



# **MERA VoIP Transit Softswitch v 2.1.8**

---

*Руководство Системного Администратора*

---

# MERA VoIP Transit Softswitch v 2.1.8

---

Документ №:	15032004SAG_Rus
Тип документа:	Руководство Системного Администратора
Дата выпуска:	15/03/2004

---

Copyright © 1999-2003 Mera Networks.  
All rights reserved.

Mera Networks оставляет за собой право вносить изменения в содержащуюся в данном документе информацию без предварительного уведомления.

## **ИНФОРМАЦИЯ О ПРАВЕ СОБСТВЕННОСТИ**

Информация, содержащаяся в данном документе, является собственностью Mera Networks. Никакая часть этого документа не может быть воспроизведена или заимствована в какой бы то ни было форме или каким-либо способом – в графическом, электронном виде или механическим путем, включая фотокопирование, запись, в том числе и на магнитные носители, или любые другие устройства, предназначенные для хранения информации – без письменного разрешения Mera Networks. Подобное разрешение не может быть выдано третьей стороной, будь то организация или частное лицо.

---

# Содержание

<b>1. ВВЕДЕНИЕ.....</b>	<b>8</b>
1.1. АННОТАЦИЯ .....	8
1.2. АУДИТОРИЯ .....	8
1.3. ОРГАНИЗАЦИЯ ДОКУМЕНТА .....	8
1.4. ТИПОГРАФИЧЕСКИЕ СОГЛАШЕНИЯ .....	9
1.5. СПИСОК ЛИТЕРАТУРЫ .....	9
<b>2. ОБЗОР MERA VOIP TRANSIT SOFTSWITCH .....</b>	<b>11</b>
2.1 НАЗНАЧЕНИЕ .....	11
2.2 ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ .....	11
2.3 ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ.....	16
2.4 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ .....	18
2.5 ПРОИЗВОДИТЕЛЬНОСТЬ .....	19
<b>3. УСТАНОВКА.....</b>	<b>21</b>
3.1. ПЛАНИРОВАНИЕ .....	21
3.1.1. АППАРАТНЫЕ СРЕДСТВА .....	21
3.1.2. ОПЕРАЦИОННАЯ СИСТЕМА .....	23
3.1.3. КОНФИГУРАЦИЯ СЕТИ.....	25
3.2. УСТАНОВКА АППАРАТНОГО ОБЕСПЕЧЕНИЯ .....	25
3.3. УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....	25
3.3.1. УСТАНОВКА ОПЕРАЦИОННОЙ СИСТЕМЫ .....	25
3.3.2. УСТАНОВКА MERA VOIP TRANSIT SOFTSWITCH.....	26
<b>4. НАСТРОЙКА.....</b>	<b>30</b>
4.1. ОБЗОР КОНФИГУРАЦИОННЫХ ФАЙЛОВ .....	30
4.2. НАСТРОЙКА ОСНОВНЫХ ПАРАМЕТРОВ .....	31
4.3. НАСТРОЙКА ФУНКЦИИ ПРИВРАТНИКА (GATEKEEPER).....	31
4.4. НАСТРОЙКА H.323 PROXY .....	31
4.5. НАСТРОЙКА СИСТЕМЫ НАЧИСЛЕНИЯ ПЛАТЫ .....	32
4.5.1. CDR-ФАЙЛЫ .....	32
4.5.2. RADIUS СЕРВИС УЧЕТА ПЛАТЫ (RADIUS ACCOUNTING).....	32
4.6. НАСТРОЙКА ПАРАМЕТРОВ АВТОРИЗАЦИИ.....	33
4.6.1. ОБЩИЙ ДОСТУП .....	33
4.6.2. АВТОРИЗАЦИЯ ПО IP АДРЕСУ .....	33
4.6.3. RADIUS (СЛУЖБА УДАЛЕННОЙ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ).....	33
4.7. НАСТРОЙКА ПРАВ ГРУПП ПОЛЬЗОВАТЕЛЕЙ.....	33
4.8. ПОДГОТОВКА ПЛАНА НАБОРА .....	34
4.9. ПОДГОТОВКА ФАЙЛА ДАННЫХ ШЛЮЗОВ .....	34
4.10. ПОДГОТОВКА ФАЙЛА ДАННЫХ RAS-ПОЛЬЗОВАТЕЛЕЙ.....	35
4.11. ПОДГОТОВКА ФАЙЛА ДАННЫХ ПРИВРАТНИКОВ (GATEKEEPERS).....	35
4.12. НАСТРОЙКА СРЕДСТВ ОБЕСПЕЧЕНИЯ УСТОЙЧИВОСТИ .....	35
4.12.1. УВЕДОМЛЕНИЯ ПО ЭЛЕКТРОННОЙ ПОЧТЕ .....	36
4.12.2. РЕЗЕРВИРОВАНИЕ СИСТЕМЫ, НАХОДЯЩЕЙСЯ ПОД КОММЕРЧЕСКОЙ НАГРУЗКОЙ .....	36

---

<b>5. АДМИНИСТРИРОВАНИЕ .....</b>	<b>39</b>
5.1. ЗАПУСК И ОСТАНОВ MVTS .....	39
5.1.1. ЗАПУСК MVTS .....	39
5.1.2. ОСТАНОВ MVTS .....	39
5.2. УПРАВЛЕНИЕ ФАЙЛОМ RAS-ПОЛЬЗОВАТЕЛЕЙ .....	39
5.2.1. КАК MVTS ИСПОЛЬЗУЕТ ФАЙЛ ДАННЫХ RAS-ПОЛЬЗОВАТЕЛЕЙ.....	39
5.2.2. СТРУКТУРА ФАЙЛА ДАННЫХ RAS-ПОЛЬЗОВАТЕЛЕЙ .....	39
5.2.3. ДОБАВЛЕНИЕ РЕГИСТРИРУЕМОГО ПОЛЬЗОВАТЕЛЯ.....	42
5.2.4. УДАЛЕНИЕ ПОЛЬЗОВАТЕЛЯ .....	43
5.2.5. ИЗМЕНЕНИЕ ПАРАМЕТРОВ.....	43
5.3. УПРАВЛЕНИЕ ФАЙЛОМ ДАННЫХ ШЛЮЗОВ.....	43
5.3.1. КАК MVTS ИСПОЛЬЗУЕТ ФАЙЛ ДАННЫХ ШЛЮЗОВ .....	43
5.3.2. СТРУКТУРА ФАЙЛА ДАННЫХ ШЛЮЗОВ .....	43
5.3.2.1. <i>ВЫБОР РЕЖИМА ПРОКСИРОВАНИЯ</i> .....	46
5.3.3. ДОБАВЛЕНИЕ ШЛЮЗА .....	46
5.3.4. УДАЛЕНИЕ ШЛЮЗА .....	46
5.3.5. ИЗМЕНЕНИЕ ПАРАМЕТРОВ.....	47
5.4. УПРАВЛЕНИЕ ФАЙЛОМ ДАННЫХ ПРИВРАТНИКОВ (GATEKEEPERS) .....	47
5.4.1. КАК MVTS ИСПОЛЬЗУЕТ ФАЙЛ ДАННЫХ ПРИВРАТНИКОВ .....	47
5.4.2. СТРУКТУРА ФАЙЛА ДАННЫХ ПРИВРАТНИКОВ .....	47
5.4.3. ДОБАВЛЕНИЕ ПРИВРАТНИКА (GATEKEEPER) .....	48
5.4.4. УДАЛЕНИЕ ПРИВРАТНИКА (GATEKEEPER) .....	48
5.4.5. ИЗМЕНЕНИЕ ПАРАМЕТРОВ ПРИВРАТНИКА.....	48
5.5. УПРАВЛЕНИЕ ПЛАНОМ НАБОРА.....	49
5.5.1. КАК MVTS ИСПОЛЬЗУЕТ ПЛАН НАБОРА .....	49
5.5.2. АЛГОРИТМ ВЫБОРА ОБЪЕКТА НАБОРА .....	50
5.5.3. СТРУКТУРА ПЛАНА НАБОРА .....	51
5.5.4. ДОБАВЛЕНИЕ ОБЪЕКТА НАБОРА .....	53
5.5.5. УДАЛЕНИЕ ОБЪЕКТА НАБОРА .....	53
5.5.6. НАСТРОЙКА ФУНКЦИИ ПЕРЕНАПРАВЛЕНИЯ (LOOK_AHEAD_ROUTING) .....	53
5.5.7. ИЗМЕНЕНИЕ ПАРАМЕТРОВ.....	56
5.6. УПРАВЛЕНИЕ НАЧИСЛЕНИЕМ ПЛАТЫ.....	56
5.6.1. ОРГАНИЗАЦИЯ НАЧИСЛЕНИЯ ПЛАТЫ .....	56
5.6.2. CDR-ФАЙЛЫ .....	56
5.6.3. RADIUS .....	57
5.7. УПРАВЛЕНИЕ ЗВОНКАМИ .....	57
5.7.1. ПРОСМОТР СПИСКА АКТИВНЫХ ЗВОНКОВ.....	57
5.7.2. ПРОСМОТР ДЕТАЛЬНОЙ ИНФОРМАЦИИ О ЗВОНКЕ .....	58
5.7.3. ПРИНУДИТЕЛЬНОЕ ЗАВЕРШЕНИЕ ЗВОНКА .....	58
5.7.4. ИМИТАЦИЯ ЗВОНКА.....	58
5.7.5. ПРОСМОТР ОБЩЕЙ СТАТИСТИКИ.....	58
5.8. ОБЕСПЕЧЕНИЕ УСТОЙЧИВОСТИ.....	59
5.8.1. ОГРАНИЧЕНИЕ ЧИСЛА ЗВОНКОВ .....	59
5.8.2. АВТОМАТИЧЕСКИЙ ПЕРЕЗАПУСК .....	59
5.8.3. ДЕЙСТВИЯ ПРИ ПЕРЕПОЛНЕНИИ ДИСКА .....	59
5.8.4. ДРУГИЕ ЗАДАЧИ ПО ОБЕСПЕЧЕНИЮ УСТОЙЧИВОСТИ.....	60
5.8.5. ПРОТОКОЛ РАБОТЫ.....	60

<b>6. СПРАВОЧНАЯ ИНФОРМАЦИЯ.....</b>	<b>62</b>
6.1. КОНФИГУРАЦИОННЫЕ ФАЙЛЫ.....	62
6.1.1. ОБЗОР .....	62
6.1.2. ФОРМАТ КОНФИГУРАЦИОННЫХ ФАЙЛОВ .....	62
6.1.3. СТРУКТУРА СИСТЕМНОГО КОНФИГУРАЦИОННОГО ФАЙЛА.....	63
6.1.4. СТРУКТУРА ФАЙЛА ДАННЫХ RAS-ПОЛЬЗОВАТЕЛЕЙ.....	105
6.1.5. СТРУКТУРА ФАЙЛА ДАННЫХ ШЛЮЗОВ .....	123
6.1.6. СТРУКТУРА ФАЙЛА ДАННЫХ ПРИВРАТНИКОВ (GATEKEEPER) .....	141
6.1.7. СТРУКТУРА ФАЙЛА С ПЛАНОМ НАБОРА .....	146
6.1.8. СТРУКТУРА ЗАПИСЕЙ ДАННЫХ О ЗВОНКЕ (CDRs).....	155
6.1.9. СТРУКТУРА ЗАПИСЕЙ ПРОТОКОЛА РАБОТЫ .....	157
6.2. ЯДРО .....	159
6.2.1. ПАРАМЕТРЫ КОМАНДНОЙ СТРОКИ .....	159
6.2.2. СКРИПТ ЗАПУСКА ЯДРА mp_kernel.sh.....	160
6.3. КОНСОЛЬ АДМИНИСТРИРОВАНИЯ .....	161
6.3.1. ВОЗМОЖНОСТИ, ПРЕДОСТАВЛЯЕМЫЕ КОНСОЛЮ .....	161
6.3.2. ПАРАМЕТРЫ КОМАНДНОЙ СТРОКИ .....	162
6.3.3. ПОДКЛЮЧЕНИЕ И ОТКЛЮЧЕНИЕ ОТ КОНСОЛИ АДМИНИСТРИРОВАНИЯ.....	162
6.3.4. КОМАНДЫ КОНСОЛИ АДМИНИСТРИРОВАНИЯ .....	163
6.4. HASP ДЛЯ ОПЕРАЦИОННОЙ СИСТЕМЫ LINUX .....	180
6.5. СИСТЕМА УДАЛЕННОГО АДМИНИСТРИРОВАНИЯ MVTs .....	181
6.5.1. УСТАНОВКА СИСТЕМЫ УДАЛЕННОГО АДМИНИСТРИРОВАНИЯ MVTs MS.....	182
6.6. WEB-ИНТЕРФЕЙС ТЕКУЩЕГО КОНТРОЛЯ WEB MONITOR.....	184
6.6.1. КОНФИГУРИРОВАНИЕ WEB-ИНТЕРФЕЙСА И РАБОТА С НИМ .....	184
6.6.2. РЕДАКТИРОВАНИЕ И ИЗМЕНЕНИЕ ЯЗЫКА ИНТЕРФЕЙСА .....	188
<b>7. ВЫЯВЛЕНИЕ И УСТРАНЕНИЕ СБОЕВ В РАБОТЕ .....</b>	<b>191</b>
<b>ПРИЛОЖЕНИЕ А: ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ .....</b>	<b>192</b>
ОПРЕДЕЛЕНИЯ.....	192
СОКРАЩЕНИЯ.....	192
СИНТАКСИС РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ .....	194
МАКРОИМЕНА В ШАБЛОНАХ ТРАНСЛЯЦИИ .....	199
АВТОРИЗАЦИЯ ПО ПРОТОКОЛУ RADIUS И ВНЕШНЯЯ МАРШРУТИЗАЦИЯ.....	201
КОНФИГУРИРОВАНИЕ ЯДРА ОПЕРАЦИОННОЙ СИСТЕМЫ .....	203
ДЕИНСТАЛЛЯЦИЯ MVTs .....	205
СОВМЕСТИМОСТЬ .....	206
<b>ПРИЛОЖЕНИЕ В: НАИБОЛЕЕ ЧАСТО ВСТРЕЧАЮЩИЕСЯ КОДЫ РАЗЪЕДИНЕНИЯ, ГЕНЕРИРУЕМЫЕ MVTs .....</b>	<b>207</b>
<b>ПРИЛОЖЕНИЕ С: ПЕРЕЧЕНЬ ПОДДЕРЖИВАЕМЫХ ИДЕНТИФИКАТОРОВ MIB ОБЪЕКТОВ.....</b>	<b>209</b>

---

## Рисунки

Рис. 2-1. Организация межоператорского обмена трафиком .....	12
Рис. 2-2. Построение частных виртуальных сетей (VPNs) .....	13
Рис. 2-3. Обеспечение безопасности операторской сети .....	14
Рис. 2-4. Совместимость с биллинговыми системами. Развитая система контроля ....	15
Рис. 3-1. Структура файловой системы .....	27
Рис. 5-1. Алгоритм выбора объекта набора.....	51
Рис. 7 Архитектура системы удаленного администрирования MVTS MS .....	181
Рис. 6-8 Диалог авторизации web-интерфейса.....	185
Рис. 9 Главное меню Администратора    Рис. 10 Гл. меню клиента .....	185

---

## Таблицы

Таблица 1-1: Соглашения и обозначения .....	9
Таблица 1-2: Справочная литература.....	9
Таблица 2-1 Производительность MVTS .....	20
Таблица 3-1 Рекомендуемая конфигурация (только привратник) .....	21
Таблица 3-2 Рекомендуемая конфигурация (без проксирования речевого трафика) ...	21
Таблица 3-3 Рекомендуемая конфигурация (полное проксирование).....	22
Таблица 3-4 Группы пользователей MVTS .....	24
Таблица 4-1: Конфигурационные файлы MVTS.....	30
Таблица 5-1: Параметры RAS-пользователя .....	40
Таблица 5-2: Параметры шлюза .....	43
Таблица 5-3: Параметры привратника .....	47
Таблица 5-4: Параметры объекта набора (dial peer) .....	51
Таблица 6-1: Структура записи с данными о звонке (CDR).....	155
Таблица 6-2: Параметры командной строки ядра MVTS (mp_kernel.d.x).....	159
Таблица 6-3: Поля, выводимые командой show call для группы звонков.....	168
Таблица 6-4: Поля, выводимые командой show call для заданного звонка .....	168
Таблица 6-5: Поля, выводимые командой show dp для каждого объекта набора .....	171
Таблица 6-6: Параметры, отображаемые командой show stat .....	174
Table 6-7: Параметры, выводимые по команде show stat gw .....	176
Таблица 6-7: Поля, выводимые командой show ep.....	179

# 1. ВВЕДЕНИЕ

## 1.1. АННОТАЦИЯ

В настоящем руководстве приводится описание системы MERA VoIP Transit Softswitch v 2.1.8 (в дальнейшем **MVTS**), ее структуры, порядка установки и настройки. В руководстве подробно отражены возможности по администрированию и настройке отдельных подсистем.

## 1.2. АУДИТОРИЯ

Руководство предназначено для системного администратора, чьей задачей является установка, настройка и эксплуатация MERA VoIP Transit Softswitch v 2.1.8

## 1.3. ОРГАНИЗАЦИЯ ДОКУМЕНТА

Документ состоит из следующих основных частей:

- **Глава 1. Введение**

Содержит сведения о назначении и организации документа.

- **Глава 2. Обзор MERA VoIP Transit Softswitch**

Содержит краткое описание продукта: его назначение, варианты применения, поддерживаемые функции и стандарты, и, кроме того, общие сведения об организации программного и аппаратного обеспечения продукта.

- **Глава 3. Установка**

Рассматривает процесс установки продукта, начиная с требований к аппаратной части системы и заканчивая вопросами установки прикладного программного обеспечения.

- **Глава 4. Настройка**

Приводит сведения о конфигурационных файлах, порядке настройки основных параметров системы, функции привратника (gatekeeper), N323 прокси (проху), настройке системы начисления платы (billing system) и прав доступа. Освещает вопросы настройки прав пользователей и средств обеспечения устойчивости, создания файлов данных.

- **Глава 5. Администрирование**

Описывает возможности администрирования и управления, предоставляемые системой.

Данная глава разделена на подглавы, позволяющие получить информацию о порядке запуска и останова системы, управлении файлами данных шлюзов, пользователей, управлении планом набора, звонками, функциями начисления платы и обеспечения устойчивости



- **Глава 6. Справочная информация**

Содержит справочную информацию, позволяющую получить представление о структуре и формате конфигурационных файлов, организации файлов данных и структуре их отдельных записей. Приводит сведения о ядре операционной системы и возможностях консоли администрирования, параметрах командной строки и отдельных команд.

- **Глава 7. Выявление и устранение сбоев в работе**

Содержит признаки, позволяющие самостоятельно определить причину недостатков в работе системы и способы устранения возникающих проблем.

## 1.4. ТИПОГРАФИЧЕСКИЕ СОГЛАШЕНИЯ

Обозначения, использованные в данном документе, приведены ниже (см. Таблицу 1-1).

**Таблица 1-1: Соглашения и обозначения**

Пример	Назначение
<b>Примечание:</b> <i>текст</i>	Информация, требующая специального внимания.
[N]	Ссылка на другой документ (см. раздел 1.5).
<code>void</code>	Примеры исходного кода, информации, выводимой программой, протоколов, конфигурационных файлов и т.п.
<b>ulimit</b>	Имя программы или файла, когда оно встречается в основном тексте документа.

## 1.5. СПИСОК ЛИТЕРАТУРЫ

**Таблица 1-2: Справочная литература**

Ссылка	Название документа
[1]	ITU-T Recommendation H.323 “Packet-based multimedia communications systems”
[2]	RFC 1889 “RTP: A Transport Protocol for Real-Time Applications. Audio-Video”
[3]	ITU-T Recommendation H.245 “Control protocol for multimedia communication”
[4]	ITU-T Recommendation H.225, “Call signalling protocols and media stream packetization for packet based multimedia communication systems”
[5]	ITU-T Recommendation E.164, “Assigned Country codes”

---

Ссылка	Название документа
[6]	Remote Authentication Dial-In User Service (RADIUS), RFC 2138, April 1997 ( <a href="http://www.pasteur.fr/cgi-bin/mfs/01/21xx/2138?8#mfs">http://www.pasteur.fr/cgi-bin/mfs/01/21xx/2138?8#mfs</a> )
[7]	RADIUS Accounting, RFC 2139, April 1997 ( <a href="http://www.pasteur.fr/cgi-bin/mfs/01/21xx/2139">http://www.pasteur.fr/cgi-bin/mfs/01/21xx/2139</a> )
[8]	Jeffrey Friedl “Mastering Regular Expressions”, O’Reilly, 1997, ISBN: 5-318-00056-8 Фридл Дж., “Регулярные выражения.” Библиотека программиста. Изд. Питер, 2001 г. FreeBSD Handbook. The FreeBSD Documentation Project. <a href="#">Copyright</a> © 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002 by The FreeBSD Documentation Project
[9]	Red Hat Linux Manuals. Red Hat Linux 7.2. “x86 Installation Guide”, “Getting Started Guide”, “Customization Guide”, “Reference Guide”. <a href="#">Copyright</a> © by Red Hat Inc.
[10]	ITU-T Recommendation T.38 “Procedures for real-time Group 3 facsimile communication over IP networks”, June 1998
[11]	Mozilla Public License, version 1.1. <a href="http://www.mozilla.org/MPL/MPL-1.1.html">http://www.mozilla.org/MPL/MPL-1.1.html</a>
[12]	ITU-T Recommendation Q.931. “ISDN user-network interface layer 3 specification for basic call control. “

## 2. ОБЗОР MERA VOIP TRANSIT SOFTSWITCH

### 2.1 НАЗНАЧЕНИЕ

MERA VoIP Transit Softswitch (сокращенно MVTS) - это программное VoIP устройство коммутации вызовов с функциями привратника (gatekeeper) и проксирования. MVTS представляет собой программно-аппаратный комплекс, подключаемый к сети IP-телефонии, с целью маршрутизации трафика IP-телефонии между шлюзами.

Внедрение MVTS позволяет провайдеру достичь следующих результатов:

- Существенно упростить процесс администрирования и наращивания сети (новые шлюзы регистрируются только на MVTS).
- Получить более гибкие возможности концентрации и маршрутизации трафика (в том числе динамическая маршрутизация, балансировка нагрузки шлюзов в пределах одного направления).
- Решить проблему совместимости оборудования разных производителей (например, Cisco и VocalTec).
- Осуществлять авторизацию и биллинг в одной точке, пользуясь одной системой стандарта RADIUS.
- Обеспечить сокрытие реальной структуры собственной сети и корпоративных сетей клиентов.
- Создавать и администрировать виртуальные частные сети компаний-клиентов с собственными планами нумерации.
- Получить надежную систему с высокой производительностью и развитой системой контроля.

### 2.2 ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ

В силу своей универсальности MVTS может иметь множество применений. С учетом аудитории, на которую рассчитано данное руководство, ниже приводятся лишь некоторые основные из возможных примеров использования программного продукта. Профессиональный поставщик услуг, искушенный во всех тонкостях рынка, несомненно определит много других выгодных вариантов практического применения продукта.

## Управление концентрацией и маршрутизацией трафика

**MVTS** предоставляет эффективные механизмы транзита сигнального (H.323) и медиа (RTP/RTCP) трафиков. Провайдер самостоятельно задает правила маршрутизации медиапотоков на основании данных об их инициаторе и направлении, доступности и приоритете шлюза места назначения. Дополнительно предоставляется возможность управления приоритетом IP-пакетов транзитных медиапотоков.

Производительность **MVTS** позволяет передавать до 3000 одновременных звонков с учетом проксирования голосового трафика на каждом сервере.

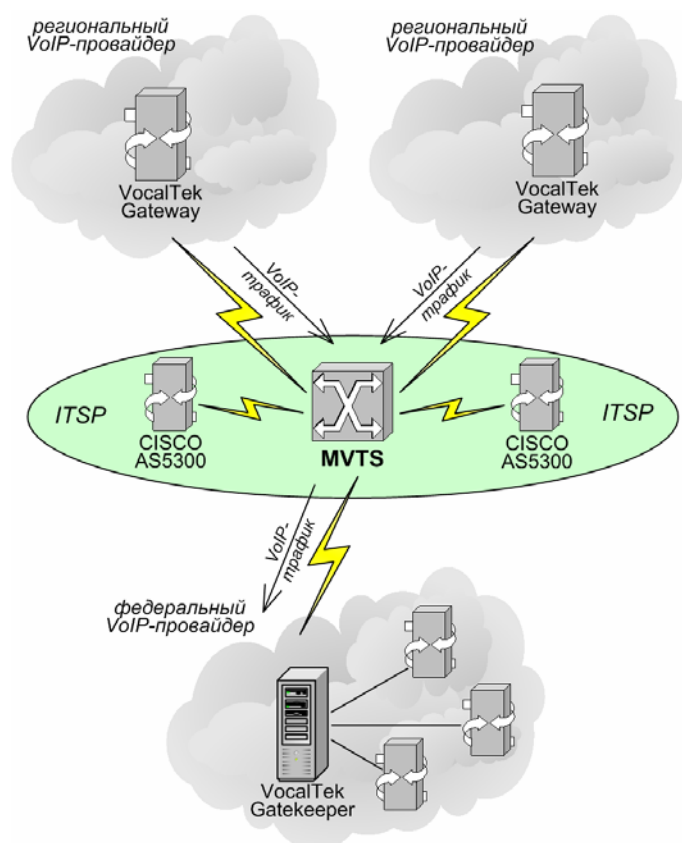


Рис. 2-2-1. Организация межоператорского обмена трафиком

### Объединение H.323-сетей различного типа

Благодаря поддержке нескольких H.323-диалектов одновременно с проксированием трафика **MVTS** позволяет производить трансляцию одного типа H.323-стандарта в другой. Таким образом, продукт выступает конвертером трафика, решая проблему объединения сетей, построенных на оборудовании разных производителей.

**MVTS**, динамически регистрирующийся на привратнике, в ведении которого находятся несколько шлюзов, решает двойную задачу: выступает в роли привратника для подконтрольных ему шлюзов и в качестве обычного шлюза для других сетей со своим привратником.

### Организация абонентских сетей (VPN)

MVTS помогает разработать собственный план нумерации и таблицу маршрутизации для каждого корпоративного клиента. В системе реализованы алгоритмы преобразования номеров набора с использованием рекуррентной трансляции и regex-шаблонов.

Маршрутизация может осуществляться в соответствии с номером вызывающего и вызываемого абонентов, идентификатором группы принадлежности инициатора звонка. В последней версии системы реализована возможность динамической маршрутизации с использованием RADIUS-сервера.

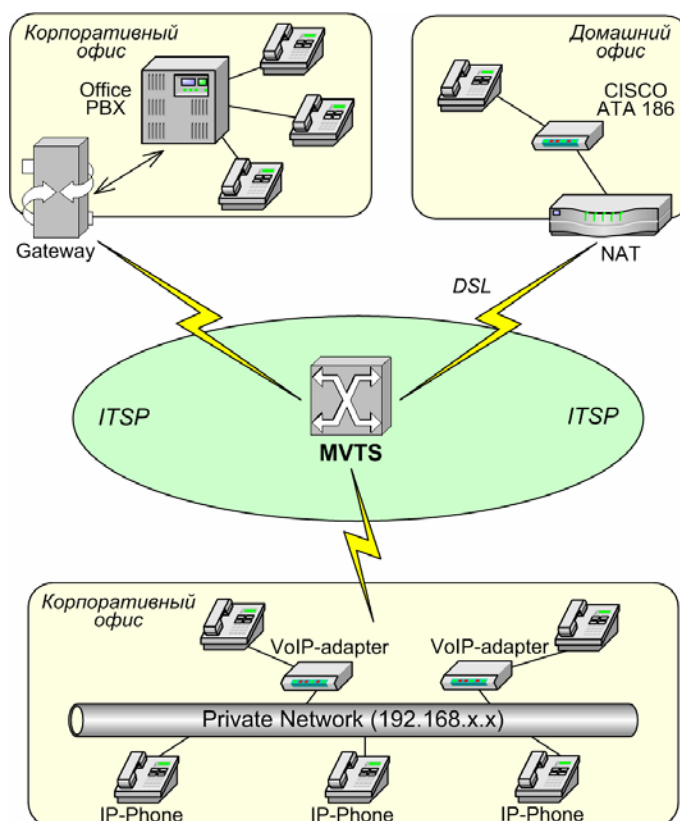


Рис. 2-2-2. Построение частных виртуальных сетей (VPNs)

### Подключение пользователей, скрытых за NAT- маршрутизаторами

MVTS делает возможными входящие звонки на шлюзы, скрытые за NAT-маршрутизаторами (Network Address Translation). Так клиент, имеющий DSL линию, может использовать CISCO ATA 186 как для входящей, так и для исходящей телефонной связи.

### Статическая и динамическая регистрация клиентов

Необходимость регистрировать большое количество VoIP-адаптеров и/или IP-телефонов на нескольких шлюзах, сама по себе, является задачей нетривиальной. Это затрудняет сбор трафика малых клиентов. **MVTS** с единственным выделенным для него IP-адресом становится изящным выходом из такого положения.

**MVTS** предоставляет возможность статической и динамической регистрации абонентов. В первом случае, авторизация исходящих и входящих звонков осуществляется по IP-адресу, во втором – регистрация происходит по протоколу RAS, обеспечивая тем самым мобильность пользователей.

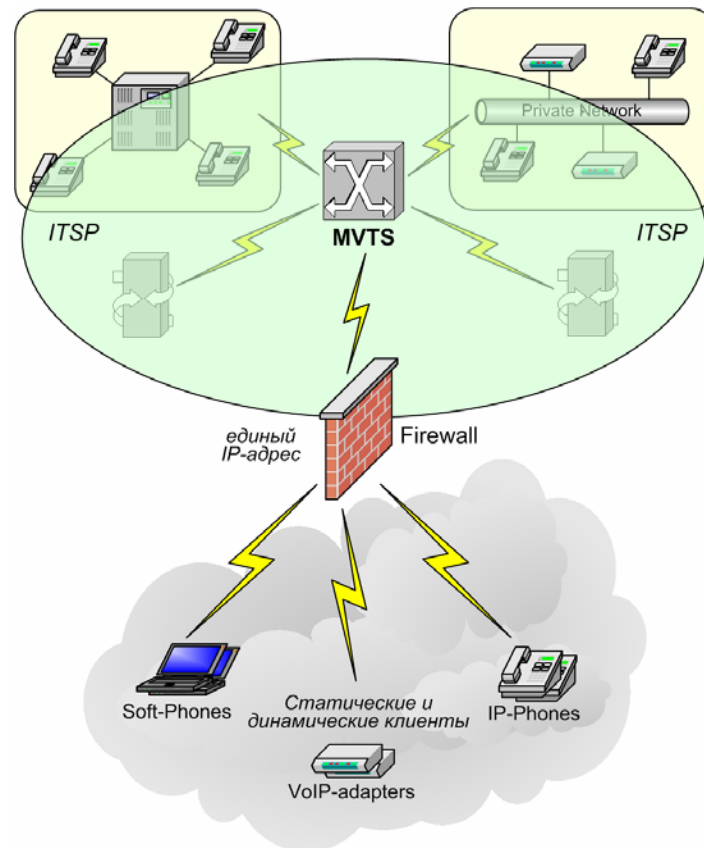


Рис. 2-2-3. Обеспечение безопасности операторской сети

### Соккрытие инфраструктуры сети

**MVTS** позволяет провайдеру реализовывать концепцию "одного окна" как для своих корпоративных клиентов, так и для себя самого по отношению к оператору. Такой подход обеспечивает надежный способ сокрытия реальной структуры сети, защищая коммерческие интересы провайдера и интересы его клиентов.

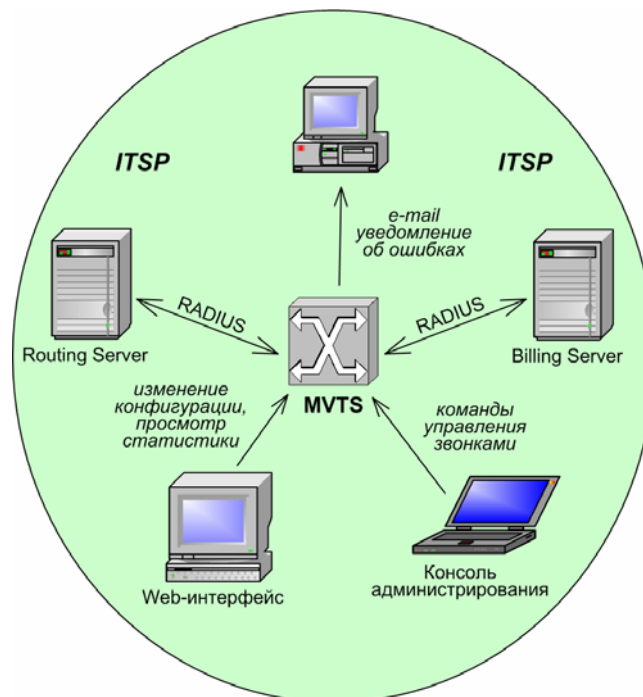
Благодаря представлению всех устройств сети посредством единого IP-адреса, обеспечивается возможность точечного преодоления Firewall и NAT. Наряду с этим, происходит существенное упрощение администрирования и повышение надежности системы в целом.

### Возможность централизованного биллинга и авторизации

Наличие в сети единой точки концентрации трафика делает возможным осуществление авторизации и биллинга с помощью единственной биллинговой системы. MVTS поддерживает протокол RADIUS, что позволяет провайдеру использовать практически любую современную систему тарификации.

MVTS генерирует детальные биллинговые записи для каждого звонка (CDR/call detailed record), которые синхронно сохраняет в структурированных текстовых файлах, удобных для последующего анализа и обработки.

Провайдеру предоставлена возможность выбора prepaid- или postpaid-метода предоставления услуг.



**Рис. 2-2-4. Совместимость с биллинговыми системами.  
Развитая система контроля**

### Мониторинг и администрирование

MVTS спроектирован и построен как надежная и защищенная от сбоев система, в которую входят локальный и удаленный мониторинг. Средства диагностики и контроля (консольная панель, интерфейс удаленного администрирования, сервер оперативной замены и др.) обеспечивают высокую работоспособность системы.

MVTS предоставляет провайдеру удобные и гибкие механизмы настройки и управления: графический интерфейс и текстовые файлы конфигурации.

Отличительной чертой системы является возможность перезапуска сервера без прерывания обслуживания.

## 2.3 ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ

### Преобразование сигнальных и мультимедийных потоков

- двухсторонняя конвертация протоколов сигнализации SIP/H.323 (модуль SIP-НIT)
- конвертация медиа-кодеков G.729, G.729A, G.723.1, G.711A-Law, G.711μ-Law (модуль SIP-НIT)
- поддержка H.323-диалектов (сопряжение оборудования разных производителей)
- сквозная передача (pass-through) протокола T.38

### Проксирование сигнального и мультимедийного трафика

- сигнальный трафик: H.225, H.245
- мультимедийный трафик: RTP/RTCP
- приоритезация IP-пакетов транзитного медиапотока

### Маршрутизация на основании

- номера вызывающего и/или вызываемого абонентов (Cg/Cd E.164, Cg/Cd H.323)
- идентификатора группы принадлежности вызывающего абонента
- загрузки шлюза и/или маршрута прохождения звонка
- приоритета шлюза места назначения
- доступности шлюза места назначения
- дня недели и времени суток
- надежности маршрута прохождения звонка - показателя ASR (интеллектуальная маршрутизация)

### Динамическая маршрутизация с использованием RADIUS-сервера (RADIUS API с внешними системами маршрутизации)

### Балансировка нагрузки в пределах одного направления (распределение нагрузки между шлюзами одной группы)

### Зональная организация и управление по принципу H.323 Gatekeeper

- RAS-регистрация окончного и иного оборудования
- адресная трансляция (преобразование телефонных номеров и



псевдонимов в сетевые адреса)

- контроль доступа к сетевому оборудованию
- опрос и индикация текущего состояния оборудования
- принудительная разрегистрация

#### **Управление доступом к сети на основании**

- данных от системы биллинга (по протоколу RADIUS)
- протоколов аутентификации MD5 и Cisco CHAP
- доступности полосы пропускания
- доступности шлюза
- возможностей (capability) шлюзов
- IP-адреса инициатора
- возможность свободного доступа (без авторизации)

#### **Биллинговые возможности**

- RADIUS-интерфейс (в том числе с использованием Cisco VSA)
- детальные CDR-записи (текстовые файлы Call Detailed Records)

#### **Обеспечение безопасности VoIP-сети**

- преодоление NAT-маршрутизаторов
- сокрытие реальной структуры сети провайдера
- противодействие DoS-атакам

#### **Контроль полосы пропускания**

- установка значения максимального объема входящего и исходящего трафика на каждый локальный IP-адрес
- выбор физического канала передачи трафика для каждого шлюза
- назначение правила выбора физического канала в условиях перегрузки

#### **Преобразование номеров набора**

- преобразование номеров с использованием регулярных выражений (regex-шаблонов)
- рекуррентная трансляция номеров (до десяти уровней вложенности)

- назначение приоритетов объектов набора (dial peer priority)

### **Сбор и отображение статистической информации**

- параметры: ASR, ACD, QoS, максимальная и текущая загрузка, количество успешных и неудачных звонков, число принятых и переданных байт пр.
- объекты статистики: шлюз, диалпир, маршрут
- инструменты отображения: консоль MVTTS, графический интерфейс MVTTS Manager, web-приложение MVTTS Web Monitor

### **Возможности управления звонками**

- вывод списка текущих звонков по заданным критериям
- вывод детальной информации о звонке
- принудительное завершение группы звонков по заданным критериям

### **Конфигурирование и администрирование**

- графический интерфейс пользователя MVTTS Manager
- текстовые файлы конфигурации
- системная консоль администрирования
- изменение конфигурации системы без прерывания сервиса

## **2.4 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**

### **Поддерживаемые протоколы**

- H.323 v.2 (в том числе H.245 v.7, H.225 v.4)
- SIP v.2 RFC 2543bis (модуль SIP-HIT)
- RTP/RTCP
- T.38, T.120
- SNMP v.1 (statistics and trap)
- MD5, CHAP
- RADIUS authentication
- RADIUS accounting (Attribute 44 and VSA)

### **Операционные системы**

- Linux RedHat 9.0
- Linux RedHat Enterprise 3.0
- Linux Fedore Core 2
- FreeBSD 4.5, 5.0+

#### **Ведение протокола работы системы**

- трассировка действий по обработке звонков с выбором уровня детализации
- извлечение логов по заданному звонку (log extractor)
- просмотр логов звонка с использованием MVTTS Manager
- автоматизация записи протокола работы: архивирование, управление размером и временем существования файлов (log rotation)

#### **Отказоустойчивость**

- удаленный SNMP-мониторинг посредством MVTTS Manager или стороннего SNMP-клиента
- встроенный 'watch dog' таймер
- уведомление администратора о системных сбоях по email
- дублирование основного MVTTS-сервера (redundancy)
- механизм поддержки альтернативного RADIUS-сервера

#### **Варианты конфигурации**

- стандартные конфигурации системы с количеством одновременных звонков, кратных потоку E1: от 30 до 1500 на один сервер
- работа на одном сервере/кластерное решение
- установка резервной копии (backup)

## **2.5 ПРОИЗВОДИТЕЛЬНОСТЬ**

MERA VoIP Transit Softswitch сочетает в себе множество функций, каждая из которых имеет свою, индивидуальную с точки зрения производительности, характеристику. В целом, производительность MVTTS существенно зависит от конфигурации аппаратной платформы, на которой установлено программное обеспечение.

Наиболее требовательной к ресурсам аппаратной части системы с точки зрения производительности является функция проксирования трафика (особенно речевого трафика). В разделе 3.1.1 приведены конкретные

рекомендации по выбору конфигурации аппаратной платформы для установки MVTС. Кроме того, производительность MVTС при проксировании речевого трафика в значительной степени зависит от типа кодека, который выбирается для передачи речевой информации.

**Таблица 2-1 Производительность MVTС**

<b>Показатель</b>	<b>Значение</b>
Скорость нарастания звонков (в секунду)	не менее 5
Количество одновременных разговоров без проксирования речевого трафика	не менее 1000
Количество одновременных разговоров с проксированием речевого трафика (G.711)	не менее 300
Количество одновременных разговоров с проксированием речевого трафика (G.729)	не менее 500

## 3. УСТАНОВКА

### 3.1. ПЛАНИРОВАНИЕ

Установка и успешная эксплуатация MVTS в значительной мере обеспечиваются планомерным подходом к процедуре предварительной подготовки аппаратных и программных средств. Ниже приводится информация, призванная помочь пользователю спланировать действия по установке MVTS и подготовке программ к работе.

#### 3.1.1. АППАРАТНЫЕ СРЕДСТВА

##### 3.1.1.1. Платформа

Аппаратная платформа, на которой планируется эксплуатировать MVTS должна выбираться с учетом требований, предъявляемым к максимальной производительности, а также в зависимости от того какие функции вы планируете использовать.

Рекомендуемая конфигурация аппаратного обеспечения для использования MVTS только в качестве привратника (gatekeeper) (без использования функции проксирования) приведена в таблице 3-1.

**Таблица 3-1 Рекомендуемая конфигурация (только привратник)**

Производительность	Рекомендуемая конфигурация
Низкая	Pentium III 833/256Mb RAM/10Gb SCSI HDD
Средняя	Pentium III 1,4/1024Mb RAM/10Gb SCSI HDD
Высокая	Dual Pentium III 1,4/1024Mb RAM/20Gb SCSI HDD

Если планируется использовать функцию проксирования сигнального трафика (без проксирования речевого трафика), то для выбора минимально-необходимой конфигурации можно руководствоваться данными из таблицы 3-2.

**Таблица 3-2 Рекомендуемая конфигурация (при проксировании только сигнального трафика)**

Пропускная способность	Конфигурация
30 звонков	Pentium III 933GHz/256Mb RAM/20 Gb HDD/100 Mb Ethernet
60 звонков	Pentium III 933GHz/512Mb RAM/20 Gb HDD/100 Mb Ethernet
120 звонков	Pentium III 1.13GHz/512Mb RAM/20 Gb HDD/100 Mb Ethernet

Пропускная способность	Конфигурация
300 звонков	Pentium III 1.2GHz/512Mb RAM/20 Gb HDD/100 Mb Ethernet
600 звонков	Pentium IV 1.4GHz/1024Mb RAM/40 Gb HDD/1 Gb Ethernet
1000 звонков	Pentium IV 1.8GHz/1024Mb RAM/40 Gb HDD/1 Gb Ethernet

При необходимости использовать возможности MVTС по проксированию сигнального и речевого трафика в максимальной степени, в выборе конфигурации аппаратной платформы в зависимости от числа одновременных звонков окажутся полезными данные из таблицы 3-3.

**Таблица 3-3 Рекомендуемая конфигурация (полное проксирование)**

Пропускная способность	Конфигурация
30 звонков	Pentium III 1.13GHz/256Mb RAM/20 Gb HDD/100 Mb Ethernet
60 звонков	Pentium III 1.2GHz/512Mb RAM/20 Gb HDD/100 Mb Ethernet
120 звонков	Pentium IV 1.4/512Mb RAM/20 Gb HDD/100 Mb Ethernet
300 звонков	Pentium IV 1.8/1024Mb RAM/40 Gb HDD/100 Mb Ethernet
600 звонков	Xeon Pentium IV 2.4/2048Mb RAM/40 Gb HDD/1 Gb Ethernet
1000 звонков	Dual Xeon Pentium IV 2.0/2048Mb RAM/40 Gb HDD/1 Gb Ethernet

*Примечание: Для достижения производительности более 300 одновременных соединений с полным проксированием, необходимо использовать сетевую карту 1Gbit Ethernet.*

### 3.1.1.2. Сетевое окружение

При установке MVTС следует помнить, что его производительность в большой степени зависит от возможностей сети передачи данных, к которой он подключен. Особую значимость этот параметр обретает при использовании функции проксирования речевого трафика.

При включении MVTС в сеть передачи данных типа Ethernet, минимальным требованием является скорость передачи данных не менее 100Mb. Кроме того, рекомендуется использование оборудования, обеспечивающего работу в полнодуплексном режиме (full-duplex).

Максимально возможная производительность MVTС при проксировании всего речевого трафика (G.729) в сети передачи данных с пропускной способностью 100Mb в полнодуплексном режиме составляет, примерно, 1100 одновременных звонков. Это требует установки и конфигурирования не менее двух сетевых интерфейсов для

приема и передачи сигнального и речевого трафика H.323 и дополнительного интерфейса для работы с сервером RADIUS.

Если вам требуется большая производительность при проксировании речевого трафика, рассмотрите возможность использования более производительных сетевых интерфейсов (например, Gigabit Ethernet).

### 3.1.2. ОПЕРАЦИОННАЯ СИСТЕМА

Программное обеспечение MVTС версии 1.1 рассчитано для работы под управлением операционных систем Linux Red Hat 6.1 (и выше), FreeBSD 4.3 (и выше).

#### 3.1.2.1. Планирование файловых систем

Для обеспечения устойчивой работы MVTС файловые системы должны быть организованы определенным образом.

В зависимости от вариантов использования MVTС, перед установкой операционной системы администратору рекомендуется зарезервировать дисковое пространство под создание нескольких файловых систем<sup>1</sup>:

- Для операционной системы и программного обеспечения MVTС рекомендуется создать файловую систему размером не менее 2Гб.
- Место под свопинг должно резервироваться исходя из объема установленной оперативной памяти в соответствии с рекомендациями руководства по установке операционной системы [9]. Однако не рекомендуется создавать раздел для свопинга размером менее 1Гб.
- В случае, когда начисление платы организовано с использованием CDR-файлов (RADIUS не используется), рекомендуется создать отдельную файловую систему размером не менее 5Гб с точкой монтирования в каталоге **billing**/. Это позволит избежать переполнения корневой файловой системы, где установлено программное обеспечение MVTС.
- Дополнительно рекомендуется создать отдельную файловую систему с точкой монтирования в каталоге **debug**/. В этом каталоге хранятся файлы с журналом работы системы и core-файлы. В случае, если MVTС функционирует в тестовом режиме, размер файлов с журналом работы может оказаться достаточно большим. В этом случае, создание отдельной файловой системы позволит защитить корневую файловую систему от переполнения.

---

<sup>1</sup> Предполагается, что программное обеспечение MVTС устанавливается на ту же файловую систему, куда установлена операционная система, т.е. в корневую файловую систему.

### 3.1.2.2. Конфигурация ядра ОС

MVTS очень активно использует ресурсы операционной системы и особенно файловые дескрипторы. На каждый открытый файл, сокет или именованный программный канал (fifo) требуется как минимум один дескриптор. Однако, стандартная конфигурация ядра операционной системы позволяет пользователю (процессу) использовать ограниченное количество файловых дескрипторов.

Поэтому, если вы планируете использовать MVTS для предоставления крупномасштабного сервиса, необходимо изменить такой параметр конфигурации ядра операционной системы, как количество одновременно используемых файловых дескрипторов. Подробная информация о конфигурации ядра операционной системы приведена в разделе “Конфигурирование ядра операционной системы”

Для расчета количества одновременно используемых дескрипторов, исходите из того, что на каждый звонок при проксировании всего трафика требуется от 6 до 20 сокетов (дескрипторов). Дополнительно необходимо зарезервировать некоторое количество файловых дескрипторов для нужд операционной системы. С учетом этих рекомендаций, для проксирования речевого трафика трехсот звонков, следует установить число одновременно используемых файловых дескрипторов из расчета 20 дескрипторов на один обслуживаемый звонок.

### 3.1.2.3. Пользователи и группы пользователей

При планировании процесса установки ОС следует учитывать необходимость создания учетных записей пользователей (accounts), которые для облегчения задач администрирования объединяются в группы пользователей с соответствующими правами.

В MVTS предусмотрены три группы пользователей, каждая из которых предполагает выполнение специфических задач (см. таблицу 3-4)

**Таблица 3-4 Группы пользователей MVTS**

Группа	Выполняемые задачи
<b>Admin</b>	Группа администраторов. Пользователи этой группы имеют доступ к файлам всех каталогов, имеют право принудительно завершить звонок с заданным идентификатором или группу звонков с участием одного шлюза, могут выполнять любые команды консоли администрирования
<b>Billing</b>	Группа учета и начисления платы. Пользователи этой группы имеют доступ к файлам статистики начисления платы из каталога <b>billing/</b> . Из команд консоли администрирования пользователям этой группы доступны только команды, позволяющие просмотр



Группа	Выполняемые задачи
<b>Support</b>	<p>информации о текущих звонках (<code>show call</code>), и просмотр успешно загруженных объектов набора (<code>show dp</code>)</p> <p>Группа технической поддержки. Пользователи этой группы имеют право на выполнение команд просмотра статистики (группа команд <code>show: show call, show dial, show stat</code> и т.д. консоли администрирования). Пользователям этой группы запрещены команды, влияющие на работу MVTС (<code>start, stop</code> и некоторые другие).</p>

Необходимые группы пользователей с соответствующими правами создаются в процессе исполнения скрипта начальной установки MVTС `setup.sh`.

Детальное описание процесса создания учетных записей пользователей и групп пользователей имеется в документации по ОС [9].

### 3.1.3. КОНФИГУРАЦИЯ СЕТИ

В случае, если MVTС устанавливается во внутренней защищенной сети, то необходимо задать соответствующие правила для сервиса брандмауэра (`firewall`), для того, чтобы MVTС имел возможность инициировать и принимать звонки извне.

Для нужд RAS трафика, для входящих звонков, а также для обмена сообщениями с RADIUS сервером в системе задействованы стандартные порты, значения которых и указаны в качестве значений по умолчанию в конфигурационных файлах.

При настройке сервиса брандмауэра следует иметь в виду, что в случае проксирования речевого трафика, порты, по которым происходит передача голосовых данных, назначаются динамически (в диапазоне от 1024 до 65535).

## 3.2. УСТАНОВКА АППАРАТНОГО ОБЕСПЕЧЕНИЯ

Установка аппаратных средств осуществляется пользователем по собственному усмотрению в соответствии с запланированным в предыдущей главе и с требованиями, предъявляемыми программным обеспечением к аппаратной части системы (см. 3.1.1)

## 3.3. УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### 3.3.1. УСТАНОВКА ОПЕРАЦИОННОЙ СИСТЕМЫ

Операционные системы Red Hat и Free BSD распространяются (в том числе и бесплатно) на условиях открытого исходного кода и отличаются

хорошо развитой поддержкой разработчиков.

Вся информация по ядру операционной системы, ее отдельным компонентам и вопросам ее установки доступна пользователям в обширной документации по операционной системе, имеющейся как в комплекте поставки ОС, так и на сайтах разработчиков [9].

Установка операционной системы осуществляется в соответствии с процедурой, описанной в руководстве по операционной системе, и соответствующей справочной документацией.

### 3.3.2. УСТАНОВКА MERA VOIP TRANSIT SOFTSWITCH

#### 3.3.2.1. Состав поставляемого пакета

Программное обеспечение MVTС поставляется в виде файла, являющегося файлом архива с окончанием **tar.gz** в имени файла. В имени архивного файла указывается, какая версия программного обеспечения находится внутри и для какой версии операционной системы предназначено поставляемое программное обеспечение. Например, архив с именем *MVTС-1.1-linux-x86-7.2.tar.gz* содержит программное обеспечение MVTС версии 2.1, скомпилированное для операционной системы RedHat Linux версии 7.2 (для процессоров семейства x86).

В состав поставляемого пакета программного обеспечения MVTС входят:

- Скрипт начальной установки **setup.sh**, который производит настройку окружения для корректной работы MVTС, т.е. создает необходимые группы пользователей, настраивает права доступа к файлам, добавляет ядро MVTС в число программ запускаемых автоматически при старте операционной системы.
- Исполняемый файл ядра MVTС и скрипт для автоматического перезапуска ядра в случае аварийного завершения (**mp\_kernel.x** и **mp\_kernel.sh**).
- Исполняемый файл консоли администрирования и скрипт для запуска консоли администрирования (**mp\_shell.x** и **mp\_shell.sh**).
- Набор шаблонов конфигурационных файлов (**meraproxy.cfg**, **gateway.cfg**, **dialpeer.cfg**, **gatekeeper.cfg**, **user.cfg**). Поставляемые шаблоны конфигурационных файлов могут использоваться в качестве примеров при начальной настройке MVTС.

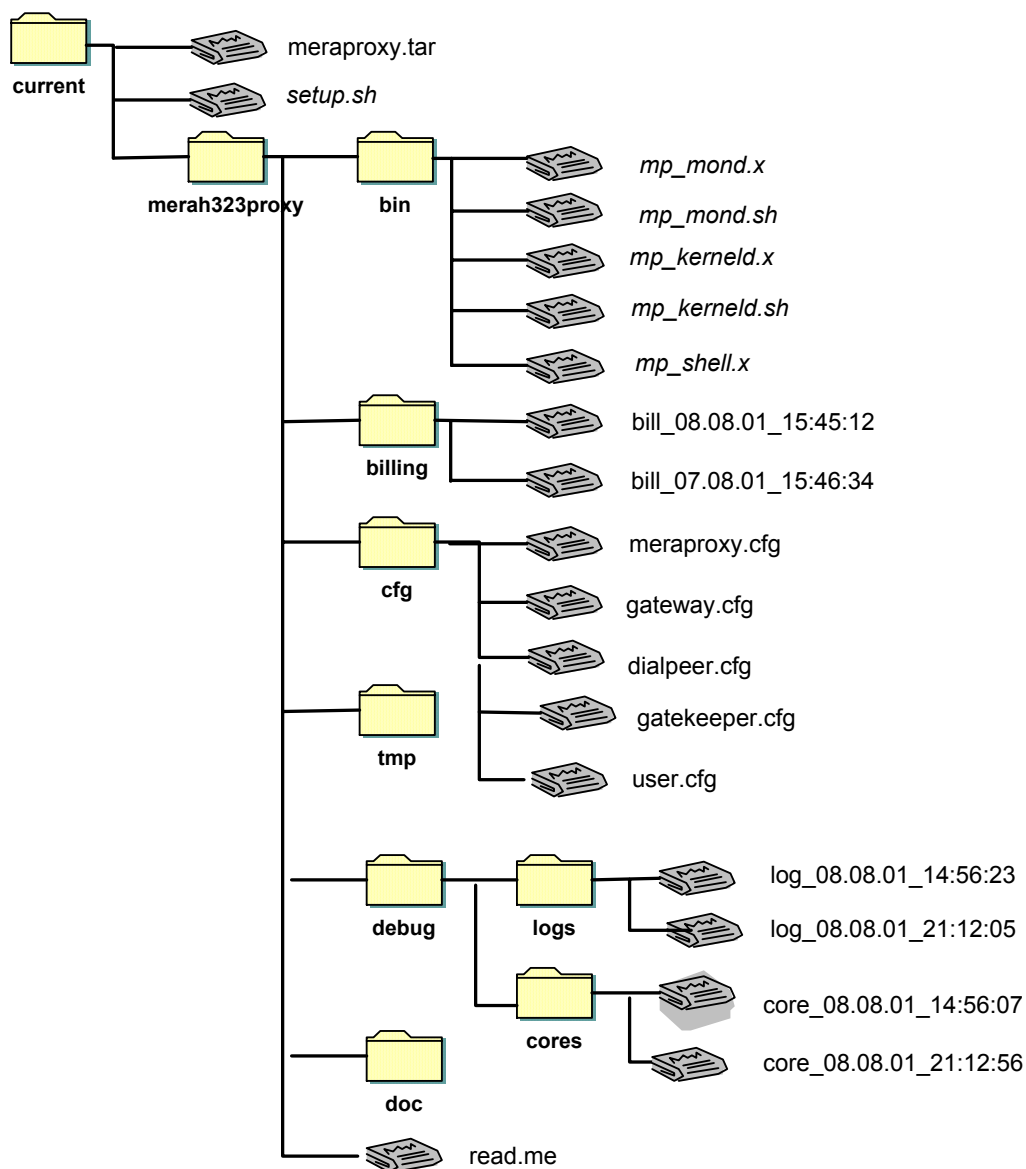
Кроме вышперечисленных компонентов в состав пакета программного обеспечения MVTС дополнительно могут входить:

- Комплект драйверов HASP-ключей для операционной системы Linux Red Hat в виде rpm-файлов
- Различная документация в электронном формате (включая данное руководство).

- Часть исходных кодов MVTС, которые являются открытыми и публикуются в соответствии с условиями *Mozilla Public License (MPL)* [11].

### 3.3.2.2. Структура каталогов пакета

При установке программного обеспечения MVTС, будет создана



структура каталогов аналогичная представленной на рисунке 3-1.

**Рис. 3-3-1. Структура файловой системы**

В каталоге **bin/** находятся все исполняемые файлы и скрипты поставляемые в составе программного пакета за исключением скрипта начальной установки **setup.sh**, который находится в корневом каталоге.

Конфигурационные файлы MVTС по умолчанию располагаются в каталоге **cfg/**. После установки MVTС в этом каталоге находятся шаблоны конфигурационных файлов (**meraproxy.cfg**, **gateway.cfg**, **dialpeer.cfg**, **user.cfg** и **gatekeeper.cfg**).

При сборе информации для начисления платы (биллинг) с помощью CDR-файлов, последние помещаются в каталог **billing/**. В зависимости от своих настроек, MVTС периодически завершает запись текущего файла и создает новый файл. Имена CDR-файлов имеют вид *bill\_<дата>\_<время>*, где *<дата>* - дата создания файла в формате: ууууммдд, *<время>* - время создания файла в формате: hhmmss, а префикс *bill* используется по умолчанию и может быть изменен администратором. Пример имени CDR файла: *bill20020327\_113000*. Файл, в который в текущий момент производится запись, называется по-другому.

Файлы с протоколом работы MVTС создаются в каталоге **debug/logs/**. Смена файла протокола работы, в который ведется запись, осуществляется аналогично CDR-файлам. Имена файлов протокола работы имеют вид *logs\_<дата>\_<время>*, где *<дата>* - дата создания файла в формате: ууууммдд, *<время>* - время создания файла в формате: hhmmss, а префикс *logs* используется по умолчанию и может быть изменен администратором. Пример имени файла протокола работы: *log20020327\_120000*. В таком формате имя присваивается файлу при смене текущего протокола работы. Файл, в который ведется запись, имеет другое имя.

В случае аварийного завершения работы ядра MVTС могут быть сгенерированы **core**-файлы, которые копируются скриптом **mp\_kernel.sh** в каталог **debug/cores/**. **Core**-файлы снабжаются пометкой времени аварийного завершения и могут быть использованы администратором или службой технической поддержки для анализа причин сбоев в работе MVTС.

Каталог **tmp/** содержит внутренние файлы, используемые MVTС. Они не должны изменяться или удаляться администратором во время работы MVTС.

В каталоге **doc/** хранится различная документация, поставляемая в электронном виде.

### 3.3.2.3. Процедура установки MVTС и драйверов HASP

Для установки программного обеспечения MVTС выполните следующие действия:

1. Войдите в систему как пользователь **root**.
2. Перейдите в каталог **/usr/local**
3. Распакуйте архив с программным обеспечением MVTС, используя следующую команду:

---

```
tar xvfz <путь_до_архива>/MVTS-1.1-linux-x86-7.2.tar.gz
```

4. Перейдите в каталог `.MVTS`
5. Запустите скрипт начальной установки **setup.sh** и следуйте инструкциям на экране. После того, как установка успешно завершена на консоль будет выведено следующее сообщение:

```
Setup finished successfully
```

6. Для установки драйверов HASP необходимо перейти в каталог, где находится файл `aksusbd-redhat-*.i386.rpm` и набрать команду:

```
> rpm -i aksusbd-redhat-*.i386.rpm
```

Авторские права на документацию по системе защиты HASP принадлежат компании **Aladdin Knowledge Systems**, и потому информация, подробно описывающая особенности драйверов и демона обращения к USB, здесь не воспроизводится.

Программа готова для начального конфигурирования.

## 4. НАСТРОЙКА

### 4.1. ОБЗОР КОНФИГУРАЦИОННЫХ ФАЙЛОВ

Существует 5 конфигурационных файлов, в которых хранятся данные, необходимые для настройки функций и работы MVTС.

**Таблица 4-1: Конфигурационные файлы MVTС**

Файл	Описание
<b>meraproxy.cfg</b>	Системный конфигурационный файл данных, содержащий настройки функций администрирования, протокола работы, консоли администрирования и начисления платы.
<b>gateway.cfg</b>	Конфигурационный файл данных, содержащий информацию о шлюзах и их настройках
<b>dialpeer.cfg</b>	Конфигурационный файл данных – план набора, содержащий идентификаторы объектов набора, данные о требованиях к вызывающим/вызываемым номерам, внутренние имена шлюзов, правила преобразования номеров
<b>gatekeeper.cfg</b>	Конфигурационный файл, содержащий информацию о привратниках (gatekeepers), на которых MVTС зарегистрирован в качестве клиента..
<b>user.cfg</b>	Конфигурационный файл данных RAS-пользователей

Во время загрузки MVTС считывает все конфигурационные файлы в кэш, и при обработке звонков вся информация берется только из кэша. Таким образом, изменения, вносимые в любой из конфигурационных файлов, будут восприняты MVTС и вступят в силу только при перезапуске программы, происходящем при аварийном или принудительном завершении работы или при вызове команды `reload config` из консоли администрирования (глава 6.3.4.2).

**Примечание:** При редактировании какого-либо из конфигурационных файлов рекомендуется сначала скопировать файл во временный, отредактировать его, а затем скопировать обратно.

## 4.2. НАСТРОЙКА ОСНОВНЫХ ПАРАМЕТРОВ

После установки MVTС файл конфигурации содержит настройки по умолчанию, а в файлах данных шлюзов и настроек плана набора находятся соответствующие шаблоны.

Перед первым запуском нужно:

1. Сконфигурировать настройки в системном файле конфигурации **meraproxy.cfg** (см. главу 6.1.3).
2. Добавить шлюзы в файл данных шлюзов **gateway.cfg** (см. главу 5.3).
3. Настроить план набора **dialpeer.cfg** (см. главу 5.5).
4. При необходимости внести изменения в файл данных RAS пользователей и файл данных привратников (**user.cfg**, **gatekeeper.cfg**).

## 4.3. НАСТРОЙКА ФУНКЦИИ ПРИВРАТНИКА (GATEKEEPER)

Для настройки MVTС на выполнение функций привратника (gatekeeper) нужно соответствующим образом задать параметры в секции [Gatekeeper] системного конфигурационного файла **meraproxy.cfg**. Описание полей секции и форматов принимаемых ими значений см. в разделе 6.1.3, посвященном описанию секции [Gatekeeper].

Для задеирования функции привратника общий порядок настройки следующий:

- Установить значение параметра `gatekeeper_enable` в 1 для включения функции привратника
- В поле `gatekeeper_id` указать идентификатор привратника. Например:

```
gatekeeper_id=OwnMVTС
```

## 4.4. НАСТРОЙКА H.323 PROXY

Для настройки функций прокси-сервера необходимо выполнить следующие действия:

В секции [H323] системного конфигурационного файла **meraproxy.cfg** указать необходимые параметры:

- Указать номер порта для установления входящих звонков, если он отличается от значения по умолчанию (1720)
- в поле `route_cause` в секции [LAR] конфигурационного файла **meraproxy.cfg** задать маской десятичного числа причину завершения звонка (Q931 cause), требующую продолжение попыток его маршрутизации.

Подробное описание полей и форматы их значений см. в разделе 6.1.3

## 4.5. НАСТРОЙКА СИСТЕМЫ НАЧИСЛЕНИЯ ПЛАТЫ

Процедура настройки системы начисления платы зависит от того, будет ли статистика начисленной платы вестись с применением CDR-файлов или предполагается пользоваться RADIUS-сервисом учета платы.

### 4.5.1. CDR-ФАЙЛЫ

Настройка системы для работы с CDR файлами заключается в указании необходимых для задействования системы начисления платы параметров:

- в системном конфигурационном файле **meraproxy.cfg** в поле `period` секции `[billing]` указать максимальное время записи в один файл статистики начисления платы (см. раздел 6.1.4)
- в поле `file` секции `[billing]` указать путь и имя файла, в который будет осуществляться запись статистики
- в поле `level` поставить значение уровня детализации записи информации

Подробные сведения о формате значений, которыми заполняются поля секции см. в разделе 6.1.3

Прочие административные функции в отношении CDR-файлов (архивирование или удаление устаревших файлов) осуществляются с помощью скрипта администрирования файлов статистики **rotate.sh**, работу которого можно настроить через редактирование конфигурационного файла **rotate.cfg**.

### 4.5.2. RADIUS СЕРВИС УЧЕТА ПЛАТЫ (RADIUS ACCOUNTING)

Настройка работы MVTС для учета платы через сервис RADIUS осуществляется заполнением соответствующих полей (адрес RADIUS сервера, поле задействования системы начисления платы и т.д.) в секции `[Radius]` системного конфигурационного файла **meraproxy.cfg**.

Минимальная настройка для работы MVTС с RADIUS-сервисом учета начисляемой платы заключается в следующем:

- Задействовать режим использования RADIUS-поддержки начисления платы, установив значение 1 в поле `acct_enable`
- Указать адрес RADIUS-сервера в поле `acct_address`
- При необходимости указать номер порта для RADIUS-сервера в поле `acct_port` (по умолчанию 1813)
- Задать номер порта для приема ответных сообщений от RADIUS-сервера в поле `local_acct_port`

Подробные сведения о формате значений, которыми заполняются поля секции см. в разделе 6.1.3



## 4.6. НАСТРОЙКА ПАРАМЕТРОВ АВТОРИЗАЦИИ

Доступ RAS-пользователей и шлюзов к MVTС может осуществляться двумя способами: в режиме свободного (общего) доступа (`public_access`) и в режиме авторизации по IP- адресу.

### 4.6.1. ОБЩИЙ ДОСТУП

Общий доступ означает отключение режима авторизации шлюзов по IP адресу (чаще всего применяется для целей отладки). Обеспечение режима общего доступа (`public_access`) достигается установкой значения 1 для соответствующего параметра в секции [Administration] системного конфигурационного файла **meraproxy.cfg** (см. `public_access`)

### 4.6.2. АВТОРИЗАЦИЯ ПО IP АДРЕСУ

Если режим общего доступа отключен (т.е. задействован режим авторизации шлюзов по IP адресу), необходимо убедиться в том, что запись о соответствующем шлюзе имеется в файле данных шлюзов.

### 4.6.3. RADIUS (СЛУЖБА УДАЛЕННОЙ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ)

При необходимости задействовать поддержку службы удаленной аутентификации пользователей RADIUS, настройка системы заключается в заполнении следующих полей в секции RADIUS, системного конфигурационного файла **meraproxy.cfg**:

- `local_address` (адрес, на который осуществляется привязка (bind) при работе с RADIUS).
- `auth_enable` поле задействования аутентификации через RADIUS.
- `secret` ключ шифрования при работе с RADIUS сервером
- `auth_address` адрес RADIUS сервера
- `auth_port` номер порта для сервера удаленной аутентификации
- `local_auth_port` порт для приема ответных сообщений сервера удаленной аутентификации

Подробное описание полей секции RADIUS, принимаемых ими значений и форматов см. в разделе [Radius] описания системного конфигурационного файла (**meraproxy.cfg**).

## 4.7. НАСТРОЙКА ПРАВ ГРУПП ПОЛЬЗОВАТЕЛЕЙ

Настройка прав групп пользователей осуществляется заполнением полей `admin_gid`, `billing_gid` и `support_gid` в секции [Console] системного конфигурационного файла **meraproxy.cfg** значениями, соответствующими номерам групп пользователей,

созданных после установки ОС.

При этом следует учитывать, что запущенный на исполнение скрипт установки программного обеспечения **setup.sh** создает необходимые группы пользователей, (если они не были созданы ранее) и задает соответствующие права, заполняя поля в секции [Console] системного конфигурационного файла соответствующими значениями.

#### 4.8. ПОДГОТОВКА ПЛАНА НАБОРА

План набора, в виде данных об объектах набора (dial peers), хранится в текстовом файле **dialpeer.cfg** или в заданном администратором с помощью системного конфигурационного файла **meraproxy.cfg**.

Настройка плана набора заключается в указании необходимых параметров объектов набора (dial peers) в секциях файла **dialpeer.cfg**.

Подробно о действиях с файлом данных плана набора см. в разделе 5.5

#### 4.9. ПОДГОТОВКА ФАЙЛА ДАННЫХ ШЛЮЗОВ

Файл данных шлюзов представляет собой текстовый файл **gateway.cfg** или назначенный администратором с помощью системного конфигурационного файла **meraproxy.cfg**.

Настройка MVTС на работу с шлюзами, через которые устанавливаются звонки, заключается в указании необходимых параметров в файле данных шлюзов:

- Задать внутреннее имя шлюза в качестве названия секции
- Указать статический IP-адрес шлюза
- Указать максимальное количество звонков одновременно поддерживаемых MVTС при работе с шлюзом
- При необходимости – указать множество поддерживаемых/неподдерживаемых шлюзом кодеков
- Определить приоритет IP-пакетов, проходящих через шлюз
- Указать телефонный номер, присвоенный шлюзу (для однопортовых шлюзов)
- При необходимости переопределить порт для входящих звонков (по умолчанию 1720)
- Если звонки через шлюз обрабатываются привратником, указать имя привратника (gatekeeper)
- Установить флаг режима проксирования трафика (0 – проксирование только сигнального трафика, 1 – проксирование сигнального и медиа (речевого, мультимедийного) трафика.

Подробно описание способов работы с файлом данных шлюзов и

форматов параметров, вносимых в поля секции, см. в разделе 5.3

#### 4.10. ПОДГОТОВКА ФАЙЛА ДАННЫХ RAS-ПОЛЬЗОВАТЕЛЕЙ

Для создания объема данных о зарегистрированных пользователях, необходимо добавить записи в файл, имеющий название, указанное в поле `user_cfg` системного конфигурационного файла **meraproxy.cfg** (по умолчанию **user.cfg**).

Для того, чтобы действия с файлом данных RAS-пользователей имели смысл, должна быть задействована функция привратника, для чего необходимо установить значение 1 в поле `gatekeeper_enable` секции `[Gatekeeper]` в системном конфигурационном файле (**meraproxy.cfg**).

Файл данных пользователей, регистрируемых по протоколу RAS, представляет собой текстовый файл **user.cfg** (или иной, заданный администратором с помощью системного конфигурационного файла **meraproxy.cfg**).

Порядок добавления записей о пользователях в файл, удаление записей из файла подробно рассматривается в разделах 5.2.3, 5.2.4. Детальная информация о полях записи, их значениях и формате данных имеется в разделе 6.1.4

#### 4.11. ПОДГОТОВКА ФАЙЛА ДАННЫХ ПРИВРАТНИКОВ (GATEKEEPERS)

Данные о привратниках, с которыми MVTs осуществляет обмен, хранятся в файле с именем **gatekeeper.cfg** (или с любым другим именем по выбору администратора, указанном в соответствующем поле системного конфигурационного файла **meraproxy.cfg**)

Чтобы сконфигурировать MVTs для работы с другими привратниками, отредактируйте файл данных привратников, пользуясь подробным его описанием, приведенным в разделе 5.4.

Необходимо учитывать, что внесение изменений в файл данных привратников может потребовать соответствующего редактирования файла данных шлюзов.

#### 4.12. НАСТРОЙКА СРЕДСТВ ОБЕСПЕЧЕНИЯ УСТОЙЧИВОСТИ

К настраиваемым параметрам средств обеспечения устойчивости относятся: приложение удаленного администрирования и контроля системы (MVTs Manager), адрес электронной почты администратора для email сообщений MVTs, маски файлов протокола работы системы и период записи протокола в один файл, определяемые в системном конфигурационном файле (**meraproxy.cfg**). При наличии программного решения, дублирующего работу основного экземпляра программы, необходимо также сконфигурировать основную и резервную системы для работы в паре.

#### 4.12.1. УВЕДОМЛЕНИЯ ПО ЭЛЕКТРОННОЙ ПОЧТЕ

MVTS имеет возможность направлять сообщения о работоспособности системы на адрес электронной почты администратора. Адрес электронной почты для сообщений MVTS указывается в системном конфигурационном файле в поле `mail_alert` (см. гл. 6 “Структура системного конфигурационного файла”)

Программный модуль, контролирующей работоспособность MVTS, отправляет сообщения о проблемах в работе системы на адрес электронной почты администратора, задаваемый в файле общих настроек `meraproxy.cfg`

#### 4.12.2. РЕЗЕРВИРОВАНИЕ СИСТЕМЫ, НАХОДЯЩЕЙСЯ ПОД КОММЕРЧЕСКОЙ НАГРУЗКОЙ

Резервирование системы, работающей с коммерческим трафиком, осуществляется за счет установки дополнительного (резервного) аппаратно-программного комплекса с запущенным на нем экземпляром MVTS, защищенным HASP ключом, который идентифицирует работающее приложение в качестве резервной системы. Продолжительность функционирования резервного комплекса под коммерческой нагрузкой ограничена 24 часами.

В такой паре программно-аппаратный комплекс «горячего резерва» конфигурируется как RAS-пользователь регистрирующийся на привратнике (`gatekeeper`), в качестве которого для него выступает система, несущая основную рабочую нагрузку.

Конфигурация резервной системы должна быть идентична конфигурации системы, находящейся под коммерческой нагрузкой, за исключением конфигурационных параметров, обеспечивающих связь основной и резервной системы как RAS пользователя и привратника (`gatekeeper`). Для обеспечения такой связи оба комплекса конфигурируются следующим образом.

В конфигурационный файл `user.cfg` главной рабочей системы заносятся параметры RAS-пользователя (резервного комплекса), а в конфигурационных файлах резервной машины «оформляются» параметры привратника (основного комплекса). Пример конфигурации связи «RAS пользователь – привратник» между основной и резервной системами:

**Конфигурационные файлы  
основного комплекса**

`user.cfg`

**Конфигурационные файлы резервной  
системы**

`gatekeeper.cfg`

**Конфигурационные файлы  
основного комплекса**

```

user.cfg

[Standby_PC]
... ..
... ..
user=user_name
password= MyPass8
... ..
... ..
... ..
... ..

```

**Конфигурационные файлы резервной  
системы**

```

gatekeeper.cfg

[Main_MERA]
type=1
user=user_name
password=MyPass8

meraproxy.cfg

[Administration]
... ..
... ..
master_gatekeeper=Main_MERA
... ..

```

В то время, когда резервный комплекс работает в режиме дублирования, вызовы через него не маршрутизируются и регистрации RAS-пользователей отклоняются. Резервная версия MVTС в режиме ожидания на запрос GRQ отвечает пакетом GRJ с причиной 'resource unavailable' ("ресурс недоступен").

В случае выхода из строя основного комплекса (выражающемся в «обрыве» регистрации на основной системе-привратнике), резервный экземпляр программы переходит в рабочий режим, начиная маршрутизировать звонки и регистрировать RAS-пользователей. Одновременно резервная MVTС система посылает соответствующее уведомление на email адрес, указанный в поле `mail_alert` секции [Administration] основного конфигурационного файла **meraproxy.cfg**.

С данного момента оператор имеет в своем распоряжении 24 часа на то, чтобы ввести в строй основную систему (устранение причин неисправности, замена аппаратного обеспечения и т.д.). При невозможности разрешить аварийную ситуацию с основным комплексом в течение 24 часов, можно перезапустить резервную машину с HASP ключом от основной, что обеспечит неограниченный запас времени на преодоление возникших трудностей.

После восстановления работоспособности основного комплекса, его можно включить в пару дублирования, в качестве резервного, запустив его с HASP ключом от дублирующего экземпляра программы (что должно быть предусмотрено наличием соответствующих записей о RAS пользователе и привратнике в конфигурационных файлах), либо по усмотрению оператора восстановить систему в первоначальном виде, существовавшем до возникновения аварийной ситуации.

При восстановлении регистрации на основной системе резервная система сбрасывает все активные регистрации RAS-пользователей, перестает принимать новые звонки и отклоняет все последующие запросы на регистрацию, но все активные звонки на момент перехода в режим дублирования принудительно не завершаются.

## 5. АДМИНИСТРИРОВАНИЕ

### 5.1. ЗАПУСК И ОСТАНОВ MVTС

В каждый момент времени может быть запущен только один экземпляр MVTС.

Команды, осуществляющие запуск или остановку MVTС, доступны только пользователям группы администраторов.

Администратор может запустить, остановить или перезапустить MVTС с помощью консоли администрирования.

#### 5.1.1. Запуск MVTС

Запуск MVTС осуществляется командой `start` командной строки консоли администрирования.

#### 5.1.2. Останов MVTС

Останов MVTС осуществляется командой `stop` командной строки консоли администрирования. После вызова команды `stop` MVTС заканчивает текущие звонки, отсоединяет консоль администрирования и прекращает работу. Для того чтобы возобновить работу после вызова данной команды, нужно запустить MVTС заново.

### 5.2. УПРАВЛЕНИЕ ФАЙЛОМ RAS-ПОЛЬЗОВАТЕЛЕЙ

#### 5.2.1. Как MVTС ИСПОЛЬЗУЕТ ФАЙЛ ДАННЫХ RAS-ПОЛЬЗОВАТЕЛЕЙ

В отличие от “статически” зарегистрированных объектов, то есть тех, записи о которых имеются в файле данных шлюзов `gateways.cfg`, IP-адрес пользовательских объектов, регистрируемых “динамически”, заранее неизвестен, поэтому такие пользователи должны периодически регистрироваться на привратнике, входящим в состав MVTС, для того, чтобы MVTС знал, что они работают, и имел информацию о том, куда необходимо отправлять звонок с их номером, а также имел возможность принимать исходящие от таких пользователей звонки. Такие объекты-пользователи регистрируются на привратнике посредством обмена с ним RAS сообщениями.

#### 5.2.2. СТРУКТУРА ФАЙЛА ДАННЫХ RAS-ПОЛЬЗОВАТЕЛЕЙ

Файл данных динамически регистрируемых пользователей представляет собой текстовый конфигурационный файл, состоящий из секций по числу пользователей.

Файл данных сообщает MVTС параметры пользователя, содержащиеся в полях каждой записи, описание которых приводится ниже.

Таблица 5-1: Параметры RAS-пользователя

Поле	Назначение
[Имя пользователя]	Имя пользователя используется MVTС только для внутренней идентификации пользователя и связи файла данных пользователя с планом набора. Назначается администратором.
<b>acct_enable</b>	Флаг задеирования RADIUS сервиса по учету начисленной платы.
<b>ani_allow</b>	Список разрешенных <b>ani</b> - номеров для данного входящего шлюза
<b>auth_enable</b>	Флаг задеирования аутентификации через RADIUS сервер
<b>capacity</b>	Максимальное количество звонков, которое MVTС может одновременно поддерживать с данным шлюзом.
<b>codec_allow</b>	Необязательное поле. Отражает множество кодеков, поддерживаемых данным шлюзом. По умолчанию все.
<b>codec_deny</b>	Необязательное поле. Содержит множество кодеков, не поддерживаемых или нестандартно воспринимаемых данным шлюзом. По умолчанию пустое множество.
<b>compatibility</b>	4-х байтная маска характеристик шлюза
<b>datacap_deny</b>	Список отбрасываемых функциональных возможностей оборудования
<b>dst_translate</b>	Правила следующего за первичным преобразования номера вызываемого абонента.
<b>fake_rtcp</b>	Флаг генерации RTCP пакетов. Используется в случае, когда шлюз не поддерживает отсылку RTCP. Необязательное поле. Применяется для совместимости с VocalTec Gateway.
<b>force_alerting</b>	Интервал времени в миллисекундах до отсылки на оригиналирующий шлюз сообщения Alerting Message с нейтральным содержанием
<b>force_call_proceeding</b>	Интервал времени в миллисекундах до отсылки пакета CallProceeding Message на оригиналирующий шлюз.
<b>gateway_type</b>	Тип шлюза (VocalTec, CISCO и т.д.)
<b>group</b>	Необязательное поле. Название группы к



Поле	Назначение
<b>h323_gw_id</b>	которой принадлежит пользователь. Задаёт значение Source H323 Alias для исходящего шлюза.
<b>in_converter</b>	Определяет имя конвертера для входящего трафика от данного RAS пользователя
<b>in_dst_translate</b> <b>in_src_translate</b>	Предварительная трансляция, выполняемая при поступлении звонка от этого шлюза, перед вызовом поиска по объектам набора (dial peers).
<b>ip_precedence</b>	Необязательное поле. Приоритет IP-пакетов, проходящих через шлюз – должен быть установлен при отсылке IP-пакетов к данному шлюзу. По умолчанию 3 (как у операционной системы).
<b>local_address</b>	Поле для задания локальных IP адресов для входящего и исходящего трафика
<b>min_asr</b>	Нижнее значение среднего показателя успешных звонков (ASR) для шлюза, достижение которого вызывает отправку уведомлений Администратору системы
<b>nat_rtp</b>	Флаг направления RTP пакетов. Используется при нахождении шлюза за устройством трансляции адресов (NAT). Необязательное поле.
<b>no_port_clean</b> <b>number</b>	Флаг высвобождения порта Телефонный номер пользователя (номер, по которому вызывается пользователь).
<b>numbering_plan</b>	значение (в десятиричном формате) параметра type:plan для обращения к шлюзу
<b>password</b>	Пароль пользователя. Используется совместно с параметром поля <b>user</b> для тех же целей.
<b>proxy_type</b>	Необязательное поле. Режим проксирования для звонков с участием данного шлюза – один из двух. Значение 0 означает проксирование только сигнального трафика, значение 1 – проксирование всего трафика. Первый режим используется между двумя шлюзами, находящимися в одной сети. Второй – между шлюзами, находящимися в разных сетях. Если у двух шлюзов разные режимы проксирования, используется более

Поле	Назначение
	строгий – проксирование всего трафика. По умолчанию 1.
<b>reg_address</b>	IP адрес для целей регистрации
<b>radius_gw_id</b>	Задаёт RADIUS VSA поля h323-gw-id и h323-remote-id, из исходного и терминирующего шлюза соответственно
<b>radius_uses_callid</b>	Флаг режима идентификации вызова
<b>rbt_timeout</b>	Заданное максимальное время ожидания пакетов Alert и ProgressIndicator от терминирующего шлюза
<b>route_cause</b>	Причины завершения звонка, требующие продолжения попыток маршрутизации
<b>src_translate</b>	Правила следующего за первичным преобразования номера вызывающего абонента.
<b>user</b>	Имя пользователя в системе.Используется для идентификации пользователя при регистрации его на привратнике MVTС и при запросах к системе удаленной аутентификации (Radius).
<b>validation_gap</b>	время задержки отправки на шлюз-инициатор звонка пакетов, полученных от терминирующего шлюза. Используется в описании терминирующих шлюзов.
<b>validation_msg</b>	тип сообщения, прибытие которого возобновляет задержанный процесс установления вызова. Используется в описании терминирующих шлюзов.

Подробная информация о значениях полей файла, и формате данных изложена в разделе 6.1.4

### 5.2.3. ДОБАВЛЕНИЕ РЕГИСТРИРУЕМОГО ПОЛЬЗОВАТЕЛЯ

Для того чтобы добавить пользователя, регистрируемого на MVTС через RAS, нужно:

- Задействовать функцию привратника, установив флаг привратника в секции [Gatekeeper] системного конфигурационного файла (**meraproxy.cfg** или другом, назначенном администратором системы) в значение 1.

Пример: `gatekeeper_enable=1`

- Добавить в файл данных пользователя запись о пользователе (в виде секции) с помощью любого текстового редактора (формат данных полей файла описан в разделе 6.1.4)
- Выполнить команду `reload config` консоли администрирования (глава 6.3.4.2)

#### 5.2.4. УДАЛЕНИЕ ПОЛЬЗОВАТЕЛЯ

Чтобы удалить пользователя:

- Удалите запись о пользователе из файла данных RAS-пользователей.
- Выполните команду `reload config` консоли администрирования (раздел 6.3.4.2)

#### 5.2.5. ИЗМЕНЕНИЕ ПАРАМЕТРОВ

Для изменения параметров динамически регистрируемого пользователя:

- Внесите необходимые изменения в поля файла данных
- Выполните команду `reload config` с консоли администрирования.

### 5.3. УПРАВЛЕНИЕ ФАЙЛОМ ДАННЫХ ШЛЮЗОВ

#### 5.3.1. КАК MVTС ИСПОЛЬЗУЕТ ФАЙЛ ДАННЫХ ШЛЮЗОВ

Если не задействован свободный режим доступа (`public_access=0`), MVTС может установить звонок между двумя клиентами, только если записи о шлюзе-инициаторе вызова и шлюзе, на который направляется звонок, имеются в файле данных шлюзов.

#### 5.3.2. СТРУКТУРА ФАЙЛА ДАННЫХ ШЛЮЗОВ

Файл данных шлюзов позволяет MVTС получить данные о шлюзе по параметрам, сгруппированным в таблице.

**Таблица 5-2: Параметры шлюза**

Поле	Назначение
[Имя шлюза]	Имя шлюза используется MVTС только для внутренней идентификации шлюза и связи файла данных шлюзов с планом набора. Назначается администратором.
<code>acct_enable</code>	Флаг RADIUS сервиса по учету начисленной платы.
<code>address</code>	Статический IP адрес шлюза.
<code>ani_allow</code>	Список разрешенных <b>ani</b> -номеров для данного

Поле	Назначение
	входящего шлюза
<b>auth_enable</b>	Флаг аутентификации через RADIUS сервер
<b>capacity</b>	Максимальное количество звонков, которое MVTС может одновременно поддерживать с данным шлюзом.
<b>codec_allow</b>	Множество кодеков, поддерживаемых данным шлюзом. По умолчанию все.
<b>codec_deny</b>	Множество кодеков, не поддерживаемых или нестандартно воспринимаемых данным шлюзом. По умолчанию пустое множество.
<b>compatibility</b>	4-х байтная маска характеристик шлюза
<b>datacap_deny</b>	Список отбрасываемых функциональных возможностей.
<b>dst_translate</b>	Правила следующего за первичным преобразования номера вызываемого абонента.
<b>fake_rtcp</b>	Флаг генерации RTCP пакетов. Используется в случае, когда шлюз не поддерживает отсылку RTCP. Необязательное поле. Используется для совместимости с VocalTec Gateway.
<b>force_alerting</b>	Интервал времени в миллисекундах до отсылки на оригинирующий шлюз сообщения Alerting Message с нейтральным содержанием
<b>force_call_proceeding</b>	Интервал времени в миллисекундах до отсылки пакета CallProceeding Message на оригинирующий шлюз.
<b>gatekeeper</b>	Имя привратника (gatekeeper). Присутствие данного параметра означает, что все звонки на данный шлюз обрабатываются привратником, и поля <b>address</b> и <b>port</b> должны быть удалены из записи о шлюзе
<b>gateway_mode</b>	Режим шлюза
<b>gateway_type</b>	Тип шлюза. Необязательное поле.
<b>group</b>	Название группы к которой принадлежит шлюз. Определяет возможность шлюза устанавливать исходящие звонки через какой-либо объект набора (dial peer).
<b>h323_gw_id</b>	Задаёт значение Source H323 Alias для исходящего шлюза.
<b>in_dst_translate</b> <b>in_src_translate</b>	Предварительная трансляция, выполняемая при поступлении звонка от шлюза перед поиском объекта набора (dial peer)
<b>ip_precedence</b>	Приоритет IP-пакетов, проходящих через шлюз – должен быть установлен при отсылке IP-пакетов к

Поле	Назначение
	данному шлюзу. По умолчанию 3 (как у операционной системы).
<b>local_address</b>	Поле для задания локальных IP адресов для входящего и исходящего трафика
<b>mask</b>	Маска входящих соединений шлюза.
<b>min_asr</b>	Нижнее значение среднего показателя успешных звонков (ASR) для шлюза, достижение которого вызывает отправку уведомлений Администратору системы
<b>nat_rpt</b>	Флаг направления RTP пакетов. Используется при нахождении шлюза за устройством трансляции адресов (NAT). Необязательное поле.
<b>no_port_clean</b>	Флаг высвобождения порта
<b>number</b>	Прямой номер шлюза для поиска по прямому соответствию и полному совпадению (регулярные выражения запрещены)
<b>numbering_plan</b>	Значение пары type:plan в десятиричном формате для обращения к шлюзу
<b>password</b>	Пароль шлюза
<b>port</b>	Порт, на котором шлюз ожидает входящие звонки. По умолчанию принимается 1720.
<b>proxy_type</b>	Режим проксирования для звонков с участием данного шлюза – один из двух. Значение 0 означает проксирование только сигнального трафика, значение 1 – проксирование всего трафика.
<b>radius_gw_id</b>	Задаёт RADIUS VSA поля h323-gw-id и h323-remote-id, из исходного и терминирующего шлюза соответственно
<b>radius_uses_callid</b>	Флаг режима идентификации вызова
<b>rbt_timeout</b>	Заданное максимальное время ожидания пакетов Alert и ProgressIndicator от терминирующего шлюза
<b>route_cause</b>	Причины завершения звонка, вызывающие продолжение попыток его дальнейшей маршрутизации
<b>src_translate</b>	Правила следующего за первичным преобразования номера вызывающего абонента.
<b>user</b>	Имя пользователя в системе.Используется для идентификации пользователя при регистрации его на привратнике MVTС и при запросах к системе удаленной аутентификации (Radius).

Поле	Назначение
<b>validation_gap</b>	время задержки отправки на шлюз-инициатор звонка пакетов, полученных от терминирующего шлюза. Используется в описании терминирующих шлюзов.
<b>validation_msg</b>	тип сообщения, прибытие которого возобновляет задержанный процесс установления вызова. Используется в описании терминирующих шлюзов.

Множество поддерживаемых кодеков и множество неподдерживаемых кодеков используются MVTS при установлении звонка. При получении запроса от одной из сторон MVTS обрабатывает множество кодеков, передаваемое шлюзом, по некоторому алгоритму. С помощью данного алгоритма из запросов шлюзов исключаются кодеки, содержащиеся в объединенном множестве неподдерживаемых кодеков шлюзов, и остаются только кодеки, поддерживаемые обеими сторонами.

#### 5.3.2.1. ВЫБОР РЕЖИМА ПРОКСИРОВАНИЯ

Система всегда проксирует сигнальный трафик звонка. Режим проксирования речевого трафика между двумя шлюзами вычисляется по наиболее строгому режиму проксирования из определенных для каждого шлюза. Таким образом речевой трафик не проксируется, только если у обоих шлюзов режим проксирования 0. Режим 0 используется между двумя шлюзами, находящимися в одной сети, режим 1 – между шлюзами, находящимися в разных сетях. Если у двух шлюзов разные режимы проксирования, используется более строгий – проксирование всего трафика. По умолчанию 1.

#### 5.3.3. ДОБАВЛЕНИЕ ШЛЮЗА

Для того чтобы добавить шлюз нужно:

- Добавить в файл данных шлюзов запись о шлюзе с помощью любого текстового редактора (структура файла описана в разделе 6.1.4)
- Отредактировать файл плана набора, если требуется (см. раздел 5.4)
- Выполнить команду `reload config` консоли администрирования (раздел 6.3.4.2)

#### 5.3.4. УДАЛЕНИЕ ШЛЮЗА

Чтобы удалить запись о шлюзе:

- Внесите необходимые изменения в файл плана набора (раздел 5.4)

- Удалите запись о шлюзе из файла данных и все ссылки на него из файла плана набора
- Выполнить команду `reload config` консоли администрирования (раздел 6.3.4.2)

### 5.3.5. ИЗМЕНЕНИЕ ПАРАМЕТРОВ

Изменения могут касаться одного или нескольких шлюзов.

Для внесения изменений в параметры шлюза:

- Внесите изменения в файл данных шлюзов
- Внесите соответствующие изменения в файл плана набора **dialpeer.cfg** (глава 5.4)
- Выполните команду `reload config` консоли администрирования (раздел 6.3.4.2)

## 5.4. УПРАВЛЕНИЕ ФАЙЛОМ ДАННЫХ ПРИВРАТНИКОВ (GATEKEEPERS)

### 5.4.1. КАК MVTС ИСПОЛЬЗУЕТ ФАЙЛ ДАННЫХ ПРИВРАТНИКОВ

Секция `[Administration]` в конфигурационном файле **meraproxy.cfg** может содержать следующее необязательное поле, задающее имя файла конфигурации привратников:

```
gatekeeper_cfg
```

Если значение этого поля не указано, используется имя по умолчанию **gatekeeper.cfg**.

Список параметров любого шлюза (в конфигурационном файле шлюзов **gateway.cfg**) также может содержать необязательное поле `gatekeeper`.

Присутствие этого параметра означает, что все звонки на этот шлюз обрабатываются привратником, имя которого определяется значением поля `gatekeeper`. В таком случае параметры шлюза, задаваемые полями `address` и `port` игнорируются.

### 5.4.2. СТРУКТУРА ФАЙЛА ДАННЫХ ПРИВРАТНИКОВ

Таблица 5-3: Параметры привратника

Поле	Назначение
[Имя]	Уникальное имя привратника.
<b>address</b>	IP адрес или DNS имя привратника. Обязательное поле.
<b>id</b>	Идентификатор привратника. Необязательное поле

Поле	Назначение
<b>keepalive</b>	Периодичность перерегистрации на привратнике. Значение по умолчанию 5 секунд.
<b>keepalive_type</b>	Тип сообщения, отсылаемого на привратник при перерегистрации
<b>local_address</b>	Реальный IP адрес, с которого будет осуществляться отправка RAS пакетов
<b>password</b>	Пароль пользователя для авторизации на привратнике
<b>port</b>	Порт привратника.
<b>prefixes</b>	Список префиксов шлюза или E.164 псевдонимов
<b>security</b>	Способ авторизации на привратнике
<b>terminal</b>	Способ представления MVTС привратнику.
<b>type</b>	Признак принадлежности MVTС к зоне привратника.
<b>user</b>	Имя пользователя для авторизации на привратнике

#### 5.4.3. ДОБАВЛЕНИЕ ПРИВРАТНИКА (GATEKEEPER)

Для того чтобы добавить данные о привратнике (gatekeeper), нужно:

- Внести в файл данных привратника запись о привратнике в виде отдельной секции файла с помощью любого текстового редактора (структура файла описана в главе 6.1.4)
- Отредактировать соответствующую запись в файле данных шлюзов (глава 5.4)
- Выполнить команду `reload config` консоли администрирования (глава 6.3.4.2)

#### 5.4.4. УДАЛЕНИЕ ПРИВРАТНИКА (GATEKEEPER)

Чтобы удалить запись о привратнике:

- Внесите необходимые изменения в файл данных шлюзов (глава 5.4)
- Удалите запись о привратнике из файла данных.
- Выполнить команду `reload config` консоли администрирования (раздел 6.3.4.2)

#### 5.4.5. ИЗМЕНЕНИЕ ПАРАМЕТРОВ ПРИВРАТНИКА

Изменения могут касаться одного или нескольких привратников.

Для внесения изменений в параметры привратника:

- Внесите необходимые изменения в запись о привратнике
- Выполнить команду `reload config` консоли



---

администрирования (раздел 6.3.4.2)

## 5.5. УПРАВЛЕНИЕ ПЛАНом НАБОРА

### 5.5.1. Как MVTS ИСПОЛЬЗУЕТ ПЛАН НАБОРА

План набора хранится в текстовом файле **dialpeer.cfg** или в заданном администратором (с помощью системного конфигурационного файла **meraproxy.cfg** (см. главу 6.1.3)).

План набора представляет собой список объектов набора (dial peers). При выборе пути звонка определяется наиболее оптимальный объект набора (по приоритету) для данной пары абонентов и из него берется информация, необходимая для установления звонка.

Изменения в файле плана набора могут касаться одного или нескольких объектов набора (dial peers). При внесении изменений убедитесь, что план набора согласуется с файлом данных шлюзов. По окончании редактирования выполните команду `reload config` консоли администрирования (глава 6.3.4.2).

При передаче звонка на шлюз назначения может потребоваться изменить следующие характеристики звонка:

- Номер вызываемого абонента  
Применяется для добавления в номер технических префиксов. Правила изменения задаются полем `dst_translate` в конфигурации объекта набора (dial peer)
- Номер вызывающего абонента  
Применяется для расширения номера до полного международного. Правила изменения задаются полем `src_translate` в конфигурации объекта набора (dial peer)
- Кодеки, используемые для передачи голоса  
Применяется для запрещения использования определенных кодеков на некоторых шлюзах. Правила задаются полями `codec_allow`, `codec_deny`

При установлении звонка участвуют два шлюза, у каждого из которых есть свои наборы `codec_allow` и `codec_deny`. Для построения соответствующих наборов для звонка наборы `codec_deny` объединяются, а наборы `codec_allow` пересекаются.

Решение о возможности использования некоторого кодека принимается на основании следующего алгоритма:

- Использование будет разрешено, если кодек перечислен в поле `codec_allow`
- Иначе, использование будет запрещено, если кодек перечислен в поле `codec_deny`

- Иначе, использование будет разрешено

Поиск кодека останавливается при обнаружении первого вхождения. В случае, если набор `codec_allow` (`codec_deny`) не задан, считается, что разрешены все кодеки (запрещенных кодеков нет).

### 5.5.2. АЛГОРИТМ ВЫБОРА ОБЪЕКТА НАБОРА

Сервер просматривает все адресуемые направления передачи звонка (объекты набора) в порядке убывания их приоритета. Объект набора (`dial peer`) считается подходящим, если он удовлетворяет следующим условиям

- номера вызываемого и вызывающего абонентов удовлетворяют условиям `dst_pattern` и `src_pattern` соответственно

Если условие `src_pattern` не сформулировано, то любой номер считается удовлетворительным для данного направления.

- группе, к которой принадлежит звонящий шлюз, не запрещены исходящие звонки через данный объект набора

Множества групп шлюзов, которым разрешено/запрещено совершать звонок через данный объект набора, определяются по следующему алгоритму:

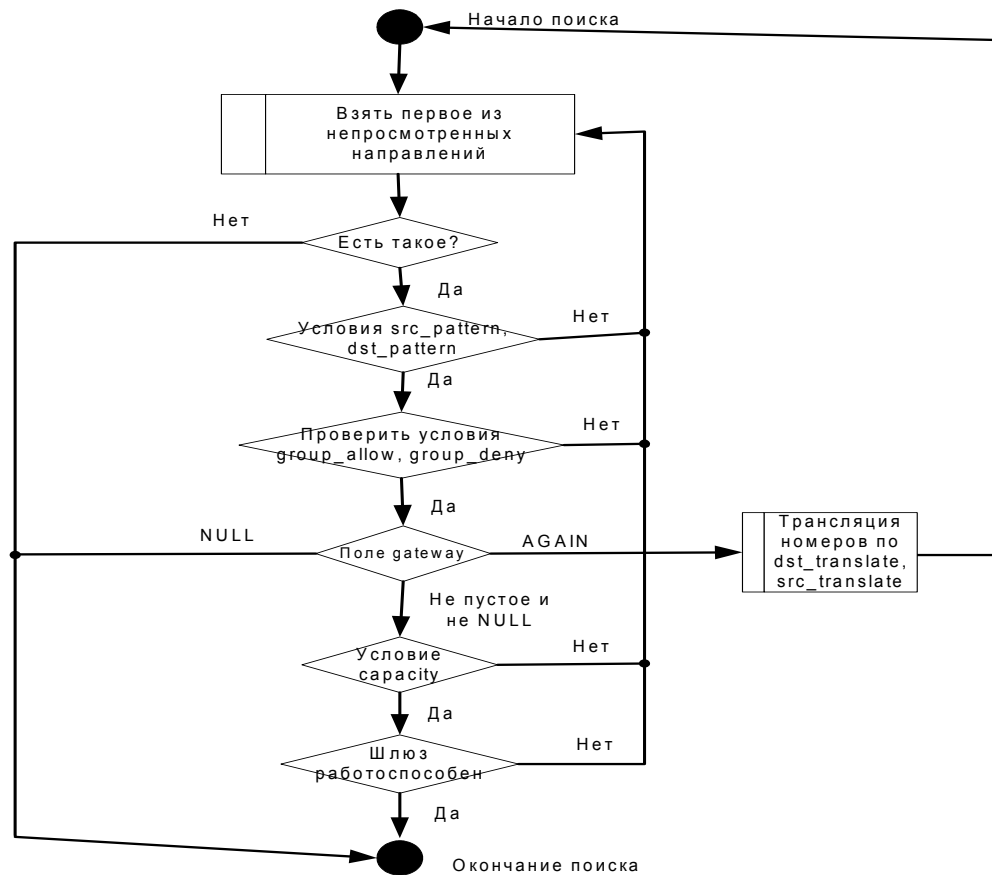
- если `group_allow` и `group_deny` - пустые, то разрешено всем группам, иначе
- если `group_deny` - не пустое и `group_allow` - пустое, то разрешено всем группам, кроме перечисленных в `group_deny`, иначе
- если `group_allow` - не пустое и `group_deny` - пустое, то разрешено только группам, перечисленным в `group_allow`, иначе
- если `group_allow` и `group_deny` - не пустые, то разрешено только группам, перечисленным в `group_allow` и не перечисленным в `group_deny`
- шлюз, указанный в поле `gateway` записи об объекте набора, не загружен полностью и не имеет признака временной неработоспособности (параметр `accessibility=1` см. раздел 6.3.4.3. Команды диагностики, *Команда show gw*)

Звонок направляется на шлюз, связанный с найденным объектом набора.

Если в записи об объекте набора, выбранном в соответствии с полями `dst_pattern` и `src_pattern`, в поле `gateway` указано служебное слово `AGAIN` (`gateway=AGAIN`), то производится

трансляция номеров, и поиск объекта начинается заново с измененными номерами (максимальный уровень вложенности поиска 10).

**Рис. 5-5-1. Алгоритм выбора объекта набора**



### 5.5.3. СТРУКТУРА ПЛАНА НАБОРА

Каждый объект набора (dial peer) имеет параметры, перечисленные в таблице.

**Таблица 5-4: Параметры объекта набора (dial peer)**

Поле	Назначение
active_time	Задание интервала работы объекта набора (dial peer)

Поле	Назначение
<b>bill_translate</b>	Шаблоны номеров в формате, удобном для системы начисления платы. (данный параметр больше не используется для целей конфигурации, так как его функции выполняются параметрами <code>dst_bill_translate</code> и <code>src_bill_translate</code> . В конфигурационных файлах оставлен для совместимости с более ранними версиями MVTС)
<b>capacity</b>	Макс. количество одновременных вызовов, направляемых через данный объект набора
<b>dst_bill_translate</b>	Шаблон для дополнительной трансляции номера вызываемого абонента в формат, удобный для системы биллинга
<b>dst_exclude</b>	Список исключенных вызывающих номеров
<b>dst_pattern</b>	Шаблоны номеров вызываемого абонента, список регулярных выражений через точку с запятой.
<b>dst_translate</b>	Трансляция номеров вызываемого абонента.
<b>gateway</b>	Имя шлюза (из файла данных шлюзов), к которому будет перенаправлен звонок.
<b>group_allow</b>	Список групп шлюзов, которым разрешены исходящие звонки через данный объект набора.
<b>group_deny</b>	Список групп шлюзов, которым запрещены исходящие звонки через данное направление.
<b>group_translate</b>	Трансляция группы пользователя.
<b>hunt_mode</b>	Способ распределения нагрузки между шлюзами.
<b>hunt_stop</b>	Флаг прекращения дальнейшего поиска в случае, когда обнаруженное направление является правильным, но связанный с ним шлюз либо не найден, не доступен или перегружен.
<b>override_proxy_mode</b>	Данный параметр позволяет изменять режим проксирования и действует только в отношении объекта набора ( <code>dial peer</code> ), в записи которого он присутствует.
<b>priority</b>	Приоритет объекта набора ( <code>dial peer</code> ). Позволяет выбрать один из группы, подходящих для данного звонка, объектов. Большие значения соответствуют большему приоритету.
<b>q931_cause</b>	задает Q931 причину завершения звонка в случае попадания объект набора с параметром <code>gateway=NULL</code> .

Поле	Назначение
<b>src_exclude</b>	Список исключенных вызывающих номеров.
<b>src_bill_translate</b>	Шаблон дополнительного приведения вызывающего номера к формату удобному для учета начисляемой платы (биллинга)
<b>src_pattern</b>	Шаблоны номеров вызывающего абонента, список регулярных выражений через точку с запятой.
<b>src_translate</b>	Трансляция номеров вызывающего абонента.

Выбор оптимального направления для звонка производится по алгоритму, описанному выше, в разделе 5.5.2.

#### 5.5.4. ДОБАВЛЕНИЕ ОБЪЕКТА НАБОРА

Добавить запись в файл плана набора можно с помощью любого текстового редактора (структура файла описана в разделе 6.1.7). По окончании редактирования выполните команду `reload config` консоли администрирования (раздел 6.3.4.2).

#### 5.5.5. УДАЛЕНИЕ ОБЪЕКТА НАБОРА

Удалите запись об объекте из файла плана набора с помощью любого текстового редактора. По окончании редактирования выполните команду `reload config` консоли администрирования (раздел 6.3.4.2).

#### 5.5.6. НАСТРОЙКА ФУНКЦИИ ПЕРЕНАПРАВЛЕНИЯ (LOOK\_AHEAD\_ROUTING)

Функция представляет собой обработку отказа или невозможности одним из шлюзов принять звонок. В этом случае анализируется причина отказа, и если она входит во множество, для которого применяется данная функция, то поиск шлюза, могущего принять звонок, продолжается со следующего за данным (в порядке приоритета). Таким образом, звонок может перенаправляться до тех пор, пока

- один из шлюзов не начнет его обработку (пришлет ответное сообщение, отличное от `ReleaseComplete`);
- не будут опрошены все шлюзы, способные принять звонок;
- звонящий не положит трубку.

Звонок также не будет перенаправляться, если в параметрах шлюза установлено свойство `hunt_stop` - отказ от дальнейшей маршрутизации звонка в случае неудачи.

Перенаправление звонка возможно, только если текущий вызываемый шлюз не начал его обработку. Если вызываемый шлюз ответил на звонок каким-либо промежуточным сообщением (`CallProceeding`,

Alerting, Progress) и затем отказался его обрабатывать(прислал ReleaseComplete), то звонок завершится.

Если у текущего объекта набора (dial\_peer) в поле gateway есть несколько шлюзов, то сначала они все будут опрошены в том же порядке и только затем (если ни один из этих шлюзов не примет звонок) прокси перейдет к рассмотрению следующего объекта набора. Если же по вызываемому номеру первым был найден шлюз, за которым закреплен этот номер (запись имеется в gateway.cfg), то данная функция вообще не будет применяться.

Для настройки нужно:

1. указать причины (поле Q931::Cause) окончания звонков, для которых следует использовать данную функцию.
2. в файле **dialpeers.cfg** прописать несколько объектов набора (их настройки должны позволять нескольким объектам обрабатывать один и тот же звонок).
3. флаг hunt\_stop должен быть сброшен (установлен в 0) для возможности перенаправления звонка.

Особенности взаимодействия функции со значением AGAIN в поле gateway.

Объект, имеющий поле gateway=AGAIN, выполняет трансляцию номеров вызывающего и вызываемого абонентов в соответствии с правилами dst\_translate, src\_translate, и поиск начинается сначала (с объекта с самым высоким приоритетом). Таким образом, следует устанавливать этому объекту (dial peer) наиболее низкий приоритет, иначе объекты с более высоким приоритетом могут быть опрошены с исходными и транслированными номерами, а с более низким - только с транслированными.

Функция LAR реализована таким образом, что при отказе текущего объекта (его шлюза) обрабатывать звонок, поиск продолжается со следующего за ним объекта набора (dial peer) с теми же условиями. Если текущий объект был найден после нескольких трансляций номеров (через AGAIN), то для поиска следующего будут использованы эти же транслированные номера. Уровень рекурсии также запоминается. Если при продолжении поиска осуществляется трансляция номеров через AGAIN, то поиск начинается с начала и при определенной конфигурации может быть выбран тот же объект. В этом случае звонок может быть направлен несколько раз на один и тот же шлюз с разными вызываемыми номерами.

Будьте внимательны при использовании AGAIN!

### Пример.

```
[TEST1_DP]
```

```
dst_pattern=1[0-9]*
src_pattern=[0-9]*
priority=100
hunt_stop=1
gateway=TEST1

[TEST2_DP]
dst_pattern=2[0-9]*
src_pattern=[0-9]*
priority=90
hunt_stop=0
gateway=TEST2

[TEST3_DP]
dst_pattern=2[0-9]*
src_pattern=[0-9]*
priority=80
hunt_stop=0
gateway=TEST3

[TEST4_DP]
dst_pattern=32[0-9]*
src_pattern=[0-9]*
dst_translate=.* /4&
src_translate=
priority=70
hunt_stop=0
gateway=TEST4

[AGAIN_DP]
dst_pattern=2[0-9]*
src_pattern=[0-9]*
dst_translate=.* /3&
src_translate=
priority=60
hunt_stop=0
gateway=AGAIN
```

Рассмотрим попытку пользователя позвонить по номеру 222222.  
Алгоритм следующий:

Первый подходящий объект набора TEST2\_DP. Предпринимается попытка направить туда звонок.

Шлюз TEST2 не принимает звонок.

Выбирается следующее направление - TEST3\_DP. Шлюз TEST3 не принимает звонок.

Очередной подходящий объект - AGAIN\_DP. Транслируется номер 222222 -> 3222222, и поиск начинается заново.

Первое подходящее направление – TEST4\_DP. Звонок направляется по нему. При этом вызываемый номер - 43222222.

Если у объекта TEST3\_DP флаг `hunt_stop=1` и шлюз TEST3 не принял звонок, то звонок завершится.

### 5.5.7. ИЗМЕНЕНИЕ ПАРАМЕТРОВ

Изменения могут касаться одного или нескольких объектов набора.

Для внесения изменений в файл плана набора:

- Внесите необходимые изменения в план набора при помощи любого текстового редактора
- Если изменения касались поля `gateway` какого-либо из объектов набора, то внесите соответствующие изменения в файл данных шлюзов

По окончании редактирования выполните команду `reload config` консоли администрирования (глава 6.3.4.2).

## 5.6. УПРАВЛЕНИЕ НАЧИСЛЕНИЕМ ПЛАТЫ

### 5.6.1. ОРГАНИЗАЦИЯ НАЧИСЛЕНИЯ ПЛАТЫ

MVTS позволяет получить информацию, необходимую для работы системы начисления платы. MVTS генерирует записи об установлении и окончании звонка в специальном текстовом формате и помещает их в файлы биллинговой статистики (см. главу 4.5.1). Эти файлы располагаются в каталоге **billing/**. Система позволяет удаленному пользователю (или сервису) получить файлы биллинговой статистики с помощью протокола FTP или SCP для дальнейшей обработки. При этом гарантируется, что файлы доступны для передачи, уже не используются системой.

Запись текущей статистики начисления платы (биллинговой статистики) ведется всегда во внутренний файл.

MVTS не заботится об удалении устаревших файлов (см. главу 5.8.4).

CDR-файл системы начисления платы представляет собой текстовый файл, организованный по принципу “один звонок – одна запись”.

---

**Примечание:** *Чтобы получить подробную информацию об отдельно взятом звонке, администратор может провести поиск в файлах протокола работы MVTS по идентификатору звонка и времени его начала.*

---

### 5.6.2. CDR-ФАЙЛЫ

Частота обновления и имена файлов начисления платы задаются администратором в системном конфигурационном файле, секция



[Billing] (глава 6.1.3).

CDR-файлы из каталога **billing/** доступны пользователям групп администраторов и оформления счетов (Admin и Billing). Пользователи этих групп должны заботиться о своевременном удалении устаревших файлов.

Подробную информацию о структуре CDR файлов см. в разделе 6.1.8.

### 5.6.3. RADIUS

С точки зрения ведения учета статистики причитающейся платы информационный обмен MVTs с сервером по протоколу RADIUS совместим с форматами CISCO (CISCO Attribute 44 и CISCO VSA [6]).

В случае, когда в учет причитающейся платы предполагается вести через систему RADIUS, настройка MVTs для работы с сервером осуществляется заполнением соответствующих полей в системном конфигурационном файле **meraproxy.cfg**.

Для работы MVTs с RADIUS сервером, в поле `acct_enable` необходимо поставить значение 1, а в поле `acct_address` (секция [Radius]) указать адрес RADIUS сервера.

Подробное описание значений полей конфигурационных файлов и форматов отдельных параметров приводится в разделе 6.1.2 Формат конфигурационных файлов

## 5.7. УПРАВЛЕНИЕ ЗВОНКАМИ

Просмотр состояния звонка и управление звонком осуществляется с помощью консоли администрирования. Возможности различных пользователей зависят от группы, к которой они принадлежат (см. разделы 4.7, 6.3.1).

Пользователи группы Admin могут принудительно завершить звонок с заданным идентификатором или группу звонков с участием одного шлюза (раздел 6.3.4.2).

### 5.7.1. ПРОСМОТР СПИСКА АКТИВНЫХ ЗВОНКОВ

Для просмотра списка активных звонков необходимо:

- Вызвать консоль администрирования, запустив на выполнение скрипт **mp\_shell.sh**
- После появления приглашения консоли выполнить команду `show call`.

Подробно о вызове консоли администрирования, о командах консоли и их параметрах см. раздел 6.3 «Консоль администрирования»

### 5.7.2. ПРОСМОТР ДЕТАЛЬНОЙ ИНФОРМАЦИИ О ЗВОНКЕ

Пользователи групп Admin и Support могут просмотреть текущее состояние одного или нескольких звонков с помощью команды `show call` (см. раздел 6.3.4.2).

Для просмотра детальной информации о звонке необходимо:

- Выполнить действия с консолью администрирования, описанные в пункте 5.7.1 выше.
- Еще раз запустить на выполнение команду `show call` с указанием в качестве параметра команды идентификационного номера нужного звонка.

Подробно о вызове консоли администрирования, о командах консоли и их параметрах см. раздел 6.3 Консоль администрирования

### 5.7.3. ПРИНУДИТЕЛЬНОЕ ЗАВЕРШЕНИЕ ЗВОНКА

Для принудительного завершения звонка, необходимо:

- запустить консоль администрирования с помощью скрипта **mp\_shell.sh**
- вызвать список активных звонков командой `show call`
- выбрав из списка номер звонка, подлежащего принудительному завершению, выполнить команду `terminate call` с указанием номера звонка в качестве параметра команды

Подробно о вызове консоли администрирования, о командах консоли и их параметрах см. раздел 6.3 Консоль администрирования

### 5.7.4. ИМИТАЦИЯ ЗВОНКА

Имитация звонка осуществляется командой `show dial`. Для имитации звонка необходимо:

- запустить консоль администрирования с помощью скрипта **mp\_shell.sh**
- выполнить команду `show dial`, задав в качестве параметра команды номер вызываемого или вызывающего телефона.

На экране должны появиться сообщения, отражающие ход и результат поиска объекта набора для звонка.

Подробно о вызове консоли администрирования, о командах консоли и их параметрах см. раздел 6.3 Консоль администрирования

### 5.7.5. ПРОСМОТР ОБЩЕЙ СТАТИСТИКИ

Для просмотра общей статистики сервера необходимо выполнить следующие действия:

- запустить консоль администрирования с помощью скрипта **mp\_shell.sh**
- выполнить команду `show stat`

Подробнее о работе с консолью администрирования, о командах консоли и примеры вывода на экран см. в разделе 6.3 Консоль администрирования

## 5.8. ОБЕСПЕЧЕНИЕ УСТОЙЧИВОСТИ

Наибольшую нагрузку на систему создают звонки с проксированием речевого и мультимедийного трафика. Кроме того, нагрузка увеличивается при выборе для передачи данных некоторых видов кодеков (например, G.711). Для обеспечения устойчивости работы системы существуют несколько простых методов: ограничение числа одновременных звонков: автоматический перезапуск системы, ведение протоколов работы, использование программы удаленного контроля за работой MVTС и т.д.

### 5.8.1. ОГРАНИЧЕНИЕ ЧИСЛА ЗВОНКОВ

MVTС ограничивает число одновременных звонков. Максимальное число возможных звонков - это динамическая величина, которая рассчитывается каждый раз при поступлении нового звонка. Если это число достигнуто, обработка нового звонка не производится. Расчет максимально допустимого количества звонков для данного момента времени производится с учетом текущих звонков. Во внимание принимаются следующие параметры:

- Количество звонков без проксирования мультимедийного трафика
- Количество звонков с проксированием мультимедийного трафика
- Используемые кодеки

### 5.8.2. АВТОМАТИЧЕСКИЙ ПЕРЕЗАПУСК

Если программа запущена с помощью скрипта **mp\_kerneld.sh** или командой `start` консоли администрирования, в случае аварийного завершения программы осуществляется запись образовавшихся **core**-файлов в специальный каталог **cores/** (при этом **core**-файл снабжается меткой – временем аварийного завершения программы). Затем программа автоматически перезапускается.

### 5.8.3. ДЕЙСТВИЯ ПРИ ПЕРЕПОЛНЕНИИ ДИСКА

Перепополнение диска может произойти, если файлы протокола работы или статистики биллинга не уничтожаются вовремя. MVTС не занимается удалением устаревших файлов (протоколы работы, файлы начисления платы и **core**-файлы) и не выдает предупреждений администратору о возможном перепополнении диска. Забота о

своевременном удалении устаревших файлов лежит на администраторе системы. Если переполнение дискового пространства все же происходит, система продолжает корректно функционировать, обслуживая звонки, однако, без записи протоколов, файлов начисления платы и **core**-файлов. Вся информация за время работы системы с момента переполнения диска будет безвозвратно утеряна.

При переполнении дискового пространства Администратору необходимо удалить с жесткого диска накопившиеся старые файлы, распорядившись ими в соответствии с принятой в компании архивной политикой и обычаями документооборота.

---

**Примечание:** *MVTS не занимается уничтожением устаревших файлов протокола работы, core-файлов и других, поэтому администратору следует позаботиться о своевременном освобождении диска. Реакция MVTS на переполнение диска описана в главе 5.8.3 “Действия при переполнении диска”.*

---

#### 5.8.4. ДРУГИЕ ЗАДАЧИ ПО ОБЕСПЕЧЕНИЮ УСТОЙЧИВОСТИ

Администратор должен проследить за соблюдением требований к системе, описанных в разделе 3.1.1 “Аппаратные средства”.

Администратору следует позаботиться о создании резервных копий конфигурационных файлов и файлов начисления платы.

Для периодической диагностики работоспособности MVTS администратор может воспользоваться утилитой MVTS Manager, запустив его на удаленном компьютере.

Рекомендуется пользоваться командой `reload config` консоли администрирования для перезагрузки конфигурационных файлов, так как при таком способе MVTS прерывает выполнение своих функций на минимальное время и через системную консоль можно получить наиболее полную информацию об ошибках.

#### 5.8.5. ПРОТОКОЛ РАБОТЫ

Протокол работы MVTS представляет собой статистику течения каждого звонка, поступившего на MVTS. Он организуется для службы технической поддержки. Протокол работы пишется MVTS в текстовом виде в файл `logs_<date>_<time>` в каталоге **logs/** (<date> и <time> указывают время начала записи в файл). Он может быть использован в целях отладки и анализа работы. При очередном старте системы открывается новый файл для записи протокола. Текущий протокол, а также архив доступны для загрузки на удаленный компьютер через протокол FTP или SCP.

Протокол работы позволяет получить для каждого звонка:

- Время получения сообщений

- тексты сообщений в абстрактной синтаксической нотации ASN.1

#### 5.8.5.1. УПРАВЛЕНИЕ ПАРАМЕТРАМИ ПРОТОКОЛА

Администратору разрешается конфигурировать следующие настройки протокола:

- Степень детализации информации. Позволяет получить протокол с разной степенью детализации или отключить его вывод полностью. По умолчанию запись протокола отключена.
- Частота смены файла протокола. Позволяет менять текущий файл протокола работы через заданные промежутки времени. По умолчанию - 120 минут.
- Имя файла протокола. По умолчанию - **log**.

Все вышеперечисленные настройки находятся в системном файле конфигурации MVTС в секции [Debug] (см. раздел 6.1.3 “Структура системного конфигурационного файла”).

Файлы протокола работы, находящиеся в каталоге **logs/**, создаются MVTС и могут быть использованы администратором по своему усмотрению.

---

**Примечание:** *Вывод подробной информации предназначен для тестирования и отладки MVTС. Поэтому в устоявшемся режиме работы его использование не рекомендуется.*

---

## 6. СПРАВОЧНАЯ ИНФОРМАЦИЯ

### 6.1. КОНФИГУРАЦИОННЫЕ ФАЙЛЫ

#### 6.1.1. ОБЗОР

Пять конфигурационных файлов (**meraproxy.cfg**, **user.cfg**, **gateway.cfg**, **gatekeeper.cfg**, **dialpeer.cfg**) выполняют двойную функцию – содержат настройки системы, используя для целей конфигурирования, администрирования системы и одновременно являются файлами хранения данных, необходимых для работы MVTС.

#### 6.1.2. ФОРМАТ КОНФИГУРАЦИОННЫХ ФАЙЛОВ

Все конфигурационные файлы имеют текстовый формат. Файлы разбиты на секции, каждая из которых представляет собой запись об объекте. Название каждой секции пишется с новой строки в квадратных скобках и может состоять только из следующих символов:

- Латинские буквы в верхнем или нижнем регистре
- Цифры ‘0’-‘9’
- Знак подчеркивания
- Точки

Первая скобка должна быть в начале строки. Пробелы после открывающей скобки, перед закрывающей скобкой и после неё игнорируются.

Пример правильного названия секции:

```
[ Section ]
```

Порядок следования параметров внутри секции не важен. Названия параметров пишутся с начала строки. Между названием параметра и его значением ставится знак “=”. Если в качестве значения выступает список, для разделения его составляющих используется ‘;’. Длинный список значений, не уместяющийся в пределах одной строки для удобства редактирования может быть перенесен на следующую строку с обязательным дублированием названия параметра.

#### Пример:

```
[ata3]  
address=183.132.44.76;183.132.44.78;183.132.44.71;183.132.44.79;  
address=183.132.44.77
```

Значение параметра не должно содержать символов:

- пробел, табуляция
- перевод строки

Регистр букв в названиях параметров и секций должен соблюдаться.

Игнорируются:

- пробелы и знаки табуляции до и после знака “=”
- пробелы между скобками и названием секции
- пустые строки
- строки, начинающиеся со знака “#” (комментарии)
- параметры с неизвестными названиями
- Целые положительные значения конфигурационных параметров могут вводиться в десятичном, шестнадцатиричном и восьмиричном представлении. Значения в шестнадцатиричном представлении должны начинаться с **0x**, значения в восьмиричном представлении должны начинаться с **0**.

### 6.1.3. СТРУКТУРА СИСТЕМНОГО КОНФИГУРАЦИОННОГО ФАЙЛА

Системные настройки хранятся в файле **\$H323PROXY\_ROOT/cfg/meraproxy.cfg** (см. главу 3.1.3 “Конфигурация сети”) или в файле, заданном администратором в командной строке. Файл может быть отредактирован любым текстовым редактором. Рекомендуется перед редактированием скопировать файл во временный и по окончании скопировать отредактированный файл обратно в исходный.

**Пример:**

```
[Administration]
gateway_cfg=gateway.cfg
dialpeer_cfg=dialpeer.cfg
user_cfg=user.cfg
gatekeeper_cfg=gatekeeper.cfg
mail_alert=skvair@mera.ru
max_call_duration=5000
min_call_duration=200
local_address=*
public_access=1
#master_gatekeeper=

[Console]
console_port=1730
admin_gid=0
billing_gid=0
support_gid=0

[H323]
port=1720
alternate_port=
#converter=

[Billing]
file=../billing/bill
```

```
bil_tmpfile_attr=644
bil_file_attr=644
cdr_format=0
period=30:00
level=3
no_empty=1

[Debug]
trace_level=3
level=1
period=1:00:00
file=./debug/logs/log
debug_tmpfile_attr=664
debug_file_attr=664
no_empty=0
max_size=
max_lifetime=

[Radius]
local_address=*
auth_enable=0
acct_enable=0
acct_type=1
acct_leg_type=2
acct_boot_msg=0
acct_stop_only=0
acct_interim_update=30
auth_password_type=1
secret=strongsecret
auth_address=195.98.57.10
auth_port=1812
auth_password_type=0
acct_address=195.98.57.10
acct_port=1813
dst_user_orig_leg=1
local_auth_port=1644
local_acct_port=1645
local_route_port=1725
repeat_time=7
repeat_quantity=5
route_address=915.98.44.23
route_enable=1
route_port=1723
q931_h323_disconnect_cause=0
suppress_Q931_cause=0
use_h323_ivr_in=1

[BackupRadius]
local_address=*
auth_enable=
auth_address=
auth_port=
acct_enable=
acct_address=
acct_port=
repeat_time=7
repeat_quantity=5
route_enable
route_address
```



```
route_port
local_auth_port=
local_acct_port=
local_route_port=
secret=strongsecret
```

```
[Gatekeeper]
gatekeeper_enable=1
gatekeeper_id=
ras_port=
mult_port=
mult_group=
```

```
[LAR]
route_cause=0-16;17-127
validation_gap=3000
```

```
[proxy]
read_route_table=0
alias_route_path=
local_ip_manager=192.168.1.1/120/120;192.168.1.2/64/64;192.168.1.3/512/256
local_ip_manager_mode=3
```

```
[SNMP]
trap_email=admin@noztel.com
trap_email_cmd=mail_script
trap_email_subject=new traps received
trap_email_from=MERA softswitch server 3
trap_email_period=02:00:00
local_port=161
contact_info=esquire@mear.com +78213995875
system_name=MAIN SILONET GK
system_location=
community=public
trap_community=
trap_address=194.23.122.14
trap_port=162
trap_level=5
trusted_address=126.44.23.15
trap_enable=1
snmp_enable=0
```

```
[Statistics]

statistics_enable=1
file=../debug/stat
period=120:30
reset_after_save=1
time_span=
no_empty=1
asr_alert_email=admin@telnoz.com
call_radix=70
suspend_time=1440
call_sampling_number=45
set_route_asr=20
```

```
stat_renewal_time=1440
smart_routing=1
route_statistics_enable=1
stat_email=admin@telnoz.com
stat_email_cmd=stat_script
stat_email_from=MERA Server 3
stat_email_subject=Latest call statistics
```

### Секция [Administration]

Эта секция содержит настройки администрирования.

#### Поле gateway\_cfg

Имя файла, содержащего данные о шлюзах. Путь указывается относительно корневого каталога или файла системных настроек.

**Формат:**

```
gateway_cfg = имя_файла
```

**Значение по умолчанию:**

```
gateway.cfg
```

**Пример:**

```
gateway_cfg = my_gateway.cfg
```

#### Поле dialpeer\_cfg

Файл, содержащий план набора. Путь указывается относительно корневого каталога или файла системных настроек.

**Формат:**

```
dialpeer_cfg = имя_файла
```

**Значение по умолчанию:**

```
dialpeer.cfg
```

**Пример:**

```
dialpeer_cfg = my_dialpeer.cfg
```

#### Поле user\_cfg

Имя файла, содержащего данные о регистрируемых пользователях. Путь указывается относительно корневого каталога или файла системных настроек.

**Формат:**

```
user_cfg = имя_файла
```

**Значение по умолчанию:**

```
user.cfg
```

**Пример:**

```
user_cfg = my_user.cfg
```

**Поле** `gatekeeper_cfg`

Имя файла, содержащего данные о привратниках. Путь указывается относительно корневого каталога или файла системных настроек.

**Формат:**

```
gatekeeper_cfg = имя_файла
```

**Значение по умолчанию:**

```
gatekeeper.cfg
```

**Пример:**

```
gatekeeper_cfg = my_gatekeeper.cfg
```

**Поле** `mail_alert`

E-mail адрес, на который будут отсылаться письма с предупреждениями о системных сбоях. Поле не обязательно для заполнения.

**Формат:**

```
mail_alert = адрес
```

где адрес - e-mail адрес, заданный в виде somebody@domain.ru.

**Пример:**

```
mail_alert = admin@mera.ru
```

**Поле** `master_gatekeeper`

Имя основной системы, выступающей в качестве привратника (gatekeeper), на котором в качестве RAS пользователя регистрируется резервная система. Данное поле используется только при конфигурировании системы, выступающей в качестве резервной.

**Формат:**

```
master_gatekeeper=<строка>
```

где <строка> – символьная строка имени привратника, совпадающего с названием соответствующей секции в файле данных привратников **gatekeeper.cfg**

**Пример:**

```
master_gatekeeper=Major_GK
```

**Поле** `max_call_duration`

... задает предел максимальной продолжительности звонка (в секундах). Допустимы значения от 100 до 1000000. Если продолжительность звонка превышает установленный предел, MVTS завершает звонок с внутренним кодом 301 (предположительно зависший звонок).

**Формат:**

```
max_call_duration=<число>
```

где <число> – положительное целое в диапазоне от 100 до 1000000.

**Значение по умолчанию:**

10000 (2ч. 46мин. 40сек.)

**Пример:**

```
max_call_duration=3000
```

**Поле** `min_call_duration`

... задает порог округления дробной части продолжительности звонка до ровных секунд. Установка параметра, скажем, равным 200 (например, `min_call_duration=200`) означает, что звонок фактической продолжительностью 45,19 сек. будет зарегистрирован как звонок, который длился ровно 45 сек., а продолжительность звонка длительностью в 45,21 сек. будет округлена до 46 сек.

**Формат:**

```
min_call_duration=<число>
```

где <число> – положительное целое в диапазоне от 10 до 999.

**Значение по умолчанию:**

500

**Пример:**

```
min_call_duration=300
```

**Поле** `local_address`

IP-адрес, с которого MVTС устанавливает звонки, если в системной таблице маршрутизации не хватает данных для установления соединения с вызываемым шлюзом. Необязательное поле.

**Формат:**

```
local_address = <address> или *
```

где <address> – IP-адрес или \* - автоматическое определение.

**Значение по умолчанию:**

Поле default gateway из системной таблицы маршрутизации.

**Пример:**

```
local_address = *
```

**Поле** `public_access`

Отключает авторизацию шлюзов по IP адресу, в этом режиме разрешены все звонки со всех шлюзов. Полезен для отладки.

**Формат:**

```
public_access = 1 | 0
```

**Значение по умолчанию:** 0

**Пример:**

```
public_access = 0
```

**Секция [H323]**

Секция содержит настройки H.323 протокола.

**Поле** `alternate_port`

... определяет дополнительный (альтернативный) порт для приема сигнального трафика.

**Формат:**

```
port = <число>
```

где <number> – положительное целое значение для номера порта

**Значение по умолчанию:**

НЕТ

**Пример:**

```
port=1770
```

**Поле** `port`

Порт для входящих звонков.

**Формат:**

```
port = число
```

где число – целое, положительное.

**Значение по умолчанию:**

1720

**Пример:**

```
port = 1770
```

**Поле** `converter`

Используйте данное поле для определения параметров модуля межпротокольной трансляции (конвертера).

**Формат:**

```
converter=<строка>;<строка>
```

где <строка> – строка данных в формате

```
converter_name/converter_ip_address/port/mode/type
```

`converter_name` – имя конвертера

`converter_ip_address` – IP адрес конвертера

`port` – номер порта

`mode` – режим трансляции (аналогично параметру `gateway_mode`)

`type` – тип протокола, поддерживаемого шлюзом (**1** – H323; **2** – SIP)

Список из нескольких конвертеров разделяется ;

**Пример:**

```
converter= SIPconv/198.98.12.50/1720/3/2;H323conv/198.98.12.12/1720/3/1
```

**Секция [Billing]**

Секция содержит настройки системы учета и начисления платы.

Поле `period`

Максимальное время записи в один файл.

**Формат:**

```
period = [[<часы >:] минуты:] секунды
```

**Значение по умолчанию:**

```
15:00
```

**Пример:**

```
period = 120:40
```

Поле `file`

Путь и имя файла, в который ведется запись статистики начисления платы (без учета метки времени и флага текущего файла).

**Формат:**

```
file = имя
```

где имя - строка из букв 'A'...'Z', 'a'...'z' и символов подчеркивания.

**Значение по умолчанию:**

```
bill
```

**Пример:**

```
file = ../billing/bill
```

Поле `bil_tmpfile_attr`

Атрибуты временного файла учета и начисления платы (биллинга). Атрибуты (разрешение на чтение файла, запись в файл или его исполнение) аналогичны атрибутам файлов в UNIX-подобных ОС.

Каждая из трех позиций числа – цифра от 0 до 7 (т.е. от 000 до 111). Цифра в первой позиции относится к правам на файл владельца файла, вторая - отражает права пользователей группы, к которой принадлежит владелец файла, третья – права всех остальных пользователей. Право на соответствующую операцию с файлом считается доступным, если соответствующий бит равен 1.

**Формат:**

```
bil_tmpfile_attr = <число>
```

где <число> атрибуты временного файла выраженные трехзначным числом

**Значение по умолчанию:**

NIL

**Пример:**

```
bil_tmpfile_attr=664
```

атрибут 664 (в двоичном представлении **110 110 100**) разрешает:

Владельцу файла **110 – 1** разрешено чтение файла; **1** разрешена запись в файл; **0** – запрещено исполнение файла

Членам группы владельца файла **110 – 1** разрешено чтение файла; **1** разрешена запись в файл; **0** – запрещено исполнение файла

Остальным пользователям системы **100 – 1** разрешено чтение файла; **0** запрещена запись в файл; **0** – запрещено исполнение файла.

**Примечание:** Независимо от значащихся в конфигурационном файле атрибутов владельцу файла всегда доступны для чтения и записи как временные, так и постоянные файлы.

**Поле `bil_file_attr`**

Атрибуты постоянного файла учета и начисления платы (биллинга). Атрибуты (разрешение на чтение файла, запись в файл или его исполнение) аналогичны атрибутам файлов в UNIX-подобных ОС. Каждая из трех позиций числа – цифра от 0 до 7 (т.е. от 000 до 111). Цифра в первой позиции относится к правам на файл владельца файла, вторая - отражает права пользователей группы, к которой принадлежит владелец файла, третья – права всех остальных пользователей. Право на соответствующую операцию с файлом считается доступным, если соответствующий бит равен 1.

**Формат:**

```
bil_file_attr = <число>
```

где <число> атрибуты временного файла выраженные трехзначным числом

**Значение по умолчанию:**

NIL

**Пример:**

```
bil_file_attr=664
```

атрибут 664 (в двоичном представлении **110 110 100**) разрешает:

Владельцу файла **110 – 1** разрешено. чтение файла; **1** разрешена запись в файл; **0** – запрещено исполнение файла

Членам группы владельца файла **110 – 1** разрешено. чтение файла; **1** разрешена запись в файл; **0** – запрещено исполнение файла

Остальным пользователям системы **100 – 1** разрешено. чтение файла; **0** запрещена запись в файл; **0** – запрещено исполнение файла.

**Примечание:** Независимо от значащихся в конфигурационном файле атрибутов владельцу файла всегда доступны для чтения и записи как временные, так и постоянные файлы.

**Поле** `cdr_format`

Флаг выбора формата CDR-файлов.

**Формат:**

```
cdr_format=<число>
```

где число – целое 0 или 1. 0 – запись в CDR-файл ведется во внутренне присущем MVTС формате, 1 – запись в CDR-файл ведется в формате MIND СТІ.

**Значение по умолчанию:**

0

**Пример:**

```
cdr_format=1
```

**Поле** `level`

Включение/выключение записи в файл начисления платы.

**Формат:**

```
level = число
```

где число – целое, положительное 0 или 1. 0 – запись в файл отсутствует, 1 – запись в файл ведется. Все значения, превышающие 1, воспринимаются системой как 1.

**Значение по умолчанию:**

нет

**Пример:**

```
level = 1
```



**Поле** `no_empty`

Флаг включения/выключения записи «пустых» файлов начисления платы. При установке флага в значение 1 прокси-сервер не оставляет пустых файлов биллинга.

**Формат:**

`no_empty = число (1 или 0)`

**Значение по умолчанию:**

нет

**Пример:**

`no_empty=1`

**Секция [Debug]**

Секция содержит настройки протокола работы.

**Поле** `max_size`

Задаёт максимальный суммарный размер файлов с протоколами работы системы (отладочными протоколами) в килобайтах. При превышении заданной величины суммарного размера система стирает старые файлы.

**Формат:**

`max_size=<число>`

где `<число>` - положительное целое значение.

**Значение по умолчанию:**

0 (контроль суммарного размера отключен)

**Пример:**

`max_size=500`

**Поле** `max_lifetime`

Задаёт максимальный срок хранения файлов с отладочными протоколами в сутках. При превышении заданного времени хранения файлов с протоколами система стирает старые файлы.

**Формат:**

`max_lifetime=<число>`

где `<число>` - положительное целое значение.

**Значение по умолчанию:**

0 (контроль длительности хранения отключен)

**Пример:**

`max_lifetime=30`

**Поле level**

Уровень детализации при записи информации в файл отладочного протокола (debug log). Чем выше значение, тем выше степень детализации выводимой информации.

**Формат:**

```
level=<число>
```

где число – целое от 0 до 3.

**Значение по умолчанию:**

3

**Пример:**

```
level=2
```

**Поле trace\_level**

Уровень детализации при записи информации в файл трассировочного протокола (trace log), содержащегося в файле с названием mp.kernel.sh.log-<date>. Чем выше значение параметра, тем выше степень детализации выводимой информации.

**Формат:**

```
trace_level=число
```

где число – целое от 0 до 3.

**Значение по умолчанию:**

3

**Пример:**

```
trace_level=2
```

**Поле period**

Максимальное время записи в один файл.

**Формат:**

```
period = [[<часы >:] минуты:] секунды
```

**Значение по умолчанию:**

120:00

**Пример:**

```
period = 120:40
```

**Поле file**

Путь и имя файла, в который ведется запись протокола работы (без учета метки времени и флага текущего файла).

**Формат:**

```
file = имя
```

где имя - строка из букв 'A'...'Z', 'a'...'z' и символов подчеркивания.

**Значение по умолчанию:**

```
log
```

**Пример:**

```
file = ../debug/logs/pr_statistics
```

**Поле** `debug_tmpfile_attr`

Атрибуты временного файла отладки (debug-file). Атрибуты (разрешение на чтение файла, запись в файл или его исполнение) аналогичны атрибутам файлов в UNIX-подобных ОС. Каждая из трех позиций числа – цифра от 0 до 7 (т.е. от 000 до 111). Цифра в первой позиции относится к правам на файл владельца файла, вторая - отражает права пользователей группы, к которой принадлежит владелец файла, третья – права всех остальных пользователей. Право на соответствующую операцию с файлом считается доступным, если соответствующий бит равен 1.

**Формат:**

```
debug_tmpfile_attr = <число>
```

где <число> атрибуты временного файла выраженные трехзначным числом

**Значение по умолчанию:**

```
NIL
```

**Пример:**

```
debug_tmpfile_attr=664
```

атрибут 664 (в двоичном представлении **110 110 100**) разрешает:

Владельцу файла **110 – 1** разрешено. чтение файла; **1** разрешена запись в файл; **0** – запрещено исполнение файла

Членам группы владельца файла **110 – 1** разрешено. чтение файла; **1** разрешена запись в файл; **0** – запрещено исполнение файла

Остальным пользователям системы **100 – 1** разрешено. чтение файла; **0** запрещена запись в файл; **0** – запрещено исполнение файла.

**Примечание:** Независимо от значащихся в конфигурационном файле атрибутов владельцу файла всегда доступны для чтения и записи как временные, так и постоянные файлы.

**Поле** `debug_file_attr`

Атрибуты постоянного файла отладки (debug-file). Атрибуты (разрешение на чтение файла, запись в файл или его исполнение) аналогичны атрибутам файлов в UNIX-подобных ОС. Каждая из трех

позиций числа – цифра от 0 до 7 (т.е. от 000 до 111). Цифра в первой позиции относится к правам на файл владельца файла, вторая - отражает права пользователей группы, к которой принадлежит владелец файла, третья – права всех остальных пользователей. Право на соответствующую операцию с файлом считается доступным, если соответствующий бит равен 1.

**Формат:**

```
debug_file_attr = <число>
```

где <число> атрибуты временного файла выраженные трехзначным числом

**Значение по умолчанию:**

NIL

**Пример:**

```
debug_file_attr =664
```

атрибут 664 (в двоичном представлении **110 110 100**) разрешает:

Владельцу файла **110 – 1** разрешено чтение файла; **1** разрешена запись в файл; **0** – запрещено исполнение файла

Членам группы владельца файла **110 – 1** разрешено чтение файла; **1** разрешена запись в файл; **0** – запрещено исполнение файла

Остальным пользователям системы **100 – 1** разрешено чтение файла; **0** запрещена запись в файл; **0** – запрещено исполнение файла.

**Примечание:** Независимо от значащихся в конфигурационном файле атрибутов владельцу файла всегда доступны для чтения и записи как временные, так и постоянные файлы.

**Поле no\_empty**

Флаг включения/выключения записи «пустых» файлов протокола. При установке флага в значение 1 прокси-сервер не оставляет пустых файлов протокола.

**Формат:**

```
no_empty = число (1 или 0)
```

**Значение по умолчанию:**

нет

**Пример:**

```
no_empty=0
```

**Секция [Console]**

Секция содержит настройки консоли администрирования.

**Поле console\_port**

Используется для задания порта для консоли администрирования,

отличного от значения по умолчанию.

**Формат:**

```
console_port = число
```

где число – целое, положительное.

**Значение по умолчанию:**

```
1730
```

**Пример:**

```
port = 1750
```

**Поле** `admin_gid`

Обязательное поле. Содержит номер группы пользователя в системе, соответствующей группе администраторов сервера.

**Формат:**

```
admin_gid = число
```

где число - целое, положительное.

**Пример:**

```
admin_gid = 101
```

**Поле** `billing_gid`

Обязательное поле. Содержит номер группы пользователя в системе, соответствующей группе тарификации.

**Формат:**

```
billing_gid = число
```

где число - целое, положительное.

**Пример:**

```
billing_gid = 101
```

**Поле** `support_gid`

Обязательное поле. Содержит номер пользовательской группы в системе, соответствующий группе инженеров службы технической поддержки.

**Формат:**

```
support_gid = число
```

где число - целое, положительное.

**Пример:**

```
support_gid = 101
```

**Секция [Gatekeeper]**

Необязательная секция. Добавляется в системный конфигурационный файл при необходимости задействовать функциональность привратника.

**Поле** `gatekeeper_enable`

В зависимости от значения данного поля включается/выключается функция привратника. 1 – функциональность привратника задействовать, 0 – функцию привратника отключить.

**Формат:**

```
gatekeeper_enable = <число>
```

где <число> - 0 или 1

**Пример:**

```
gatekeeper_enable=1
```

**Поле** `ras_port`

Порт привратника для RAS-сообщений .

**Формат:**

```
port = число
```

где число – целое, положительное.

**Значение по умолчанию:**

```
1719
```

**Пример:**

```
port = 1771
```

**Поле** `mult_port`

Порт привратника для многоадресных RAS-сообщений .

**Формат:**

```
port = число
```

где число – целое, положительное.

**Значение по умолчанию:**

```
1718
```

**Пример:**

```
port = 1770
```

**Поле** `mult_group`

Адрес группы многоадресной рассылки RAS-сообщений (RAS multicast).

**Формат:**

```
mult_group= <адрес>
```

где <адрес> – адрес в формате oct.oct.oct.oct.

**Значение по умолчанию:**

```
224.0.1.41
```

**Пример:**

```
mult_group=224.0.135.17
```

**Поле** `gatekeeper_id`

Символьный (произвольный) идентификатор привратника.

**Формат:**

```
gatekeeper_id = <строка символов>
```

где < строка символов > - произвольный набор символов

**Значение по умолчанию:**

```
МераГК: <имя_машины>
```

**Пример:**

```
gatekeeper_id=OwnMVTs
```

**Секция [Radius]**

Секция содержит настройки для работы с сервером удаленной аутентификации пользователей и учета причитающейся платы (RADIUS сервером)

.

**Поле** `local_address`

IP адрес, на который осуществляется привязка (bind) при работе с RADIUS.

**Формат:**

```
local_address = <address>
```

где <address> – IP адрес в формате oct.oct.oct.oct или \*

**Пример:**

```
local_address = *  
local_address = 199.99.99.99
```

**Поле** `auth_enable`

Поле задеирования аутентификации через RADIUS. Обязательное для заполнения поле. 1 – аутентификацию через RADIUS включить, 0 – аутентификацию через RADIUS отключить.

**Формат:**

```
auth_enable = <число>
```

где <число> - 0 или 1

**Пример:**

```
auth_enable=0
```

**Поле** `acct_enable`

Поле задействия RADIUS сервиса по учету начисленной платы. Обязательное для заполнения поле. 1 – использовать RADIUS сервис по начислению платы, 0 – отключить RADIUS сервис по начислению платы.

**Формат:**

```
acct_enable = <число>
```

где <число> - 0 или 1

**Пример:**

```
acct_enable=0
```

**Поле** `acct_type`

Поле форматов файлов учета начисленной платы. Принимаемые значения 0, 1. 0 – старый формат CISCO (так называемый, overloaded attribute 44), 1 – CISCO VSA формат совместимости.

**Формат:**

```
acct_type = <число>
```

где <число> - 0 или 1

**Значение по умолчанию:**

0

**Пример:**

```
acct_type=1
```

**Поле** `acct_leg_type`

Тип эмулируемых при начислении платы RADIUS записей. Принимаемые значения 1, 2, 3, 4 или 5

**1** – использовать start/stop записи, относящиеся к участку ‘answer telephony’;

**2** – использовать start/stop записи, относящиеся к участку ‘originateVoIP’;

**3** – использовать start/stop записи, относящиеся одновременно и к ‘answer telephony’, и к ‘originate VoIP’

**4** – при значении 4 изменяются следующие поля account-пакетов для RADIUS:

incoming leg:

```
h323-call-origin=answer
```



h323-call-type=VoIP  
 h323-remote-address=IP-адрес оригинирующего шлюза  
 h323-remote-id=ID оригинирующего шлюза  
 h323-gw-id=ID терминирующего шлюза  
 h323-setup-time=время получения setup'a от оригинирующего шлюза  
 outgoing leg:  
 h323-call-origin=originate  
 h323-call-type=VoIP  
 h323-remote-address=IP-адрес терминирующего шлюза или gatekeeper'a  
 h323-remote-id=ID терминирующего шлюза или gatekeeper'a  
 h323-gw-id=ID оригинирующего шлюза  
 h323-setup-time=время отправки setup'a терминирующему шлюзу

5 – отличается от предыдущего значения 4 наличием только одного комплекта account-пакетов для участка 'answer leg'. Например, если вызов будет иметь три перенаправления (reroute) то последовательность отправки будет следующей:

```

start answer leg record
start originate leg record 1
stop originate leg record 1
start originate leg record 2
stop originate leg record 2
start originate leg record 3
stop originate leg record 3
stop answer leg record
  
```

**Формат:**

acct\_leg\_type = <число>

где <число> - 1, 2, 3, 4 или 5

**Значение по умолчанию:**

2

**Пример:**

acct\_leg\_type=2

**Поле** acct\_boot\_msg

Флаг отправки сообщений о включении/выключении сервиса начисления платы (Accounting-on/Accounting-off).

**Формат:**

acct\_boot\_msg = <число>

где <число> - 0, 1

**Значение по умолчанию:**

1

**Пример:**

```
acct_boot_msg=0
```

**Поле** `acct_stop_only`

Флаг отправки на RADIUS-сервер только stop-записей.

**Формат:**

```
acct_stop_only = <число>
```

где <число> - 0, 1

**Значение по умолчанию:**

0

**Пример:**

```
acct_stop_only=1
```

**Поле** `acct_interim_update`

Это поле используется для задания в секундах интервала периодичности отправки на RADIUS-сервер помимо start и stop записей еще и промежуточных account сообщений.

**Формат:**

```
acct_interim_update = <число>
```

где <число> целое число секунд, задающее периодичность отправки промежуточных сообщений. При значении 0, промежуточные сообщения не отсылаются. Положительные значения менее 10, независимо от конкретной величины, приводят к минимально возможному интервалу 10 секунд.

**Значение по умолчанию:**

0

**Пример:**

```
acct_interim_update=0
```

**Поле** `auth_password_type`

Поле для указания типа пароля. Принимаемые значения 0, 1. 0 – MD5 хэш пароль (RFC 2138). 1 – пароль в обычном формате ASCII текста (для совместимости с MindCti).

**Формат:**

```
auth_password_type = <число>
```

где <число> - 0 или 1

**Значение по умолчанию:**

0

**Пример:**

```
auth_password_type=1
```

**Поле** `secret`

Обязательное поле. Ключ шифрования (стандарт shared secret) при работе с RADIUS сервером.

**Формат:**

```
secret = <строка>
```

где <строка> - символьное значение ключа.

**Пример:**

```
secret = strongsecret
```

**Поле** `auth_address`

IP адрес сервера удаленной аутентификации RADIUS.

**Формат:**

```
auth_address = oct.oct.oct.oct
```

где oct.oct.oct.oct - IP адрес RADIUS сервера.

**Пример:**

```
auth_address = 195.88.97.20
```

**Поле** `auth_port`

Порт для сервера удаленной аутентификации RADIUS.

**Формат:**

```
auth_port = число
```

где число – целое, положительное.

**Значение по умолчанию:**

1812

**Пример:**

```
port = 1813
```

**Поле** `acct_address`

IP адрес RADIUS сервера для целей начисления платы.

**Формат:**

```
acct_address = oct.oct.oct.oct
```

где oct.oct.oct.oct - IP адрес RADIUS сервера.

**Пример:**

```
acct_address = 195.88.97.20
```

**Поле** `acct_port`

Порт RADIUS сервера для целей начисления платы.

**Формат:**

```
acct_port = число
```

где число – целое, положительное.

**Значение по умолчанию:**

1813

**Пример:**

```
acct_port = 1813
```

**Поле** `dst_user_orig_leg`

Данный флаг используется для управления подстановкой имени пользователя в поле `UserName` пакетов биллинга оригинирующего участка (accounting packets of the originating leg), отсылаемых на RADIUS сервер

**Формат:**

```
dst_user_orig_leg=<number>
```

где <number> – 0 или 1

**Значение по умолчанию:**

0

**Пример:**

```
dst_user_orig_leg=1
```

**Примечание:** Данный параметр функционален только при значениях в поле `acct_leg_type` 4 и 5

**Поле** `q931_h323_disconnect_cause`

Назначение этого флага задействовать подстановку в поле Cisco VSA(30) h323-disconnect-cause вместо реального кода причины разъединения (Q931 disconnect cause), значение кода, полученное в AccessReject от RADIUS-сервера, так как в случае получения отказа (reject) от RADIUS сервера Q931 всегда устанавливается в 3 (destination unreachable).

**Формат:**

```
q931_h323_disconnect_cause=<число>
```

где <число> 0 или 1 . 1 включает функцию подстановки, 0 отменяет подстановку.

**Значение по умолчанию:**

0

**Пример:**

```
q931_h323_disconnect_cause=1
```

**Примечание:** Поскольку этот Q931 cause также заносится и в Release Complete, то следует использовать эту опцию с **ОСТОРОЖНОСТЬЮ**, так как, из опыта реакция некоторых шлюзов на нестандартные значения может приводить к зависанию последних.

**Поле** `local_auth_port`

Порт для приема ответных сообщений с RADIUS сервера.

**Формат:**

```
local_auth_port = число
```

где число – целое, положительное.

**Значение по умолчанию:**

1812

(2812 для секции [BackupRadius])

**Пример:**

```
local_auth_port=1644
```

**Примечание:** Важно, чтобы значение порта, заданное здесь, отличалось от значения порта, заданного в одноименном поле в секции [BackupRadius]

**Поле** `local_acct_port`

Порт для приема ответных сообщений RADIUS сервера для целей начисления платы.

**Формат:**

```
local_acct_port=<число>
```

где <число> – целое, положительное, номер порта

**Значение по умолчанию:**

1813

(2813 для секции [BackupRadius])

**Пример:**

```
local_acct_port=1845
```

**Примечание:** Важно, чтобы значение порта, заданное здесь, отличалось от значения порта, заданного в одноименном поле в секции [BackupRadius]

Поле `local_route_port`

Номер локального порта, для отправки запросов на внешнюю маршрутизацию.

**Формат:**

```
local_route_port=<число>
```

где <число> – целое, положительное, обозначающее номер порта

**Значение по умолчанию:**

1814

(2814 для секции [BackupRadius])

**Пример:**

```
local_route_port=1844
```

**Примечание:** Важно, чтобы значение порта, заданное здесь, отличалось от значения порта, заданного в одноименном поле в секции [BackupRadius]

Поле `repeat_time`

Периодичность повторения отсылаемых на RADIUS-сервер пакетов в секундах.

**Формат:**

```
repeat_time=<число>
```

где <число> – целое, положительное, в диапазоне от 1 до 9

**Значение по умолчанию:**

3

**Пример:**

```
repeat_time=8
```

Поле `repeat_quantity`

Количество попыток отправки отсылаемых на RADIUS-сервер пакетов.

**Формат:**

```
repeat_quantity=<число>
```

где <число> – целое, положительное, в диапазоне от 1 до 9

**Значение по умолчанию:**

5

**Пример:**

```
repeat_quantity=7
```

**Поле** `route_address`

Адрес, на который следует отсылать запросы по внешней маршрутизации.

**Формат:**

```
acct_address = oct.oct.oct.oct
```

где oct.oct.oct.oct - IP адрес сервера внешней маршрутизации.

**Пример:**

```
route_address=195.88.97.20
```

**Поле** `route_enable`

Флаг включения (выключения) режима внешней маршрутизации с адреса отличного от адреса, с которого осуществляется авторизация вызовов.

**Формат:**

```
route_enable=<число>
```

где <число> – 0 или 1. 1 режим внешней маршрутизации включить, 0 – отключить внешнюю маршрутизацию.

**Значение по умолчанию:**

0

**Пример:**

```
route_enable=1
```

**Поле** `route_port`

Номер порта в дополнение к значению поля `route_address`.

**Формат:**

```
route_port=<число>
```

где <число> – целое положительное значение, обозначающее номер порта

**Значение по умолчанию:**

Значение порта из поля `auth_port`

**Пример:**

```
route_port=1798
```

**Поле `suppress_Q931_cause`**

Флаг включения передачи на RADIUS-сервер значения 16 в качестве Q931-кода завершения звонка, при котором продолжительность разговора ненулевая. Используется в случае, когда система учета и начисления платы не воспринимает отличные от 16 коды Q931 завершения звонков нормальной (ненулевой) продолжительности. Порт для приема ответных сообщений RADIUS сервера для целей начисления платы.

**Формат:**

```
suppress_Q931_cause=<число>
```

где число – 0 или 1

**Значение по умолчанию:**

0

**Пример:**

```
suppress_Q931_cause=1
```

**Поле `use_h323_ivr_in`**

Флаг управления подстановкой имени пользователя в поле `UserName` биллинговых пакетов, отправляемых на RADIUS сервер. Подставляемое имя – это имя из поля `h323_ivr_in` Cisco AV-pair(1) из ответа `AccessAccept`. Подставляемое имя используется только для данного направления и исключительно в пакетах биллингового обмена.

**Формат:**

```
use_h323_ivr_in=<число>
```

где <число> – 0 или 1

**Значение по умолчанию:**

0

**Пример:**

```
use_h323_ivr_in=1
```



**Примечание:** Данный параметр функционален только при значениях в поле `acct_leg_type` от 1 до 4

### Секция [BackupRadius]

Секция [BackupRadius] служит для дублирования отправки RADIUS пакетов на другой адрес, если не получено подтверждения о их получении от RADIUS сервера, расположенного по основному адресу (указанному в параметрах `auth_address` и `auth_port`). В отличие от основной секции [Radius], дублирующая секция содержит только следующие параметры:

```
acct_enable
acct_address
acct_port

auth_enable
auth_address
auth_port

local_auth_port
local_acct_port
local_route_port

route_enable
route_address
route_port
secret
repeat_time
repeat_quantity
```

значения и формат которых аналогичны параметрам основной секции.

**Примечание:** Важно, чтобы значения портов, задаваемые в полях `local_auth_port`, `local_acct_port` и `local_route_port`, отличались от значений портов, заданных в одноименных полях в секции [Radius]

### Секция [LAR]

Секция настроек функции перенаправления (Look\_ahead\_routing)

Поле `route_cause`

Причины завершения звонка, вызывающие продолжение попыток его маршрутизации.

**Формат:**

```
route_cause = <число>; [< число >]
```

где <число> – цифра(ы) в десятичном формате, выражающая(ие) код причин, по которым происходит возобновление попыток маршрутизации звонка. Элементы списка разделяются ';'. Для срабатывания функции по любой причине значение параметра `route_cause=0-127`

**Значение по умолчанию:**

Нет. Перенаправление звонка не задействовано

**Пример:**

```
route_cause=0-16;18-127
```

**ВВ!** В данном случае, если от terminating шлюза приходит `release_complete` с кодом 17 («номер занят»), то попытки дальнейшей маршрутизации звонка прекращаются

**Поле `validation_gap`**

Общее (для всех случаев) определяемое администратором, время задержки отправки пакетов на шлюз-инициатор вызова.

В процессе установления звонка все пакеты, полученные от terminating шлюза сохраняются в буфере MVTs, содержимое которого будет отправлено на шлюз-инициатор вызова либо по истечении периода задержки, установленного в поле `validation_gap` времени, либо по получении от terminating шлюза одного из сообщений, определенного в поле `validation_msg` (CallProceeding, Progress, Facility, Alerting).

(Подобная организация процесса установления вызова объясняется тем, что некоторые виды шлюзов допускают лишь однократное получение ими пакета CallProceeding, в результате чего, если после CallProceeding вдруг последует сигнал Release\_complete, дальнейший процесс установления звонка (перенаправление вызова) с таким шлюзом становится невозможен, так как он не будет воспринимать повторные сообщения CallProceeding)

**Формат:**

```
validation_gap=<число>
```

где <число> – время задержки в миллисекундах до начала отправки на шлюз-инициатор звонка пакетов, полученных от terminating шлюза.

**Значение по умолчанию:**

0

**Пример:**

```
validation_gap=3000
```

**Секция [proxy]**

Секция настроек функциональности proxy

**Поле `read_route_table`**

Флаг принимающий значения 0 и 1. При установке в значение 0 блокирует чтение таблицы маршрутизации ядра ОС. При значении 1 чтение таблицы разблокируется. Если чтение таблицы заблокировано, то маршрутизация должна вводиться вручную в поле `alias_route_path` в

этой же секции..

**Формат:**

```
read_route_table=<число>
```

где <число> – цифра 0 или 1.

**Значение по умолчанию:**

0

**Пример:**

```
read_route_table = 0
```

**Поле** `alias_route_path`

Поле служит для ручного ввода альтернативной схемы маршрутизации.

**Формат:**

```
alias_route_path=<path>;<path>;<path>
```

где <path> означает альтернативную схему маршрутизации введенную в следующем формате: *адрес сети/маска сети/собственный IP-адрес*, последовательность таких блоков, разделяется ';'

**Значение по умолчанию:**

НЕТ

**Пример:**

```
alias_route_path=1.1.1.1/255.255.255.255/123.345.678.12;1.2.1.1/255.255.255.255/211.123.111.4
```

**Поле** `local_ip_manager`

Данный параметр предназначен для задания максимальной ширины полосы пропускания на каждом локальном IP-адресе. Расчет нагрузки на адрес ведется приблизительно раз в 30 секунд на основе данных от системы проксирования. При значении 0 (значение по умолчанию) ограничения на максимальную ширину полосы пропускания на трафик не накладываются.

**Формат:**

```
local_ip_manager=<ip_address>/<RxMax>/<TxMax>/TcpPingAddress:Port
```

где <ip\_address> – локальный IP адрес, <RxMax> – максимальная ширина полосы пропускания, задаваемая для приема трафика (Кбит/сек), <TxMax> – максимальная ширина полосы пропускания, задаваемая для отправки трафика (Кбит/сек), **TcpPingAddress:Port** адрес и порт, используемые для проверки успешности TCP-соединения на данный адрес. Проверка выполняется каждые 30 секунд.. Последний аргумент может быть опущен, в таком случае проверка соединения выполняться не будет. В одном поле разрешено указывать нескольких адресов через точку с запятой.

**Значение по умолчанию:**

0

**Пример:**

```
local_ip_manager=192.168.1.1/120/120;192.168.1.2/64/64;192.168.1.3/512/256
```

**Поле `local_ip_manager_mode`**

Данный параметр помогает определить поведение системы в ситуациях, когда все локальные адреса, указанные в поле `local_address`, оказываются перегруженными.

**Формат:**

```
local_ip_manager_mode=<число>
```

где <число> – 0, 1, 2, или 3 при следующих значениях

0 – использовать адрес по умолчанию;

1 – завершить звонок;

2 – использовать наименее загруженный адрес;

3 – баланс в процентах от максимальной загрузки адреса;

**Значение по умолчанию:**

0

**Пример:**

```
local_ip_manager_mode=3
```

**Секция [SNMP]****Поле `local_port`**

... задает локальный порт для работы по протоколу SNMP.

**Формат:**

```
local_port=<число>
```

где <число> – положительное целое, обозначающее номер порта

**Значение по умолчанию:**

161

**Пример:**

```
local_port=161
```

**Поле** `contact_info`

Контактная информация о владельце или операторе системы..

**Формат:**

```
contact_info=<строка> <строка>
```

где <строка> <строка> – последовательность буквенно-цифровых строк, составляющих «текст» контактной информации.

**Пример:**

```
contact_info=esquire@mear.com +78213995875
```

**Поле** `system_name`

Имя системы.

**Формат:**

```
system_name=<строка> <строка>
```

где <строка> <строка> – произвольная последовательность буквенно-цифровых строк, обозначающих название (имя) системы.

**Пример:**

```
system_name=MAIN SILONET GK/PROXY
```

**Поле** `system_location`

... информация о местоположении системы.

**Формат:**

```
system_location=<строка> <строка>
```

где <строка> <строка> – произвольная последовательность буквенно-цифровых строк, обозначающих название (имя) системы.

**Пример:**

```
system_location=SEATTLE OFFICE, 5-th ALLEY
```

**Поле** `community`

... задает имя допустимого SNMP сообщества (SNMP community) для окружения доступа (access environment).

**Формат:**

```
community=<строка>
```

где <строка> – символьная строка задающая имя «сообщества» (SNMP community), включаемое в SNMP запросы. Устройства «не знающие» заданного имени сообщества из оперирования с SNMP исключаются.

**Пример:**

```
community=transCity
```

**Поле** trap\_community

... задает имя сообщества получателей trap сообщения.

**Формат:**

```
trap_community=<строка>
```

где <строка> – символьная строка, задающая имя сообщества.

**Пример:**

```
trap_community=municipal
```

**Поле** trap\_address

... задает IP адрес для отсылки trap сообщений.

**Формат:**

```
trap_address=<адрес>
```

где <адрес> – 32-битный IP адрес

**Пример:**

```
trap_address=194.23.122.14
```

**Поле** trap\_port

... назначает порт для отправки trap сообщений.

**Формат:**

```
trap_port=<number>
```

где <number> – положительное целое обозначающее номер порта.

**Значение по умолчанию:**

0

**Пример:**

```
trap_port=162
```

**Поле** `trap_level`

...задает уровень «значимости» отправляемых trap сообщений.

**Формат:**

```
trap_level=<число>
```

где <число> – 0, 1, 2, 3, 4 или 5

- 0 – trap сообщения не отсылаются
- 1 – критические ошибки (critical errors)
- 2 – некритические ошибки (non-critical errors)
- 3 – предупреждения (warnings)
- 4 – информация (information)
- 5 – уведомления (notification)

**Значение по умолчанию:**

0

**Пример:**

```
trap_level=1
```

**Поле** `trusted_address`

список “надежных” IP-адресов, с которых могут приниматься SNMP-запросы. Элементы списка должны разделяться ‘;’.

**Формат:**

```
trusted_address=<адрес>
```

где <адрес> – 32-битный IP адрес

**Пример:**

```
trusted_address=126.44.23.15
```

**Поле** `trap_email`

В данном поле указывается адрес эл. почты для сводок с trap-сообщениями.

**Формат:**

```
trap_email=<строка>
```

где <строка> – символьная строка адреса эл. почты, помещаемая в переменную \$EMAIL\$

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
trap_email=admin@noztel.com
```

**Поле** `trap_email_cmd`

В поле указывается команда (скрипт), отвечающая за отправку эл. почты.

**Формат:**

```
trap_email_cmd=<строка>
```

где <строка> - команда с параметрами или имя скрипта для отправки эл. почты.

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
trap_email_cmd=((printf "From: mvts@\nTo: $EMAIL$\nSubject:
$SUBJECT$\n\n";cat $FILE$) | /usr/sbin/sendmail -B8BITMIME -f mvts@ -
F$FROM$ $EMAIL$)
```

**Поле** `trap_email_subject`

Данное поле используется для указания содержимого строки “Тема:” (Subject) сообщения, отправляемого эл. почтой.

**Формат:**

```
trap_email_subject=<строка>
```

где <строка> – произвольная последовательность символов (включая пробелы), передающая содержание строки “Тема:” (Subject) и помещаемая в переменную \$SUBJECT\$.

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
trap_email_subject=Routine trap report
```

**Поле** `trap_email_from`

Поле для указания содержимого строки «От кого» (“From”) эл. сообщения.

**Формат:**

```
trap_email_from=<строка>
```



где <строка> – произвольная последовательность символов, допускающая пробелы и помещаемая в переменную \$FROM\$.

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
trap_email_from=MVTS server No.3
```

**Поле** `trap_email_period`

...используется для задания периодичности отправки сообщений эл. почтой.

**Формат:**

```
trap_email_period=<[[<hours>:] minutes:] seconds>
```

где <[[<hours>:] minutes:] seconds> периодичность отправки.

**Значение по умолчанию:**

00:00:00

**Пример:**

```
trap_email_period=02:00:00
```

**Поле** `trap_enable`

Флаг включения/выключения режима отправки trap-сообщений.

**Формат:**

```
trap_enable=<число>
```

где <число> – 0 или 1. 1 отправка сообщений включена, 0 – отправка сообщений выключена.

**Значение по умолчанию:**

0

**Пример:**

```
trap_enable=1
```

**Поле** `snmp_enable`

Флаг включение/выключение режима ответа на SNMP-запросы

**Формат:**

```
snmp_enable=<число>
```

где <число> – 0 или 1. 1 включает ответ на SNMP-запросы, 0 выключает SNMP обмен.

**Значение по умолчанию:**

0

**Пример:**

```
snmp_enable=1
```

**Note:** Список идентификаторов поддерживаемых объектов MIB см. в Приложении С

**Секция [Statistics]**

Данная секция содержит параметры управления статистикой.

**Поле** `statistics_enable`

Флаг включения/отключения подсчета статистики.

**Формат:**

```
statistics_enable=<число>
```

где <число> – 0 или 1.

0 – подсчет статистики отключен

1 – подсчет статистики включен

**Значение по умолчанию:**

1

**Пример:**

```
statistics_enable=1
```

**Поле** `call_radix`

количество звонков, используемое для анализа успешности данного направления (маршрута).

**Формат:**

```
call_radix=<число>
```

где <число> – положительное целое в диапазоне от 10 до 500.

**Значение по умолчанию:**

500

**Пример:**

```
call_radix=200
```

**Поле** `call_sampling_number`

Количество звонков, используемое для анализа успешности временно разблокированного направления (маршрута).

**Формат:**

```
call_sampling_number=<число>
```

где <число> – целое положительное в диапазоне от 1 до значения, заданного в поле `call_radix`.

**Значение по умолчанию:**

50

**Пример:**

```
call_sampling_number=70
```

**Поле** `file`

Путь и имя файла, в который ведется запись статистики (без учета метки времени и флага текущего файла).

**Формат:**

```
file=<имя>
```

где <имя> - строка из букв 'A'...'Z', 'a'...'z' и символов подчеркивания.

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
file=../debug/stat
```

**Поле** `period`

Максимальное время записи в один файл, по истечении которого файл закрывается и открывается новый файл записи.

**Формат:**

```
period = [[<часы >:] минуты:] секунды
```

**Значение по умолчанию:**

00:00

**Пример:**

period=120:30

**Поле** `reset_after_save`

Флаг необходимости обнуления статистики после записи в файл.

**Формат:**`reset_after_save=<значение>`

где <значение> 0 или 1. Выставление флага (`reset_after_save=1`) означает, что после записи в файл текущая статистика будет обнуляться. При сбросе флага (`reset_after_save=0`) статистика сохраняется даже после записи в файл.

**Значение по умолчанию:**

0

**Пример:**`reset_after_save=1`**Поле** `route_statistics_enable`

Флаг включения/выключения подсчета статистики по маршрутам.

**Формат:**`route_statistics_enable=<число>`

где <число> 0 или 1. 1 – подсчет статистики включить, 0 – подсчет статистики выключить.

**Значение по умолчанию:**

0

**Пример:**`route_statistics_enable=1`**Поле** `set_route_asr`

... задает минимальное допустимое значение ASR (средний показатель успешности звонков) для направлений (маршрутов).

**Формат:**

```
set_route_asr=<число>
```

где <число> – целое положительное значение в диапазоне от 1 до 100.

**Значение по умолчанию:**

```
20
```

**Пример:**

```
set_route_asr=15
```

**Поле** `time_span`

`time_span` – конфигурационный параметр, при помощи которого задается максимальное время отсутствия обновлений статистики, после которого имеющаяся статистика оп данному объекту (шлюзу или объекту набора) обнуляется.

**Формат:**

```
time_span=[[<hours>:] minutes:] seconds
```

**Значение по умолчанию:**

```
00:00:00
```

**Пример:**

```
time_span=120:00
```

**Поле** `no_empty`

Флаг включения/выключения записи «пустых» файлов статистики. При установке флага в значение 1 прокси-сервер не оставляет пустых файлов статистики.

**Формат:**

```
no_empty = число (1 или 0)
```

**Значение по умолчанию:**

```
0
```

**Пример:**

```
no_empty=1
```

**Поле** `asr_alert_email`

Поле для адреса электронной почты, на который система отправляет уведомления администратору о снижении ASR (показателя числа

успешных звонков).

**Формат:**

```
asr_alert_email=<строка>
```

где <строка> адрес электронной почты администратора системы.

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
asr_alert_email=admin@noztel.com
```

**Поле** `smart_routing`

Включение/выключение функции интеллектуальной маршрутизации по данным статистики.

**Формат:**

```
smart_routing=<число>
```

где <число> – 0 или 1. 0 отключает функцию интеллектуальной маршрутизации (smart routing).

**Значение по умолчанию:**

0

**Пример:**

```
smart_routing=1
```

**Поле** `stat_email`

Поле для адреса электронной почты получателя отчетов со статистикой по звонкам.

**Формат:**

```
stat_email=<строка>
```

где <строка> - символьная строка с адресом электронной почты, помещаемым в переменную \$EMAIL\$.

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
stat_email=admin@noztel.com
```

**Поле** `stat_email_cmd`

В поле указывается команда (скрипт), отвечающая за отправку эл. почты.

**Формат:**

```
stat_email_cmd=<строка>
```

где <строка> - команда с параметрами или имя скрипта для отправки эл. почты.

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
stat_email_cmd=((printf "From: mvts@\nTo: $EMAIL$\nSubject:
$SUBJECT$\n\n";cat $FILE$) | /usr/sbin/sendmail -B8BITMIME -f mvts@ -
F$FROM$ $EMAIL$)
```

**Поле** `stat_email_from`

Поле для указания содержимого строки «От кого» (“From”) эл. сообщения.

**Формат:**

```
stat_email_from=<строка>
```

где <строка> – произвольная последовательность символов, допускающая пробелы и помещаемая в переменную \$FROM\$.

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
stat_email_from=MVTS Server Statistics Manager
```

**Поле** `stat_email_subject`

Данное поле используется для указания содержимого строки “Тема:” (Subject) сообщения, отправляемого эл. почтой.

**Формат:**

```
stat_email_subject=<строка>
```

где <строка> – произвольная последовательность символов (включая пробелы), передающая содержание строки “Тема:” (Subject) и помещаемая в переменную \$SUBJECT\$.

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
stat_email_subject=Routine call statistics
```

**Поле** `stat_renewal_time`

интервал времени в минутах, через который осуществляется очистка статистики от данных, подлежащих удалению (например, данных по шлюзам, которых уже нет в конфигурации MVTs).

**Формат:**

```
stat_renewal_time=<число>
```

где <число> – положительно целое значение в минутах в диапазоне от 0 до 525599 (т.е. 1 год). При значении 0 функция отключается.

**Значение по умолчанию:**

720 (12 часов)

**Пример:**

```
stat_renewal_time=1440
```

**Поле** `suspend_time`

... задает длительность периода временного блокирования направления (маршрута) с низким значением среднего показателя успешности вызовов (ASR) в минутах. Диапазон значений от 1 до 525599 (т.е. один год).

**Формат:**

```
suspend_time=<число>
```

где <число> положительное целое, определяющее длительность периода блокирования направления (в минутах).

**Значение по умолчанию:**

30

**Пример:**

```
suspend_time=50
```



#### 6.1.4. СТРУКТУРА ФАЙЛА ДАННЫХ RAS-ПОЛЬЗОВАТЕЛЕЙ

Файл данных RAS-пользователей (динамически регистрируемых пользователей) представляет собой текстовый файл с именем **user.cfg** или любым другим, назначенным администратором (см. главу 3.1.3). Администратор может вносить изменения в файл данных с помощью любого текстового редактора.

**Пример:**

```
[default]
user=default
ip_precedence=3
group=RAS
proxy_type=1
codec_allow=g7231;g711U64k
validation_gap=2000

[ata1]
user=ata
password=test
number=78312778790
ip_precedence=3
group=RAS
reg_address=255.125.125.046
capacity=40
compatibility=0x208
datacap_deny=2048
ani_allow=78312[0-9]*;7812[0-9]*
acct_enable=0
auth_enable=0
nat_rtp=1
fake_rtcp=0
force_alerting=0
force_call_proceeding=4000
gateway_type=1
proxy_type=1
route_cause=0-16;18-127
radius_remote_address=0
radius_uses_callid=0
rbt_timeout=
dst_translate=
src_translate=
in_dst_translate=77|.|.|32/774\332
in_src_translate=
number=78312775825
numbering_plan=33
dialpeer=ATA1
codec_allow=g7231;g711U64k
codec_deny=g729
validation_gap=3500
validation_msg=7
```

**Примечание:** При обновлении файла данных администратору рекомендуется сначала скопировать файл данных во временный файл,

---

*отредактировать его и перекопировать обратно.*

---

В файле данных динамически регистрируемых пользователей вся информация об отдельном пользователе хранится в виде секции. Имя пользователя (псевдоним) назначается администратором и используется MVTS только для внутренней идентификации пользователя и связи файла данных пользователей с планом набора (**dialpeer.cfg**). Псевдоним служит названием секции. Пример псевдонима пользователя:

[ **ata1** ]

Существует возможность авторизации пользователя только при помощи RADIUS, для чего в файле данных RAS-пользователей создается секция [default]. В таком случае в секции имеется поле `user` со значением `user=default` (см. пример выше), и прокси-сервер регистрирует всех пользователей с авторизацией через RADIUS (если такой сервис задействован; см. значение поля `auth_enable=` в секции [Radius] файла **meraproxy.cfg**). Для всех динамически регистрируемых пользователей, записи о которых не обнаруживаются в файле, будут использоваться параметры, заданные в секции [default].

При авторизации RAS-пользователей через секцию с `user=default` на RADIUS-сервере, RADIUS-сервер может вернуть в пакете AccessAccept поле Cisco AV-Pair(1) `xrpgk-ep-number=<список номеров>`. Благодаря такому механизму теперь имеется возможность направлять вызовы на таких RAS-пользователей, а не только принимать от них звонки.

Например, если в ответ на AccessRequest при авторизации RAS-пользователя будет получен AccessAccept с полем `xrpgk-ep-number=78312778708;78312788865`, то в последующем при приходе звонка с конечным номером 78312778708 или 78312788865 при попадании в объект набора, содержащий параметр `gateway=ENDPOINTS`, будет предпринята попытка терминировать звонок на этого RAS-пользователя.

## Поле `user`

Имя пользователя в системе. Используется для идентификации пользователя при регистрации его на привратнике MVTS и при запросах к системе удаленной аутентификации RADIUS.

Если поле `user` присутствует в секции [default]

### Формат:

`user = <строка>`

где <строка> – строка символов, назначенная администратором, в качестве внутреннего имени пользователя

### Пример:

`user = anyone`

**Поле password**

Пароль пользователя. Поле обязательно для заполнения. Используется системой совместно с внутренним именем пользователя для тех же целей.

**Формат:**

```
password = <строка символов >
```

где <строка символов> – пароль пользователя, подчиняющийся общепринятым правилам использования паролей в ОС. Это поле также воспринимает в качестве параметра метасимвол '\*'. В конфигурации `password=* пароль пользователя будет извлечен из запроса на регистрацию и без изменений направлен на RADIUS-сервер. Таким образом, можно не указывать пароли для RAS-пользователей в файле user.cfg. Этот режим работы действителен для регистраций типа <user>|<passwd>.`

**Пример:**

```
password = Key0598E  
password=*
```

**Поле ip\_precedence**

Содержит приоритет IP-пакетов, проходящих через шлюз. Необязательное поле.

**Формат:**

```
ip_precedence = <число>
```

где <число> – приоритет пакетов [0..7]

**Значение по умолчанию:**

3 (эквивалентно операционной системе)

**Пример:**

```
ip_precedence = 2
```

**Поле group**

Название группы к которой принадлежит шлюз. Необязательное поле.

**Формат:**

```
group = <строка>
```

где <строка> - имя группы.

**Значение по умолчанию:**

RAS

**Пример:**

```
group = GROUP1
```

**Поле** `h323_gw_id`

Задаёт значение Source H323 Alias, параметр, который иногда необходимо задавать для удаленного шлюза. Имеет смысл только для исходящих шлюзов.

**Формат:**

```
h323_gw_id=<строка символов>
```

где < строка символов> – произвольный символьный идентификатор.

**Значение по умолчанию:**

Нет.

**Пример:**

```
h323_gw_id=DistantGate
```

**Поле** `reg_address`

IP адрес RAS-пользователя.

**Формат:**

```
reg_address = oct.oct.oct...
```

где oct.oct.oct.oct - IP адрес

**Пример:**

```
reg_address = 195.128.97.20
```

**Поле** `capacity`

Максимальное количество звонков, которое MVTs может установить с участием данного шлюза.

**Формат:**

```
capacity = число
```

где число – целое, неотрицательное.

**Значение по умолчанию:**

```
65535
```

**Пример:**

```
capacity = 30
```

**Поле** `compatibility`

4-х байтная маска характеристик шлюза. Сопоставление значений этого параметра у оригиналирующего и терминирующего шлюзов позволяет

выбирать маршрут терминции вызова. Сравнение выполнется побайтно по логическому И. Если логическое И хотя бы по одной паре байтов окажется ненулевым, шлюз может выбираться для терминции вызова. В данном поле администратор задает параметры совместимости, произвольно назначая в качестве битов либо кодеки, протоколы (T120 или T38) либо иные характеристики, важные для совместной работы пары шлюзов.

Например, определим следующие соответствия в младшем байте:

0x01 – G711 A-law

0x02 – G711 U-law

0x04 – G729

0x08 – G723

а во втором байте:

0x01 - T120

0x02 - T38

0x40 – прочие типы совместимости по протоколам

#### Формат:

compatibility=<число>

где <число> – шестнадцатиричное целое типа **unsigned int**, отражающее характеристики совместимости.

#### Значение по умолчанию:

0xFFFFFFFF

#### Пример 1:

```
[originator_GW]
... ..
compatibility=0x0105 (G711 A-law, G729, T120)
... ..
```

```
[terminating_GW]
... ..
compatibility=0x30C (G723, G729, T120, T38)
... ..
```

0x0105 AND 0x30C=0x0104

таким образом, terminating\_GW может использоваться для терминции вызова с originator\_GW.

#### Пример 2:

```
[originator_GW]
... ..
compatibility=0x0105 (G711 A-law, G729, T120)
```

... ..

```
[terminating_GW]
... ..
compatibility=0x208 (G711 A-law, T38)
... ..
```

0x0105 AND 0x208=0x0000

таким образом, terminating\_GW не будет выбран для терминирования вызова с originator\_GW.

### Поле alert\_prog\_ind\_orig

... задает значение ProgressIndicator в сообщениях Alert и Progress. Значение поля ProgressIndicator, заданное в данном параметре будет безусловно подставлено в вышеуказанные сообщения в пакетах направляемых от терминирующей стороны орIGINАТОРУ. Если в пакетах от терминирующей стороны поле ProgressIndicator отсутствует, MVTС его вставит в заданным значением.

#### Формат:

alert\_prog\_ind\_orig=<число>

где <число> – 0, 1, 2, 3, 4, 5, или 8. При значении 0 функция отключается.

#### Значение по умолчанию:

0

#### Пример:

alert\_prog\_ind\_orig=2

### Поле alert\_prog\_ind\_term

... задает значение ProgressIndicator в сообщениях Alert и Progress. MVTС подставит отсутствующее поле ProgressIndicator с заданным значением в пакеты, направляемые терминирующей стороне от орIGINАТОРА. Если такое поле присутствует в пакетах, оно будет оставлено без изменений.

#### Формат:

alert\_prog\_ind\_term=<число>

где <число> – 0, 1, 2, 3, 4, 5, или 8. При значении 0 функция отключается.

#### Значение по умолчанию:

0

#### Пример:

alert\_prog\_ind\_term=2

**Поле** `ani_allow`

Список разрешенных номеров АОН для данного входящего шлюза, т.е. звонок с этого шлюза будет пропущен, если АОН-номер, который он предъявил подходит по маске `ani_allow`. Для разрешения любых АОН это поле надо не указывать. Проверка выполняется по `src_number`, который прислал шлюз без каких либо предварительных трансляций значения `ani_allow`.

**Формат:**

```
ani_allow = <маска номера>
```

где <маска номера> - регулярное выражение, задающее маску телефонных номеров.

**Пример:**

```
ani_allow=78312[0-9]*;7812[0-9]*
```

**Поле** `acct_enable`

Поле задействия RADIUS сервиса по учету начисленной платы. Обязательное для заполнения поле. 1 – использовать RADIUS сервис по начислению платы, 0 – отключить RADIUS сервис по начислению платы. (Окончательный результат задействия RADIUS сервиса применительно к данному RAS-пользователю получается перемножением значения этого флага на значение такого же, но общего параметра в секции [Radius] системного конфигурационного файла **meraproxy.cfg**)

**Формат:**

```
acct_enable = <число>
```

где <число> - 0 или 1

**Пример:**

```
acct_enable=0
```

**Поле** `auth_enable`

Поле задействия аутентификации через RADIUS. Обязательное для заполнения поле. 1 – аутентификацию через RADIUS включить, 0 – аутентификацию через RADIUS отключить. (Окончательный результат задействия RADIUS сервиса применительно к данному RAS-пользователю получается перемножением значения этого флага на значение такого же, но общего параметра в секции [Radius] системного конфигурационного файла **meraproxy.cfg**)

**Формат:**

```
auth_enable = <число>
```

где <число> - 0 или 1

**Пример:**

```
auth_enable=0
```

**Поле `local_address`**

...служит для задания списка локальных адресов для входящего и исходящего трафика. Система всегда выбирает первый наименее загруженный IP адрес. Если все локальные адреса из списка оказываются перегруженными выбор осуществляется в соответствии со значением поля `local_ip_manager_mode` в секции `[проху]` конфигурационного файла **meraproxy.cfg**. Если поле `local_address` не задано, то локальный адрес будет выбираться, исходя из локальной таблицы маршрутизации. Для входящего трафика, если поле `local_address` не задано, будет использоваться адрес, на который пришел звонок, если же поле определено, из списка будет выбран первый неперегруженный на данный момент адрес, который и будет использоваться для H245 сессии и для приема/передачи RTP и RTCP пакетов. При регистрации RAS пользователя, локальный адрес, на который пришла регистрация, автоматически добавляется в конец списка адресов, определенных в поле `local_address` для данного RAS-шлюза.

**Формат:**

```
local_address=<адрес>
```

где <адрес> – IP адрес, заданный в виде oct.oct.oct.oct (oct=0...255)

**Пример:**

```
local_address=195.88.57.123
```

**Поле `min_asr`**

Служит для задания нижнего предела ASR (усредненного показателя количества успешны вызовов), достижение которого будет вызывать отправку уведомлений администратору по эл. почте.

**Формат:**

```
min_asr=<число>
```

где <число> - целое положительное значение в пределах от 0 до 100, означающее мин. допустимый уровень ASR

**Значение по умолчанию:**

0

**Пример:**



```
min_asr=50
```

**Поле** `nat_rtp`

Флаг направления RTP пакетов. Используется при нахождении шлюза за устройством трансляции адресов (NAT). Необязательное поле.

**Формат:**

```
nat_rtp = <number>
```

где <number> - 0 или 1. При значении 1 прокси будет посылать RTP пакеты не на адреса, сообщенные шлюзом в H.245 сессии, а на адреса, с которых RTP пакеты поступают реально.

**Значение по умолчанию:**

0

**Пример:**

```
nat_rtp = 1
```

**Поле** `no_port_clean`

Флаг, используемый для управления процессом освобождения портов. Этот параметр применяется когда шлюз, несмотря на изменения в режиме работы (например при переключении на отправку факса), продолжает отправку RTP-пакетов на прежние порты, которые по стандарту H.323 должны были быть высвобождены. (Такая особенность функционирования, например, наблюдалась в работе шлюзов MediaRing)

**Формат:**

```
no_port_clean = <число>
```

где <число> 0 или 1. Когда флаг установлен (значение `no_port_clean=1`) процесс высвобождения портов блокируется, при сброшенном флаге (0) высвобождение портов возобновляется.

**Значение по умолчанию:**

0

**Пример:**

```
no_port_clean=1
```

**Поле** `fake_rtcp`

Флаг генерации RTCP пакетов. Используется в случае, когда шлюз не поддерживает отсылку RTCP. Необязательное поле. Используется для совместимости с VocalTec Gateway. (Необходимо отметить, что использование этой функции может мешать прохождению факсов.)

**Формат:**

```
fake_rtcp = <number>
```

где <number> - 0 или 1. При значении 1 прокси будет посылать RTCP пакеты вместо шлюза.

**Значение по умолчанию:**

0

**Пример:**

```
fake_rtcp = 1
```

**Поле** `force_alerting`

Параметр, определяющий интервал времени в миллисекундах, по прошествии которого на оригиналирующий шлюз будет отослан пакет Alerting Message с нейтральным содержанием.

**Формат:**

```
force_alerting=<number>
```

где <number> - 0 или любое положительное целое. При значении 0 отсылка пакета не происходит.

**Значение по умолчанию:**

0

**Пример:**

```
force_alerting=4000
```

**Примечание:** Это поле используется, если шлюз некорректно функционирует в отсутствии пакета Alerting Message. Например, VocalTec 2000 (как, впрочем и некоторые модификации шлюзов Cisco) без этого сообщения часто передают звук только в одну сторону или не передают его вовсе.

**Поле** `force_call_proceeding`

Параметр, определяющий интервал времени в миллисекундах, по прошествии которого на оригиналирующий шлюз будет отослан пакет CallProceeding Message с нейтральным содержанием, если до того времени MVTs не получит ни одного пакета от терминирующего шлюза в ответ на отправку ему Setup Message.

**Формат:**

```
force_call_proceeding=<number>
```

где <number> - 0 или любое положительное целое. При значении 0 отсылка пакета не происходит.

**Значение по умолчанию:**

0

**Пример:**

```
force_call_proceeding=4000
```

**Примечание:** Это поле используется, когда задержка отправки сообщения `CallProceeding Message` терминирующим звонком шлюзом не соответствует стандарту H.323 (4 секунды), что приводит к завершению звонка, который при других условиях мог бы состояться (подобная ситуация, в частности, может возникать при работе шлюзов SMG с операторами из Зап. Европы)

**Поле** `gateway_type`

Тип шлюза. Необязательное поле.

**Формат:**

```
gateway_type = <number>
```

где `<number>` - 0, 1, 2, 3 или 4. 0 – тип шлюза неизвестен. 1 – CISCO, 2 – VocalTec, 3 – Samsung, 4 – Media Ring.

**Значение по умолчанию:**

0

**Пример:**

```
gateway_type=1
```

**Поле** `proxy_type`

Необязательное поле. Содержит признак проксирования медиа трафика от данного шлюза (0 – проксировать только сигнальный трафик, 1 – проксировать весь трафик). Первый режим используется между двумя шлюзами, находящимися в одной сети, второй – между шлюзами, находящимися в разных сетях. Если у двух шлюзов разные режимы проксирования, используется более строгий – проксирование всего трафика. По умолчанию 1.

**Формат:**

```
proxy_type = <flag>
```

где `<flag>` – признак проксирования [0,1]

**Значение по умолчанию:**

1

**Пример:**

```
proxy_type = 0
```

**Поле** `radius_gw_id`

Задаёт RADIUS VSA поля `h323-gw-id` и `h323-remote-id`, из исходного и терминирующего шлюзов соответственно. Если параметр `radius_gw_id` шлюза не задан, то `h323-gw-id` и `h323-remote-id` заполняются IP адресами соответствующих шлюзов.

**Формат:**

```
Radius_gw_id=<строка символов>
```

где < строка символов> – произвольный символьный идентификатор.

**Значение по умолчанию:**

Нет.

**Пример:**

```
radius_gw_id=DistantGate
```

**Поле** `radius_remote_address`

Поле используется для подстановки в сообщения, направляемые RADIUS-серверу IP-адрес привратника (gatekeeper) вместо IP-адреса удаленного шлюза. При значении 0 в сообщениях h323-remote-address и h323-remote-id указывается IP-адрес удаленного шлюза, при значении 1 - IP-адрес привратника (gatekeeper)..

**Формат:**

```
radius_remote_address=<число>
```

где < число> – 0 или 1.

**Значение по умолчанию:**

0

**Пример:**

```
radius_remote_address=1
```

**Поле** `radius_uses_callid`

Некоторые системы биллинга используют для идентификации звонка идентификатор конференции. Для решения проблем, которые могут возникать в случае использования отдельных видов шлюзов (шлюзы Vocaltec, например, могут генерировать одинаковый идентификатор конференции для нескольких звонков), введен параметр `radius_uses_callid`.

Флаг `radius_uses_callid` используется для замены в сообщениях Radius серверу идентификатора конференции (`confID`) на идентификатор звонка (`callID`). Флаг принимает значения 0 и 1. Когда флаг сброшен (0) в сообщениях `h323-incoming-conf-id` и `h323-conf-id` используется идентификатор конференции, если флаг установлен (1) и параметр `call ID` не пустой, то в этих сообщениях идентификатор конференции и идентификатор звонка меняются местами.

**Формат:**

```
radius_uses_callid=<number>
```

где <number> 0 или 1

**Значение по умолчанию:**

0

**Пример:**

```
radius_uses_callid=1
```

**Поле** `rbt_timeout`

В описании шлюза задает максимальное время ожидания пакетов Alert и/или ProgressIndicator от терминирующего шлюза (в сек.). Параметр предназначен для тонкой настройки режима работы. **Совместное использование с параметрами validation\_gap и validation\_msg требует внимания к деталям!**

Например, при значениях параметров `validation_gap=10000`, `validation_msg=7` (т.е. возобновление обмена между шлюзами по истечению 10 сек. или по приходу сообщения Connect) и с параметром `rbt_timeout=9` (9 сек.), в случае отсутствия сообщения Connect в течение 9 сек. после отправки Setup звонок будет завершен с причиной eRBTimeout.

**Формат:**

```
rbt_timeout=<число>
```

где <число> – положительное целое значение в секундах

**Значение по умолчанию:**

0, т.е. функциональность не задействована

**Пример:**

```
rbt_timeout=10
```

**Поле** `route_cause`

Причины завершения звонка, вызывающие продолжение попыток его маршрутизации.

**Формат:**

```
route_cause = <число>; [< число >]
```

где <число> – цифра(ы) в десятиричном формате, выражающие код причин, по которым происходит возобновление попыток маршрутизации звонка. Элементы списка разделяются ';'. Для срабатывания функции по любой причине значение параметра `route_cause=0-127`

**Значение по умолчанию:**

Нет. Перенаправление звонка не задействовано

**Пример:**

```
route_cause=0-16;18-127
```

**NB!** В данном случае, если от терминирующего шлюза приходит `release_complete` с кодом 17 («номер занят»), то попытки дальнейшей

---

маршрутизации звонка прекращаются

Поля `src_translate`, `dst_translate`

Содержат правила следующего за первичным преобразованием номера (followup translation) вызывающего (`src_translate`) или вызываемого (`dst_translate`) объекта. Как правило, состоит из двух частей, разделенных символом “/”

- выражение `regex`, задающее строку поиска
- строка замены

Строка поиска, для удобства, может быть разделена на секции символом “|”. В строке замены можно использовать подстроку, удовлетворяющую условию секции, указав номер секции после символа “\”. Для использования в строке замены подстроки, удовлетворяющей всем секциям одновременно служит символ “&”. Необязательные поля.

**Формат:**

```
src_translate =  
regex1[|regex2[|regex3]]/[str][\1][str][\2][str][\3][str][&]  
dst_translate =  
regex1[|regex2[|regex3]]/[str][\1][str][\2][str][\3][str][&]  
где regex - строка поиска в формате regex, str - строка символов
```

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
Расширение городского номера в полный интернациональный  
dst_translate = ...../78312&  
778899 -> 78312778899  
Изменение определенных цифр номера  
src_translate = 77|.|.|32/774\332  
778832 -> 774832
```

Поле `dialpeer`

При регистрации шлюза под этим именем, шлюз будет добавлен в файл плана набора (`dialpeer.cfg`) с этим именем в поле `gateway`, что обеспечивает возможность присваивать динамически регистрируемому шлюзу объект набора так же, как и статическому

**Формат:**

```
dialpeer = <строка>
```

где <строка> – строка символов

**Пример:**

```
dialpeer=ATA1
```

**Поле** `codec_allow`

Список кодеков, поддерживаемых шлюзом. Кодеки перечисляются через точку с запятой. Пробелы игнорируются. Регистр имеет значение.

**Формат:**

```
codec_allow = кодек1; кодек2; кодек3
```

где кодек# – мнемоническое обозначение кодека. Одно из

- g711A64k
- g711A56k
- g711U64k
- g711U56k
- g722\_64k
- g722\_56k
- g722\_48k
- g7231
- g728
- g729
- g729A
- is11172
- is13818
- g729B
- g729AwB
- g7231C
- gsmFR
- gsmHR
- gsmEFR
- NS:<нестандартный\_кодек>

**Значение по умолчанию:**

Пустое.

**Пример:**

```
codec_allow = g729; g7231; g711U56k; NS:g729ar8
```

---

**Примечание:** Значение по умолчанию подразумевает все кодеки, кроме указанных в `codec_deny`.

---

**Поле** `codec_deny`

Список кодеков, которые шлюз не поддерживает или интерпретирует нестандартным образом. Кодеки перечисляются через точку с запятой. Пробелы игнорируются. Регистр имеет значение.

**Формат:**

```
codec_deny = кодек1; кодек 2; кодек 3
```

где кодек# – мнемоническое обозначение кодека. Одно из значений, указанных в описании `codec_allow`.

**Значение по умолчанию:**

Пустое множество.

**Пример:**

```
codec_deny = g729; g7231; g711U56k
```

**Поле** `datacap_deny`

Перечень оборудования не поддерживаемого данным шлюзом. Отсутствие информации об оборудовании, не поддерживаемого шлюзом, может привести к отказу в установлении звонка.

**Формат:**

```
datacap_deny=<число>
```

где <число> целое десятичное значение представляющее двоичную маску поддерживаемого оборудования.

```
enum DataCapability {
    e_nonStandard=1,
    e_t120=2,
    e_dsm_cc=4,
    e_userData=8,
    e_t84=16,
    e_t434=32,
    e_h224=64,
    e_nlpid=128,
    e_dsxdControl=256,
    e_h222DataPartitioning=512,
    e_t30fax=1024,
    e_t140=2048,
    e_t38fax=4096,
    e_genericDataCapability=8192
};
```

**Значение по умолчанию:**

нет

**Пример:**

```
datacap_deny=2048
```

**Поле** `validation_gap`

Определенное для данного терминирующего объекта, время задержки отправки пакетов на шлюз-инициатор вызова.

В процессе установления звонка все пакеты, полученные от терминирующего шлюза сохраняются в буфере MVTs, содержимое которого будет отправлено на шлюз-инициатор вызова либо по



истечении периода задержки, установленного в поле `validation_gap` времени, либо по получении от терминирующего шлюза одного из сообщений, определенного в поле `validation_msg` (`CallProceeding`, `Progress`, `Facility`, `Alerting`).

(Подобная организация процесса установления вызова объясняется тем, что некоторые виды шлюзов допускают лишь однократное получение ими пакета `CallProceeding`, в результате чего, если после `CallProceeding` вдруг последует сигнал `Release_complete`, дальнейший процесс установления звонка (перенаправление вызова) с таким шлюзом-инициатором становится невозможным, так как он не будет воспринимать повторные сообщения `CallProceeding`).

Если для данного пользователя продолжительность периода задержки не определена, то используется общесистемное значение параметра, заданное в конфигурационном файле `meraproxy.cfg`

**Формат:**

```
validation_gap=<число>
```

где <число> – время задержки в миллисекундах до начала отправки на шлюз-инициатор звонка пакетов от терминирующего шлюза.

**Значение по умолчанию:**

0

**Пример:**

```
validation_gap=3500
```

**Поле `validation_msg`**

Определенный для данного терминирующего пользователя тип сообщения, до получения которого приостанавливается процесс обмена пакетами со стороной-инициатором вызова, (другим условием возобновления процесса отправки пакетов является истечение периода времени, определенного параметром `validation_gap`) Если вызываемая сторона успевает отказать в установлении соединения до выполнения обоих этих условий, звонок будет перенаправлен.

---

**Примечание:** *Перенаправление звонка окажется невозможным, если вызываемая сторона уже установила TCP-соединение для H.245.*

---

**Формат:**

```
validation_msg = <число>
```

где <число> – тип сообщения от вызываемой стороны,

**Значение по умолчанию:**

по умолчанию, Alerting=1, CallProcessing=2, Progress=3, Connect=7(обязательное), Facility=98. (для задействия значений по умолчанию достаточно установить любой не нулевой параметр validation\_gap)

**Пример:**

```
validation_msg=7
```

**Поле in\_converter**

Данное поле позволяет задать конвертер для входящего трафика, оригинатором которого является данный RAS объект.

**Формат:**

```
converter=<строка>
```

где <строка> – имя конвертера, заданное в поле converter секции [H323] в файле **meraproxy.cfg**

**Пример:**

```
converter=SIPconv
      or
converter=H323conv
```

**Поля in\_dst\_translate, in\_src\_translate**

Предварительная трансляция, выполняемая при поступлении звонка от этого шлюза, перед вызовом поиска по объектам набора (dial peers). Параметры полезны, когда какой-либо шлюз присылает номера не в формате E164.

Правила преобразования номера идентичны правилам, приведенным в описании полей src\_translate, dst\_translate.

**Формат:**

```
src_translate =
regexpr1[|regexp2[|regexp3]]/[str][\1][str][\2][str][\3][str][&]
dst_translate =
regexpr1[|regexp2[|regexp3]]/[str][\1][str][\2][str][\3][str][&]
где regexpr - строка поиска в формате regexpr, str - строка символов
```

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
Расширение городского номера в полный интернациональный
dst_translate = ...../78312&
778899 -> 78312778899
Изменение определенных цифр номера
src_translate = 77|.|.|32/774\332
778832 -> 774832
```

---

**Поле number**

Прямой телефонный номер шлюза. Регулярные выражения не используются. Применяется для поиска по прямым соответствиям в файле плана набора **dialpeer.cfg**, объекта набора ссылающегося на макроимя ENDPOINTS, например вида:

```
[DIRECTNUMBERS]
dst_pattern=[0-9]*
priority=1000
gateway=ENDPOINTS
```

**Формат:**

```
number = <строка>
```

где <строка> – набор цифровых символов, представляющих собой телефонный номер.

**Пример:**

```
number = 78312775825
```

**Поле numbering\_plan**

В десятичном формате задает значение параметра type:plan для обращения к шлюзу, например, National number + ISDN/telephony numbering plan см. [5] Необязательное поле.

**Формат:**

```
numbering_plan = <number>
```

где <number> -

**Значение по умолчанию:**

0

**Пример:**

```
numbering_plan = 33
```

### 6.1.5. СТРУКТУРА ФАЙЛА ДАННЫХ ШЛЮЗОВ

Файл данных шлюзов представляет собой текстовый файл **gateway.cfg** или назначенный администратором (см. главу 3.1.3). Администратор может вносить изменения в файл данных с помощью любого текстового редактора.

---

**Примечание:** При обновлении файла данных администратору рекомендуется сначала скопировать файл данных во временный файл, отредактировать его и перекопировать обратно.

---

В файле данных шлюзов вся информация об отдельном шлюзе хранится в виде секции. Названием секции служит внутренний псевдоним шлюза.

Пример записи о шлюзе:

```
[NNOV_KIS]
address=127.0.0.1
port=1720
capacity=30
ip_precedence=3
proxy_type=1
codec_allow=g711A64k; g729
codec_deny= g711U56k
datacap_deny=2048
compatibility=0x0105
group=GROUP1
acct_enable=1
auth_enable=1
nat_rtp=1
fake_rtcp=1
force_alerting=0
force_call_proceeding=4000
gatekeeper=nnov
gateway_type=2
gateway_mode=3
mask=255.255.255.0
rbt_timeout=10
route_cause=0-16;18-127
radius_remote_address=1
radius_uses_callid=0
dst_translate=
src_translate=
number=78123323232
in_dst_translate=77|.|.|32/774\332
in_src_translate=
validation_gap=3500
validation_msg=7
user=test
password=unknown
```

[ h323 ]      Название секции. Пример псевдонима шлюза

Поле `address`

IP адрес или DNS-имя шлюза. Поле обязательно для заполнения. В случае с входящим шлюзом, когда поле `address` содержит список значение параметра `mask` будет применяться ко всем элементам списка.

**Формат:**

`address` = адрес

где адрес – IP адрес, заданный в виде oct.oct.oct.oct (oct=0...255), или DNS имя хоста. **Примечание:** Если поле `address` содержит список адресов, общее количество элементов в списке не может превышать 127.

**Пример:**

```
address=196.88.57.12
address=computer.mera.ru
```

**Поле** `alert_prog_ind_orig`

... задает значение ProgressIndicator в сообщениях Alert и Progress. Значение поля ProgressIndicator, заданное в данном параметре будет безусловно подставлено в вышеуказанные сообщения в пакетах направляемых от терминирующей стороны оригинатору. Если в пакетах от терминирующей стороны поле ProgressIndicator отсутствует, MVTС его вставит в заданным значением.

**Формат:**

```
alert_prog_ind_orig=<число>
```

где <число> – 0, 1, 2, 3, 4, 5, или 8. При значении 0 функция отключается.

**Значение по умолчанию:**

0

**Пример:**

```
alert_prog_ind_orig=2
```

**Поле** `alert_prog_ind_term`

... задает значение ProgressIndicator в сообщениях Alert и Progress. MVTС подставит отсутствующее поле ProgressIndicator с заданным значением в пакеты, направляемые терминирующей стороне от оригинатора. Если такое поле присутствует в пакетах, оно будет оставлено без изменений.

**Формат:**

```
alert_prog_ind_term=<число>
```

где <число> – 0, 1, 2, 3, 4, 5, или 8. При значении 0 функция отключается.

**Значение по умолчанию:**

0

**Пример:**

```
alert_prog_ind_term=2
```

**Поле** `port`

Порт шлюза для приема сигнального трафика.

**Формат:**

```
port = число
```

где число – целое, положительное.

**Значение по умолчанию:**

1720

**Пример:**

port = 1780

Поле `capacity`

Максимальное количество звонков, которое MVTС может установить с участием данного шлюза.

**Формат:**`capacity = число`

где число – целое, неотрицательное.

**Значение по умолчанию:**

65535

**Пример:**`capacity = 30`Поле `ip_precedence`

Содержит приоритет IP-пакетов, проходящих через шлюз. Необязательное поле.

**Формат:**`ip_precedence = <number>`

где number – приоритет пакетов[0..7]

**Значение по умолчанию:**

3 (эквивалентно операционной системе)

**Пример:**`ip_precedence = 2`

---

**Примечание:** *Рекомендуется назначать `ip_precedence` выше 3 минимально возможному числу шлюзов, так как большое количество звонков с приоритетом больше 3 может вызвать нестабильность ядра операционной системы.*

---

Поле `codec_allow`

Список кодеков, поддерживаемых шлюзом. Кодеки перечисляются через точку с запятой. Пробелы игнорируются. Регистр имеет значение.

**Формат:**`codec_allow = кодек1; кодек 2; кодек 3`

где кодек – мнемоническое обозначение кодека. Одно из

- g711A64k
- g711A56k
- g711U64k

- g711U56k
- g722\_64k
- g722\_56k
- g722\_48k
- g7231
- g728
- g729
- g729A
- is11172
- is13818
- g729B
- g729AwB
- g7231C
- gsmFR
- gsmHR
- gsmEFR
- NS:<нестандартный\_кодек>

**Значение по умолчанию:**

Пустое.

**Пример:**

```
codec_allow = g729; g7231; g711U56k; NS:g729ar8
```

---

**Примечание:** Значение по умолчанию подразумевает все кодеки, кроме указанных в `codec_deny`.

---

**Поле** `codec_deny`

Список кодеков, которые шлюз не поддерживает или интерпретирует нестандартным образом. Кодеки перечисляются через точку с запятой. Пробелы игнорируются. Регистр имеет значение.

**Формат:**

```
codec_deny = кодек1; кодек 2; кодек 3
```

где кодек – мнемоническое обозначение кодека. Одно из значений, указанных в описании `codec_allow`.

**Значение по умолчанию:**

Пустое множество.

**Пример:**

```
codec_deny = g729; g7231; g711U56k
```

**Поле** `compatibility`

4-х байтная маска характеристик шлюза. Сопоставление значений этого параметра у originирующего и терминирующего шлюзов позволяет выбирать маршрут терминирования вызова. Сравнение выполняется побайтно по логическому И. Если логическое И хотя бы по одной паре байтов

окажется ненулевым, шлюз может выбираться для терминции вызова. В данном поле администратор задает параметры совместимости, произвольно назначая в качестве битов либо кодеки, протоколы (T120 или T38) либо иные характеристики, важные для совместной работы пары шлюзов.

Например, определим следующие соответствия в младшем байте:

0x01 – G711 A-law

0x02 – G711 U-law

0x04 – G729

0x08 – G723

а во втором байте:

0x01 - T120

0x02 - T38

0x40 – прочие типы совместимости по протоколам

**Формат:**

compatibility=<число>

где <число> – шестнадцатиричное целое типа **unsigned int**, отражающее характеристики совместимости.

**Значение по умолчанию:**

0xFFFFFFFF

**Пример 1:**

```
[originator_GW]
... ..
compatibility=0x0105 (G711 A-law, G729, T120)
... ..
```

```
[terminating_GW]
... ..
compatibility=0x30C (G723, G729, T120, T38)
... ..
```

0x0105 AND 0x30C=0x0104

таким образом, terminating\_GW может использоваться для терминции вызова с originator\_GW.

**Пример 2:**

```
[originator_GW]
... ..
compatibility=0x0105 (G711 A-law, G729, T120)
... ..
```

```
[terminating_GW]
```



```
... ..
compatibility=0x208 (G711 A-law, T38)
```

```
0x0105 AND 0x208=0x0000
```

таким образом, `terminating_GW` не будет выбран для терминции вызова с `originator_GW`.

### Поле `datacap_deny`

Перечень оборудования не поддерживаемого данным шлюзом. Отсутствие информации об оборудовании, не поддерживаемого шлюзом, может привести к отказу в установлении звонка.

#### Формат:

```
datacap_deny=<число>
```

где <число> целое десятичное значение представляющее двоичную маску поддерживаемого оборудования.

```
enum DataCapability {
    e_nonStandard=1,
    e_t120=2,
    e_dsm_cc=4,
    e_userData=8,
    e_t84=16,
    e_t434=32,
    e_h224=64,
    e_nlpid=128,
    e_dsxdControl=256,
    e_h222DataPartitioning=512,
    e_t30fax=1024,
    e_t140=2048,
    e_t38fax=4096,
    e_genericDataCapability=8192
};
```

#### Значение по умолчанию:

нет

#### Пример:

```
datacap_deny=2048
```

### Поле `group`

Название группы к которой принадлежит данный шлюз. Необязательное поле.

#### Формат:

```
group = <строка>
```

где <строка> - имя группы.

**Значение по умолчанию:**

DEFAULT

**Пример:**

```
group = GROUP1
```

**Поле** `acct_enable`

Флаг задействия RADIUS сервиса по учету начисленной платы. Обязательное для заполнения поле. 1 – использовать RADIUS сервис по начислению платы, 0 – отключить RADIUS сервис по начислению платы. (Окончательный результат задействия RADIUS сервиса применительно к данному шлюзу получается перемножением значения этого флага на значение подобного общего параметра в секции [Radius] системного конфигурационного файла **meraproxy.cfg**)

**Формат:**

```
acct_enable = <число>
```

где <число> - 0 или 1

**Пример:**

```
acct_enable=1
```

**Поле** `auth_enable`

Флаг задействия аутентификации через RADIUS. Обязательное для заполнения поле. 1 – аутентификацию через RADIUS включить, 0 – аутентификацию через RADIUS отключить. (Окончательный результат задействия RADIUS сервиса применительно к данному шлюзу получается перемножением значения этого флага на значение подобного общего параметра в секции [Radius] системного конфигурационного файла **meraproxy.cfg**)

**Формат:**

```
auth_enable = <число>
```

где <число> - 0 или 1

**Пример:**

```
auth_enable=1
```

**Поле** `nat_rtp`

Флаг направления RTP пакетов. Используется при нахождении шлюза за устройством трансляции адресов (NAT). Необязательное поле.

**Формат:**

```
nat_rtp = <number>
```

где <number> - 0 или 1. При значении 1 прокси будет посылать RTP пакеты не на адреса, сообщенные шлюзом в H.245 сессии, а на адреса с

которых RTP пакеты поступают реально.

**Значение по умолчанию:**

0

**Пример:**

```
nat_rtp = 1
```

**Поле `no_port_clean`**

Флаг, используемый для управления процессом освобождения портов. Этот параметр применяется когда шлюз, несмотря на изменения в режиме работы (например при переключении на отправку факса), продолжает отправку RTP-пакетов на прежние порты, которые по стандарту H.323 должны были быть высвобождены. (Такая особенность функционирования, например, наблюдалась в работе шлюзов MediaRing)

**Формат:**

```
no_port_clean = <число>
```

где <число> 0 или 1. Когда флаг установлен (значение `no_port_clean=1`) процесс высвобождения портов блокируется, при сброшенном флаге (0) высвобождение портов возобновляется.

**Значение по умолчанию:**

0

**Пример:**

```
no_port_clean=1
```

**Поле `fake_rtcp`**

Флаг генерации RTCP пакетов. Используется в случае, когда шлюз не поддерживает отсылку RTCP. Необязательное поле. Используется для совместимости с VocalTec Gateway.

**Формат:**

```
fake_rtcp = <number>
```

где <number> - 0 или 1. При значении 1 прокси будет посылать RTCP пакеты вместо шлюза.

**Значение по умолчанию:**

0

**Пример:**

```
fake_rtcp = 1
```

**Поле** `force_alerting`

Параметр, определяющий интервал времени в миллисекундах, по прошествии которого на оригиналирующий шлюз будет отослан пакет Alerting Message с нейтральным содержанием.

**Формат:**

```
force_alerting=<number>
```

где <number> - 0 или любое положительное целое. При значении 0 отсылка пакета не происходит.

**Значение по умолчанию:**

0

**Пример:**

```
force_alerting=4000
```

**Примечание:** Это поле используется, если шлюз некорректно функционирует в отсутствии пакета Alerting Message. Например, VocalTec 2000 (как, впрочем и некоторые модификации шлюзов Cisco) без этого сообщения часто передают звук только в одну сторону или не передают его вовсе.

**Поле** `force_call_proceeding`

Параметр, определяющий интервал времени в миллисекундах, по прошествии которого на оригиналирующий шлюз будет отослан пакет CallProceeding Message с нейтральным содержанием, если до того времени MVTs не получит ни одного пакета от терминирующего шлюза в ответ на отправку ему Setup Message.

**Формат:**

```
force_call_proceeding=<number>
```

где <number> - 0 или любое положительное целое. При значении 0 отсылка пакета не происходит.

**Значение по умолчанию:**

0

**Пример:**

```
force_call_proceeding=4000
```

**Примечание:** Это поле используется, когда задержка отправки сообщения `CallProceeding Message` терминирующим звонок шлюзом не соответствует стандарту H.323 (4 секунды), что приводит к завершению звонка, который при других условиях мог бы состояться (подобная ситуация, в частности, может возникать при работе шлюзов SMG с операторами из Зап. Европы)

#### Поле `gatekeeper`

Имя привратника. Необязательное поле, однако наличие этого поля предполагает удаление из записи о шлюзе полей `port` и `address`, во избежание нарушения логики работы программы.

**Формат:**

`gatekeeper = <строка>`

где `<строка>` - строка символов, представляющая собой название секции из файла данных привратников (`gatekeeper.cfg`)

**Значение по умолчанию:**

Нет

**Пример:**

`gatekeeper=nnov`

#### Поле `gateway_type`

Тип шлюза. Необязательное поле.

**Формат:**

`gateway_type = <number>`

где `<number>` - 0, 1, 2, 3 или 4. 0 – тип шлюза неизвестен. 1 – CISCO, 2 – VocalTec, 3 – Samsung, 4 – Media Ring.

**Значение по умолчанию:**

0

**Пример:**

`gateway_type = 1`

#### Поле `gateway_mode`

Режим функционирования шлюза. Необязательное поле.

**Формат:**

`gateway_mode = <number>`

где `<number>` - 0, 1,2 или 3. 0 – звонки не могут направляться на шлюз и приниматься от него; 1 – звонки не могут направляться на шлюз, но могут приниматься от него; 2 – звонки могут направляться на шлюз, но

не принимаются от него; 3 – звонки могут направляться на шлюз и могут приниматься от него;.

**Значение по умолчанию:**

3

**Пример:**

```
gateway_mode = 3
```

**Поле** `h323_gw_id`

Задаёт значение Source H323 Alias, параметр, который иногда необходимо задавать для удаленного шлюза. Имеет смысл только для исходящих шлюзов.

**Формат:**

```
h323_gw_id=<строка символов>
```

где < строка символов> – произвольный символьный идентификатор.

**Значение по умолчанию:**

Нет.

**Пример:**

```
h323_gw_id=DistantGate
```

**Поле** `local_address`

...служит для задания списка локальных адресов для входящего и исходящего трафика. Система всегда выбирает первый наименее загруженный IP адрес. Если все локальные адреса из списка оказываются перегруженными выбор осуществляется в соответствии со значением поля `local_ip_manager_mode` в секции `[ргоху]` конфигурационного файла **meraproxy.cfg**. Если поле `local_address` не задано, то локальный адрес будет выбираться, исходя из локальной таблицы маршрутизации. Для входящего трафика, если поле `local_address` не задано, будет использоваться адрес, на который пришел звонок, если же поле определено, из списка будет выбран первый неперегруженный на данный момент адрес, который и будет использоваться для H245 сессии и для приема/передачи RTP и RTCP пакетов.

**Формат:**

```
local_address=<адрес>
```

где <адрес> – IP адрес, заданный в виде oct.oct.oct.oct (oct=0...255)

**Пример:**

```
local_address=195.88.57.123
```

**Поле** `mask`

Маска входящих соединений шлюза. Параметры этого шлюза будут применимы ко всем соединениям, удовлетворяющим значениям пары `address/mask`.

**Формат:**

```
mask = <number>
```

где `<number>` - значение в формате `oct.oct.oct.oct`.

**Значение по умолчанию:**

```
255.255.255.255
```

**Пример:**

```
mask=255.255.255.0
```

**Поле** `min_asr`

Служит для задания нижнего предела ASR (усредненного показателя количества успешны вызовов), достижение которого будет вызывать отправку уведомлений администратору по эл. почте.

**Формат:**

```
min_asr=<число>
```

где `<число>` - целое положительное значение в пределах от 0 до 100, означающее мин. допустимый уровень ASR

**Значение по умолчанию:**

```
0
```

**Пример:**

```
min_asr=50
```

**Поле** `proxy_type`

Содержит признак проксирования медиа трафика от данного шлюза (0 – проксировать только сигнальный трафик, 1 – проксировать весь трафик). Необязательное поле.

**Формат:**

```
proxy_type = <flag>
```

где `flag` – признак проксирования [0,1]

**Значение по умолчанию:**

```
1
```

**Пример:**

```
proxy_type = 0
```

**Поле** `radius_gw_id`

Задаёт RADIUS VSA поля `h323-gw-id` и `h323-remote-id`, из исходного и терминирующего шлюзов соответственно. Если параметр `radius_gw_id` шлюза не задан, то `h323-gw-id` и `h323-remote-id` заполняются IP адресами соответствующих шлюзов.

**Формат:**

```
Radius_gw_id=<строка символов>
```

где < строка символов> – произвольный символьный идентификатор.

**Значение по умолчанию:**

Нет.

**Пример:**

```
radius_gw_id=DistantGate
```

**Поле** `radius_remote_address`

Поле используется для подстановки в сообщения, направляемые RADIUS-серверу IP-адрес привратника (`gatekeeper`) вместо IP-адреса удаленного шлюза. При значении 0 в сообщениях `h323-remote-address` и `h323-remote-id` указывается IP-адрес удаленного шлюза, при значении 1 - IP-адрес привратника (`gatekeeper`).

**Формат:**

```
radius_remote_address=<число>
```

где < число> – 0 или 1.

**Значение по умолчанию:**

0

**Пример:**

```
radius_remote_address=1
```

**Поле** `radius_uses_callid`

Некоторые системы биллинга используют для идентификации звонка идентификатор конференции. Для решения проблем, которые могут возникать в случае использования отдельных видов шлюзов (шлюзы Vocaltec, например, могут генерировать одинаковый идентификатор конференции для нескольких звонков), введен параметр `radius_uses_callid`.

Флаг `radius_uses_callid` используется для замены в сообщениях Radius серверу идентификатора конференции (`confID`) на идентификатор звонка (`callID`). Флаг принимает значения 0 и 1. Когда флаг сброшен (0)



в сообщениях h323-incoming-conf-id и h323-conf-id используется идентификатор конференции, если флаг установлен (1) и параметр call ID не пустой, то в этих сообщениях идентификатор конференции и идентификатор звонка меняются местами.

**Формат:**

```
radius_uses_callid=<number>
```

где <number> 0 или 1

**Значение по умолчанию:**

0

**Пример:**

```
radius_uses_callid=1
```

**Поле rbt\_timeout**

В описании шлюза задает максимальное время ожидания пакетов Alert и/или ProgressIndicator от терминирующего шлюза (в сек.). Параметр предназначен для тонкой настройки режима работы. **Совместное использование с параметрами validation\_gap и validation\_msg требует внимания к деталям!**

Например, при значениях параметров validation\_gap=10000, validation\_msg=7 (т.е. возобновление обмена между шлюзами по истечению 10 сек. или по приходу сообщения Connect) и с параметром rbt\_timeout=9 (9 сек.), в случае отсутствия сообщения Connect в течение 9 сек. после отправки Setup звонок будет завершен с причиной eRBTTimeout.

**Формат:**

```
rbt_timeout=<число>
```

где <число> – положительное целое значение в секундах

**Значение по умолчанию:**

0, т.е. функциональность не задействована

**Пример:**

```
rbt_timeout=10
```

**Поле route\_cause**

Причины завершения звонка, вызывающие продолжение попыток его маршрутизации.

**Формат:**

```
route_cause = <число>; [< число >]
```

где <число> – цифра(ы) в десятиричном формате, выражающая(ие) код(ы) причин, по которым происходит возобновление попыток маршрутизации звонка. Элементы списка разделяются ';'. Для

срабатывания функции по любой причине значение параметра  
`route_cause=0-127`

**Значение по умолчанию:**

Нет. Перенаправление звонка не задействовано

**Пример:**

`route_cause=0-16;18-127`

**ВНИМАНИЕ!** В данном случае, если от терминирующего шлюза приходит `release_complete` с кодом 17 («номер занят»), то попытки дальнейшей маршрутизации звонка прекращаются

## Поля `src_translate`, `dst_translate`

Содержат правила преобразования номера вызывающего (`src_translate`) или вызываемого (`dst_translate`) абонента. Как правило, состоит из двух частей, разделенных символом “/”

- выражение `regex`, задающее строку поиска
- строка замены

Строка поиска, для удобства, может быть разделена на секции символом “|”. В строке замены можно использовать подстроку, удовлетворяющую условию секции, указав номер секции после символа “\”. Для использования в строке замены подстроки, удовлетворяющей всем секциям одновременно служит символ “&”. Необязательные поля.

**Формат:**

```
src_translate =
regex1[|regex2[|regex3]]/[str][\1][str][\2][str][\3][str][&]
dst_translate =
regex1[|regex2[|regex3]]/[str][\1][str][\2][str][\3][str][&]
где regex - строка поиска в формате regex, str - строка символов
```

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
Расширение городского номера в полный интернациональный
dst_translate = ...../78312&
778899 -> 78312778899
Изменение определенных цифр номера
src_translate = 77|.|.|32/774\332
778832 -> 774832
```

## Поле `number`

Прямой телефонный номер шлюза. Регулярные выражения не используются. Применяется для поиска по прямым соответствиям в файле плана набора `dialpeer.cfg`, объекта набора ссылающегося на макроязык ENDPOINTS, например вида:

[DIRECTNUMBERS]

```
dst_pattern=[0-9]*
priority=1000
gateway=ENDPOINTS
```

**Формат:**

```
number = <строка>
```

где <строка> – набор цифровых символов, представляющих собой телефонный номер.

**Пример:**

```
number = 78312775825
```

**Поля** `in_dst_translate`, `in_src_translate`

Предварительная трансляция, выполняемая при поступлении звонка от этого шлюза, перед вызовом поиска по объектам набора (dial peers). Параметры полезны, когда какой-либо шлюз присылает номера не в формате E164.

Правила преобразования номера идентичны правилам, приведенным в описании полей `src_translate`, `dst_translate`.

**Формат:**

```
src_translate =
regexpl[|regexp2[|regexp3]]/[str][\1][str][\2][str][\3][str][&]
dst_translate =
regexpl[|regexp2[|regexp3]]/[str][\1][str][\2][str][\3][str][&]
где regexp - строка поиска в формате regexp, str - строка символов
```

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
Расширение городского номера в полный интернациональный
dst_translate = ...../78312&
778899 -> 78312778899
Изменение определенных цифр номера
src_translate = 77|.|.|32/774\332
778832 -> 774832
```

**Поле** `validation_gap`

Определенное для данного направления администратором, время задержки отправки пакетов на шлюз-инициатор вызова.

В процессе установления звонка все пакеты, полученные от terminating шлюза сохраняются в буфере MVTS, содержимое которого будет отправлено на шлюз-инициатор вызова либо по истечении периода задержки, установленного в поле `validation_gap` времени, либо по получении от terminating шлюза одного из сообщений, определенного в поле

`validation_msg` (CallProceeding, Progress, Facility, Alerting).

(Подобная организация процесса установления вызова объясняется тем, что некоторые виды шлюзов допускают лишь однократное получение ими пакета CallProceeding, в результате чего, если после CallProceeding вдруг последует сигнал Release\_complete, дальнейший процесс установления звонка (перемаршрутизация) с таким шлюзом-инициатором становится невозможен, так как он не будет воспринимать повторные сообщения CallProceeding).

Если для данного пользователя продолжительность периода задержки не определена, то используется общесистемное значение параметра, заданное в конфигурационном файле **meraproxy.cfg**

**Формат:**

`validation_gap = <число>`

где <число> – время задержки в миллисекундах до начала срабатывания функции перенаправления звонка.

**Значение по умолчанию:**

0

**Пример:**

`validation_gap=3500`

**Поле** `validation_msg`

Определенный для данного терминирующего шлюза тип сообщения, до получения которого приостанавливается процесс обмена пакетами со стороной-инициатором вызова, (другим условием возобновления процесса отправки пакетов является истечение периода времени, определенного параметром `validation_gap`) Если вызываемая сторона успевает отказать в установлении соединения до выполнения обоих этих условий, звонок будет перенаправлен.

---

**Примечание:** *Перенаправление звонка окажется невозможным, если вызываемая сторона уже установила TCP-соединение для H.245.*

---

**Формат:**

`validation_msg = <число>`

где <число> – тип сообщения от вызываемой стороны,

**Значение по умолчанию:**

по умолчанию Connect=7, Alerting=1, Progress=3, Facility=98. (для задействования значений по умолчанию достаточно установить любой не нулевой параметр `validation_gap`)

**Пример:**

`validation_msg=7`

## Поле `user`

Имя пользователя в системе. Используется для идентификации пользователя при запросах через шлюз к системе удаленной аутентификации RADIUS.

### Формат:

```
user=<строка>
```

где `<строка>` – строка символов, назначенная администратором, в качестве внутреннего имени пользователя и, при необходимости, одно из макроимен `$ip$`, `$group$`, либо макроимя `$ani$`, означающее входящий номер абонента.

Если в поле в качестве параметра указывается макроподстановка `$ani$`, то в запросе к RADIUS серверу вместо имени пользователя будет содержаться номер инициатора вызова.

Макроимена `$ip$` (IP адрес), `$group$` (имя группы) применяются для трансляции имени пользователя при включении в запрос к RADIUS серверу. При поступлении вызова с адреса 127.0.0.2 и наличии в поле `user` имени пользователя и, например, макроимени `$ip$ (v:$ip$@)`, результат трансляции будет `v:127.0.0.2@`

### Пример:

```
user=anyone
user=$ani$
user=v:$ip$@
```

## Поле `password`

Пароль шлюза. Используется системой совместно с внутренним именем для тех же целей.

### Формат:

```
password = <строка символов >
```

где `<строка символов>` – пароль, подчиняющийся общепринятым правилам использования паролей в ОС.

### Пример:

```
password = Key0598E
```

## 6.1.6. СТРУКТУРА ФАЙЛА ДАННЫХ ПРИВРАТНИКОВ (GATEKEEPER)

Файл данных привратников представляет собой текстовый файл с именем **gatekeeper.cfg** или любым другим, назначенным

администратором (см. главу 5.4.2). Администратор может вносить изменения в файл данных с помощью любого текстового редактора.

---

**Примечание:** При обновлении файла данных администратору рекомендуется сначала скопировать файл данных во временный файл, отредактировать его и перекопировать обратно.

---

В файле данных привратников вся информация об отдельном привратнике хранится в виде секции с параметрами привратника в виде полей секции, перечисленными ниже.

Пример записи о привратнике:

```
[ nnov_kis ]
address= 215.24.79.120
port= 1719
type= 1
user=skvair
password=
local_address=215.24.79.122
security=2
terminal=0
keepalive=30
prefixes=789#;889#
```

[ **mera** ] – Пример названия секции, уникальное имя привратника

Поле `address`

IP адрес или DNS-имя привратника. Поле обязательно для заполнения.

**Формат:**

```
address = <адрес>
```

где <адрес> – IP адрес, заданный в виде oct.oct.oct.oct (oct=0...255), или имя хоста.

**Пример:**

```
address = 196.88.57.12
address = computer.mera.ru
```

Поле `id`

Идентификатор привратника. Необязательное поле

**Формат:**

```
id = <тип> (число, имя, и т.д.)
```

где тип – (целое, неотрицательное и т.д.).

**Пример:**

```
id=
```

**Поле** `keepalive`

Периодичность регистрации на привратнике.

**Формат:**

`keepalive = < число >`

где `<число>` – целое положительное значение периодичности в секундах. Значение по умолчанию 5.

**Пример:**

`keepalive = 10`

**Поле** `keepalive_type`

Тип сообщения, отсылаемого на привратник при перерегистрации.

**Формат:**

`keepalive_type=< число >`

где `<число>` – 0 или 1. 0 – на привратник будет отсылаться «сокращенное» сообщение о перерегистрации, 1 – на привратник будет отсылаться полный запрос на регистрацию.

**Значение по умолчанию:**

0

**Пример:**

`keepalive_type=1`

**Поле** `local_address`

Поле для явного задания реального IP адреса, с которого будет осуществляться отправка RAS сообщений.

**Формат:**

`local_address=<адрес>`

где `<адрес>` – IP адрес, заданный в виде oct.oct.oct.oct (oct=0...255)

**Пример:**

`local_address=196.88.57.12`

**Поле** `password`

Пароль пользователя для авторизации на привратнике.

**Формат:**

`password=< строка символов >`

где `< строка символов >` - пароль пользователя

**Пример:**

```
password = 04efd7
```

**Поле** `port`

Порт привратника для приема RAS трафика.

**Формат:**

```
port = <число>
```

где число – целое, положительное.

**Значение по умолчанию:**

```
1719
```

**Пример:**

```
port = 1780
```

**Поле** `prefixes`

Список префиксов шлюза, сообщаемый привратнику (если значение поля `terminal = 0`) и список список E.164 псевдонимов (если значение поля `terminal = 1`).

**Формат:**

```
prefixes = < список >
```

где <список> – список префиксов шлюза или E.164 псевдонимов, элементы которого разделяются ‘;’

**Пример:**

```
prefixes=789#;889#;775#
```

**Поле** `security`

Способ, авторизации на привратнике.

**Формат:**

```
security = <число >
```

где <число> – 0, 1, 2 или 3

0 – формирование регистрационного параметра по формуле `source_alias="user|password"`

1 - VocalTec

2 – CISCO

3 – CHAP

Значение по умолчанию: 0

**Пример:**

```
security=0
```



**Поле** `terminal`

Способ представления MVTС привратнику. Принимаемые значения 1 или 0. Если 1, представляться привратнику как терминал, если 0 – как шлюз. Необязательное поле.

**Формат:**

`terminal = <число >`

где <число> – способ представления привратнику [0,1]

**Значение по умолчанию:**

0

**Пример:**

`terminal = 1`

**Поле** `type`

Признак принадлежности MVTС к зоне привратника.

При значении 1, MVTС регистрируется на привратнике и посылает ARQ для исходящих звонков. Для входящих звонков предполагается Pre-granted ARQ и привратник не информируется.

При значении 0, MVTС запрашивает у привратника адрес назначения при помощи LRQ.

**Формат:**

`type = <число >`

где <число> – признак принадлежности [0,1]

**Значение по умолчанию:**

0 (MVTС запрашивает у привратника адрес назначения при помощи LRQ)

**Пример:**

`type = 1`

**Поле** `user`

Имя пользователя для авторизации на привратнике

**Формат:**

`user = < имя >`

где <имя> – строка символов

**Пример:**

`user = anyone`

### 6.1.7. СТРУКТУРА ФАЙЛА С ПЛАНОМ НАБОРА

План набора представляет собой текстовый файл **dialpeer.cfg** или назначенный пользователем (см. главу 3.1.3). Администратор может вносить изменения в файл плана набора с помощью любого текстового редактора.

---

**Примечание:** При обновлении плана набора администратору рекомендуется сначала скопировать файл плана набора во временный файл, отредактировать его и перекопировать обратно.

---

В файле плана набора вся информация об отдельном объекте набора (**dial peer**) хранится в виде секции. Названием секции служит внутренний идентификатор объекта.

Пример записи об объекте набора:

```
[KIS]
dst_pattern=78312[0-9]*;78319[0-9]*
dst_exclude=
source_exclude=
src_pattern=[0-9]*
dst_translate= [0-9]*/78312778899
src_translate= 7831277|.|.|32/78312774\332
dst_bill_translate=
bill_translate=
hunt_stop=0
priority=10
gateway=NNOV_KIS
group_allow= GROUP1; GROUP2
group_deny= GROUP3
group_translate=NEWGROUP/OLDGROUP
capacity=40
override_user=OtherUser
q931_cause=1

[KIS1]
active_time=1/08:00-5/20:00
dst_pattern =78312[0-9]*;78319[0-9]*
gateway =NNOV_KIS1
src_exclude=78312[0-9]*
dst_exclude=78095[0-9]*
```

[ **moscow** ] Пример идентификатора направления:

Поле `active_time`

С помощью поля `active_time` можно задать временной интервал работы конкретного объекта набора (**dial\_peer**).

**Формат:**

```
active_time=<DAY>/<HH:MM>-<day>/<hh:mm>
```

где DAY - номер дня недели начала интервала (пн=1, вт=2 ... вс=7),

`day` - номер дня недели конца интервала (пн=1, вт=2 ... вс=7)

`hh:mm` - начало интервала (часы, минуты)

`hh:mm` - конец интервала (часы, минуты)

**Пример:**

```
active_time=1/08:00-5/20:00
```

Объект набора с такими параметрами будет срабатывать только начиная с 8 часов понедельника до 20 часов пятницы.

**Примечание.** Как и в других полях в `active_time` можно задавать несколько интервалов, разделяя их точкой с запятой.

## Поле `capacity`

Позволяет ограничить число одновременных звонков через объект набора. Диапазон допустимых значений от 0 до 65535.

**Формат:**

```
capacity=<число>
```

где < число > положительное целое от 0 до 65535

**Значение по умолчанию:**

```
65535
```

**Пример:**

```
capacity=200
```

## Поле `bill_translate`

Содержит правила преобразования номера для целей системы учета и начисления платы. Часто для системы биллинга необходим номер не в том виде, в каком он «уходит» к удаленному шлюзу, а в более удобном формате, например, E.164. При входе в поиск по объектам набора (`dial-peers`) значение параметра `bill_number` равно значению параметра `dst_number`, однако во время процедуры поиска преобразование осуществляется не по правилам, заданным в поле `dst_translate`, а по правилам, содержащимся в поле `bill_translate`, в результате чего можно получить номер в любом удобном для системы начисления платы виде. Синтаксис правил преобразования идентичен синтаксису поля `dst_translate`.

При задании правил преобразования для обозначения «пустого» номера применяется ключевое слово `empty`.

Во всех полях преобразования номера, (*кроме `group_translate` и только в пределах файла `dialpeer.cfg`*) допускается применение

следующих макроимен:

**\$ani\$** - ANI-номер вызывающей стороны

**\$dnis\$** - номер вызываемой стороны

**\$bill\_ani\$** - ANI-номер для биллинга

**\$bill\_dnis\$** - номер вызываемой стороны для биллинга

**\$user\$** - имя пользователя

**\$id\$** – уникальный идентификатор вызова (извлеченный из CDR записи) в формате <time stamp момента запуска MVTs>#<порядковый номер звонка>#

#### Формат:

```
bill_translate =
regexpl[|regex2[|regex3|]/[str][\1][str][\2][str][\3][str][&]
dst_translate =
regexpl[|regex2[|regex3|]/[str][\1][str][\2][str][\3][str][&
```

где regex - строка поиска в формате regex, str - строка символов

#### Значение по умолчанию:

Пустая строка

#### Примеры:

```
[DIALPEER_TEST]
dst_pattern=[0-9]*
src_pattern=[0-9]*
dst_translate=.../1234#8812788&
bill_translate=.../7812788&
Номер набираемый 3264
Итог:
dst_number=1234#88127883264
bill_number=7127883264
```

#### Использование ключевого слова и макроимен в преобразовании:

1. подстановка вместо пустого номера имени пользователя  
src\_translate=empty/\$user\$
2. подстановка вместо пустого номера номера вызывающей стороны:  
src\_translate=empty/\$ani\$

**Примечание:** параметр `bill_translate` больше не используется для целей конфигурации, так как его функции выполняются параметрами `dst_bill_translate` и `src_bill_translate`. В конфигурационных файлах оставлен для совместимости с более ранними версиями MVTs.

#### Поле `dst_bill_translate`

В этом поле задается шаблон дополнительного преобразования номера вызываемого абонента для приведения его к виду удобному для системы учета начисляемой платы (биллинг).

Синтаксис трансляции идентичен полю `dst_translate`

При задании правил преобразования для обозначения «пустого» номера применяется ключевое слово `empty`.

Во всех полях преобразования номера, (*кроме* `group_translate` и *только в пределах файла* `dialpeer.cfg`) допускается применение следующих макроимен:

**\$ani\$** - ANI-номер вызывающей стороны

**\$dnis\$** - номер вызываемой стороны

**\$bill\_ani\$** - ANI-номер для биллинга

**\$bill\_dnis\$** - номер вызываемой стороны для биллинга

**\$user\$** - имя пользователя

**\$id\$** – уникальный идентификатор вызова (извлеченный из CDR записи) в формате `<time stamp момента запуска MVTs>#<порядковый номер звонка>#`

#### Формат:

```
dst_bill_translate =
regexpr1[|regexp2[|regexp3]]/[str][\1][str][\2][str][\3][str][&]
dst_translate =
regexpr1[|regexp2[|regexp3]]/[str][\1][str][\2][str][\3][str][&]
```

где `regexpr` - строка в формате регулярных выражений, `str` - строка символов

#### Значение по умолчанию:

Пустая строка

**Примечание:** Когда в одной и той же записи об объекте набора одновременно присутствуют поля с шаблонами трансляции `dst_bill_translate` и `bill_translate`, параметр `dst_number` будет транслирован в соответствии с шаблоном заданным в поле `dst_bill_translate`

#### Поля `dst_exclude`, `src_exclude`

Списки объектов набора (тел. номеров), исключенных для установления соединений и приема входящих звонков.

#### Формат:

```
src_exclude=<строка>; <строка>; ...
dst_exclude=<строка>; <строка>; ...
```

где

`<строка>` - `regexpr` выражение.

**Значение по умолчанию:**

src\_exclude - пустая строка

dst\_exclude - пустая строка

**Пример:**

```
dst_exclude = 78312.... ; 7095[0-9]*
src_exclude = 78414....
```

**Поля** dst\_pattern, src\_pattern

Требования к номеру вызываемого (dst\_pattern) или вызывающего (src\_pattern) абонента. Требования задаются в виде regexp выражения. src\_pattern - необязательное поле.

**Формат:**

```
src_pattern = <строка>; <строка>; ...
dst_pattern = <строка>; <строка>; ...
```

где

&lt;строка&gt; - regexp выражение.

**Значение по умолчанию:**

dst\_pattern - Пустая строка

src\_pattern - Все номера

**Пример:**

```
dst_pattern = 78312.... ; 7095[0-9]*
src_pattern = 78414....
```

**Поля** dst\_translate, src\_translate

Содержат правила преобразования номера вызывающего (src\_translate) или вызываемого (dst\_translate) абонента. Как правило, состоит из двух частей, разделенных символом “/”

- выражение regexp, задающее строку поиска
- строка замены

Строка поиска, для удобства, может быть разделена на секции символом “[|]”. В строке замены можно использовать подстроку, удовлетворяющую условию секции, указав номер секции после символа “\”. Для использования в строке замены подстроки, удовлетворяющей всем секциям одновременно служит символ “&”. Необязательные поля.

При задании правил преобразования для обозначения «пустого» номера применяется ключевое слово empty.

Во всех полях преобразования номера, (кроме group\_translate и только в пределах файла *dialpeer.cfg*) допускается применение следующих макроимен:

**\$ani\$** - ANI-номер вызывающей стороны

**\$dnis\$** - номер вызываемой стороны

**\$bill\_ani\$** - ANI-номер для биллинга

**\$bill\_dnis\$** - номер вызываемой стороны для биллинга

**\$user\$** - имя пользователя

**\$id\$** – уникальный идентификатор вызова (извлеченный из CDR записи) в формате <time stamp момента запуска MVTs>#<порядковый номер звонка>#

**Формат:**

```
src_translate =
regexpl[|regex2[|regex3|]/[str][\1][str][\2][str][\3][str][&]
dst_translate =
regexpl[|regex2[|regex3|]/[str][\1][str][\2][str][\3][str][&]
где regex - строка поиска в формате regex, str - строка символов
```

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
Расширение городского номера в полный интернациональный
dst_translate = ...../78312&
778899 -> 78312778899
Изменение определенных цифр номера
src_translate = 77|.|.|32/774\332
778832 -> 774832
```

**Поле** `src_bill_translate`

В этом поле задается шаблон дополнительного преобразования номера вызывающего абонента для приведения его к виду удобному для системы учета начисляемой платы (биллинг).

Синтаксис трансляции идентичен полю `src_translate`

**Формат:**

```
src_bill_translate =
regexpl[|regex2[|regex3|]/[str][\1][str][\2][str][\3][str][&]
dst_translate =
regexpl[|regex2[|regex3|]/[str][\1][str][\2][str][\3][str][&]
```

где regex - строка в формате регулярных выражений, str - строка символов

**Значение по умолчанию:**

Пустая строка

**Поле** `gateway`

Содержит внутренний псевдоним шлюза, куда будет направлен звонок в

случае выбора данного объекта (dial peer). Если в поле установлено зарезервированное значение AGAIN, то при выборе данного объекта набора осуществляется трансляция номеров и заново начинается поиск объекта, удовлетворяющего транслированным номерам. Если в поле установлено зарезервированное значение NULL, то при выборе данного объекта поиск прекращается и звонок заканчивается. Значениями этого поля также могут являться макроимена ENDPOINTS (поиск шлюзов с точным совпадением номера назначения) и NEXT (переход к следующему dial-peer). Если макроименя ENDPOINTS явным образом не указывается поиск шлюзов с точным совпадением номеров осуществляться не будет. Макроименя NEXT в отличие от AGAIN вместо рекурсивного поиска инициирует преобразование номеров, групп и переход к следующему объекту набора (dial peer). Макроименя EXTERNAL означает использование динамической маршрутизации от RADIUS сервера.

В качестве своего значения поле может также принимать имена секций из файлов **gateway.cfg** и **user.cfg**, что накладывает ограничения на имена данных секций, которые должны быть уникальны в обоих файлах.

Обязательное поле.

**Формат:**

```
gateway=<string >
```

где string - внутреннее имя шлюза, присвоенное ему в файле данных шлюзов либо одно из макроимен (AGAIN, ENDPOINTS, NULL, NEXT, EXTERNAL)

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
gateway=Moscow
```

## Поля `group_allow`, `group_deny`

Списки групп шлюзов, которым разрешены или запрещены исходящие звонки через данное направление. Используются при поиске объекта набора. Необязательные поля.

**Формат:**

```
group_allow = <string1>; <string2>; ...  
group_deny = <string3>; <string4>; ...
```

где <stringN> - имя группы.

**Значение по умолчанию:**

Пустая строка



**Пример:**

```
group_allow = GROUP1; GROUP2  
group_deny = GROUP3
```

**Поле** `group_translate`

Трансляция группы пользователя. Необязательное поле.

**Формат:**

```
group_translate=<string1>/<string2>; ...
```

где <строка> - имя группы.

**Значение по умолчанию:**

Пустая строка

**Пример:**

```
group_translate=OLDGROUP/NEUGROUP
```

**Поле** `hunt_stop`

Флаг, останавливающий дальнейший просмотр объектов набора, если данное направление подходит, но соответствующий ему шлюз недоступен или перегружен (1- просмотр останавливается, 0 – не останавливается). Необязательное поле.

**Формат:**

```
hunt_stop = значение  
где значение - 1 или 0
```

**Значение по умолчанию:**

0

**Пример:**

```
hunt_stop = 1
```

**Поле** `hunt_mode`

Данный параметр помогает задать способ распределения нагрузки между шлюзами, перечисленными в поле `gateway` данной секции. Принимаемые значения 0, 1, 2, 3.

`hunt_mode=0` функция отключена;

`hunt_mode=1` каждые 10 секунд шлюз, стоящий первым в списке, переставляется в конец.

`hunt_mode=2` каждые 10 секунд шлюзы списка сортируются по текущему значению абсолютной загрузки.

`hunt_mode=3` каждые 10 секунд шлюзы списка сортируются по отношению текущего значения абсолютной загрузки к максимальной пропускной способности шлюза.

**Формат:**

```
hunt_mode = <value>
```

где <value> – 0, 1, 2 или 3

**Значение по умолчанию:**

0

**Пример:**

```
hunt_mode=1
```

**Поле** `override_proxy_mode`

Поле позволяет изменить режим проксирования применительно к данному конкретному шлюзу. Принимаемые значения 0, 1, 2, 3. Необязательный параметр.

**Формат:**

```
override_proxy_mode=<number>
```

где <number> 0, 1, 2, or 3

0 – проксирование только сигнального трафика

1 – полное проксирование (сигнальный и мультимедийный трафик)

2 – использовать режим проксирования вызывающей стороны

3 – использовать режим проксирования вызываемой стороны

**Значение по умолчанию:**

НЕТ

**Пример:**

```
override_proxy_mode=2
```

**Поле** `override_user`

Используется для подстановки (замены) имени пользователя (присутствующего в параметрах шлюза) при возникновении необходимости осуществить терминацию звонка в каком-либо направлении, авторизуемую по имени другого пользователя.

**Формат:**

```
override_user=<строка>
```

где <строка> - символьная строка, обозначающая подставляемое имя авторизуемого пользователя.

**Значение по умолчанию:**

НЕТ

**Пример:**

```
override_user=OtherUser
```

**Поле** `priority`

Содержит приоритет объекта набора. Позволяет выбрать один из группы подходящих объектов для данного звонка. Больше значение соответствует большему приоритету. Необязательное поле.

**Формат:**

```
priority=<number>
```

где `number` - приоритет [0..65535]

**Значение по умолчанию:**

```
1
```

**Пример:**

```
priority=4
```

**Поле** `q931_cause`

С помощью поля `q931_cause`, можно задавать Q931 причину завершения звонка в случае попадания на объект набора с параметром `gateway=NULL`. Например, можно задать причину завершения = 1 для звонков, которые не подходят ни к одному объекту набора, т.е. имеют, например, неверный номер.

**Формат:**

```
q931_cause=<число>
```

где `<число>` - целое положительное от 0 до 127.

**Пример:**

```
q931_cause=0
```

**6.1.8. СТРУКТУРА ЗАПИСЕЙ ДАННЫХ О ЗВОНКЕ (CDRs)**

Файл для систем учета и начисления причитающейся платы (billing system) представляет собой текстовый файл, организованный по принципу “один звонок – одна запись”.

**Таблица 6-1: Структура записи с данными о звонке (CDR)**

Поле	Назначение
<b>CALLID</b>	Идентификатор вызова
<b>CONFID</b>	Идентификатор конференции в которой участвовал данный звонок

Поле	Назначение
<b>CONNECT-TIME</b>	Время начала разговора
<b>DIALPEER-NAME</b>	Имя секции с описанием задействованного объекта набора
<b>DISCONNECT-CODE-LOCAL</b>	Причина разъединения
<b>DISCONNECT-CODE-Q931</b>	Код разъединения по Q931
<b>DISCONNECT-TIME</b>	Время окончания звонка
<b>DST-BYTES-IN</b>	Количество полученных байт от вызываемого абонента
<b>DST-BYTES-OUT</b>	Количество переданных байт вызываемому абонента
<b>DST-CODEC</b>	Кодеки, использованные вызываемым абонентом
<b>DST-IP</b>	IP адрес шлюза на который отправлен звонок
<b>DST-NAME</b>	Имя секции с описанием терминирующего шлюза
<b>DST-NUMBER-BILL</b>	Номер вызываемого абонента после трансляции, для системы начисления платы (биллинга)
<b>DST-NUMBER-IN</b>	Номер вызываемого абонента до трансляции
<b>DST-NUMBER-OUT</b>	Номер вызываемого абонента после трансляции
<b>DST-USER</b>	Имя вызываемого пользователя
<b>ELAPSED-TIME</b>	Общее время разговора в секундах
<b>HOST</b>	Локальный адрес MVTС
<b>INITIAL-INCOMING-LOCAL-ADDRESS</b>	Локальный IP-адрес, на который был принят Setup от originатора
<b>LAR-FAULT-REASON</b>	Код ошибки по LAR (функция перенаправления звонка)
<b>OUTGOING-LOCAL-ADDRESS</b>	Выбранный локальный IP-адрес для обмена с терминатором
<b>PROXY-MODE</b>	Режим проксирования
<b>QoS</b>	Качество сервиса
<b>RADIUS-USER</b>	Имя пользователя, полученное от RADIUS сервера при включенном режиме <code>use_h323_ivr_in</code> и используемое в accounting-пакетах (см. секция [Radius] в файле <code>meraproxy.cfg</code> )
<b>RECORD-ID</b>	Уникальный идентификатор звонка в формате <code>&lt;start-time&gt;-&lt;call-number&gt;</code> , где <code>&lt;start-time&gt;</code> - штамп времени текущей сессии MVTС, выраженный в количестве секунд с 1-го января 1970г до момента последнего запуска MVTС, а

Поле	Назначение
	<call-number> - порядковый номер звонка с момента последнего запуска MVTTS.
REMOTE-GATEKEEPER-IP	IP-адрес привратника (gatekeeper)
ROUTE-RETRIES	Количество попыток перенаправления звонка
SELECTED-INCOMING-LOCAL-ADDRESS	Выбранный локальный IP-адрес для обмена с оригинатором
SETUP-TIME	Время установления звонка
SRC-BYTES-IN	Количество полученных байт от вызывающего абонента
SRC-BYTES-OUT	Количество переданных байт вызываемому абонента
SRC-CODEC	Кодеки, использованные вызывающим абонентом
SRC-IP	IP адрес шлюза с которого получен звонок
SRC-NAME	Имя секции с описанием оригинирующего шлюза
SRC-NUMBER-BILL	Номер вызывающего абонента после трансляции, для системы начисления платы (биллинга)
SRC-NUMBER-IN	Номер вызывающего абонента до трансляции
SRC-NUMBER-OUT	Номер вызывающего абонента после трансляции
SRC-USER	Имя вызывающего пользователя

**Пример:**

```
Fri Oct 12 17:13:50 2001, HOST=195.98.57.10, SRC-NUMBER=78312778794,
DST-NUMBER=78312788847, SRC-IP=195.98.57.27:1993, DST-
IP=215.24.78.120:1720, SRC-USER=skvair, ELAPSED-TIME=6, SETUP-
TIME=17:13:38.000 +0400 Fri Oct 12 2001, CONNECT-TIME=17:13:40.000
+0400 Fri Oct 12 2001, DISCONNECT-TIME=17:13:46.000 +0400 Fri Oct 12
2001, DISCONNECT-CODE-LOCAL=2, DISCONNECT-CODE-Q931=8090, SRC-BYTES-
IN=5394, DST-BYTES-IN=3863, SRC-BYTES-OUT=3707, DST-BYTES-OUT=4823,
SRC-CODEC=g7231 , DST-CODEC=g7231 ,
CALLID=5108205609bfd511a30300002125184a,
CONFID=5208205609bfd511a30300002125184a, PROXY-MODE=0, ROUTE-
RETRIES=3
```

Перечень значений поля DISCONNECT-CODE-LOCAL (внутренний код причины разъединения) см. в [Error! Not a valid bookmark self-reference.](#) **Приложение В: Наиболее часто встречающиеся коды разъединения, генерируемые MVTTS**

**6.1.9. СТРУКТУРА ЗАПИСЕЙ ПРОТОКОЛА РАБОТЫ**

Протокол работы содержит записи о смене состояний каждого звонка.

При наибольшем уровне детализации выводятся тексты полученных и переданных сообщений. Уровень детализации протокола работы

устанавливается администратором в системном конфигурационном файле ( глава 6.1.3).

При уровне детализации 1 запись протокола содержит только время события, тип сообщения и IP адрес компьютера, с которого/на который оно было отправлено.

**Формат:**

<время> <дата> <Recv/Sent> <IP адрес> <протокол> <тип сообщения>  
, где:

<время> и <дата> - текущее время

<Recv/Sent> - Recv - если сообщение было получено, Sent - отослано

<IP адрес> - адрес компьютера, с которого/на который было отправлено сообщение

<протокол> - протокол сообщения

<тип сообщения> - тип сообщения (в соответствии с протоколом H.323 [1])

**Пример:**

```
12:16:47 12/09/2001 Recv 192.168.5.1:1813      Q.931 Setup
12:16:47 12/09/2001 Sent 192.168.5.1:1720      Q.931 Setup
12:16:47 12/09/2001 Recv 192.168.5.1:1720      Q.931 Facility
12:16:47 12/09/2001 Sent 192.168.5.1:1813      Q.931 Facility
12:16:47 12/09/2001 Recv 192.168.5.1:1720      Q.931 Connect
12:16:47 12/09/2001 Sent 192.168.5.1:1813      Q.931 Connect
12:16:47 12/09/2001 Recv 192.168.5.1:1813      Q.931 Facility
```

При уровне детализации 2 выводится текст сообщения.

**Формат:**

<время> <дата> <Recv/Sent> <IP адрес> <протокол> <текст сообщения>  
, где:

<время> и <дата> - текущее время

<Recv/Sent> - Recv - если сообщение было получено, Sent - отослано

<IP адрес> - адрес компьютера, с которого/на который было отправлено сообщение

<протокол> - протокол сообщения

<текст сообщения> - текст сообщения (может содержать несколько строк)

**Пример:**

```
15:05:05 12/09/2001 Recv 192.168.5.1:2883      Q.931
{
  protocolDiscriminator = 8
  callReference = 1
  from = originator
  messageType = Setup
  IE: Bearer-Capability = {
    88 c0 a5
  }
  ...
}
```

```

IE: Display = {
  4d 45 52 41 20 70 68 6f 6e 65 20 37 37 37 37 30 MERA phone
  30 77770
  } 0
}

```

## 6.2. Ядро

В редких случаях необходимости выполнения действий непосредственно с ядром MVTs (запуск/останов), администратор может запустить консоль администрирования и выполнить необходимые манипуляции с указанием при запуске (команда `start`) в командной строке необходимых параметров.

Ядро MVTs можно также запустить скриптом, что имеет свои преимущества. Подробно о скрипте запуска см. ниже

### 6.2.1. ПАРАМЕТРЫ КОМАНДНОЙ СТРОКИ

Команда на запуск ядра MVTs должна содержать один или несколько параметров, значения которых приведены в таблице.

**Таблица 6-2: Параметры командной строки ядра MVTs (mp\_kerneld.x)**

Поле команды <code>start</code>	Пояснение
<code>-h --help</code>	Вывод подсказки и завершение работы
<code>-v --version</code>	Вывод информации о номере версии и завершение работы
<code>-d --daemon</code>	Запуск в качестве фонового процесса (демона)
<code>-u --uid uid</code>	Запуск в качестве процесса с идентификатором пользователя
<code>-g --gid gid</code>	Запуск в качестве процесса с идентификатором группы пользователей
<code>-p --pid-file</code>	Запуск с указанием имени или полного пути для файла идентификатора процесса (pid-file)
<code>-t --terminate</code>	Запуск с нормальным завершением процесса в файле идентификатора (pid-file)
<code>-k --kill</code>	Запуск с предварительным завершением процесса в файле идентификатора (pid-file)
<code>-c --console</code>	Запуск с выводом сообщений в

Поле команды start	Пояснение
	стандартный поток вывода (консоль) вместо журнала syslog
<b>-l --log-file</b>	Запуск с выводом сообщений в файл с именем или каталог вместо журнала syslog
<b>-x --execute</b>	Выполнять как обычную программу
<b>-i --ini-file</b>	Запуск с указанием имени (пути) файла инициализации или списка каталогов для поиска файла инициализации, разделенного ':'
<b>-C --core-size</b>	Запуск с заданием максимального размера core-файла

Попытка запустить ядро без указания параметров команды приведет к выводу на экран подсказка о параметрах командной строки и завершению работы.

Пример вывода на экран при попытке запустить ядро без указания параметров:

```
bash-2.05$ ./mp_kernel.d.x
error: must specify one of -v, -h, -t, -k, -d or -x
usage: [-c] -v|-d|-h|-x
  -h --help          output this help message and exit
  -v --version       display version information and exit
  -d --daemon        run as a daemon
  -u --uid uid       set user id to run as
  -g --gid gid       set group id to run as
  -p --pid-file      name or directory for pid file
  -t --terminate     orderly terminate process in pid file
  -k --kill          preemptively kill process in pid file
  -c --console       output messages to stdout rather than
syslog
  -l --log-file file output messages to file or directory
instead of syslog
  -x --execute       execute as a normal program
  -i --ini-file      set the ini file to use, may be explicit
file or
                    a ':' separated set of directories to
search.
  -C --core-size     set the maximum core file size
```

### 6.2.2. СКРИПТ ЗАПУСКА ЯДРА MP\_KERNEL.SH

Данный способ запуска системы обеспечивает:

- сохранение core-файлов



- внесение записей в протокол работы при аварийном завершении программы
- автоматический перезапуск программы при аварийном завершении

Запуск осуществляется командой

```
mp_kernel.d.sh [cfg_file]
```

, выполненной из каталога **bin/**.

**cfg\_file** - системный конфигурационный файл (по умолчанию **meraproxy.cfg**).

Данный способ позволяет запустить MVTs, не подключая консоль администрирования.

### 6.3. КОНСОЛЬ АДМИНИСТРИРОВАНИЯ

В данной главе описываются возможности, предоставляемые системной консолью: подключение и отключение консоли, назначение и форматы команд.

#### 6.3.1. ВОЗМОЖНОСТИ, ПРЕДОСТАВЛЯЕМЫЕ КОНСОЛЬЮ

Консоль администрирования является вспомогательным приложением, предназначенным для мониторинга работы и администрирования MVTs. Одновременно может быть запущено несколько экземпляров консоли.

Для общения с пользователем реализуется командный интерфейс. Команды поступают через стандартный ввод, результаты выводятся на стандартный вывод.

Консоль администрирования должна авторизовать пользователя перед исполнением команды. Для этого приложение разделяет пользователей на следующие группы:

- **Admin**  
Пользователь, принадлежащий к группе администраторов, имеет право выполнить любую команду консоли администрирования.
- **Support**  
Пользователю из группы технической поддержки разрешен просмотр разнообразной статистики и запрещены команды, влияющие на работу MVTs.
- **Billing**  
Пользователь группы учета начисляемой платы может только просмотреть информацию о текущих звонках.

Для идентификации пользователя приложение проверяет GID пользователя на принадлежность к той или иной группе. Консоль администрирования позволяет:

- Остановить и запустить MVTs

- Загрузить конфигурационные файлы в кэш MVTС, не прекращая выполнение программы
- Получить некоторую статистику работы MVTС
- Принудительно завершить текущие звонки, удовлетворяющие определенным условиям

### 6.3.2. ПАРАМЕТРЫ КОМАНДНОЙ СТРОКИ

В качестве параметра командной строки при запуске консоли администрирования могут указываться одна из команд консоли.

Если консоль запускается просто командой на исполнение двоичного файла без указания каких-либо параметров, консоль запускается с появлением ее командного приглашения.

### 6.3.3. ПОДКЛЮЧЕНИЕ И ОТКЛЮЧЕНИЕ ОТ КОНСОЛИ АДМИНИСТРИРОВАНИЯ

Обычно запуск консоли производится с помощью вызова программы **mp\_shell.sh** (см. раздел 3.3.2.1). При вводе команды консоль администрирования пытается подсоединиться к MVTС. В случае неудачного соединения консоль выводит соответствующее сообщение:

```
Proxy server not found. Use command "start"
to run it.
```

Вывод консоли администрирования направляется в стандартный поток вывода.

При входе консоли в режим ожидания команд появляется приглашение на ввод команд:

```
#>
```

Несколько пользователей могут пользоваться консолью администрирования в одно и то же время.

Для отключения от консоли администрирования воспользуйтесь командой **quit**.

Командная строка консоли может содержать одну команду или параметр циклического повторения `repeat:`, за которым следует команда, для которой требуется циклическое исполнение. Если команда задана без параметра цикличности, по окончании ее выполнения работа консоли администрирования завершается. Результат работы передается в коде возврата:

- 0 – команда выполнена
- 1 – не удалось подключиться к MVTС

#### Формат:

```
mp_shell.sh [command]
, где command – команда для исполнения.
```

#### Пример:

```
mp_shell.sh reload config
```

При запуске скрипта с указанием параметра циклического выполнения `repeat`, команда, которая следует за параметром `repeat:`, будет выполняться с периодичностью в 10 секунд, до тех пор пока работа программы не будет прервана нажатием сочетания клавиш `Ctrl+C`. Периодичность выполнения команды при необходимости меняется редактированием скрипта.

**Формат:**

```
mp_shell.sh repeat: [command]
, где command – команда для исполнения.
```

**Пример:**

```
mp_shell.sh repeat: help
```

– каждые 10 секунд выводит экран подсказки

### 6.3.4. КОМАНДЫ КОНСОЛИ АДМИНИСТРИРОВАНИЯ

Данная версия MVTС содержит несколько типов команд:

- Справочные команды: *help*
- Команды администрирования и конфигурирования: *reload config, reset statistics, start, stop, stop gracious, terminate call, disable gatekeeper*
- Команды тестирования: *show call, show dial, , show disconnect, show dp, show gw, show stat, show route, show stat route, show ep, show incoming, sh ipload, show stat file, show stat param*

Перед вводом команды убедитесь, что консоль администрирования находится в состоянии ожидания команд – в начале строки находится приглашение пользователя:

```
#>
```

Консоль не воспринимает новую команду до тех пор, пока не обработана старая.

#### 6.3.4.1. СПРАВОЧНЫЕ КОМАНДЫ

**Команда** `help`

Данная команда без аргументов выводит список возможных команд с краткими пояснениями. Если в качестве аргумента задать имя любой команды, **help** выведет подробное описание, формат и пример использования команды. Команда доступна пользователям любой из групп.

**Формат:**

```
help
```

**Пример:**

```
#>help
List of command:
  start [cfgfile]           - starts proxy server
  stop                     - stops proxy server
  reload config            - reloads new configuration from config
  ...
  terminate call call_number - terminates call with call_number
  terminate call [-src oct.oct.oct.oct] [-dst -src oct.oct.oct.oct]
                          - terminates selected calls
  quit                    - quits from console
(for more detail information see system administrator guide)
#>
```

**6.3.4.2. КОМАНДЫ АДМИНИСТРИРОВАНИЯ И КОНФИГУРИРОВАНИЯ**

**Команда** `reload config [-d] [-ras]`

Команда заставляет MVTС заново считать все конфигурационные файлы.

При выполнении с аргументом `-d` сообщения об ошибках в конфигурации выводятся на экран.

Поция `-ras` вызывается перепривязку RAS сокета.

Если в процессе считывания конфигурации обнаруживаются ошибки, информация о них выводится на консоль администрирования и MVTС возвращается к предыдущей конфигурации.

Команда доступна пользователям группы Admin .

**Формат:**

```
reload config
```

**Пример:**

```
#>reload config
Configuration reloading ...
Reloading successful
```

**Команда** `reset statistics (re st) [all|src|dst|gw|dp]`

Команда выполняет сброс текущей статистики (количество принятых, отвергнутых звонков, величина входящего/выходящего трафика и т.п).

`reset stat all` – команда очищает всю отведенную под статистику область памяти

`reset stat src|dst|gw|dp` – обнуляет статистику по категориям объектов IP-телефонии (по оригиналирующим, терминирующим, по шлюзам, по объектам набора)

`reset stat src|dst|gw|dp <name>` – обнуляет статистику указанного объекта

#### **Команда** `start`

Команда запускает MVTС, если он не был запущен. Команда доступна пользователям группы Admin.

##### **Формат:**

`start [имя_файла]`

, где `имя_файла` – имя системного конфигурационного файла (по умолчанию **meraproxy.cfg**)

##### **Пример:**

```
#> start myproxy.cfg
Try to start Mera H323 Proxy Server ...
Mera H323 Proxy Server started successfully.
#>
```

#### **Команда** `stop`

Команда завершает текущие звонки и работу MVTС. Команда доступна пользователям группы Admin.

##### **Формат:**

`stop`

##### **Пример:**

```
#> stop
Starting shutdown system...
#>
```

#### **Команда** `stop gracious (stop gra, stop gr)`

Команда ожидает окончания текущих звонки, не принимая новых и только после этого завершает работу MVTС,. Команда доступна пользователям группы Admin.

##### **Формат:**

`stop gracious` **или** `stop gra` **или** `stop gr`

##### **Пример:**

```
#> stop gr
Starting shutdown system...
#>
```

#### **Команда** `terminate call`

Консоль администрирования позволяет администратору принудительно

завершить определенный звонок или группу звонков, удовлетворяющих заданным фильтрам.

Исполнение команды разрешено пользователям из группы Admin.

**Формат:**

```
terminate call <ICID>|<[-dst <oct.oct.oct.oct>] [-src  
<oct.oct.oct.oct>]>
```

, где:

ICID – уникальный номер звонка

-dst - фильтровать по адресу вызываемого шлюза

-src - фильтровать по адресу вызывающего шлюза

oct - октет [0..255]

**Пример:**

```
#>terminate_call 125  
Call #125 has been terminated.  
#>
```

**Команда** `disable gatekeeper (di gk)`

Команда, решающая проблему некорректного завершения звонков, требующих авторизации на привратнике (gatekeeper), при выполнении команды перезагрузки конфигурации из консоли администрирования (reload config).

Команда прекращает использование привратника (gatekeeper). Регистрация на привратнике не закрывается до тех пор, пока не завершатся все текущие звонки через данный привратник (gatekeeper)

**Формат:**

```
disable gatekeeper <gkname1> <gkname2> ...
```

, где:

gkname – имя привратника (gatekeeper), регистрацию на котором необходимо закрыть

Для возобновления регистрации на таком привратнике (gatekeeper) требуется:

- а. Завершение всех активных звонков через данный привратник
- б. Наличие описания привратника в конфигурационном файле (напр. gatekeeper.cfg)
- в. Выполнение команды перезагрузки конфигурации reload config

**Команда** `unregister endpoint [num]`

Команда принудительной «дерегистрации» RAS-пользователя, зарегистрированного на привратнике. При использовании на месте параметра num номера RAS шлюза (из первого столбца таблицы, выводимой по команде show ep) можно осуществить разрегистрацию соответствующего RAS пользователя.

Исполнение команды разрешено пользователям из группы Admin.

**Формат:**

```
unregister endpoint(id endpoint_id | addr endpoint_call_addr[:port])
```

**Сокращенная форма:**

```
un ep(id endpoint_id | addr endpoint_call_addr[:port])
```

, где:

`id` – идентификатор

`addr` – адрес

**Пример:**

```
#> un ep id 1:mera
#> un ep addr 195.99.45.12:1720
#> un ep addr 195.99.45.12
```

Если порт в параметре `addr` не указан, то используется 1720

### 6.3.4.3. КОМАНДЫ ДИАГНОСТИКИ

**Команда** `show call [table][name] (sh ca [ta] [na])`

Команда позволяет просмотреть текущие звонки, а также их подмножество, выделенное по некоторым критериям. Команда доступна пользователям групп Admin, Support и Billing. При введении команды со опцией `[table]` информация отображается в форме таблицы.

(Во избежание искажения таблицы длина строки в консоли должна составлять не менее 112 символов)

```
#> sh ca ta na
```

						source		destination	
call	source_number	dest_number	source	destination	time	from	to	from	to
5	555512345678901>	41512114	h323_in_test_t>	h323_out_out_o>	17	41K	41K	41K	41K
6	555512345678901>	41518258	h323_in_test_t>	h323_out_out_o>	11	27K	27K	27K	27K
7	555512345678901>	41518994	h323_in_test_t>	h323_out_out_o>	5	13K	13K	13K	13K

Если значение поля превышает ширину столбца, значение выводится в урезанной форме и оканчивается на символ >. Величина трафика звонка выводится в килобайтах.

С опцией `[name]` для шлюзов вместо IP адресов выводятся имена шлюзов.

**Формат:**

```
show call [-dst <oct.oct.oct.oct>] [-src
```

```
<oct.oct.oct.oct>]
```

,где:

-dst - фильтровать по адресу вызываемого шлюза

-src - фильтровать по адресу вызывающего шлюза

oct - октет [0..255]

Выдает список звонков, удовлетворяющих условию в виде CSV. Для каждого звонка выводятся поля.

**Таблица 6-3: Поля, выводимые командой show call для группы звонков**

Поле	Пояснение
<b>ICID</b>	Уникальный номер звонка
<b>Number_in</b>	Номер вызывающего абонента
<b>Number_out</b>	Номер вызываемого абонента
<b>Ip_in</b>	IP адрес шлюза, с которого получен звонок
<b>Ip_out</b>	IP адрес шлюза, на который отправлен звонок
<b>Talk_time</b>	Общее время разговора в секундах

```
show call <идентификатор>
```

Выводит активный звонок с заданным идентификатором в виде CSV. Для звонка выводятся поля.

**Таблица 6-4: Поля, выводимые командой show call для заданного звонка**

Поле	Пояснение
<b>ICID</b>	Уникальный номер звонка
<b>Number_in</b>	Номер вызывающего абонента
<b>Number_out</b>	Номер вызываемого абонента
<b>Ip_in</b>	IP адрес шлюза, с которого получен звонок
<b>Ip_out</b>	IP адрес шлюза, на который отправлен звонок
<b>CallID</b>	Идентификатор звонка
<b>ConfID</b>	Идентификатор конференции, в которой участвовал данный звонок
<b>Time_setup</b>	Время начала звонка
<b>Time_connect</b>	Время начала разговора
<b>Talk_time</b>	Общее время разговора в секундах
<b>Bytes_in_src</b>	Количество полученных байтов от вызывающего абонента



Поле	Пояснение
<b>Bytes_out_src</b>	Количество переданных байтов от вызывающего абонента
<b>Bytes_in_dst</b>	Количество полученных байтов от вызываемого абонента
<b>Bytes_out_dst</b>	Количество переданных байтов от вызываемого абонента
<b>Codecs_used_in</b>	Кодеки, использованные вызывающим абонентом
<b>Codecs_used_out</b>	Кодеки, использованные вызываемым абонентом

**Пример:**

```
#> show call -dst 195.98.57.10
124, 78312778899, 70951231212, 195.98.57.11, 195.98.57.10, 400,
100543, 100453
125, 78127867234, 78312778890, 195.98.57.103, 195.98.57.10, 120,
2543, 3443
126, 70953456732, 78312778895, 195.98.57.12, 195.98.57.10, 7300,
500346, 500143
#> show call 125
```

```
Call #125
Number in : 78127867234
Number out : 78312778890
Ip in : 127.0.0.3
Ip Out : 127.0.0.1
CallID: 0102030405060708090a0b0c0e0f
ConfID: 0102030405060708090a0b0c0e0f
Time setup: 10:45:15
Time connect: 10:45:18
Talk time : 120
Bytes in : 2543
Bytes out: 3443
Codec used in: G.729
Codec used out: G.723
#>
```

**Примечание:** при использовании команды `show call table (sh ca ta)`, рекомендуемая ширина экрана консоли должна составлять не менее 112 символов.

**Команда** `show dial`

Команда позволяет проследить путь маршрутизации звонка с заданными вызываемым (вызывающим) номерами. Команда ищет оптимальный **dial peer** для данного сочетания номеров в кэшированном плане набора. В процессе исполнения команды система сообщает о принятых решениях работы алгоритма, найденный шлюз, а также всю результирующую информацию, необходимую для установления звонка.

Команда доступна пользователям групп Admin, Support.

**Формат:**

```
show dial <number_out> [number_in] [number_in group_in]
,где:
```

number\_out - номер вызываемого абонента (обязательный параметр)  
number\_in - номер вызывающего абонента (необязательный параметр)  
group\_in - группа вызывающего абонента (необязательный параметр)

**Пример:**

```
>show dial 5555 3333 TEST
Seach dialpeer for
  Number In :3333
  Number Out :5555
  Group :TEST

Check dial peer MOBILE
  failed
Check dial peer STOP
  failed
Check dial peer KASAN
  failed
Check dial peer KIS
  failed
Check dial peer MOSCOW
  failed
Check dial peer SAMARA
  failed
Check dial peer TEST
  valid
  try to find gateway (STATIC:TEST)
  gateway found
Gateway/Endpoint : Alias: TEST
  group           : DEFAULT
  ip_precedence: 3
  proxy_type      : 1
  capacity        : 300
  current_load    : 0
  accessibility   : 1
  target_ip       : 195.98.57.10:1720
  translate addresses
  SUCCESS
Number In :3333
Number Out :5555
#>
```

**Команда** `show disconnect (sh dc)`

...отражает статистику причин завершения звонков с краткой расшифровкой кодов завершения. Применяется в отладочных целях.

**Команда** `show dp`

Команда позволяет просмотреть все объекты набора, успешно загруженные из плана набора, либо конкретное направление. Команда доступна пользователям групп Admin , Support и Billing.

**Формат:**

```
show dp [имя]
```

,где имя – имя конкретного объекта набора

Выдает один объект или список объектов набора в формате, указанном в таблице.

**Таблица 6-5: Поля, выводимые командой `show dp` для каждого объекта набора**

Поле	Пояснение
<code>dialpeer</code>	Псевдоним объекта набора
<code>gateway</code>	Псевдоним шлюза
<code>priority</code>	Приоритет шлюза
<code>hunt_stop</code>	Флаг, останавливающий дальнейший просмотр объектов, если данное направление подходит, но соответствующий ему шлюз недоступен или перегружен.
<code>dst_pattern</code>	Шаблоны номера вызываемого и вызывающего абонентов в формате regex.
<code>src_pattern</code>	
<code>src_translate</code>	Трансляции номеров вызываемого и вызывающего абонентов.
<code>dst_translate</code>	
<code>group_allow</code>	Группа разрешенных шлюзов
<code>group_deny</code>	Группа запрещенных шлюзов

**Пример:**

```
>show dp MOBILE
Dialpeer      : MOBILE
gateway       : STATIC:NNOV_KIS
prio          : 2
hunt_stop     : 1
dst_pattern   : 78312778797
src_pattern   : [0-9]*
dst_translate : 78312778797/78319041303
...
>
```

**Команда** `show gw`

Команда позволяет получить информацию о текущем состоянии шлюза если его адрес задан иначе, выводится информация обо всех шлюзах, содержащихся в файле данных шлюзов.

Команда доступна пользователям групп Admin, Support .

**Формат:**

```
show gw [ip_address]
```

где `ip_address` – адрес шлюза (необязательный параметр)

**Пример:**

```
show gw
Gateway/Endpoint : Alias: SU29 E164: 78312778795
group            : DEFAULT
ip_precedence    : 3
```

```

    proxy_type      : 1
    capacity        : 2
    current_load    : 0
    accessibility   : 1
    target_ip       : 212.45.29.51:1720
Gateway/Endpoint  : Alias: TEST
    group           : DEFAULT
    ip_precedence   : 3
    proxy_type      : 1
    capacity        : 300
    current_load    : 0
    accessibility   : 1
    target_ip       : 195.98.57.10:1720
#>

```

**Команда** `show incoming <IP address> (sh in <IP address>)`

Команда выводит идентификатор окончного пункта (endpoint ID) динамически регистрирующегося пользователя либо имя секции (записи) в файле статически зарегистрированных шлюзов (gateway.cfg), содержащей параметры оригинирующего шлюза.

Команда доступна пользователям групп Admin, Support .

**Формат:**

```

show incoming <ip_address>
show in <ip_address>

```

где `ip_address` – адрес шлюза (обязательный параметр)

**Пример:**

```

#> sh in 195.27.88.12

Incoming Endpoint resolution for address 195.27.88.12
endpoint_id : 0:gway

#>

```

**Команда** `show ipload (sh ipload)`

... показывает расчет нагрузки по локальным IP адресам.

**Формат:**

```

show ipload

```

**Пример:**

```

#> sh ipload

```

```

#> sh ipload
Traffic load statistics for local addresses:
Mode:  if all local address overloaded use default address
Address :175.50.98.121
    Rx (kb/s): 1 - unlimited
    Tx (kb/s): 1 - unlimited
Address :175.50.98.122
    Rx (kb/s): 1 - unlimited
    Tx (kb/s): 1 - unlimited

```

**Команда** `show route`

...produces the loaded table of call routing and can also show the local address chosen by the proxy server for the called party

**Формат:**

```
show route
```

**Пример:**

```
sh route 195.99.55.23
```

**Команда** `show stat [full | src | ds t | gw | dp ]`

С помощью следующей команды администратор может получить статистику работы сервера с точностью до отдельного шлюза (оригинирующего или терминирующего) и объекта набора (dialpeer).

Исполнение команды разрешено пользователям из групп Admin, Support.

**Формат:**

```
#> show stat full
```

Статистика выводится следующими вариантами команды:

```
show stat full
```

выводит полную статистику.

```
show stat src|dst|gw|dp
```

отображает статистику по категориям интересующих объектов (оригинирующие, терминирующие шлюзы и объекты набора)

```
show stat src|dst|gw|dp <name>
```

подкачать подробную статистику по конкретному объекту.

Командами `show stat full` и `show stat src|dst|gw|dp`. Статистика выводится в табличной форме.

Ниже приводятся примеры таких таблиц

```
#>show stat full
```

----- Originators -----									
	ASR %	ACD	QoS	MaxL	ActC	TotalTime	TotalCal	Normal	Failed
h323_in	100.00	0:00:04	0	1	0	0:00	1	1	0
----- Terminators -----									
	ASR %	ACD	QoS	MaxL	ActC	TotalTime	TotalCal	Normal	Failed
h323_out	100.00	0:00:04	0	1	0	0:00	1	1	0

```

----- Dialpeers -----
DP          ASR %   ACD    MaxL ActC TotalTime TotalCal  Normal  Failed
100.00    0:00:04  1      0    0:00      1       1       0

#>show stat src
          ASR %   ACD    QoS MaxL ActC TotalTime TotalCal  Normal  Failed
h323_in  100.00  0:00:04  0    1    0    0:00      1       1       0

#>show stat dst
          ASR %   ACD    QoS MaxL ActC TotalTime TotalCal  Normal  Failed
h323_out 100.00  0:00:04  0    1    0    0:00      1       1       0

#>show stat gw
          ASR %   ACD    QoS MaxL ActC TotalTime TotalCal  Normal  Failed
h323_in  100.00  0:00:04  0    1    0    0:00      1       1       0
h323_out 100.00  0:00:04  0    1    0    0:00      1       1       0

#>show stat dp
DP          MaxL ActC TotalTime TotalCal  Normal  Failed
1           1      0    0:00      1       1       0

```

Таблица 6-6: Параметры, отображаемые командой `show stat`

Параметр	Описание
<b>ASR (Average Success Rate)</b>	Средний показатель успешных звонков. MVTS рассчитывает ASR как процент полученный в результате умножения отношения числа успешных вызовов к общему числу вызовов на 100
<b>ACD</b>	Средняя продолжительность звонков
<b>QoS (Quality of Service)</b>	Качество канала, рассчитываемое как отношение количества потерянных RTP пакетов к общему количеству принятых RTP пакетов, умноженное на 100. Так как информация о количестве потерянных и принятых пакетов получается из RTCP пакетов, поступающих ежеминутно, система вычисляет этот параметр для каждой минуты длящегося соединения и в качестве QoS канала выдает наибольший из полученных

Параметр	Описание
	результатов. (следует помнить, что чем меньше значение показателя, тем лучше QoS соединения)
<b>MaxL (Max. Load)</b>	Отмеченный максимум одновременных звонков
<b>ActC (Active Calls)</b>	Кол-во активных звонков на данный момент
<b>TotalTime</b>	Суммарное время звонков (часы/мин.)
<b>TotalCal</b>	Суммарное кол-во входящих и исходящих звонков
<b>Normal</b>	Число успешных вызовов
<b>Failed</b>	Кол-во неуспешных вызовов

Подробную информацию по объекту можно получить, если в команде указать название секции, содержащей описание объекта IP-телефонии.

```
#>show stat gw h323_out
h323_out as gateway
Statistic start time 15.07.03 13:21:19
Last update was at 15.07.03 13:21:23
Current ASR 100.00 %
Current ACD 0:00:04
Average QoS 0
Active calls maximum 1
Current active calls 0
Total calls duration 4
Total calls received 1
Normal calls 1
Zero normal calls 0
Failed calls 0
Zero failed calls 0
Total KBytes received 9
Total KBytes sent 9

Disconnect codes statistics:
Local disconnect code <-> Q931 cause
Originator normal (1) <-> No route to destination (3) 10
Originator normal (1) <-> Normal call clearing (16) 2008
Originator normal (1) <-> No answer from user (user alerted) (19) 56
Originator normal (1) <-> Call rejected (21) 13
Originator normal (1) <-> Normal, unspecified (31) 28
Originator normal (1) <-> Temporary failure (41) 1
Terminator normal (2) <-> Requested circuit/channel not available (44) 722
```

```
Terminator normal (2) <-> Resource unavailable, unspecified (47) 7782
Terminator normal (2) <-> Bearer capability not implemented (65) 5
Dangling call (301) <-> Normal, unspecified (31) 1
```

Вывод статистики на экран завершается парами локальных кодов разъединения с кодами Q931.

**Table 6-7: Параметры, выводимые по команде `show stat gw`**

Параметр	Описание
<b>Statistics start time</b>	Время начала сбора статистики
<b>Last update was at</b>	Время последнего обновления информации
<b>Current ASR</b>	Текущий средний показатель успешных звонков (ASR)
<b>Current ACD</b>	Средняя продолжительность звонка (мин:сек)
<b>Average QoS</b>	Средний показатель качества канала (QoS)
<b>Active calls maximum</b>	Пиковое значение активных звонков
<b>Current active calls</b>	Текущее количество активных звонков
<b>Total calls duration</b>	Общее время вызовов, прошедших через объект IP телефонии (час:мин.)
<b>Total calls received</b>	Суммарное кол-во входящих вызовов
<b>Normal calls</b>	Количество успешных звонков
<b>Zero normal calls</b>	Кол-во успешных звонков с нулевой продолжительностью
<b>Failed calls</b>	Кол-во неуспешных звонков

**Команда** `show stat route [all] [-dst] [-src] [-dp]`  
`(sh st rt [all] [-dst] [-src] [-dp])`

... выводит статистическую информацию по маршрутам (оригинатор → объект набора → терминатор) вызовов. Содержимое столбцов таблицы аналогично статистическим таблицам для оринаторов, объектов набора и терминаторов, за исключением последнего столбца. В нем отражается текущее состояние данного маршрута и процент успешных звонков среди последних 200 звонков. Состояние маршрута отображается следующими кодами:

FA (fully accessible) – нет ограничений на прохождение звонка по этому маршруту

PA (partially accessible) – прохождение звонка по такому маршруту временно разрешено

NA (non-accessible) – прохождение звонка по данному маршруту



заблокировано

Правила перехода маршрута из одного состояния в другое.

FA → NA – ASR по последним 200 звонкам ниже 20%

NA → PA – прошло 30 минут с момента перехода маршрута в состояние NA

PA → NA – ASR последних 20 звонков ниже 20%

PA → FA – ASR последних 20 звонков выше 20%

**Примечание:**

1. Указанные константы (30 минут, 200 и 20 звонков, 20% на данный момент жестко прописаны в MVTS, планируется реализовать их задание в конфигурации)
2. На данный момент механизм такого интеллектуального блокирования маршрута работает только виртуально, лишь сохраняя в журнале записи о моменте возможного перехода из одного состояния в другое и причине этого перехода. Реально на данный момент времени маршруты не блокируются.
3. При исполнении команда выводит данные только по 250 направлениям, и, если информация по необходимым шлюзам отсутствует, уточните свой запрос при помощи одного из следующих аргументов.

Аргумент `-src` используется для задания начала имени оригиналирующего шлюза на данном маршруте (направлении)

Аргумент `-dst` используется для задания начала имени терминирующего шлюза на данном маршруте (направлении)

Аргумент `-dp` применяется для задания начала имени объекта набора на данном маршруте (направлении)

Задавая команду с опцией `all` (`sh st rt all`), можно вывести подробную статистику по маршрутам. Поскольку выводимой информации может быть много, то рекомендуется сохранять вывод этой команды в файл, а уже потом анализировать. Как это сделать:

```
#> ./mp_shell.x sh st rt all > route_stat.txt
```

Пример:

```
#> sh st rt -src TORONTO
```

			ASR % (std)	ACD	QoS	MaxL	ActC	TotalTime	Normal	FailedStatus
TORONTO	MVKS_SYNTET	mvks_syntet	100 ( 88)	0:04:23	46	2	1	5:50	80	0 FA(88)
TORONTO	R192_NNOV	DELOVAYA	88 ( 81)	0:10:23	0	2	0	16:57	98	13 FA(81)
TORONTO	RODION_192	DELOVAYA	100 ( 50)	0:01:38	0	1	0	0:03	2	0 FA(50)
TORONTO	TORONTO	TORONTO	0 ( 0)	0:00:00	0	1	0	0:00	0	3 FA(0)

**Примечание:** MVTS рассчитывает ASR (средний показатель успешных вызовов) как процент, получаемый в результате умножения соотношения общего числа успешных звонков к общему числу неудачных вызовов на 100.

Общепринятый показатель ASR (выводимый в скобках рядом с ASR полученным MVTS) рассчитывается как отношение числа звонков ненулевой продолжительности к общему количеству вызовов.

**Команда** `show stat file (sh st file)`

При выполнении команда генерирует файл с полной статистической информацией и выводит на экран имя этого файла.

**Команда** `show stat param (sh st pa)`

выводит значения всех параметров работы статистики, в том числе и количество шлюзов, объектов набора и направлений в статистике

```
#>sh st pa
Statistics parameters:
file period      : 86400
file name       : /usr/local/xpgk/stat/stat
file attribute   : 660
no empty flag   : 1
reset after save : 0
span time       : 0
alert ASR email : vozyakov@mera.ru
email alert ASR : 30
statistics email : vozyakov@mera.ru
stat email command : ((printf "From: mvts@\nTo: $EMAIL$\nSubject:
$SUBJECT$\n\n";cat $FILE$) | /usr/sbin/sendmail -B8BITMIME -f mvts@ -F$FROM$
$EMAIL$)
stat email subject : MVTS_statistic
stat email from    : MVTS_KIS_rack
route stat enable  : 1
route call radix   : 500
```

```

route suspend time : 1800
route sampl.number : 50
route minimum ASR  : 20
stat renewal time  : 43200
smart routing      : 0
--- Total objects in statistics ---
originators       : 15
terminators       : 12
gateways          : 20
dialpeers         : 21
routes            : 91

```

**Команда** `show ep [number]`

С помощью команды `show ep` введенной без аргумента администратор может получить список всех конечных пунктов (end points) в виде таблицы. При указании аргумента `[number]` будет выведена таблица, содержащая информацию только того конечного пункта, номер которого соответствует числу, указанному в виду аргумента.

**Формат:**

```
show ep [number]
```

**Таблица 6-8: Поля, выводимые командой `show ep`**

Поле	Пояснение
<b>Num</b>	порядковый номер
<b>Gateway/Endpoint</b>	псевдоним конечного пункта
<b>Endpoint ID</b>	идентификатор конечного пункта
<b>Username</b>	Имя пользователя
<b>dst/src IP</b>	Входящий/исходящий IP-адрес

**Пример:**

```

#>sh ep
Num : Gateway/Endpoint : Endpoint ID : Username : dst/src IP
-----
 8  : E164 : 757      : ilsy      : ilsy      : 193.28.55.102:1720
15  : E164: 78312773704 :           :           :           :
   : E164: 3000      : nokia1    : nokia1    : :24.101.126.44:1720
16  :           : nokia2    : nokia2    : :24.101.126.43:1721
22  :           : phone1    : phone1    : :192.93.59.100:1720
28  : E164: 3005      : kramnik   : kramnik   : :24.101.4.76:1720
34  : E164: 3003      : my_home   : my_home   : :24.103.152.108:1720

```

## 6.4. HASP для ОПЕРАЦИОННОЙ СИСТЕМЫ LINUX

С апреля 2003г. подавляющее большинство программных комплексов MVTС распространяется с HASP ключами компании **Aladdin Knowledge Systems**, которые обеспечивают контроль за типом программной системы (рабочая/резервная версия), количеством одновременно запущенных экземпляров приложения и их лицензионным статусом.

Версии программного обеспечения, предусматривающие наличие резервного сервера, поставляются с двумя HASP ключами: с ключом для системы, несущей основную рабочую нагрузку и с отдельным ключом для резервного комплекса. Ключи основного и резервного комплексов должны иметь явные внешние отличия, так как продолжительность работы приложения с коммерческим трафиком на аппаратной части с HASP ключом для резервного экземпляра системы ограничена 24 часами непрерывной работы.

Использование HASP ключей делает программный продукт независимым от аппаратной части, которая теперь может быть в любой момент заменена владельцем по собственному усмотрению на новую с аналогичными или лучшими параметрами. Для нормального функционирования программного обеспечения достаточно будет вставить HASP ключ в USB порт нового компьютера и запустить приложение.

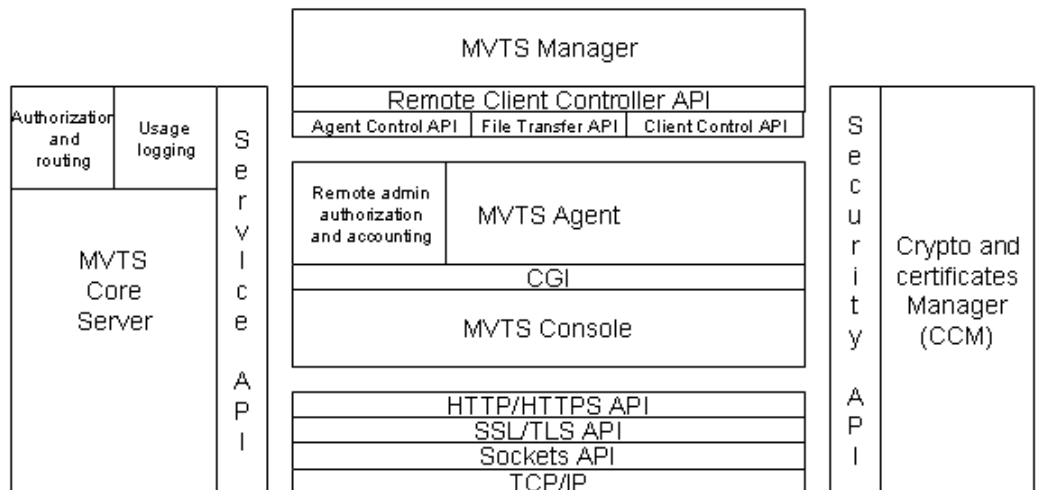
При запуске MVTС время инициализационного обращения программы к ключу составляет 25-30 секунд, по прошествии которого система становится полностью функциональной.

## 6.5. СИСТЕМА УДАЛЕННОГО АДМИНИСТРИРОВАНИЯ MVTS

MVTS Management System (MVTS MS) или система администрирования MVTS представляет собой набор программных средств для удаленного администрирования контроллера соединений MVTS. MVTS MS разрабатывался с целью перенести все задачи по администрированию, управлению работой и настройке системы из традиционного для UNIX подобных операционных систем интерфейса командной строки (консоль `mp_shell`) в дружелюбную для пользователя среду графического интерфейса, основанного на технологии WIN32.

Основными функциональными частями MVTS MS являются клиентское приложение, устанавливаемое на удаленной системе и MVTS агент, находящийся на одном сервере с MVTS.

Клиентское приложение системы удаленного администрирования MVTS MS называется MVTS Manager и работает под управлением операционной системы Windows. Серверная часть MVTS MS, работающая под ОС Linux, называется MVTS агентом. MVTS агент действует в качестве посредника, обеспечивающего связь между MVTS клиентом и MVTS. Связь между клиентской и серверной частью осуществляется по протоколу TCP/IP. Небольшая консоль, прилагаемая в качестве надстройки к клиенту служит для тестирования связи между клиентским приложением и сервер-агентом.



**Рис. 6-1** Архитектура системы удаленного администрирования MVTS MS

Система удаленного администрирования позволяет выполнять следующие задачи:

- Изменение параметров настройки MVTS без прерывания работы

контроллера соединений (редактирование конфигурационных файлов gateway.cfg, gatekeeper.cfg, dialpeer.cfg, user.cfg)

- Создание групп пользователей с разделением функций администрирования и прав доступа (admin, billing, support)
- Работа в консоли администрирования
- Вывод статистики по шлюзам, объектам набора (dial peers) и направлениям в табличной форме (макс. и текущая нагрузка, общее число обработанных вызовов и их суммарная продолжительность, ASR, ACD и пр.)
- Вывод списков активных звонков и данных об активных звонках (оригинирующий и терминирующий шлюзы, продолжительность звонка, применяемый кодек и т.д.)
- Информация о текущей загрузке
- Списки активных привратников (gatekeepers) и терминальных устройств
- Отображение статистических данных в виде графиков (с возможностью выбора вида графика, настраиваемостью масштаба, отображения цифровых значений кривой, распечатки данных и т.д.)
- SNMP запросы на получение текуще статистики от MVTС сервера
- SNMP мониторинг MVTС сервера
- Шаблоны реуглярных выражений (для наиболее часто встречающихся вариантов трансляции номеров)
- Табличное отображение записей о параметрах звонка (CDR)
- Просмотр трассировочных и отладочных журналов (в формате html)

### **6.5.1. УСТАНОВКА СИСТЕМЫ УДАЛЕННОГО АДМИНИСТРИРОВАНИЯ MVTС MS**

Для развертывания системы удаленного администрирования установите MVTС агента на сервер MVTС и клиентское приложение на компьютере с операционной системой Windows.

Серверную часть системы следует устанавливать пользователю с правами root.

Последовательность установки:

1. Войдите в систему с правами root
2. Распакуйте архив с программным обеспечением

3. Перейдите в каталог с названием `mvts_management/` и запустите инсталляционный скрипт набрав команду:  

```
#> ./configure
```
4. Чтобы сконфигурировать серверную часть программы, следуйте указаниям, появляющимся на экране
5. Завершите установку набрав команду:  

```
#> make install
```
6. Чтобы запустить сервер-агента MVTs MS наберите:  

```
#> /sbin/service mvtsagnt start
```

Процедура установки графического интерфейса и клиентской части системы практически ничем не отличается от действий по установке любых других приложений для ОС Windows, которые вы, несомненно, выполняли много раз.

Установите графический интерфейс MVTs manager, запустив на исполнение программу инсталляции **SetupMVTsmgr.exe**. Следуйте указаниям инсталлятора на экране.

Во время установки инсталлятор по умолчанию помещает файлы программы в папку `C:\Program Files\MVTs Manager folder`.

Ссылки на исполняемые файлы из пакета графического интерфейса (консоль и сам MVTs Manager) появятся в главном системном меню. Эти программы можно запускать на исполнение как любое приложение Windows указав при помощи мыши путь: **Start -> Programs -> MVTs Manager -> mpsh** (для запуска консоли администрирования) или **Start -> Programs -> MVTs Manager -> MVTs manager** (для запуска графического интерфейса).

## 6.6. WEB-ИНТЕРФЕЙС ТЕКУЩЕГО КОНТРОЛЯ WEB MONITOR

Web-интерфейс текущего контроля MVTС Web Monitor представляет собой web-приложение удаленного доступа к оперативной статистике MVTС сервера при помощи web-клиентов (например, MS Internet Explorer или Opera Web browser). Для обеспечения гибкости применения система учетных записей пользователей в Web Monitor никак не связана с учетными записями графического интерфейса MVTС Manager. Учетные записи в Web-интерфейсе создаются и редактируются независимо от учетных записей GUI.

Web Monitor для MVTС включен в комплект поставки программного обеспечения и устанавливается вместе с серверной частью графического интерфейса администратора MVTС. Вы имеете полный доступ к Web-интерфейсу сразу же после установки и запуска MVTС агента. Во время установки приложения обратите внимание (запомните или запишите) на IP адрес и порт, которые появятся на экране среди сообщений инсталлятора. Именно этот порт будет «слушать» MVTС агент (по умолчанию 1730) при работе с Web-интерфейсом.

### 6.6.1. КОНФИГУРИРОВАНИЕ WEB-ИНТЕРФЕЙСА И РАБОТА С НИМ

После установки Web Monitor'a имеется единственная учетная запись admin с паролем admin, которая предоставляет доступ к интерфейсу и права на уровне администратора для работы в нем. Права администратора включают неограниченный доступ к статистике по всем видам VoIP объектам, сконфигурированных в MVTС, а также права на создание, удаление и редактирование учетных записей пользователей.

Для доступа к web-интерфейсу MVTС введите адрес (или доменное имя) MVTС и номер порта в строке Address web-браузера. Обратите внимание на правильное указание протокола перед группой адрес:порт, например:

**https://<server IP address>:1730** или **https://<servername.com>:1730**

После ввода адреса в окне web-браузера появится форма для ввода имени и пароля на вход как на Рис. 6-2.





Рис. 6-2 Диалог авторизации web-интерфейса

При правильно введенных имени и пароле вы попадаете на страницу общей статистики, на которой расположены три таблицы:

**Статистика оринаторов**

**Статистика терминаторов и**

**Статистика объектов набора (dialpeers)**

В целом web-интерфейс доступа к статистике MVTС интуитивно понятен, а потому ограничимся замечаниями по функциям и возможностям не очевидным с первого взгляда.

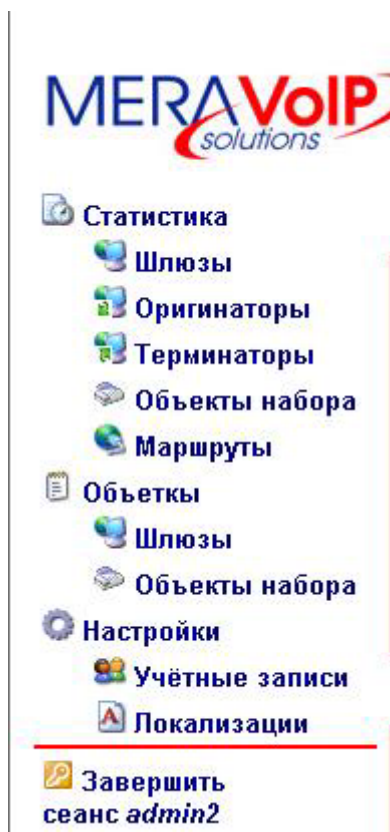


Рис. 6-3 Главное меню Администратора

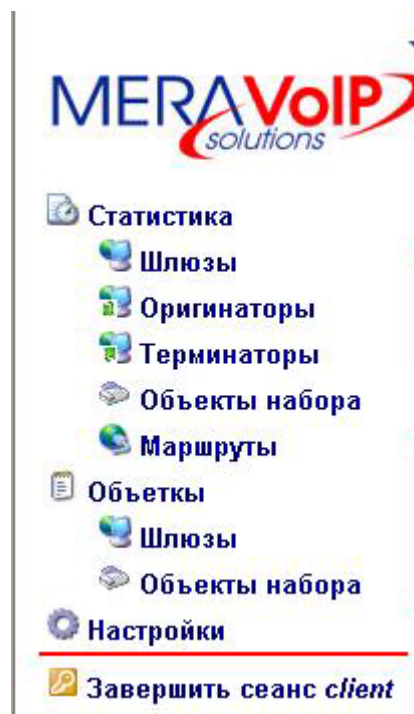


Рис. 6-4 Гл. меню клиента

Главное меню с левой стороны от таблиц общей статистики различается при входе с правами Администратора и клиента. Вид на Главное меню администратора и клиента показан на рис. 9 и 10. В меню клиента отсутствует пункт **Учетные записи**.

В последующих пояснениях имеется в виду интерфейс Администратора (интерфейс при входе с правами клиента будет идентичен за исключением тех ограничений в доступе к количеству объектов, которые определены Администратором).

Пункт меню **Статистика** служит для вызова страницы с полными данными по всем VoIP объектам, сконфигурированным в MVTs, представленными в табличной форме с информацией по шлюзам, инициаторам вызовов, терминирующим пунктам и объектам набора (dial peers). Подпункты пункта **Статистика (Шлюзы, Оригинаторы, Терминаторы, Объекты набора (dial peers))** открывают страницы с таблицами статистики по соответствующим объектам. (Пользователь с правами клиента будет иметь доступ к статистике только по тем объектам, которые разрешит Администратор при создании/редактировании учетной записи клиента)

Пункт **Объекты** вызывает страницу со списком объектов доступных для просмотра статистики. Подпункты выполняют ту же функцию, но только в отношении одноименной категории.

Пункт меню **Настройки** вызывает диалог персональных настроек, который позволяет сменить пароль доступа и выбрать язык интерфейса. Доступные варианты языка интерфейса можно создать, путем редактирования содержимого текстового файла (подробно см. в разделе 6.6.2).

Пункт **Локализация** вызывает страницу с шаблонами языкового оформления web-интерфейса.

Пункт **Учетные записи** – это инструмент администратора, при помощи которого он создает учетные записи пользователей, определяет их права и решает какие объекты будут доступны клиенту для просмотра статистики.

<b>Учётные записи</b>	
Учётная запись	Описание
admin client admin2	MVTs administrator account

[Создать учётную запись](#)

**Рис. 6-5** Список существующих учетных записей

Для создания новой учетной записи щелкните по кнопке **Создать учетную запись**.

Появляющийся диалог позволяет определить атрибуты и права присущие учетной записи. Завершив настройки щелкните по кнопке **Применить**, если желаете сохранить запись в том виде, каком она была

создана.

Диалог учетной записи кроме того можно вызвать двойным щелчком мыши по любой из уже существующих записей в списке.

**client :: Редактирование учётной записи**

Общие настройки	
Имя	client
Описание	
<input checked="" type="checkbox"/>	Включена
Пароль	
<input type="checkbox"/>	Сменить пароль
Права доступа	
<input type="checkbox"/>	Администратор сервера MVT5
<input type="checkbox"/>	Может сбрасывать статистику
Шлюзы	
<input type="checkbox"/>	LANIT
<input type="checkbox"/>	LAnck
<input checked="" type="checkbox"/>	MERANETWORKS_OFFICE
<input checked="" type="checkbox"/>	MERATRON
<input checked="" type="checkbox"/>	MOSTCOM
<input type="checkbox"/>	PROXY_123
<input checked="" type="checkbox"/>	RODIONOVA192
<input checked="" type="checkbox"/>	ROSNET
<input type="checkbox"/>	SDV
<input type="checkbox"/>	SKVAIR
<input type="checkbox"/>	SKVAIR2
<input type="checkbox"/>	SOLOSH
Планы набора	
<input type="checkbox"/>	778705
<input type="checkbox"/>	778709
<input type="checkbox"/>	BOSANOVA_test
<input checked="" type="checkbox"/>	COMPTOK
<input type="checkbox"/>	ENDPOINTS
<input type="checkbox"/>	IMARSAT
<input type="checkbox"/>	KIS
<input checked="" type="checkbox"/>	MOSTCOM
<input type="checkbox"/>	MOSTCOM_INT
<input checked="" type="checkbox"/>	MVKS_DEMO
<input type="checkbox"/>	MVKS_DEMO2
<input type="checkbox"/>	MVKS_SDV
<input checked="" type="checkbox"/>	MVKS_SYNTETICA

**Рис. 6-6** Диалог редактирования учетной записи

В режиме редактирования диалог учетной записи появляется с уже оформленными свойствами, изменить которые можно выставляя или сбрасывая отметку в соответствующих флаговых кнопках (checkboxes). Основными настройками в диалоге учетной записи являются:

- Опция Активировать/Заблокировать учетную запись (флажок **Включена**)
- Предоставление/ликвидация прав на просмотр статистики по объектам (флаговые кнопки у названий шлюзов, объектов набора, маршрутов)
- Предоставление/ликвидация права на изменение пароля (флажок

**Смена пароля)**

- Предоставление/ликвидация права на сброс статистики по объектам (флажок **Может сбрасывать статистику**)

Флажок **Администратор сервера MVTС** – это атрибут учетной записи, определяющий полноту прав пользователя и являющийся признаком учетной записи пользователя с правами Администратора (т.е. по умолчанию обладающего неограниченными правами доступа к информации и возможностью создавать, удалять и редактировать все записи).

**6.6.2. РЕДАКТИРОВАНИЕ И ИЗМЕНЕНИЕ ЯЗЫКА ИНТЕРФЕЙСА**

Языковое оформление web-интерфейса легко разнообразить или изменить применительно к индивидуальным запросам клиента. Это достигается за счет редактирования файла-шаблона подписей и всплывающих подсказок интерфейса. Редактировать файл следует в текстовом редакторе, позволяющем записывать данные в кодировке UNICODE/UTF-8, UNICODE/UTF-16.

Структура файла оформления интерфейса:

```
LANGUAGE =
ACD =
ASR =
Account =
Accounts =
Active_calls =
Administrator_of_MVTS_server =
Apply =
Are_you_sure =
Can_reset_statistics =
Cancel =
Change_password =
Common_settings =
Common_statistics =
Create_account =
Creating_account =
Delete =
Delete_account =
Description =
Detailed_dialpeer_statistics =
Detailed_gateway_statistics =
.....
..... и т.д.
```

Название языкового варианта интерфейса, появляющееся в выпадающем меню, должно находиться в строке LANGUAGE.

Чтобы изменить язык оформления интерфейса и всплывающих подсказок перейдите на страницу **Локализация**, щелкнув по одноименной ссылке в главном меню.

Локализация	
Модуль	Язык
Шаблон	
en.txt	English
ru.txt	Russian
esp.txt	Spanish

Ваш модуль:

Рис. 6-7 Страница локализации web-интерфейса

Щелчком мыши по названию **Шаблон** в первой колонке таблицы вызовите шаблон подписей и всплывающих подсказок и запишите его в любом удобном для работы каталоге.

Отредактируйте файл, вводя необходимые названия и тексты подсказок в поля шаблона после символа '='.

#### Например:

LANGUAGE	= Spanish
ACD	= ACD
ASR	= ASR
Account	= Cuenta
Accounts	= Cuentas
Active_calls	= Llamadas activas
Administrator_of_MVTS_server	= Administrador del servidor
MVTS	
Apply	=
Are_you_sure	= Esta seguro?
Can_reset_statistics	= Puede reajustar estadística
Cancel	= Cancelación
Change_password	= Cambiar la contraseña
... ..	
и т.д.	

Название варианта локализации, которое появляется в качестве одного из пунктов выпадающего списка на странице **Персональные настройки**, во избежании путаницы, рекомендуется всегда вводить в поле LANGUAGE на английском языке (напр. LANGUAGE = Russian).

Запишите готовый файл в кодировке UNICODE/UTF-8,-16 под любым другим именем. Для того, чтобы загрузить файл обратно на сервер щелкните мышью по кнопке **Browse** (эта кнопка не поддается локализации, так как не является частью Web Monitor'a) на странице **Локализация** и, выбрав вновь отредактированный файл языкового оформления интерфейса в вызванном файловом диалоге, загрузите его на сервер щелчком по кнопке **Загрузить**.

После успешной загрузки имя нового варианта оформления интерфейса должно появиться в колонке **Модуль** в таблице **Локализация**.

Для того чтобы изменить оформление интерфейса, перейдите на страницу **Персональные настройки** (вызываемую щелчком по ссылке **Настройки**), выберите из выпадающего списка нужный язык

интерфейса и нажмите на кнопку **Сохранить настройки**.

**Персональные настройки**

Язык интерфейса	Russian
<input type="checkbox"/>	English
	Russian
	Spanish

**Рис. 6-8** Выпадающий список вариантов языка

**Configuración personal**

Idioma de interfaz	Spanish
<input type="checkbox"/>	Cambiar la contraseña

**Рис. 6-9** Новый язык интерфейса (испанский)

## 7. ВЫЯВЛЕНИЕ И УСТРАНЕНИЕ СБОЕВ В РАБОТЕ

Проблема	Вероятная причина	Способы устранения
Звонки через MVTС не проходят, несмотря на то, что флаг общего доступа установлен в значение 1 (public_access=1) и корректность записей в файле плана набора и файле данных о шлюзах.	Вероятно включен режим авторизации через RADIUS сервер (auth_enable=1)	Отключите режим авторизации через RADIUS сервер, сбросив флаг авторизации (auth_enable=0)
MVTС не запускается. Код завершения 0.	Данная копия программного обеспечения рассчитана на другую платформу.	Используйте программное обеспечение, на компьютере, для которого оно предназначено.

## ПРИЛОЖЕНИЕ А: ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

### ОПРЕДЕЛЕНИЯ

<b>CDR-файл</b>	Файл, содержащий записи о параметрах звонков (время установления звонка, время окончания звонка и т.д.).
<b>RAS-пользователь</b>	Абонент сети IP телефонии (шлюз, терминал), регистрирующийся на привратнике по RAS протоколу.
<b>MVTS (MERA VoIP Transit Softswitch)</b>	Дополнительное устройство, подключаемое к сети IP-телефонии, основной задачей которого является маршрутизация трафика IP-телефонии между шлюзами.
<b>Кодек (codec)</b>	Алгоритм кодирования голоса.
<b>Объект набора (dial peer)</b>	Адресуемая точка звонка. В цифровой телефонии существуют два вида объектов набора (dial peers): объект обычной телефонной сети (POTS dial peer) и адресуемый объект голосовой связи по Интернет протоколу (VoIP dial peer)
<b>Система начисления платы (billing system)</b>	Система, ведущая статистику использования услуг MVTS клиентами и начисляющая по результатам этой статистики причитающуюся плату.
<b>Файл протокола работы (runtime journal)</b>	Текстовый файл, содержащий записи о событиях, состоянии MVTS, смене состояний звонков с указанием времени.
<b>Шлюз (gateway)</b>	Устройство, подключенное к IP-сети и к телефонной сети (PBX/PSTN) и выполняющее следующие функции: - ответ на вызов вызывающего абонента PBX/PSTN - установление соединения с удаленным шлюзом - установление соединения с вызываемым абонентом PBX/PSTN - сжатие, пакетирование и восстановление голоса (или факс-сигнала)
<b>Привратник (gatekeeper)</b>	H.323 устройство в локальной сети, которое осуществляет трансляцию адресов и контролирует доступ в сеть H.323 терминалов и шлюзов. Привратник может предоставлять H.323 терминалам и шлюзам и другие виды сервиса, такие как управление пропускной способностью и обнаружение шлюзов. Привратник ведет реестр устройств в мультимедийной сети. При запуске устройства регистрируются на привратнике и запрашивают у привратника разрешение на звонок.

### СОКРАЩЕНИЯ

<b>АОН</b>	<b>Автоматическое определение номера</b>
<b>ACD</b>	(Average Call Duration) Средняя продолжительность вызовов
<b>ASR</b>	(Average Success Rate) Средний показатель успешных вызовов



	(см. Таблица 6-6)
<b>CDR</b>	<b>Call Detail Record</b> (Запись о параметрах звонка)
<b>CSV</b>	<b>Coma Separated Values</b> (Значения, разделенные запятыми)
<b>FTP</b>	<b>File Transfer Protocol</b> (Протокол передачи файлов)
<b>Oct</b>	<b>Octet</b> – октет
<b>QoS</b>	(Quality of Service) Качество канала, рассчитываемое как отношение количества потерянных RTP пакетов к общему количеству принятых RTP пакетов, умноженное на 100. Так как информация о количестве потерянных и принятых пакетов получается из RTCP пакетов, поступающих ежеминутно, система вычисляет этот параметр для каждой минуты длящегося соединения и в качестве QoS канала выдает наибольший из полученных результатов. (следует помнить, что чем меньше значение показателя, тем лучше QoS соединения)
<b>RAS</b>	<b>Registration, Admission, and Status protocol.</b> (протокол регистрации, допуска и состояния). H.225 сигнальный протокол, по которому осуществляется управляющий обмен между конечными точками сети и привратником. RAS сигнальная функция предусматривает процедуры обмена между шлюзом и привратником по поводу регистрации, допуска, управления пропускной способностью, статуса и рассоединения.
<b>RTCP</b>	<b>Real-time Transport Control Protocol</b> – управляющий протокол реального времени (протокол управления передачей в реальном времени)
<b>RTP</b>	<b>Real-time Transport Protocol</b> – Транспортный протокол реального времени
<b>SCP</b>	<b>Secure Copy</b> (Защищенный/закодированный экземпляр)
<b>VSA</b>	<b>Vendor Specific Attribute</b> – провайдер-зависимый (провайдер-специфический) признак (атрибут)
<b>MVTS</b>	<b>MERA VoIP Transit Softswitch</b>

## СИНТАКСИС РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

В двойных кавычках далее будут употребляться значения, выдаваемые регулярными выражениями, а в одинарных - синтаксис регулярных выражений.

### Искомые выражения

Выражением может быть один символ или последовательность символов, заключенных в круглые или квадратные скобки. Особенности использования скобок будут описаны ниже.

### Классы символов (Character class)

Используя квадратные скобки, можно указать группу символов (это называют классом символов), для поиска. Например, конструкция '1[23]45' соответствует словам <1245> и <1345>, т.е. словам, начинающимся с <1>, за которым следуют <2> или <3>, и заканчивающимся на <45>.

Возможно и обратное, то есть, можно указать символы, которых не должно содержаться в найденной подстроке. Так, '[^1-6]' находит все символы, кроме цифр от 1 до 6.

### Квантификаторы, они же умножители (Quantifiers)

Если неизвестно, сколько именно знаков должна содержать искомая подстрока, можно использовать спецсимволы, именуемые мудреным словом квантификаторы (quantifiers).

Например, можно написать "1234+5", что будет означать слово, начинающееся с "123", со следующими за ним одно или несколько "4", и заканчивающееся на "5". Следует понять, что квантификатор относится к предшествующему выражению, а не отдельному символу.

:

### Таблица: Список квантификаторов

Символ	Описание
*	Соответствует 0 или более вхождений предшествующего выражения. Например, '12*' соответствует "1" и "122".
+	Соответствует 1 или более предшествующих выражений. Например, "12+" соответствует "12" and "122", но не "1".
?	Соответствует 0 или 1 предшествующих выражений.



## Вариации и группировка

Символ '|' можно использовать для перебора нескольких вариантов. Использование этого символа совместно со скобками - '(...|...|...)' - позволяет создать группы вариантов.

Скобки используются для "захвата" подстрок для дальнейшего использования и сохранения их во встроенных переменных \$1, \$2, ..., \$9.

## Обратные ссылки

Мы уже говорили об одной из важнейших возможностей регулярных выражений – способность сохранения части соответствий для дальнейшего использования. Кстати, избежать этого можно с помощью использования '?:'.

Например,

```
$test = "Today is monday the 18th.";
$test =~ m/([0-9]+)th/
сохранит "18" в $1, а
```

```
$test = "Today is monday the 18th.";
$test =~ m/[0-9]+th/
ничего не станет сохранять - из-за отсутствия скобок.
```

```
$test = "Today is monday the 18th.";
$test =~ m/(?:[0-9]+)th/
также ничего не станет сохранять благодаря использованию оператора '?:'.
```

Следующий пример демонстрирует, как можно использовать эту возможность в операции замены:

```
$test = "Today is monday the 18th.";
$test =~ s/ the ([0-9]+)th/, and the day is $1/
приведет к записи "Today is monday, and the day is 18." в переменную $test.
```

Можно ссылаться на подстроки, уже найденные данным запросом, используя \1, \2, ..., \9.

Следующее регулярное выражение удалит повторяющиеся слова:

```
$test = "the house is is big";
$test =~ s/\b(S+)\b(\s+\1\b)+/$1/
записывает "the house is big" в $test.
```

-----

## 2. Примеры использования регулярных выражений при настройке MVTS

Регулярные выражения применяются при конфигурировании следующих параметров:

### **dialpeer.cfg**

- dst\_pattern, src\_pattern,
- dst\_translate, src\_translate
- dst\_bill\_translate, src\_bill\_translate
- user\_translate

### **user.cfg; gateway.cfg**

- dst\_pattern, src\_pattern,
- dst\_translate, src\_translate,
- in\_dst\_translate, in\_src\_translate

Система проверяет регулярные выражения в сконфигурированных правилах трансляции на присутствие запрещенных символов и удаляет их. В правой части выражения трансляции можно использовать следующие символы: `^0123456789*#\&`

При возникновении ошибки (недопустимый символ в регулярном выражении) результат проверки выводится в трассировочный журнал, который записывается в файл с названием `mp.kernel.sh.log-<date>`

### 2.1 Префиксы в параметрах `dst_patern`, `src_pattern`

Наиболее часто используемые конструкции:

```
dst_pattern=777[0-9]+
```

**(комментарий:** номера, начинающиеся с 777 и состоящие далее из произвольных цифр)

**удачные примеры:** 77711, 777922

**неудачные примеры:** 77811, 7771

```
dst_pattern=777[0-5].[0-9]+
```

**(комментарий:** номера начинающиеся с 777, далее следует любая цифра в диапазоне от 0 до 5 и далее любые цифры)

**уд. примеры:** 77711, 777422

**неуд. примеры:** 777, 77811, 77761, 7775,

```
dst_pattern=.....
```

**(комментарий:** любой шестизначный номер)

**уд.примеры:** 123456, 976065, 1  
**неуд.примеры:** 1111111, 111

## 2.2 Трансляция номеров

При трансляции номеров наиболее часто используется добавление, отрезание и замена префикса.

*Добавить префикс 78 к номеру 12345:*

`dst_translate=12345/78&`    **результат:** 12345 → 7812345

*Добавить префикс 78312 к любому шестизначному номеру:*

`dst_pattern=.....`  
`dst_translate=.*78312&`    **результат:** 123456 → 78312 123456  
**результат:** 654321 → 78312 654321

*Добавить префикс 78312 к номеру, начинающемуся с 777:*

`dst_pattern=777.....`  
`dst_translate=[0-9]*/78312&`  
**результат:** 777123456 → 78312777123456  
**результат:** 777121212 → 78312777121212

*Отрезание префикса*

`dst_translate=095|.*\/\2`  
*(комментарий: «отрезать» префикс 095, от последующей (второй) подстроки)*  
**результат:** 09512345 → 095 | 123456 → 123456

`dst_pattern=8182.....`  
`dst_translate=8182|[0-9]*/\2`  
*(комментарий: «отрезать» префикс 8182, т.е. подстроку 8182, предшествующую второй подстроке)*  
**результат:** 8182123456 → 8182 | 123456 → 123456

*Отрезать префикс и заменить его на иной*

`dst_pattern=8182.....`  
`dst_translate=8182|[0-9]*/777\2`  
*(комментарий: «отрезать» префикс 8182 и вставить подстроку 777, перед второй подстрокой)*  
**результат:** 8182123456 → 8182 | 123456 → 777 123456

`bill_translate=1212|.* /1718\2`

( *комментарий:* «отрезать» префикс 1212 и заменить его на префикс 1718, перед второй подстрокой для целей биллинга)

**результат:** 121212345 → 1212 | 12345 → 1718 | 12345 → 171812345

#### **Убрать символ # из середины строки**

`src_translate=[0-9]*|#|[0-9]* /\1\3`

( *комментарий:* «вырезать» символ #, расположенный между 1-й и 3-й подстроками)

**результат:** 123#45 → 123 | # | 45 → 12345

#### **Отрезать символ #, завершающий строку**

`dst_bill_translate=[0-9]*|#/\1`

**результат:** 123456# → 123456 | # → 123456

#### **Добавить символы в конец строки**

`dst_translate=...|/\177`

( *комментарий:* подставить 77 в конец первой подстроки)

**результат:** 1234 → 123477

## **МАКРОИМЕНА В ШАБЛОНАХ ТРАНСЛЯЦИИ**

В шаблонах трансляции, задаваемых в конфигурационных файлах, могут использоваться следующие макроподстановки:

**\$ani\$** - подставить ANI номер вызывающей стороны

**\$dnis\$** - подставить DNIS данные

**\$user\$** - подставить имя пользователя

**\$bill\_ani\$** - подставить ANI номер для системы биллинга

**\$bill\_dnis\$** - подставить номер вызываемой стороны для системы биллинга.

**\$id\$** – уникальный идентификатор вызова (извлеченный из CDR записи) в формате <time stamp момента запуска MVTs>#<порядковый номер звонка>#

### **Пример:**

---

(поменять местами номер вызывающей и вызываемой стороны)

**dst\_translate=.\*/\$ani\$**  
**src\_translate=.\*/\$dnis\$**

Макроимена **\$bill\_ani\$**, **\$bill\_dnis\$** не могут использоваться в трансляции, предшествующей поиску объекта набора, т.е. в полях, начинающихся с префикса `in_`, (например, `in_dst_translate`, `in_src_translate`)



## АВТОРИЗАЦИЯ ПО ПРОТОКОЛУ RADIUS И ВНЕШНЯЯ МАРШРУТИЗАЦИЯ

### Дополнительные поля в запросах к RADIUS серверу

- **h323-call-id** (идентификатор вызова)
- **h323-gw-id** (идентификатор оригинирующего шлюза)
- **h323-gw-address** (IP адрес оригинирующего шлюза)
- **h323-remote-id** (идентификатор терминирующего шлюза)
- **h323-remote-address** (IP адрес терминирующего шлюза)
- **xpgk-destination-user** (имя терминирующего пользователя)
- **xpgk-h323-id** (H.323 алиас оригинатора звонка)
- **xpgk-src-number-in** (номер оригинатора на входе MVTС)
- **xpgk-src-number-out** (номер оригинатора на выходе MVTС)
- **xpgk-dst-number-in** (номер терминатора на входе MVTС)
- **xpgk-dst-number-out** (номер оригинатора на выходе MVTС)
- **xpgk-initial-incoming-local-address** (локальный IP-адрес, на который был принят Setup от оригинатора)
- **xpgk-selected-incoming-local-address** (выбранный локальный IP-адрес для обмена с оригинатором)
- **xpgk-outgoing-local-address** (выбранный локальный IP-адрес для обмена с терминатором)

### Маршрутизация по запросам к RADIUS серверу

Если в файле **dialpeer.cfg** в поле `gateway` стоит служебное значение `EXTERNAL`, означающее использование динамической маршрутизации от Radius-сервера, то для объекта набора с таким значением перед авторизацией пользователя на Radius-сервер будет посылаться пакет для получения маршрута с полем AV-PAIR "**xpgk-routing-request=1**".

Возможны три варианта ответов от Radius-сервера: -

- получен `reject`, в этом случае звонок завершается; -
- получен `accept` без пути, в этом случае отрабатываются последующие правила планов набора; -
- получен `accept` с путем, такой пакет должен содержать поля со следующими ID: -
  - CISCO VSA ID=251 формате `new_username/new_password` (необязательное поле, работает аналогично полю `override_user v`)

файле **dialpeer.cfg**). Например, если значение поля будет равно `user01/qwerty`, то имя пользователя и пароль для данного звонка будут заменены соответственно на `user01` и `qwerty` –

- CISCO VSA ID=252 в в 5-ти, 6-ти или семизначном формате:

`gateway/proxy_mode/source/dest/src_bill/dst_bill/ip-address[:port]`

, где: `gateway` – имя шлюза (секции) из файла **gateway.cfg**;

`proxy_mode` – режим проксирования:

0 - нет проксирования медиа трафика

1 - есть проксирование медиа трафика;

2 - использовать режим проксирования оригиналирующего шлюза;

3 - использовать режим проксирования терминирующего шлюза;

`source` – номер вызывающего абонента (`src_number`)

`dest` – номер вызываемого абонента, который будет отправлен на терминирующий шлюз (`dst_number`)

`src_bill` – номер вызываемого абонента для системы биллинга;

`dst_bill` – номер вызывающего абонента для системы биллинга;

`ip-address[:port]` – IP адрес для установления соединения, номер порта `[:port]` является необязательным параметром. Если номер порта не указан, то порт по умолчанию - 1720

Первые 5 полей являются обязательными.

Поле с ID=251 должно быть одно. Полей с ID=252 может быть несколько, указанные в них маршруты будут обрабатываться последовательно.

## КОНФИГУРИРОВАНИЕ ЯДРА ОПЕРАЦИОННОЙ СИСТЕМЫ

Пересборка ядра ОС для работы MVTs необходима в связи с тем, что система, как уже упоминалось выше, должна позволять пользователю сервера открывать одновременно не менее 6000 файловых дескрипторов, в то время как это значение ядра в поставляемом пакете операционной системы по умолчанию равняется 1024.

Простая процедура конфигурирования ядра, предусмотренная разработчиками для оптимизации работы ОС с различными устройствами и новыми вариантами операционного окружения, не позволяет изменить это значение, что вынуждает воспользоваться возможностью перекомпиляции ядра.

В настоящем документе будет описана лишь общая последовательность действий по перекомпиляции ядра с учетом того, что подробные инструкции по данной процедуре в деталях приведены в документации по операционной системе.

Для изменения количества одновременно открываемых файловых дескрипторов последовательность действий при перекомпиляции ядра ОС Red Hat Linux следующая:

1. Создать резервную копию системы (загрузочный диск), если таковая отсутствует
2. В исходных кодах заменить значения нужных параметров на новые:
  - a. В каталоге `/usr/src/linux-X.X.XX-X/include/linux` в файле **limits.h** исправить `NR_OPEN`, а в файле **fs.h** исправить `INR_OPEN`, на `8192`.
  - b. в файле **posix\_types.h** исправить `_FD_SETSIZE` на `8192`
3. Скомпилировать ядро с новыми параметрами
4. Обеспечить загрузку вновь скомпилированного ядра при запуске системы.

В отличие от ОС Red Hat, в ОС FreeBSD количество одновременно открываемых файловых дескрипторов (`$maxfiles`) непосредственным образом не задается, а связано со значением `$maxusers`. Зависимость количества одновременно открытых файловых дескрипторов от числа пользователей выражается следующим образом:

$$\text{\$maxfiles} = (20 + \text{\$maxusers} * 16) * 2$$

Таким образом, чтобы ОС FreeBSD имела возможность открывать от 8000 файловых дескрипторов одновременно и более (из расчета 20 сокетов на один звонок), нужно перекомпилировать ядро ОС, определив значение `maxusers` равным не менее 249, т.е.

$$\$maxfiles = (20+249*16)*2 = 8008$$

---

*Примечание: Максимальное допустимое значение переменной `maxusers` в FreeBSD – 384*

---

Для конфигурирования ядра ОС под крупномасштабный сервис с большим количеством одновременно открываемых файловых дескрипторов измените значение параметра `maxusers` в файле `/usr/src/sys/i386/conf/GENERIC` на необходимое, руководствуясь формулой, приведенной выше и перекомпилируйте ядро.

## ДЕИНСТАЛЛЯЦИЯ MVTS

В редких случаях, когда необходимо полностью удалить MVTS из системы выполните в системной консоли следующую последовательность команд:

```
#> rm -rf $H323PROXY_ROOT
#> vi /etc/profile #remove $H323PROXY_ROOT
#> runlevel
#> cd /etc/rc5.d
#> ls |grep mvts
#> rm -rf S50mvts
#> cd /etc/init.d
#> rm -rf mvts
```

## СОВМЕСТИМОСТЬ

### Шлюзы и гейткиперы VoIP

Контроллер соединений *MERA VoIP Transit Softswitch* успешно прошел тестирование со многими решениями ведущих разработчиков H.323-оборудования, среди которых:

- CISCO (IOS-based, ATA 186)
- VocalTec (VGW 1.4f+, VGK 1.3+)
- Samsung (SMG 400, SMG 3200)
- Nortel Networks (BCM, Succession)
- Clarent (Clarent GK)
- D-Link (DG-10xSH)
- BosCom (Bosanova)
- Well-Tech (SmartNode 1200, 1400, 2300, 2400)
- AudioCodes (Mediant-2000, MP-104)
- Quintum (Tenor Digital MultiPath Switch)
- Network Systems Group (NSGate)
- soft-телефоны (Microsoft NetMeeting, VocalTec Phone Lite)

### Биллинговые системы

Поддержка MVTs протокола *RADIUS* позволяет провайдерам самостоятельно выбирать биллинговую систему исходя из множества представленных на рынке решений. Ниже приводится список биллинговых решений, чаще всего используемых нашими клиентами:

- IPSoft Billing
- MIND CTI
- Advanced VoIP Billing System
- T-Soft Billing
- PortaBilling
- прочие биллинговые системы, совместимые с Cisco

## ПРИЛОЖЕНИЕ В: НАИБОЛЕЕ ЧАСТО ВСТРЕЧАЮЩИЕСЯ КОДЫ РАЗЪЕДИНЕНИЯ, ГЕНЕРИРУЕМЫЕ MVTs

### *eCallerNormal=1 eCalledNormal=2*

Звонок завершился нормально с посылкой Release complete от  
оригинирующей/терминирующей стороны соответственно

### *eCallerDropTCP=3 eCalledDropTCP=4*

Звонок завершился без получения Release complete в результате  
обрыва TCP соединения оригинирующей/терминирующей стороной  
соответственно

### *eForceTerminateCall=100,*

Звонок был завершен в принудительном порядке (останов MVTs,  
команда terminate call)

### *eTimeoutTCPConnectH225=101*

Не было установлено H225 соединение с терминирующей стороной в  
течение 3 секунд

### *eTimeoutConnectMsg=102*

Не получено Connect Message в течении 120 секунд

### *eTimeoutRBT=103*

Не получено Alerting Message в течении 30 секунд

### *eDestinationUnreachable=110*

Не найден соответствующий объект набора (dial peer) или ни один из  
требуемых шлюзов не был доступен

### *eFailedTCPConnectH225=112*

Сбой при установлении H225 сессии с терминирующей стороной

### *eInvalidTPKTCaller=115 eInvalidTPKTCalled=119*

Ошибка в заголовке пакета, полученного от оригинатора/терминатора  
звонка. Встречалась в случае некорректной работы соответствующего  
шлюза.

### *eDuplicateCallId=118*

Был получен звонок с Call ID, уже используемом в одном из активных  
соединений (защита от заикливания звонков)

### *eTimeoutSetupMsg=121*

Не был получен Setup Message в течении 15 секунд.

### *eTimeoutRouteAttempt=120*

Не было установлено соединение с терминирующим шлюзом в  
течении 10 секунд после начала роута на этот шлюз.

### *eTimeoutRTPidle=122*

При полном проксировании не было голосового трафика в течении

180 секунд. В этом случае звонок считается зависшим и завершается.

***eFailedTCPConnectH245=123***

Сбой при установлении H245 сессии

***eInvalidSetupMsg=124,***

Неверный Setup Message (в пакете отсутствуют UserUserПоле или первым пришел не Setup Message)

***eMaxRerouteRetries=125,***

Было совершено более 10 попыток рероута (защита от закливания при процедуре look ahead routing)

***eRadiusAdmissionCallerReject=200***

***eRadiusAdmissionCalledReject=207***

Radius-сервером отклонена авторизация оигинатора/терминатора звонка

***eGkClientAdmissionTimeout=201,***

на запрос Admission Request от удаленного привратника (gatekeeper) не получен ответ в течении 10 секунд

***eGkClientAdmissionReject=203,***

запрос Admission Request отклонен удаленным привратником (gatekeeper)

***eSourceGatewayAniReject=205,***

Параметр src\_number оигинатора не совпадает с номерами, указанными в поле ani\_allow описания шлюза или RAS-пользователя

***eRadiusAdmissionTimeout=206,***

Radius-сервер не ответил на запрос в течении 10 секунд

***eRadiusAdmissionRouteReject=208,***

Radius-сервер отказал во внешней маршрутизации

***eOutgoingDestNumberEmpty=212,***

Предпринята попытка отправить Setup с пустым полем CalledStationId.

***eMaxSessionTime=300,***

Превышено максимальное время звонка (выдается Radius-сервером как CISCO\_H323\_CREDIT\_TIME)

***eDanglingCall=301***

Превышена максимальная длительность звонка (10000 секунд, около 3-х часов), предполагается, что это зависший звонок.

NB!

---

<sup>NB!</sup> Описание кодов причин разъединения Q931 см. в публикации «ITU-T Recommendation Q.850»



## **ПРИЛОЖЕНИЕ С: ПЕРЕЧЕНЬ ПОДДЕРЖИВАЕМЫХ ИДЕНТИФИКАТОРОВ MIB ОБЪЕКТОВ**

- 1.3.6.1.2.1.1.1.0 – описание системы (sysdescr)
- 1.3.6.1.2.1.1.2.0 – ветка параметров MVTs (sysobjectid)
- 1.3.6.1.2.1.1.4.0 – контактная информация (syscontact)
- 1.3.6.1.2.1.1.5.0 – имя системы (sysname)
- 1.3.6.1.2.1.1.6.0 – местоположение системы (syslocation)
- 1.3.6.1.2.1.1.7.0 – номер сервиса (syservices)
- 1.3.6.1.2.1.1.8.0 – тики времени
- 1.3.6.1.4.1.999.10.1.0 – количество активных звонков (SNMPGetActiveCalls)
- 1.3.6.1.4.1.999.10.2.0 – среднее количество активных звонков за последние 5 минут (SNMPGetAvg5Calls)
- 1.3.6.1.4.1.999.10.3.0 – максимальное количество одновременных звонков (SNMPGetMaxCalls)
- 1.3.6.1.4.1.999.10.4.0 – продолжительность непрерывной работы (SNMPGetUpTime)
- 1.3.6.1.4.1.999.10.5.0 – продолжительность подсчета статистики (SNMPGetStatTime)
- 1.3.6.1.4.1.999.10.6.0 – общая продолжительность звонков (SNMPGetCallDuration)
- 1.3.6.1.4.1.999.10.7.0 – количество принятых звонков (SNMPGetReceivedCalls)
- 1.3.6.1.4.1.999.10.8.0 – количество успешных звонков (SNMPGetNormalCalls)
- 1.3.6.1.4.1.999.10.9.0 – количество неуспешных звонков (SNMPGetFailedCalls)
- 1.3.6.1.4.1.999.10.10.0 – количество отвергнутых (rejected) вызовов (SNMPGetRejectedCalls)
- 1.3.6.1.2.1.1.3.0 – продолжительность работы системы (sysuptime)