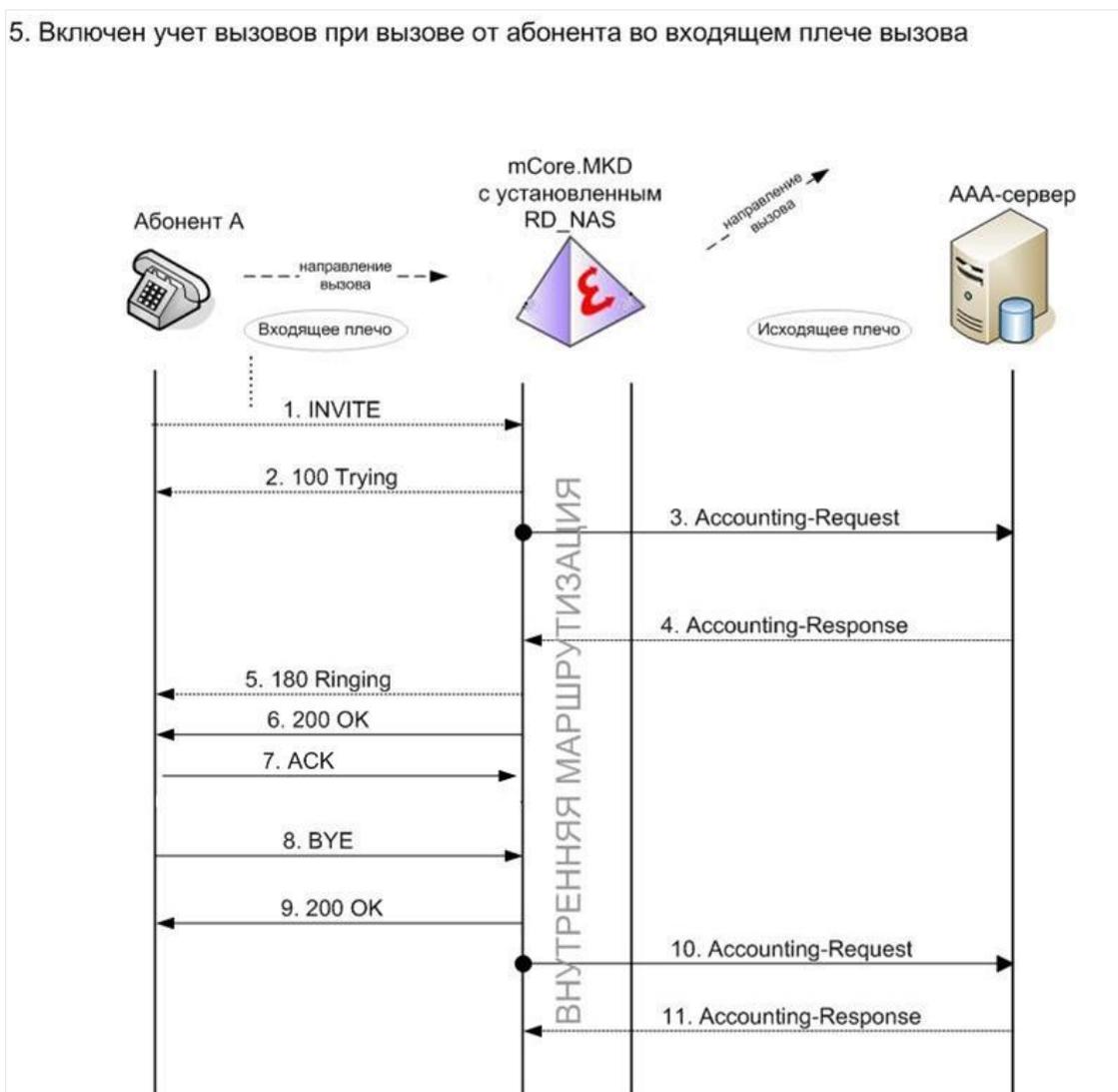


Рисунок 7. Сценарий обмена сообщениями

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 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 ОК.

10. модуль RD\_NAS посылает на AAA-сервер стоп-пакеты Accounting-Request;
11. AAA-сервер подтверждает получение стоп-пакетов, возвращая пакеты Accounting-Response.

## Начисление платы при вызове на абонента в исходящем плече вызова

Сценарий обработки вызова представлен на рисунке 8. Операция «Начисление платы» за вызовы, исходящие из МКД, производится по номеру вызываемого абонента.

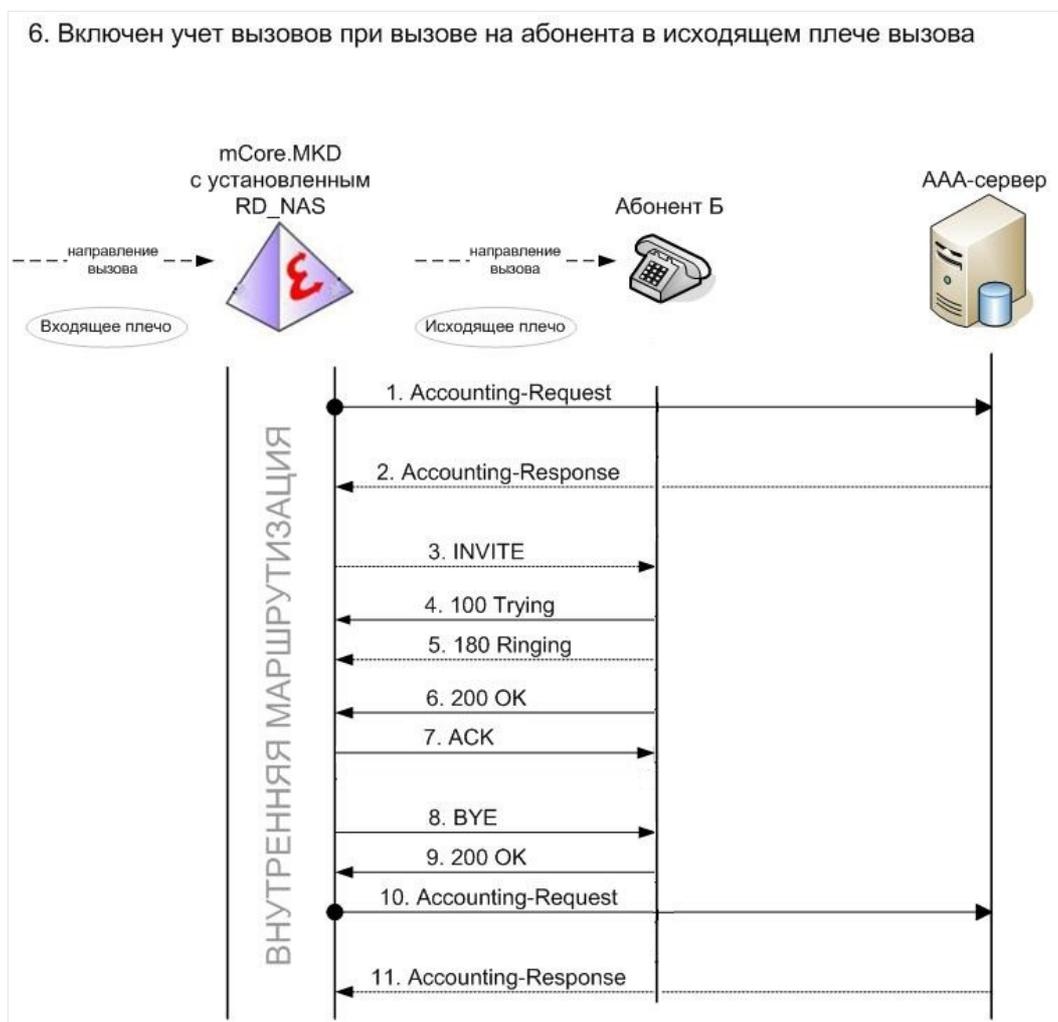


Рисунок 8. Сценарий обмена сообщениями

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 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.
10. модуль RD\_NAS посылает на AAA-сервер стоп-пакеты Accounting-Request;
11. AAA-сервер подтверждает получение стоп-пакетов, возвращая пакеты Accounting-Response.

### Начисление платы при вызове на абонента во входящем плече вызова

Сценарий обработки вызова представлен на рисунке 9. Операция «Начисление платы» за входящие на МКД вызовы производится по номеру вызываемого абонента.

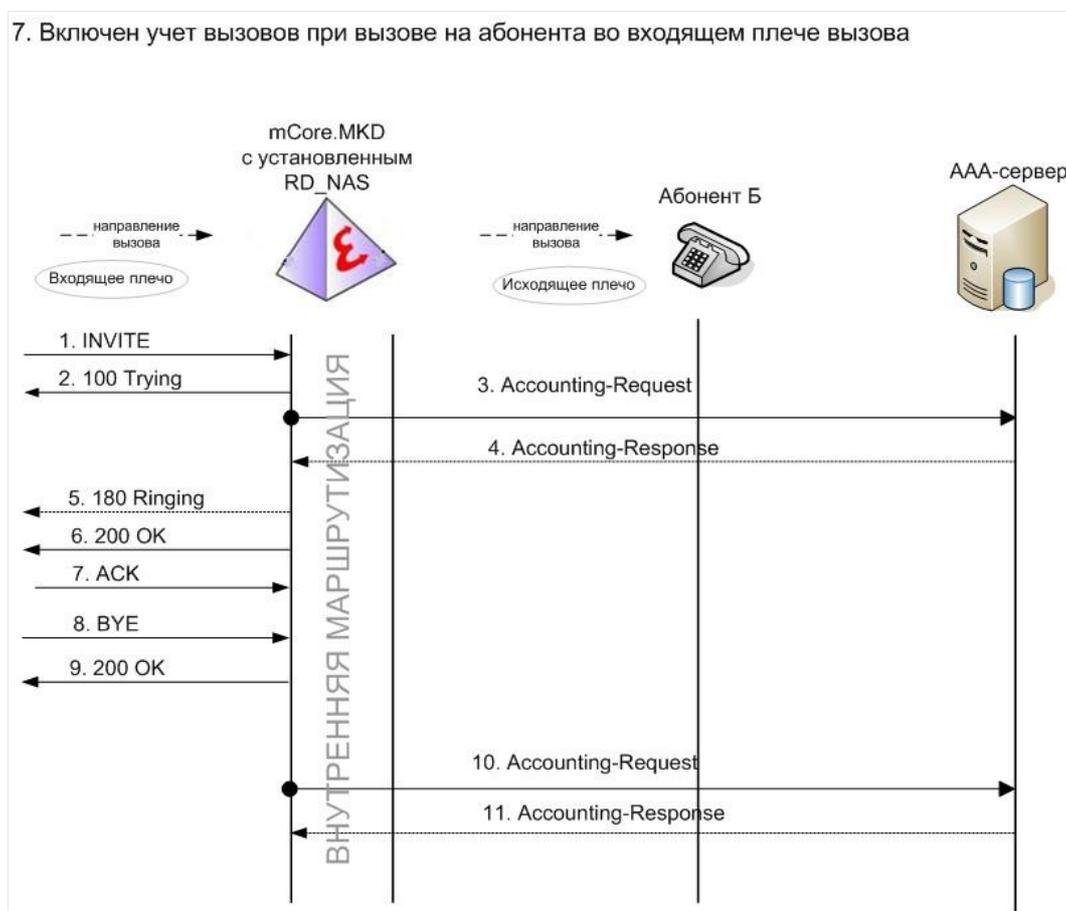


Рисунок 9. Сценарий обмена сообщениями

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.

### Начисление платы при вызове от абонента в исходящем плече вызова

Сценарий обработки вызова представлен на рисунке 10. Операция «Начисление платы» за вызовы, исходящие из МКД, производится по номеру вызывающего абонента.

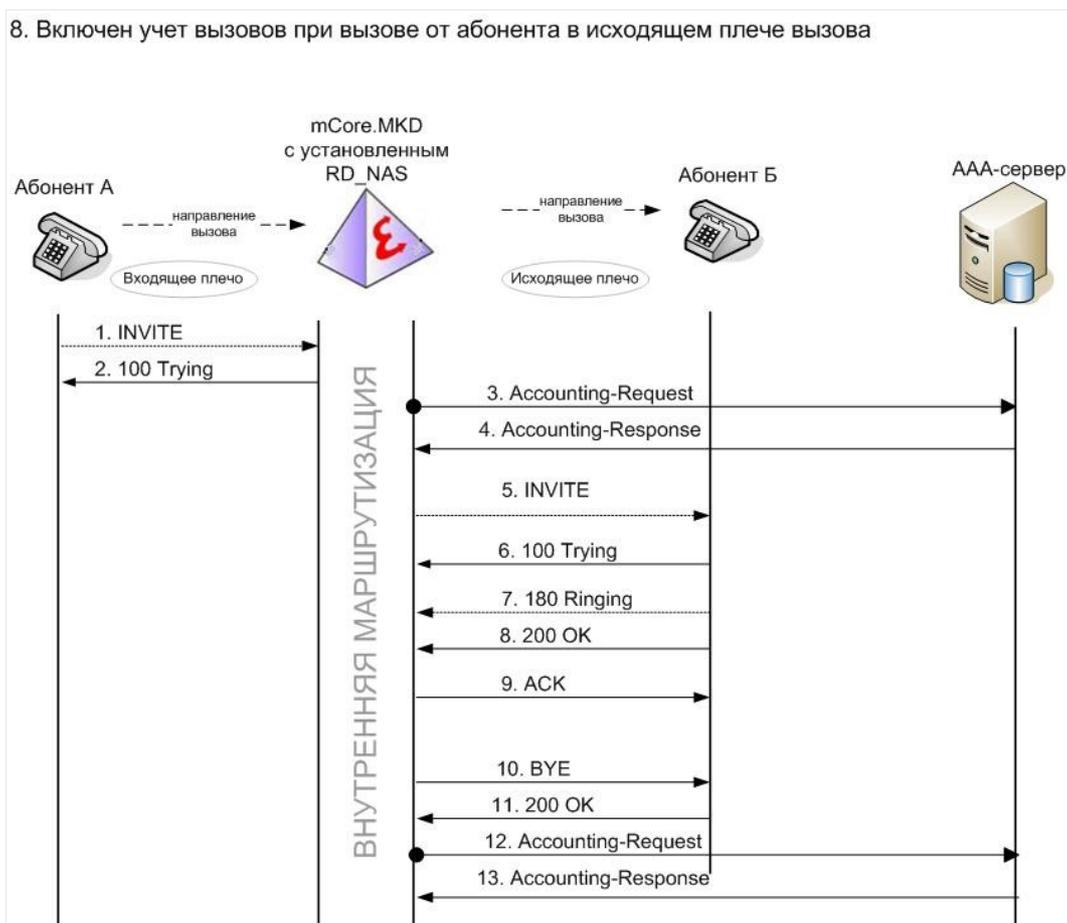


Рисунок 10. Сценарий обмена сообщениями

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 АСК, 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.

### 3.3 Описание perl-скрипта translate.pl

Конвертация запросов mCore.MKD в сообщения RADIUS построена на базе perl-скрипта — файл 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 translate.pl содержит в себе секции, отвечающие за каждое из сообщений RADIUS, которые можно отправлять на сервер RADIUS (см. таблицу 1). Каждая из секций определяет содержимое RADIUS-запроса, устанавливает соответствие между параметрами сообщений RADIUS.

Таблица 1.

Секция в файле translate.pl	Сообщения отправляемые на сервер RADIUS
sub AccessReq	Access Request
sub AccessAccept	Access Acception
sub AcctStartReq	Accounting-Request (Start)
sub AcctStopReq	Accounting-Request (Stop)

Секция в файле translate.pl	Сообщения отправляемые на сервер RADIUS
sub InterimUpdateReq	Interim Update Request

### Формат задания параметров:

```
$perlContext->{"<параметр сообщения RADIUS>"} = NumTrans($callContext->{"<параметр сообщения от mCore.MKD>"});
```

1. callContext – контекстные данные вызова, содержащие параметры пользовательского запроса, инициировавшего услугу (то есть данные, передаваемые mCore.MKD в RD\_NAS по OMI-протоколу) (см. таблицу 3);
2. perlContext – значения параметров сообщений RADIUS (то есть 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 (пример строк из таблицы параметров см. таблицу 2):

Таблица 2.

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";
}

```

### 3.3.1 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, приведены в таблицах 3 и 4. Конкретные значения параметров, принятых и переданных RD\_NAS протоколу OMI, можно увидеть в отдельном журнале perl.log. Данный журнал расположен в каталоге /usr/protei/RD\_NAS/logs.

Таблица 3. 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 (содержит значения "CallCompletion" или "AutoDial", если вызов осуществляется в рамках этих услуг (для обоих абонентов));

<b>Параметр</b>	<b>Описание</b>
RdPN	последний переадресованный вызов
RdPN_<1...n>	цепочка переадресаций, где 1 - OgPN, а n - RdPN
CgPC	категория абонента
Call_ID	идентификатор вызова
CallLeg_ID	идентификатор соединения
Domain	домен абонента (домен vPBX(МКД)) - может использоваться для заполнения некоторых параметров
ACCESS_T	время отправки ACCES_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 предполагаемых потерь за последний отчетный период(0-255)
Lost	количество потерянных пакетов от MCU за вызов.
CName	каноническое имя устройства с которым работало MCU.
RTCP_Addr	адрес устройства, отправителя RTCP
Cause	причина отбоя
Side	сторона соединения: «Originate» - входящая на МКД, «Answer» - исходящая с МКД h323-call-origin (Cisco) = 26

Параметр	Описание
Local_HP	IP-адрес и порт сигнализации для взаимодействия с абонентом данного соединения
Remote_HP	IP-адрес и порт сигнализации для взаимодействия с встречным абонентом соединения
Insider	является ли данный абонент внутренним
URI	если абонент внутренний, то здесь будет его URI
REGISTER	если это ACCESS на SIP::REGISTER
Password	пароль из профиля абонента. Если он не задан пишется пробел « ».
Contact	передается contact, указанный абонентом при регистрации

Таблица 4.Список параметров, передаваемых от RD\_NAS к mCore.MKD

Параметр	Описание
Limit_time	максимальная продолжительность вызова для абонента

### 3.3.2 Журнал perl.log

Для мониторинга взаимодействия RD\_NAS с mCore.MKD и AAA-сервером, существует дебаговый 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 каких сообщений RADIUS запрашивает mCore.MKD.

Каждая запись выводится в одной строке, значения разделены пробелом или знаком «;» и располагаются в следующем порядке:

Параметр	Описание
2009-11-27 16:51:16.683	Дата и время запроса
1	Номер запроса по порядку
CallProcedure AccessReq callContext	Название вызываемой процедуры или
<b>Параметр-&gt;s=Значение параметра</b>	Параметры передаваемые от mCore.MKD к RD_NAS. Параметры перечислены через запятую

---

### 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 Конфигурация системы

Настройка системы осуществляется с помощью 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 подсистемы сбора аварий.

### Условные обозначения

Графа «Значимость параметра/перезапуск» в таблицах конфигурации содержит буквенные коды.

Код	Описание
O	Optional. Опциональный параметр, может отсутствовать в конфигурации, тогда значение параметра заполняется значением по умолчанию.
M	Mandatory. Обязательный параметр. При его отсутствии в конфигурации система не стартует, а при перезагрузке конфигурации выдается ошибка.
X	Параметр зарезервирован. Использование запрещено.
N	Non reloadable. Параметр не перезагружается динамически. Считывается при старте системы.

### 4.1 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:

Обязательно в конфигурации должно присутствовать направление с именем default.

Секция **[Directions]**

Параметр	Значимость параметра	Описание
default	M	Направление <DirectionName::string>
aaa_server	M	IP-адрес Radius-сервера
auth_port	M	Строка. Номер порта для авторизации

Параметр	Значимость параметра	Описание
acct_port	М	Строка. Номер порта для аутентификации
aaa_nas_ip	М	IP-адрес NAS (свой)
aaa_secret	М	Строка. Ключ, согласованный с сервером, для декодирования пароля

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;

};
```

## 4.2 Настройка параметров сервера RD\_NAS (rd\_nas\_server.cfg)

Расположение: /usr/protei/RD\_NAS/config/rd\_nas\_server.cfg

Параметры файла конфигурации:

Секция **[General]**

Параметр	Значимость параметра	Описание
HandlersCount	О	Число. Количество логик
OM_ServiceName	О	Строка. Имя сервиса в OMI v 4.1 (автоматические соединения)
RD_NAS_DI_Name	О	используемые DataInterface

Пример файла конфигурации:

```
[General]

HandlersCount = 50;
OM_ServiceName = "RD_NAS";
RD_NAS_DI_Name = perl;
```

### 4.3 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:

Секция **[Automatic]**

Параметр	Значимость параметра	Описание
ConnectionLogics		Обязательная подсекция. Задаются параметры сетевой логики
SCL.RD_NAS	M	имя серверной сетевой логики;
port	M	Число. Порт данной серверной сетевой логики.
Services		подсекция
Service.RD_NAS	M	Имя сервиса
Connections	M	Список сетевых логик

Пример файла конфигурации:

```
[Automatic]

ConnectionLogics =
{
    SCL.RD_NAS =
    {
        Port = 35555;
    }
}

Services =
{
    Service.RD_NAS =
    {
        Connections = {SCL.RD_NAS;};
    }
}
```

## 4.4 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:

Секция **[Trace]**

Подсекция **Common**. В данной секция 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.

Подсекция **logs**. В данной секции задаются конфигурации журналов.

Параметр	Значимость параметра	Описание
Описание каждого журнала является опциональным. Параметры следующего формата: имя_журнала = { список_параметров };		
имя_журнала		Зарезервированы следующие журналы: <ul style="list-style-type: none"><li>• stdout - стандартный вывод;</li><li>• trace - журнал "по умолчанию"</li><li>• info — журнал для OMI-соединение</li></ul>

Параметр	Значимость параметра	Описание
		<p>вывода дополнительной информации;</p> <ul style="list-style-type: none"> <li>• warning - журнал предупреждений;</li> <li>• config — лог, отображающий все информацию, считанную с конфигурационных файлов;</li> <li>• perl — журнал параметров передаваемых от mCoreMKD к RD_NAS</li> <li>• alarm_trace – детальная информация об авариях системы</li> <li>• alarm_cdr — журнал типа cdr, содержит информацию об авариях системы.</li> </ul>
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><b>Пример:</b> cdr/%Y/%m/%d/%H_%M_%S.log преобразуется в cdr/2004/07/07/13_54_31.log.</p>
mask	O	<p>Маска формата вывода автоматических полей в журнале. Пример маски: date &amp; time &amp; tick &amp; state &amp; pid &amp; tid &amp; level &amp; file</p> <p><b>Модификаторы:</b></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>

Параметр	Значимость параметра	Описание
		<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	О	Число. Уровень журнала. Специфично для журнала. Сообщения с уровнем, большим, чем level, игнорируются.
tee	О	<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 &amp; stderr &amp; trace &amp; info &amp; warning or any your log.</p>

**Пример файла конфигурации:**

```
#possible mask : date & time & tick & pid & tid & level & file;
#possible tee  : stdout & stderr & trace & info & warning or any your log;

[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;
    };
}
```

## 4.5 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/.